Error Recognition and Parsing of Syntactically mildly ill-formed Natural Language

Veit Reuer

Humboldt-Universität zu Berlin

Proceedings of the LFG00 Conference

University of California, Berkeley

Miriam Butt and Tracy Holloway King (Editors)

2000

CSLI Publications

http://csli-publications.stanford.edu/

Error Recognition and Parsing of Syntactically mildly ill-formed Natural Language

Veit Reuer
vreuer@compling.hu-berlin.de
Humboldt-Universität zu Berlin

1 Introduction

This paper describes the error-recognition module of an interactive CALL-system. Using the system the language learner is invited to produce complete written sentences in little dialog tasks with the computer. This setting challenges the learner to use language interactively in order to enhance the development of 'communicative competence'. Emphasis is put on the possibility to give adequate feedback to the learner if an ill-formed sentence is encountered. As mentioned in e.g. Menzel and Schröder (1998), usually a program does not support both requirements at the same time. If free formed input is allowed, the system is not able to do a detailed analysis of the input while if the error-recognition capabilities are advanced for satisfying feedback, the choice of exercises is in most cases quite limited. Thus the main goal in developing a CALL-system should be to provide a stimulating environment *and* to give adequate feedback. The preconditions therefore are robust parsing and error recognition, which are both handled by the system described below.

The paper begins with a short description of the system's design. Various modules analyse the learner's input sentence for orthographic, syntactic and semantic errors. Finally a dialog component produces output either reacting to an error or continuing the dialog.

Then some introductory remarks about error-recognition follow. Mainly two ways to provide feedback to the learner have been applied in CALL-systems. One way is to introduce 'mal-rules' into the grammar to get structural descriptions for erroneous input (e.g. Schneider and McCoy (1998)). The other way is to develop costly parsing methods (Menzel, 1992). Alternatively the system described here is using a more traditional grammar-theory (Lexical Functional Grammar) without any anticipation of errors and uses a modified unification algorithm to analyse the learner-input and to provide methods to increase the efficiency of the parsing process.

The main part of the paper decribes the parser and the methods for error recognition. The parser is an Earley style chart-parser (Earley (1970); implemented following Naumann and Langer (1994)) and the grammar is an unaugmented LFG-grammar. The term 'mild syntactic errors' is used because the error analysis works until now only for errors encodable in the f-structures of a sentence. Therefore, for example, linearisation errors will not be recognised. I will show how a modified unification process can relax constraints if the values contradict each other on the one hand and on the other hand encode errors in the functional structure (f-structure) for feedback purposes. For efficiency purposes some methods have been integrated affecting the structures resulting from the unification as well as the chart-items entered into the chart.

Furthermore some arguments for the usage of LFG in this setting are discussed. Essentially LFG has proven to be useful in applied computational linguistics and provides concepts understandable by language learners.

2 System Design

While using the system every input sentence (usually an answer to a question asked by the system) is first passed to a module checking for orthographic and syntactic errors. The *orthographic check* is done based on methods developed by Oflazer (1996). With the help of a finite state recognizer mildly deviating strings are identified and correct versions are presented to the learner if necessary. The methods for a *syntactic analysis* will be described below. As a next step the analysis of the sentence is checked against a *world knowledge base*, from which feedback is given to the learner if the sentence contains an error of sortal restrictions. If the learner has made an error, the system provides feedback to support the learner in typing a syntactically correct or semantically more plausible sentence. After this step the *dialog module* tries to find a reaction to continue the dialog.

The main focus in all the analyses is to continue the dialog without ignoring the errors made by the learner. Only the orthographic check will actually interrupt the dialog with a suggestion of correct words for the misspelled items. In all other cases either the 'dialog partner' will react to the erroneous input or the system will display an error message, mark the sentence incorrect and simply continue the dialog. This means that either the dialog partner states that it was difficult to interpret the input and might ask for a rephrasal or the sentence was interpretable for the continuation of the dialog by the modules following the syntactic analysis.

As an example the dialog might take the following form.

(1) `system:` Polizeirevier. Ja bitte?
Police station. May I help you?
`user:` * Ich habe ein Unfall gesehen.
I have seen an accident.
`system:` Was ist denn genau passiert?
What exactly happened?

Following the analysis of the user-input an error-message is displayed.

(2) `error message:` Es gibt vermutlich einen Fehler im direkten Objekt: das Genus ist nicht korrekt: masc vs. neut.
There is possibly an error in the direct Object: the Gender is incorrect: masc vs. neut.

This message has to be deleted to continue the dialog as shown in example (1).

