

SENSITIVE PARSING: ERROR ANALYSIS AND EXPLANATION IN AN INTELLIGENT LANGUAGE TUTORING SYSTEM

Camilla Schwind

C.N.R.S. / G.R.T.C.
31, chemin Joseph Aiguier
13402 MARSEILLE CEDEX 9 - France

ABSTRACT

We present a uniform framework for dealing with errors in natural language sentences within the context of automated second language teaching. The idea is to use a feature grammar and to analyse errors as being sentences where features have other values than those they should have. By using a feature grammar it is possible to describe various types of errors (agreement, syntactic and semantic errors) in a uniform framework, to define in a clear and transparent way what an error is and - this is very important for our application - to analyse errors as arising from a misunderstanding or ignorance of grammatical rules on the part of students.

1. INTRODUCTION

In this paper, we present a uniform framework for dealing with errors in natural language sentences within the context of automated second language teaching. The idea is to use a feature grammar and to analyse errors as being sentences where features have other values than those they should have. By using a feature grammar it is possible to describe various types of errors (agreement, syntactic and semantic errors) in a uniform framework, to define in a clear and transparent way what an error is and -this is very important for our application- to analyse errors as arising from a misunderstanding or ignorance of grammatical rules on the part of the students.

The treatment and even the definition of errors within a language tutoring context is very different from the context of other applications of natural language understanding (c. f. approaches as described in [WEISCHEDEL 1983]), because the goal is different. In most applications, the goal is to understand a sentence despite any errors, i. e. to somehow analyse the sentence. In a language tutoring system, the goal is to understand what the student wanted to do, where he went wrong and what grammar rules he misunderstood or was unaware of. In this respect, error treatment is more difficult in the context of language tutoring than in other contexts, because errors do not need to be just ignored or sentences to be parsed despite any

errors, but the errors and the reasons for them have to be understood. To formulate this concisely, we could say : we are attempting to perform sensitive parsing rather than robust parsing. This means that we aim to achieve a system which is very sensitive to all possible kinds of errors and which moreover tries to find the "why" and the "how" of errors.

Related research has been carried out by [MENZEL 1986; PULMAN 1984; WEISCHEDEL et al. 1978]. Menzel's system is much more limited than ours, because he handles only fragmentary utterances whereas we try to analyse the correctness of complete sentences freely formed by students. Weischedel's treatment of syntactic and agreement errors is very similar to ours but less general and less thorough (they only recognize an error without explaining the reasons for it).

Here it is proposed to describe how we deal with agreement, syntactic and semantic errors within a German language tutoring system [SCHWIND 1986].

1. Agreement errors

In German, articles, adjectives and nouns within a noun group have to agree in gender, number and case, and verbs have to agree in person and number with the subject noun phrase of the sentence. The object complements of verbs take certain cases and so do prepositions. Agreement errors are errors on the syntactic level, but they do not concern the order of the words in a sentence, hence they can be corrected by changing the case, the number, the person or the gender of the noun phrases or parts of them.

2. Syntactic errors

We consider two types of syntactic errors, the first involving words which have been omitted or added, e.g. when an article or a preposition is missing or superfluous, and the second involving the permutation of words or syntactic groups. The latter error is very frequent in German because here the possible places of verbs in a sentence can differ from many other languages : for example, the verb can go to the very end of a sentence or to the very beginning. Some syntactic errors have to be partly

analysed at the morphological level because they involve both word construction and word order : for an example, in German there are verbs which have a prefix which in certain cases has to be detached and placed at the very end of the sentence.

(1) Er kommt zurück (He comes back).

This is the correct formulation of the sentence. Students of German tend typically not to detach the prefix *zurück* and construct the ill-formed sentence

(2) * Er zurückkommt (He backcomes).

The word *zurückkommt* does not exist in German (although the infinitive is *zurückkommen*) and this error has to be recognized at the word level, because *zurückkommt* is an ill-formed word, although the underlying error is a syntactic error.

3. Semantic Errors

We have actually been working on one type of semantic errors, namely errors in the semantic verb cases which arise from a misunderstanding of the meaning of verbs and nouns and their semantic relationships, thus posing lexical problems. For example, when a student forms the sentence

(3) *Das Heft arbeitet (The notebook works).

he has not understood that *arbeiten* requires an animate subject and that *Heft* is not animate, i.e. he has a lexical problem.

In what follows, we will first introduce, in a informal way, the theoretical background for error definition and then discuss the treatment of syntactic and semantic errors within the language teaching system.

