

ERROR DIAGNOSING AND SELECTION IN A TRAINING SYSTEM
FOR SECOND LANGUAGE LEARNING

Wolfgang Menzel

Zentralinstitut für Sprachwissenschaft
Akademie der Wissenschaften der DDR
Prenzlauer Promenade 149-152
Berlin, 1100, DDR

Abstract

A diagnosing procedure to be used in intelligent systems for language instruction is presented. Based on a knowledge representation scheme for a certain class of syntactic correctness conditions the system carries out a thorough analysis of possible error hypotheses and their consequences. A comparison with earlier attempts shows a clearly improved precision of diagnostic results. First of all, the procedure concentrates on an exact localization of rule violations, but - if desired - is able to infer information about factual faults as well.

I. INTRODUCTION

Error anticipation is a key issue so far, as systems for computer assisted language learning are to be enhanced with diagnostic and explanatory capabilities. Traditional approaches do always rely on the central assumption that any mistake possibly done by a student can be foreseen by the author of a teaching exercise in order to supply the tutoring system with a collection of adequate responses to all the different errors and error combinations which might occur in real training sessions.

Considering the enormous flexibility of natural language, however, this assumption is only justified for small classes of very simple type exercises, and most efforts are spent into devising sensible teaching aids despite the absence of appropriate diagnostic techniques. The author is forced to concentrate on single teaching programs without almost any chance to generalize from his results and to reuse parts of it in other contexts later.

More ambitious solutions certainly can be expected, if it becomes feasible to derive the diagnostic and explanatory abilities of the desired system directly from rules

for what is considered to be a correct construction in a sensible restricted subset of natural language, instead of using pre-selected lists of possibly occurring answers. In this respect, research in intelligent systems for second language learning sooner or later will follow similar lines as a parallel development in the field of expert systems, where a drastically improved diagnostic behaviour is attempted by modelling structure and function of the application domain (yielding so called "deep modelled" expert systems, which then are able to investigate faults by simulating them) rather than collecting huge amounts of symptom-oriented regularities to be used in a straightforward inference procedure on a surface level (c.f. for instance Davis 1984).

By including semantic, syntactic and morpho-syntactic regularities into a single solution Weischedel et al. (1978) proposed a system for intelligent language instruction that likewise turned out to be the most ambitious approach so far. Even today, almost exactly ten years later, its main premises and challenging predictions obviously do not fully match the real prospects in the field. Even worse, this as well as more recent attempts (Barchan et al. 1985, Schwind 1986) are strongly oriented on the notion of a "typical error". Thus they clearly fall short of the deep-modelling ideal and are connected with a number of critical shortcomings. These shortcomings can be avoided, if it becomes possible to identify limited application domains within the area of natural language instruction which allow a training system to be strictly based on the following basic principles:

- (1) Supply the system with only knowledge about correctness.
- (2) Devise a diagnosis procedure which is independent of the content of the knowledge base.

Currently used semantic representation schemes in general seem much too rough to capture all the rather subtle distinctions

between an acceptable natural language expression and a deviant one in a sufficiently systematic way. Even if the necessary means would be available the problem of actually coding the desired information for a realistic domain remains to be solved.

Better prospects for a successful implementation of these two principles can certainly be expected on the syntactic level. Nevertheless, in order to diagnose structural errors all currently known solutions rely exclusively on the use of "marked" rules, representing typical mistakes and being intended to invoke appropriate error messages. This practice is clearly inconsistent with principle (1) and comes out as nothing more than a simple shift of the necessity to anticipate student behaviour onto a new - not necessarily more perspicuous - level. Moreover, it leaves the system with all the consequences of a parsing grammar bloated with numbers of fault-specific rules.

The domain I believe is currently ripest for a strictly deep-modelled prototype is the area of morpho-syntactic correctness conditions which have to be satisfied in a given (if necessary artificially fixed) syntactic environment. These consistency constraints for well-formed syntactic trees comprise agreement relations as well as case government and, as Weischedel et al. (1978) acknowledge, constitute at least for languages like German a major error source. They can easily be expressed as predicates over structured sets of morpho-syntactic features and, if taken together, form simple networks.

Here again, all current approaches fall back on either short-cut solutions (e.g. take the first predicate evaluating to false) or simple heuristic guesses when trying to diagnose (esp. to locate) consistency errors. Heuristics (or the use of rules-of-thumb) typically is the method of one's choice when being faced with a, by its nature, ill-defined or extremely complex problem. Obviously the domain of morpho-syntactic regularities belongs to neither the one nor the other category.