Error Analysis

Error diagnosis depends on sensitive parsing for the identification of errors. As opposed to robust parsing with heuristics (e.g. Alexandersson et al. (1994)) the task is not to get the largest chunk possible of the sentence, but to achieve an analysis spanning the complete sentence including a recognition of a possibly minimal error.

Efficient error-sensitive parsing has to make some assumption about the possible errors it will encounter, otherwise no solution would be found as mentioned above. One choice could be the integration of the so-called 'mal-rules' into the grammar, which are used to analyse erroneous learner input. This has been done as early as 1978 (Weischedel et al. (1978) uses 'meta-rules' outside the core grammar) and in various other approaches. It does allow for a precise recognition of errors but unfortunately mal-rules may increase the search space in a grammar dramatically and additionally rely on the foresight of the linguist writing the grammar. As the error-analyses by Heringer (1995) in the computer program "Aus Fehlern lernen" shows, about half of the thirteen most common errors made by learners of German appear in areas covered by the f-structure (e.g. agreement, subcategorisation for certain preposition, valency). Foreseeing every possible error a system might encounter is therefore difficult to realise. It seems more plausible to avoid the usage of anticipation of errors and introduce some methods for error-sensitive parsing.

Another way of coping with ungrammatical input could be the usage of a probabilistic grammar. Probabilistic grammars can achieve a higher degree of robustness by using variable levels of correctness instead of clear-cut right-wrong decisions. However to be able to diagnose an error these systems would have to be trained on large amounts of *erroneous* annotated texts produced by language learners. Annotation of erroneous text corpora taken from language learning tasks hasn't been done to my knowledge.

A further difficulty in the diagnosis of an error is the inclusion of relevant information from different levels of the analysis. In case of an error a purely syntax-driven diagnosis might suggest more than one alternative for correction (example taken from Menzel and Schröder (1998)).

(3) * The cars drives fast.

To correct this sentence, the noun could be changed to singular or alternatively the verb could be plural. Here the system should ideally provide a method for finding a single diagnosis, for example decide that - based on a semantic representation of the current situation - there should only be one car hence the noun should be singular. Albeit - as is well known -, modelling the situation at hand and obtaining the correct inferences from the knowledge base is a difficult task. In the case at hand based on a syntactic analysis the computer would state that there is a disagreement in number/person between the subject noun-phrase and the verb-phrase (3sg vs. non-3sg). Besides, leaving the decision for the correction to the user has the advantage of producing a learning effect.

In the system at hand, errors can theoretically occur at all features of an f-structure. Since this is unlikely and even implausible, only linguistically relevant features may be relaxed in the unification process. Therefore the concept of this system starts with an unrestricted approach and adds methods for limiting the complexity instead of opening up a very restricted system.

There have been various previous attempts to deal with error analysis in a CALL-context. Schwind's system (1988) follows a dual strategy in using constraint relaxation for agreement errors and error-rules for e.g. linearisation errors. Unrestricted constraint relaxation as in this case works for smaller grammars, but, used in larger grammars, leads to a large computational overhead. As mentioned above, error-rules seem unsuitable for more complex systems because of the enlargement of the search space and of the fact that their limited ability can only account for some errors.

This means that also no Optimality Theory-style additions should be used as suggested e.g. in Kuhn and Rohrer (1997). Using OT-type markings with ps-rules to achieve robust parsing of erroneous text would require the addition of rules describing ungrammatical constructions which in turn would lead to the mentioned problems.

Menzel's system (1992) avoids anticipation, but uses a model-theoretic approach, which only allows for small grammars to be used due to computational complexity.

A different approach is described in Menzel and Schröder (1998). Their system allows for errors on the syntactic, semantic and pragmatic level due to the structure of the grammar. The system includes a constraint based dependency grammar with weighted constraints using the method of *partial constraint satisfaction* and more specifically solving the *minimal violation problem*. The grammar concept allows for the recognition not only of syntactic errors but also of linearisation errors, since these rules are included in the constraint structures as well. However, as the authors concede, some problems remain. The grammar formalism only allows for binary branching, which is rather artificial for some linguistic structures, even though the expressive power of the formalism is not limited by this restriction. Another difficulty is the scoring of constraints. Although this allows one to set preferences for single constraints, is not clear that a meaningful error-analysis can be reached. A certain final structural description of a sentence might have the lowest scoring from the set of structures even though a human corrector would choose another solution accidently having a higher score. Once every single constraint in a sufficiently large-scale grammar has received a scoring, the effects on the results of a parse might be difficult to determine.

To this end a system was developed that is able to analyse sentences with mild syntactic errors, identify the error and present meaningful feedback to the learner. In order to avoid the anticipation of (mild) syntactic errors the unification process was modified to allow for a relaxation of features, valuation of the error type and a subsequent encoding of the error in the f-structure. Additionally a common linguistic framework was chosen which allows for the encoding of various grammatical phenomena.