Our system is implemented in PROLOG II and has been tested with various dictionaries and by different users (adult language students, pupils).

2. THEORETICAL BACKGROUND

In this chapter, the concepts of *feature grammar* and *unification* are introduced informally. We provide a slightly modified definition of unification where the result of the unification is the unified elements and the set of the pairs of elements for which the unification did not work out. This set is necessary for interpreting and explaining errors. Complex features have been used by most schools of linguistics [KAPLAN R.M., BRESNAN J. 1983; KARTUNEN 1984]. The whole process of syntactic analysis is governed by features and their values. Not only the lexical elements are classified by features but also the syntactic categories. For example, the category sentence is subclassified by the features *satzstellung*, whose values indicate whether the sentence has a normal word order or has the verb at the very end or at the beginning, (this

corresponds in German to different types of embedded phrases), and by the feature **embedded** with values + and - indicating whether a sentence or a noun phrase contains embedded phrases. We have constructed a grammar using 25 syntactic and 40 semantic features. To our knowledge, until now feature grammars have never been applied to the problem of analysing illformed sentences, nor within the context of language teaching.

A feature grammar is an extension of a CHOMSKY-grammar. The alphabet consists of *structured symbols* which are sets of pairs (feature,value). In the rest of this paper, a structured symbol *a* will be written as a tuple $a=[f_1(v_1),\dots,f_n(v_n)]$ where the f_i denote feature names and the v_i their values. No feature can occur twice within a structured symbol. The set of features occurring in *a*, $\{f_1,\dots,f_n\}$ is called the *domain* of *a*, and is written $d(a)$. The value of $f \in d(a)$ in *a* is denoted by $a(f)$. Hence $a(f) = v$ iff $f(v) \in a$. In many cases it is useful to introduce a more concise notation for sets of structured symbols. We need to denote such sets, because many words are ambiguous and have to be described by a set of structured symbols rather than by a single one. Most current theories also allow features that have complex values. By using disjunction and negation of values many structured symbols can be written much more economically. For example, the German noun *Kind* can have three cases (nominative, dative and accusative), but in this formalism it is denoted by just one symbol

[gender(neutr),case(neg(genitive)),number(singular)]. Let us call sets of structured symbols *complex symbols*. Structured symbols are used in a formal grammar for natural language sentences in the following way: there is one feature, **cat** (category), that plays a special part and whose values are the categories usually needed in a natural language grammar: *sent* (sentence), *np* (noun phrase), *vp* (verb phrase), etc. Further features characterize properties according to which categories are subclassified; e.g. **vcat** (verb category) is a feature whose values are *intrans* (intransitive) and *trans* (transitive) and *prep* (prepositional complement); the feature **place** subclassifies verbs and its possible values are the numbers 1, 2, 3, standing for the number of complements of a verb. **tense** is a feature with the properties *pluperfect*, *imperfect*, *perf*, *pres*, *fut* specifying the time of a verb. These three features all subclassify verbs. Other more frequently cited features are **case**, **number**, **gender** which characterize articles, nouns, adjectives, but also noun phrases and noun groups. Semantic properties of categories are equally characterized by features and formally these "semantic" features are not distinguished from "syntactic" features; e.g. **animate** is a semantic feature whose values are + and - and which belongs to nouns, **durative** and **static** and **action** are features which classify verbs.

All possible types of errors have been defined by means of features.

The definition of unification is slightly different from the usual definition (see [KARTUNEN 1984]), because in our application we need not only to find whether two symbols can be unified but also for what reasons they might possibly not be unified. Hence we need to have all the pairs of elements which cannot be unified.

Let $a = [f_1(v_1), \dots, f_n(v_n)]$ and $b = [g_1(w_1), \dots, g_m(w_m)]$, where v_i and w_j are sets of values. Then we define a predicate

$unify(a, b, r, e)$

where r is the result of the unification and e is the set consisting of all the pairs of value sets for which a and b could not be unified, together with all the symbols contained in the symmetrical difference between a and b .