Accordingly, an investigation into the problem of diagnosing consistency violations has shown that an efficient and extremely precise diagnostic procedure can be devised, which provides a sound basis for the generation of comprehensible explanations (including proposals for a proper solution), for a valuation of the learning progress and for an adaptive determination of the further training strategy. Moreover, it turned out that non-trivial training exercises based exclusively on simple morpho-syntactic correctness conditions can well be used independently of an error sensitive parser and additionally are connected with a number of pedagogical and performance gains (for a detailed discussion see Menzel, 1987 and 1988). In this case the student can be guided by an appropriate sentence context in a smooth and unconstrained way to exactly produce those frag-

mentary utterances which have the desired fixed syntactic structure.

II. CONSISTENCY CONSTRAINTS

Consistency constraints to be used as a basis for diagnosing students' faults are expressed in a quite common way as binary predicates which hold over structured feature sets of syntactically related word forms (source and destination). Feature sets, as usual, are attached to word forms via the dictionary or a morphological analysis. To specify the part of a feature set relevant to a specific correctness condition correctness predicates are augmented by a third argument, a category:

agreef <source>, <destination>, <category>].

Simple examples of correctness predicates define symmetrical relations between equivalent parts of the two involved feature sets, as does the number agreement between the determiner and the governing noun in a German noun phrase:

agreef det, noun, number].

Other predicates are needed to account for agreement between different parts of two feature sets. One example are selectional demands of a source to be matched with the corresponding properties of a destination, another is given by predicates coping with the distinction between e.g. different gender values belonging to stem or ending of German possessive pronouns.

A basically different type of predicates does not specify a source but instead gives a condition to be fulfilled by the destination:

satisfy[<destination>, <condition>].

This type is always used in one of the following cases:

(1) Conditions of pure structural origin so that no particular source can be given, e.g. the nominative of the subject:

satisfy[noun-of-subject, case=nom].

(2) Case government of verbs, e.g.:

satisfy[noun-of-dative-object, case=dat].

(3) Ordinary agreement relations where, because of the limits of a specific exercise, it is not necessary or not possible to specify the source of the condition explicitly (cut-off arcs in the constraint network).

(4) Word class restrictions, e.g. a noun phrase determiner can be an article as well as a possessive or demonstrative pronoun:

satisfy[det, cat=(art, poss-p, dem-p)] .

To achieve a uniform treatment, internally the satisfy-predicates are converted into ordinary agreement-predicates between an artificially created feature expression at a nowhere source (indicated by ***) and the desired destination:

```
satisfy[ noun, case=acc ]  
=>  
agree[ ***, noun, case ] .
```

A single training exercise usually is described by a number of correctness predicates which can be combined to form a simple constraint network. The arcs in this network represent correctness predicates and the connected nodes are identified by the source and destination arguments. A rather simple type exercise is, for example, the correct insertion of a German possessive pronoun into a carefully generated context like:

"Ich habe Schal und Mütze verloren.
Hat jemand ... Sachen gesehen?"

The exercise asks the student to have regard to the agreement of the pronoun with both, the subsequent noun ("Sachen") and the corresponding antecedent ("Ich"). It results in a constraint network consisting of three nodes and nine arcs. Two of the nodes (antecedent and noun) are associated with the context whereas the third (possessive) represents the student's response. The network is equivalent to the following list of correctness predicates:

```
satisfy[ possessive, cat=poss-p ]  
satisfy[ noun, cat=noun ]  
satisfy[ antecedent, cat=( noun, pers-p) ]  
agree[ antecedent, possessive, stem-number ]  
agree[ antecedent, possessive, stem-gender ]  
agree[ antecedent, possessive, stem-person ]  
agree[ noun, possessive, case ]  
agree[ noun, possessive, number ]  
agree[ noun, possessive, gender ] .
```

As a slightly more ambitious example may serve the formation of a German local PP according to the fixed structural pattern

(preposition determiner adjective noun)

to be inserted into sentences like

"Das Geld habe ich ... gelegt."