At the heart of the system is an Earley-based chart parser and an LFG-type unification grammar. Elements of both parts (chart-parsing and unification) are used in combination to identify errors and control the parsing-process. With the help of an error-value the structure describing a sentence with minimal errors is identified. This means, that also other unification based grammar theories could be used. Up to now the parsing technique of the ps-rules is rather traditional but it is planned to enhance this in order to allow for recognition of linearisation-errors as well (e.g. Kato (1994)).

3 Modified Error-Sensitive Parsing

The main idea is to relax constraints in order to continue the unification process in case the values of an attribute contradict each other. Additionally the number of relaxed constraints is used to mark the resulting structure with an error-measure. For feedback purposes the error-attributes can then be interpreted and a message given to the learner.

Unification

The difference between the definition for this type of unification in Schwind (1988) and mine is that in my system the resulting structure does not contain value sets with the mismatching values but instead a new attribute (`err`) is introduced into the resulting structure. This feature in turn has a complex value containing the non-unifiable feature and value. Also Schwind's concepts does not include the handling of complex values and has no mechanism for stating preferences, which I consider important (see below).

The type of unification can also be compared with the so-called *default unification*. It is similar to Carpenter's (1993) *credulous default unification* in that it also returns sets of feature structures. But unlike this definition, my version of unification always returns a tuple perhaps containing the mismatching value in an extra error-feature. Subsumption is then defined as follows (simplified; see Carpenter (1993) for details):

Definition 1 *A f-structure* F *subsumes another f-structure* F' *if and only if* F' *provides at least as much information about path values* or atomic values and their immediate attributes contained in error-attributes *as* F.

It must be noted, that complex/complex clashes are not handled by this definition. The defintion of unification based on subsumption is than straightforward without any modification:

Defintion 2 *The unification of two f-structures* F *and* F' *is taken to be the least upper bound of* F *and* F' *in the collection of f-structures ordered by subsumption.*

As a result of the expanded definition of subsumption two structures will result from the unification, if values do not match.

(4) $[F : a] \sqcup_{err} [F : b] = \left\{ \begin{bmatrix} F{:}a \\ err{:}[F{:}b] \end{bmatrix}, \begin{bmatrix} F{:}b \\ err{:}[F{:}a] \end{bmatrix} \right\}$

As opposed to other treatments of inconsistencies in feature-structures this procedure is monotonic, symmetrical and results in completely consistent feature-structures (cf. Vogel and Cooper (1995)). If multiple errors occur, these are treated in the same way as adjuncts or coordination are handled in LFG. Sets are introduced into the f-structure to contain all the elements of an error-unification. The implementation of the coherence and completeness condition is slightly modified to account for error-features.

In the following example two f-structures (5) and (6) result in the structures (7) and (8) when unified (*Mice likes milk*).

(5)
$$\begin{bmatrix} \texttt{pred:} & \texttt{`like}\left\langle\left(\uparrow\texttt{subj}\right),\left(\uparrow\texttt{obj}\right)\right\rangle\texttt{'} \\ \texttt{tense:} & \texttt{pres} \\ \texttt{subj:} & \begin{bmatrix} \texttt{pers:} & \texttt{3sg} \end{bmatrix} \\ \texttt{obj:} & \begin{bmatrix} \texttt{pred:} & \texttt{`MILK'} \end{bmatrix} \end{bmatrix}$$

(6)
$$\begin{bmatrix} \texttt{subj:} & \begin{bmatrix} \texttt{pers:} & \texttt{-3sg} \\ \texttt{pred:} & \texttt{`MICE'} \end{bmatrix} \end{bmatrix}$$

(7)
$$\begin{bmatrix} \texttt{pred:} & \texttt{`like}\left\langle\left(\uparrow\texttt{subj}\right),\left(\uparrow\texttt{obj}\right)\right\rangle\texttt{'} \\ \texttt{tense:} & \texttt{pres} \\ \texttt{subj:} & \begin{bmatrix} \texttt{pers:} & \texttt{3sg} \\ \texttt{pred:} & \texttt{`MICE'} \\ \texttt{err:} & \begin{bmatrix} \texttt{pers:} & \texttt{-3sg} \end{bmatrix} \end{bmatrix} \\ \texttt{obj:} & \begin{bmatrix} \texttt{pred:} & \texttt{`MILK'} \end{bmatrix} \end{bmatrix}$$