$r = \{f(v) : f(v) \in a \div b \text{ or } (f \in d(a) \cap d(b) \text{ and } v = a(f) \cap b(f) \text{ whenever } a(f) \cap b(f) \neq \emptyset)\}$

(\div denotes the symmetrical difference between sets)

$e = \{f(v) : f(v) \in a \div b \text{ or } (f \in d(a) \cap d(b) \text{ and } a(f) \cap b(f) = \emptyset \text{ and } v = \langle a(f), b(f) \rangle)\}$

The unification is defined on sets of complex symbols. Let $a = \{a_1, a_2, \dots, a_n\}$ and $b = \{b_1, b_2, \dots, b_m\}$ where all a_i and b_j are of the form $[f_1(v_1), \dots, f_n(v_n)]$. Then the predicate unification of a and b (with the results r and e) is defined as the union set of all elements which unify(a_i, b_j, r, e)

$unification(a, b, r, e)$

$r = \cup \{r : unify(a_i, b_j, r, e) \text{ and } a_i \in a \text{ and } b_j \in b\}$

$e = \cup \{e : unify(a_i, b_j, r, e) \text{ and } a_i \in a \text{ and } b_j \in b\}$

The unification is obviously successful when $r \neq \emptyset$.

Example 1:

The definite article *der* is described by the complex symbol

$c = \{[Art-cat(def), Gender, case(genitive), number(plural)], [Art-cat(def), Gender(fem), case(or(genitive, dative)), number(singular)], [Art-cat(def), Gender(masc), case(nominative), number(singular)]\}$.

The noun *Lehrer* (teacher) has the representation

$n1 = \{[Gender(masc), case(neg(genitive)), number(singular)], [Gender(masc), case(neg(dative)), number(plural)]\}$

$unification(c, n1, r, e)$ evaluates to

$r = \{[Art-cat(def), Gender(masc), case(genitive), number(plural)], [Art-cat(def), Gender(masc), case(nominative), number(singular)]\}$

$e = \{[Art-cat(def), Gender(masc), case(<genitive, neg(genitive)>), number(<plural, singular>)], [Art-cat(def), Gender(<fem, masc>), case(genitive), number(<singular, plural>)], [Art-cat(def), Gender(<fem, masc>), case(dative), number(singular)], [Art-cat(def), Gender(masc), case(nominative), number(<singular, plural>)]\}$

Example 2:

The noun *Kind* (child) has the representation

$n2 = \{[Gender(neutr), case(neg(genitive)), number(singular)]\}$

$unification(c, n2, r, e)$ gives

$r = \emptyset$

$e = \{[Art-cat(def), Gender(neutr), case(<genitive, neg(genitive)>), number(<plural, singular>)], [Art-cat(def), Gender(<fem, neutr>), case(dative), number(singular)], [Art-cat(def), Gender(<masc, neutr>), case(nominative), number(singular)]\}$

Feature grammars are defined as formal grammars manipulating strings of complex symbols and the derivability concept is modified according to the structures of the complex symbols. To each production rule belongs an operation on the feature sets of the symbols involved.

3. THE TREATMENT OF ERRORS

The whole syntactic analysis is using the metamorphosis grammar formalism [COLMERAURER 1978], enriched with unification predicates for syntactic and semantic agreement.

3.1. Agreement errors.

The analysis of agreement errors in German is complex because morphologically, the words are highly ambiguous. There are 24 different definite articles (4 cases, 3 genders and 2 numbers) but there are only 6 different words for them all, each of which can have between 2 and 8 interpretations (or meanings). In the same way, every noun has at most four different forms which can have 8 different morphosyntactic meanings. Adjectives are even more ambiguous, because there are (at least) 4 different declinations according to their context within a sentence: preceded by a definite or an indefinite article, by no article or by a negation. Our grammar contains these four declinations, i.e. $4 \cdot 3 \cdot 2 \cdot 4$ adjective meanings and only 5 forms for them (ending by "e", "en", "em", "er", "es"). But the case and number of a noun phrase within a sentence depend on the verb, since a verb takes a certain case and determines the number. Hence an error in the number of a

noun phrase could also be an error in the number of the verb. Moreover, when two elements of a phrase do not agree, there are frequently several possible ways of analysing and explaining the disagreement. For this reason, the definition of unification has been slightly modified so as to produce all the pairs of features which do not agree as to their values. Consider example 2. The noun phrase *der Kind* cannot be unified and we want to explain to a student why. In the above example, three different analyses have been found. It depends on the context within a sentence which explanation is the right one. We have found that case filtering gives plausible explanations. In German, verb complements have cases. Hence, for any noun phrase in a sentence, there is an expectation as to the case. Consider the following sentences :

- (4) *Der Kind spielt (The child plays).
- (5) *Er gibt der Kind Milch (He gives milk to the child).
- (6) *Sie kennt der Kind (She knows the child).