This time the network consists of five nodes and 16 edges. All the nodes correspond to word forms in the student's response, but depending on the sentence context only a subset of exactly four nodes (e.g. (prep-4 det adj noun)) is instantiated to the incoming word forms:

```
satisfy[ prep-3, cat=prepl ]  
satisfy[ prep-3, select=location ]  
satisfy[ prep-4, cat=prepl ]  
satisfy[ prep-4, select=direction ]
```

```
satisfy[ noun, cat=noun ]  
satisfy[ det, cat=( art, poss-p, dem-p) ]  
satisfy[ adj, cat=adj ]  
agree[ prep-3, noun, case ]  
agree[ prep-4, noun, case ]  
agree[ noun, det, case ]  
agree[ noun, det, number ]  
agree[ noun, det, gender ]  
agree[ noun, adj, case ]  
agree[ noun, adj, number ]  
agree[ noun, adj, gender ]  
agree[ det, adj, inflectional-degree ] .
```

III. DIAGNOSIS

Diagnosis is mainly based on constraint propagation techniques. The nodes in the constraint network are treated as variables which receive their values (feature sets) by means of a pattern matching procedure on the student's input.

The procedure consists of four parts being invoked sequentially. Two of them (the hard core of diagnosis) are intended to detect and - if desired - to locate errors in the student's response. The other two refine diagnostic results by applying transformational rules and preference criteria in order to provide a sound basis for the generation of appropriate explanations.

The reason to draw a clear distinction between the error detection and localization components is one of mere technical nature. The separation has been introduced to considerably speed up the handling of error-free utterances, since it allows the time expensive localization procedure to be activated only if indeed an error occurred and the student actually did ask for an explanation of his mistake.

(A) error detection

Error detection, in fact, is a direct proof procedure for the correctness of the utterance. Trying to show the absence of errors, it evaluates the relevant predicates of the constraint network, taking into consideration all the morpho-syntactic readings of the word forms concerned. The values of the network variables are constantly updated according to the results of an ordinary feature set unification until finally a state is reached which satisfies all relevant predicates simultaneously.

Given the case this proof cannot be established, one or possibly several mistakes of the student have to be assumed and upon request a detailed analysis of the error reasons may become necessary.

(B) error localization

Error localization is carried out by a simulation of constraint violations. For

e a c h predicate in the network the procedure follows up the consequences of assuming the student did ignore the existence of exactly this particular regularity. This assumption is modelled by temporarily adding the negated predicate to the knowledge base and following up its consequences around the network. Theoretically such a modification of the knowledge base is equivalent to reasoning based on Reiter's famous "closed world assumption" (Reiter, 1978): If you cannot proof P, add not(P) to the set of premises.

By adding it as a premise the diagnosis procedure treats a negated predicate as nothing more than merely an error hypothesis. It can be raised to the level of a confirmed error description only after its consequences for other predicates in the network have carefully been investigated and properly recorded. In contrast to a typical short-cut solution or a heuristics-based approach the diagnosis procedure thus carries out a thorough analysis of all possible error hypotheses. Hence, it is particularly qualified to diagnose multiple faults as well as errors with an ambiguous interpretation.

Technically this kind of reasoning is achieved by simply resuming error detection with just the logical complement of the usual feature unification results. Error detection provides for this purpose and collects all necessary data on a separate stack.

Despite the rather tiny size of a typical constraint network, the simulation of constraint violations results in a procedure of high complexity and effective measures are required to keep the size of the search space limited. Hence, in a widely accepted way error localization is restricted to the search of minimal diagnoses (that is, results including only a minimum number of constraint violation hypotheses).

(C) error transformation

According to the fundamental distinction between rules and facts in a knowledge base two different types of misconceptions have to be distinguished in the diagnostic results:

(a) rule violations: the ignorance of correctness conditions (expressed by predicates in the network) and

(b) factual faults: the lack of knowledge about specific morpho-syntactic properties of a word form (expressed by a feature set in the dictionary).

An integration of both types of misconceptions into a single diagnosing procedure neither yields an efficient solution nor turns out to be really necessary. In principle, rule violations and factual faults represent different views on one and the same phenomenon and a particular error can equally well be explained in terms of rules or in terms of facts. Both types should be convertible into each other without much difficulty.

Accordingly, error localization has been designed to exclusively concentrate on the analysis of rule violations and to leave open the problem of factual faults. A total ignorance of factual faults, of course, would be an intolerable restriction on the diagnostic capabilities of the system, since then it interprets errors of whatever kind in terms of rule violations only.