In (4), *Der Kind* is the subject of the sentence and the expected case is the nominative. Case filtering gives the third error analysis : disagreement in the gender, since *der* is masculine whereas *Kind* is neuter. In (5), *der Kind* is the indirect object of the sentence and the expected case is the dative. Case filtering gives us the second error analysis : disagreement in the gender, since *der* is feminine whereas *Kind* is neuter. In (6), *der Kind* is the direct object of the sentence and the expected case is the accusative. By case filtering, we find that *der* cannot be accusative.

The most likely strategy for analysing of agreement errors consists of placing an error as high as possible within a syntax tree. But this procedure can be eliminated in the following situation. Take the sentence :

- (7) * Der Götter zürnen (The gods are angry).

Götter has the representation

$n = \{ \{ \text{Gender}(\text{masc}), \text{case}(\text{neg}(\text{dative})), \text{number}(\text{plural}) \} \}$

unification(a,n,r,e) gives

$r = \{ \{ \text{Art-cat}(\text{def}), \text{Gender}(\text{masc}), \text{case}(\text{genitive}), \text{number}(\text{plural}) \} \}$ and

$e = \{ \{ \text{Art-cat}(\text{def}), \text{Gender}(\text{<fem,masc>}), \text{case}(\text{genitive}), \text{number}(\text{<singular,plural>}), \text{Art-cat}(\text{def}), \text{Gender}(\text{masc}), \text{case}(\text{nominative}), \text{number}(\text{<singular,plural>}) \} \}$

Der Götter is the subject of the sentence and the expected case is the nominative, *der Götter* is genitive plural and this is the error signalled (disagreement on the case). But this analysis is not at all plausible. It is very unlikely that a student should try to construct a genitive plural, which is a "difficult" case, when the nominative is required., which is

the "easiest" case. People make errors in order to make their lives easier! Hence, the strategy of analysing errors as high as possible is not applicable when a subject noun phrase, which should be in the nominative case, could be analysed as having another case whereas parts of it are in the nominative. Now, we have seen, that in the definition of unification even when the unification is successful, the set of nonunifiable elements is produced. Besides computational issues, because the algorithm runs only once through the lists, this set is very useful when a noun phrase already analysed, such as the one in our example, has to be reviewed in order to find a possible disagreement between its parts. Case filtering of the disagreeing interpretations gives us the correct error analysis : disagreement in the number, since *der* is singular and *Götter* is plural.

During our numerous essays of the system, these explanations of agreement errors have always turned out to be plausible.

3.2. Syntactic Errors.

We distinguish between low level and high level syntactic errors. Low level syntactic errors involve the omission or addition of functional words such as articles or prepositions, and the permutation of words on the lexical level. High level syntactic errors involve the permutation of groups of words. High level errors are mostly due to non application of obligatory transformational rules or to application of the wrong rules, usually derived from the native language of the student. In [SCHUSTER 1986] this relationship between errors made by second language students and the grammar of their first language is systematically used for error handling. We will show by giving two examples how such types of errors can be clearly represented in PROLOG. All types of syntactic errors are treated by error rules.

I. In German, adjectives precede the noun group, whereas in French, they frequently follow it. This is described by the following rules (formulated as PROLOG clauses):

```

np(X,X0) :- art(X,X1), ng(X1,X0,F),F.
ng(X,X0,correct) :- ag(X,X1), noun(X1,X0);
noun(X,X0).
ng(X,X0,error(noun,ag)) :- noun(X,X1), ag(X1,X0).
ag(X,X0) :- adj(X,X0);
adj(X,X1), ag(X1,X0).
art("das".Y,Y).
noun("Auto".Y,Y).
adj("blaue".Y,Y).
correct.
error(noun,ag) :- error-message.

```

For the sake of clarity, we have simplified these rules by suppressing all terms relating to the morphological and semantic analysis and properties of the categories. The noun phrase *das blaue Auto* would be analysed correctly as np("das"."blaue"."Auto".nil,nil,correct) whereas the incorrect noun phrase *das Auto blaue* is analysed as np("das"."Auto"."blaue".nil,nil,error(noun,ag)). The np-rule treats the error predicate F, which is a PROLOG term, by calling it.

II. In German, verb groups in the perfect tense are frequently split up. The auxiliary takes the place of the verb, and the participle goes to the end of the sentence, as, for example in :

(8) Ich habe dem Baby Milch gegeben (I have to the baby milk given).

French (and equally English) students of German might say

(9) *Ich habe gegeben dem Baby Milch.