Hence, a transformational component has been added which is able to infer factual faults from the hitherto obtained diagnostic results. The necessary information is usually contained in typical structural error configurations involving the violation of adjoining predicates for one and the same morpho-syntactic category. One or several simultaneous constraint violations

```
and[not[agree[X1, Y, C]],
... ,
not[agree[Xn, Y, C]]]
```

or

```
and[not[agree[Y, X1, C]],
... ,
not[agree[Y, Xn, C]]]
```

can be transformed into a factual fault description, iff

(1) the set of constraint violations is complete with respect to Y, i.e. there is no other X which appears as source or destination argument of a predicate agree[X, Y, C] or agree[Y, X, C] within the knowledge base, and

(2) consistency holds for all neighbouring nodes of Y: agree[Xi, Xj, C].

Under these circumstances Y is said to be isolated with respect to C and without loss of generality a new description can be generated, stating the student's misconception to assume a wrong feature value concerning category C at the word form Y:

```
not[agree[***, Y, C]]
```

resp.

```
not[agree[Y, ***, C]]
```

again using *** to indicate a non-specified source or destination. In particular three types of structural configurations have to be distinguished:

(1) A single constraint violation concerning two isolated nodes X and Y can be transformed into two alternative factual error descriptions:

```
not[agree[X, Y, C]]
==>
exor[not[agree[***, Y, C]],
not[agree[X, ***, C]]].
```

(2) A single constraint violation with only

one isolated node Y is equivalent to a single factual error description:

$\text{not}[\text{agree}[X, Y, C1]] \implies \text{not}[\text{agree}[***, Y, C1]].$

(3) Several constraint violations for an isolated node Y can be summarized by a single factual error description (c.f. the general case above).

The decision whether to apply a transformation or not is influenced by two criteria, the tolerable complexity of diagnostic results (it is raised by (1), maintained by (2) and reduced by (3)) and the intended teaching strategy (c.f. section IV).

(4) error generalization

Error generalization aims at an as high as possible aggregation of elementary errors (constraint violations or factual faults) into a more concise description. Generalization schemata can be defined for all three argument positions of a violated correctness predicate, e.g. for a category generalization:

$\text{and}[\text{not}[\text{agree}[X, Y, C1]],$
 $\text{not}[\text{agree}[X, Y, C2]]]$
 \implies
 $\text{not}[\text{agree}[X, Y, (C1, C2)]]$

which for the purpose of explanation later might be paraphrased like

"Missing C1 and C2 agreement between X and Y."

Other generalization rules are intended to collapse alternative diagnostic readings which merely are artifacts resulting from the technical layout of dictionary entries.

Besides these stringent generalization schemes sometimes very weak ones turn out to be useful as well. They are particularly suited to generate an at least partially comprehensible explanation which in an opaque error situation may give a rough indication of the error location instead of annoying the student with plenty of detail:

$\text{and}[\text{not}[\text{agree}[X1, Y, C1]],$
 $[\text{not}[\text{agree}[X2, Y, C2]]]$
 \implies
 $\text{not}[\text{agree}[***, Y, (C1, C2)]].$

The result of a weak generalization then may be explained to the student perhaps in a sentence like:

"There is something wrong with C1 and C2 at Y."

Since it is logically incomplete, however, a weak generalization can no longer serve as a reliable basis to derive repair suggestions from, neither for internal use nor for a presentation to the student.

There is some reason to assume the general four step scheme of diagnosis not being restricted to violations of consistency constraints, but rather being adaptable to other types of errors, namely structural ones, as well. Again, the procedure has to be based on a detection and proper localization of elementary error types with respect to a simple model of correctness, e.g. a recursion-free PSG. In this context the elementary error types are substitution, insertion and omission of single word forms. Other types (e.g. extraposition or mutual interchange of forms) can later be inferred from the primary results by means of transformation rules. Generalization finally allows to condense error descriptions into more complex structural units (e.g. constituent based errors).

IV. ERROR SELECTION

Error selection is an integral part of several steps of the diagnosis procedure. It tries to keep the number of concurrent error descriptions as small as possible. Error localization selects on the number of primary errors per error description, but since transformation as well as generalization can influence the resulting complexity, their results in turn are subject to a decision with regard to a minimal number of errors.

Although the side effect of further reducing the complexity of minimal diagnoses is normally welcomed, error transformations have to be sensitive to the intended teaching strategy. Diagnosing a constraint violation exactly pinpoints the conceptual mistake of the student but does not always give useful hints about where to correct the utterance. In these cases, despite being more complex, an indication of factual faults may sometimes be more helpful, since it better points out the existing possibilities for an improvement of the partially wrong solution.