This transformation rule, as well as its erroneous omission, is represented in PROLOG as follows:

```
vp(X,XE,correct) :-
    verb(X,X1,t,XH) , compls(X1,X0),eq(X0,XH.XE).
vp(X,X0,error(verb,part-perf)):-
    verb(X,X1,perf,XH),
    freeze(X2,compls(X2,X0)),eq(X1,XH.X2).
verb("fährt".Y,Y,pres,0).
verb("ist".Y,Y,perf,"gefahren").
```

Again, this description has been simplified in order to make clear how these transformation rules function in PROLOG. freeze is a predefined predicate of PROLOG II [PrologIA] . freeze(X,P) delays the evaluation of P until X takes a value. compls analyses the verb complements of the sentence. The order of the sentence parts is produced by the equations between them (predicate eq).

3.3.Semantic errors.

The only type of semantic errors on which we have been working so far concerns the violation of semantic restrictions on verbs and their complements. Consider the sample sentence (3) in chapter 1. The semantic relationships between verbs and their complements as well as their semantic features are all described by semantic predicates.

```
subj-sem(arbeiten (work), human).
subj-sem(v,n) :- sup(n',n), subj-sem(v,n').
obj-sem(schreiben (write), text).
...
sup(human,individual).
sup(human,group).
sup(human,humanized).
...
sup(text,Heft (notebook)).
```

In the grammar rules, the semantic predicates are called as follows :

```
sg(...<v,n>...) :-
    np(...n...), vp(...v...),
    default(subjsem(v,n),sem-error(v,n)).
```

This is a rule analysing a sentence (sg). "default(p,q)" is a predefined predicate of PROLOG II first evaluating all possibilities for p, and only when none of these succeeds is q evaluated. sem-err produces an explanation of the type : *arbeiten* requires a human subject, *Heft* is not human but an written object.

4. CONCLUSION

The results of our experiments can be summarized as follows :

1. Agreement errors can perfectly well be handled in a very general way.
2. High and low level syntactic errors as well as lexical (semantic) errors can be satisfactorily dealt with but high level syntactic errors have to be anticipated, so that their treatment is not very general. Consequently, totally disordered sentences cannot be analysed (but should they be?).
3. Ambiguously interacting errors present a serious problem. Consider the following example.

(10) *Er schreibt dem Heft (He writes to the notebook).

The error could be analysed as a semantic error (*schreiben* requires a human dative object) or as a low level syntactic error (*schreiben* requires the preposition *an*). Obviously, there is no means of deciding which error the student has committed if there is no contextual information, which is generally the case in a language teaching environment.

REFERENCES

- COLMERAURER A. : Metamorphosis Grammar. In : Natural Language Communication with Computers (L. Bolc ed.), Lecture Notes in Computer Science 63, Springer Verlag, 1978.
- KAPLAN Ronald M. and BRESNAN Joan : Lexical Functional Grammar: A Formal System for Grammatical Representation. Ch.4 in : The Mental Representation of Grammatical Relations. J.Bresnan, (ed.), The MIT Press, Cambridge, Massachusetts, 1982.
- KARTUNEN Lauri : Features and values. In : Proceedings of the International Conference on Computational Linguistics 1984, Stanford, CA, pp. 28-33.

MENZEL Wolfgang : Automated Reasoning about Natural Language Correctness. In : Proceedings of the second Conference of the european chapter of the ACL., Kopenhagen, 1986.

PrologIA : PROLOG II, Manuel de référence, Marseille.

PULMAN S.G. : Limited Domain System for Language Teaching. In : Proceedings of the International Conference on Computational Linguistics 1984, Stanford, CA.

SCHWIND Camilla B. : A Formalism for the description of Question Answering Systems. In : Natural Language Communication with Computers (L. Bolc ed.), Lecture Notes in Computer Science 63, Springer Verlag, 1978.

SCHWIND Camilla B. : Overview of an intelligent language tutoring system. - Colloque International d'intelligence artificielle de Marseille, CIAM, Marseille, 1986, ed. Hermes, Paris.

SCHUSTER Ethel : The role of native grammars in correcting errors in second language learning. In : Computational Intelligence, Vol.2, No. 2, 1986.

WEISCHEDEL R.M. and SONDEHEIMER N.K. : Meta-rules as a Basis for Processing Ill-formed Input. American Journal of Computational Linguistics. Vol. 9, No. 3-4, 1983.

WEISCHEDEL R.M., VOGEL W.M. and JAMES M. : An Artificial Intelligence Approach to Language Teaching. Artificial Intelligence 10, 1978.