On the other hand, even less complex factual faults may be quite undesired if they result in doubtful correction proposals (e.g. in a given semantic context a gender-of-the-noun error can hardly be repaired by simply exchanging the noun in question).

Despite the selection of minimal diagnoses on several levels, sometimes the diagnosing component provides a number of alternative minimal diagnoses. They finally are evaluated and weighted according to a number of preference criteria:

- (a) a structural measure which depending on whether a conceptual description or a repair suggestion is preferred may select errors higher up or deeper down in a hierarchical syntactic representation of the utterance,
- (b) the degree to which an error explanation

can be assumed to be helpful for the student (if possible, certain types of explanations are suppressed),

(c) hints, resulting from an obstinate repetition of one and the same error type.

V. CONCLUSIONS

Although it is connected with a number of interesting advantages, deep modelled reasoning clearly should not be considered the sine qua non of future tools for language instruction. Based on the very idea of a closed world, it is by definition limited to the rather narrow domains of knowledge about natural language which allow a highly reliable and complete description of the necessary correctness conditions. Unfortunately, neither high precision nor completeness can well be attributed to major parts of currently available grammar models.

Nevertheless, two possibilities for an advantageous application of the presented diagnosis procedure can be identified even today:

(1) "stand alone" use in simple agreement exercises similar to the examples presented in section II.

(2) integration of the procedure as a specialized subroutine into an error sensitive parser (whether it be surface oriented or deep modelled) to replace currently used heuristics for consistency check.

Following the first opening a prototype system has been realized which integrates the diagnosing procedure into a learning environment. Besides the guiding advice of the sentential context additional measures have been taken to avoid the student feeling uneasy about the limitations of the fixed structural input pattern. Using a menu-based input mode it is possible not only to inspect the dictionary at run time but also to select appropriate word forms from it. Thus a quick and convenient interaction facility is offered and the student is encouraged to construct his solution in a toy kit manner from the available stock of word forms.

To evaluate the diagnostic capabilities of the approach a quality measurement has been carried out using the constraint network of the second exercise sample of section II (formation of a local PP). The valuation has been based on the observation that for practical purposes the ability to precisely locate single word form errors is of most importance. An exhaustive analysis of all instances of artificially implanted single word form substitutions (including rather "exotic" ones as well) yielded the following results:

absolutely exact localization of an error either by pointing out the erroneous form (plus the categories and values involved) or by properly reflecting an ambiguous correction possibility: 73 percent.

approximative localization by only indicating the violated constraint predicate without being able to decide between source or destination: 23 percent.

bad localization either by returning several rule violations or by ignoring an alternative correction possibility: 4 percent.

Notice that not a single instance of wrong, that is, misleading diagnostic results occurred.

The system currently runs on 16-bit-micros (with drastic reductions on 8-bit-micros, as well) and has been tested on a number of non-trivial exercises for German. Experience has shown that the approach allows diversified and intensive training and supplies surprisingly precise explanations for most of the erroneous utterances.

References

- Barchan, J; Woodmansee, B. and Yazdani, M (1985) Computer Assisted Instruction using a French Grammar Analyser. Research Report 126, Department of Computer Science, University of Exeter.
- Davis, R. (1984) Diagnostic Reasoning Based on Structure and Behaviour, Artificial Intelligence, vol. 24, no. 1-3: 347-410.
- Menzel, W. (1987) Automated Reasoning about Natural Language Correctness, Proceedings 3rd European Chapter of the ACL, Copenhagen, April 1987.
- Menzel, W. (1988) Wissensbasierte Lehrsysteme für den Sprachunterricht, Zeitschrift für Phonetik, Sprachwissenschaft und Kommunikationsforschung, vol. 41, Berlin: in press.
- Reiter, R. (1978) On Closed World Data Bases, in Gallaire, H., and Minker, J. (eds.), Logic and Data Bases. Plenum Press, New York: 55-76.
- Schwind, C. B. (1986) Overview of an Intelligent Language Tutoring System, Proceedings 2nd International Conference on Artificial Intelligence, CIIAM '86, Marseille, December 1986, Editions Hermes, Paris: 389-409.
- Weischedel, R. M.; Voge, W. M. und James, M. (1978) An Artificial Intelligence Approach to Language Instruction, Artificial Intelligence, vol. 10, no. 3: 225-240.