(8)

$$
\begin{bmatrix} \texttt{pred:} & \texttt{`like}\left\langle\left(\uparrow\texttt{subj}\right),\left(\uparrow\texttt{obj}\right)\right\rangle\texttt{'} \\ \texttt{tense:} & \texttt{pres} \\ \texttt{subj:} & \begin{bmatrix} \texttt{pers:} & \texttt{-3sg} \\ \texttt{pred:} & \texttt{`MICE'} \\ \texttt{err:} & \begin{bmatrix}\texttt{pers:} & \texttt{3sg}\end{bmatrix}\end{bmatrix} \\ \texttt{obj:} & \begin{bmatrix}\texttt{pred:} & \texttt{`MILK'}\end{bmatrix}\end{bmatrix}
$$

Since structures (7) and (8) subsume each other, only one resulting structure will be added to the chart. Notice, that even though an implemented program seems more complicated than the simple standard unification, unification of two error-less unifiable structures does not take more steps than with the standard program.

Some method has to be integrated to allow for a selection from the multiple resulting descriptions of the sentence. Therefore a scoring mechanism is introduced. Scoring the number of conflicting values gives the 'distance' between the resulting f-structure and an error-less f-structure. The unification is completely sucessful when the error-value equals 0. Otherwise the f-structure with the lowest error-value is chosen to continue the dialog. If for example the f-structure in example (7) is the structure with the lowest error-value spanning the complete sentence, the learner is informed of a number-agreement error.

Using this mechanism in a small grammar without the advanced naming of 'relaxable' features seems feasable because the encoding of the error brings about the possibility for adequate feedback to the learner. Unrestricted usage of this mechanism in larger grammars would probably lead to large amounts of structures, which slow down the analysis considerably and do not contribute to the final result. Therefore some measures are taken to decrease the processing load.

One method is to restrict the types of features which actually may be relaxed. It seems sensible to restrict the constraint relaxation mechanism to only the linguistically relevant features, i.e. especially agreement features. Features which for example only enhance the parsing process will result in a failed unification and thus do not produce an additional item to be parsed. These irrelevant features are of course more a kind of linguistically *less* relevant features. Usually they represent a kind of redundancy to guide the parsing process and make the grammar more stable (c.f. Butt et al. (1999)).

A second method is to limit the maximum number of features which may fail. The unification then fails if too many constraints have to be relaxed in order to produce a resulting f-structure, and an additional entry in the chart is prevented. As is shown in Menzel and Schröder (1998), sentences with too many recognised errors are not interpretable anymore. So this is a plausible strategy if one is able to find a suitable threshold.

Chart-Parsing

The methods to constrain the production of items during the unificaton process can and should be continued in the parsing process with ps-rules to increase the efficiency, since the parsing is already in the standard case at worst exponential.

After the unification one of the resulting items is being tested against the chart with the modified subsumption test and might be entered. The error-measure is also entered into the chart along with the chart-item, according to the number of relaxations, i.e. the number of error-features.

To further reduce the number of items in the chart, the error-value can be utilized. The chart is divided into two parts with 'active' and 'passive' items. The number of items in the chart with only passive items can be reduced by adding only items whose error value is smaller than or equal to the error value of items of the same category already in the chart. This reduces the number of items mainly if an almost correct sentence is parsed.

A subsumption-check including the use of the error-value could also be done on the chart with active items. As experiments have shown subsumption on the already found elements of an active chart-item does indeed reduce the overall number of items. Nevertheless the gain is lost especially with short sentences due to the costly subsumption-procedure and the comparatively few items that contain partial findings.

4 LFG and CALL

There are several reasons, why LFG should be chosen in an CALL-environment. More specifically I should talk about 'intelligent' CALL (ICALL), since almost all commercial language learning programs do not make use of any method of natural language processing and rely purely on anticipation-based methods for implementing exercises.

Firstly LFG has proven to be useful not only as a grammar theory but also as a computationally sensible choice in NLP as shown e.g. in the ParGram-project (Butt et al. (1999)). This of course depends partly on the general preference for unification based grammars, where implementational issues are better understood than the implementation of e.g. movement of nodes in a tree as in Government&Binding-based concepts.

Secondly LFG seems to be useful in large-scale grammar development even though the concept in the ParGram-project has been modified in some ways from the standard LFG. The introduction of the m-structure for example is owed to the overall goal of *parallel* grammar development.

Finally and most importantly LFG uses concepts and terminology which can easily be transformed into meaningful concepts for language learners. The grammar by Schwarze (1995) using LFG for a description of Italian demonstrates this.

One example is the functional approach to grammar theory. Concepts like 'subject' or 'direct object' are easy to understand for language learners. Thus messages can be created directly from the f-structure. As shown in the previous example (7) the f-structure can be used to create a message saying that the subject does not agree with the verb in number (3sg vs. non-3sg).

Another example might be the direct coding of surface-structure with ps-rules. This conforms much more to the ways used in language teaching than other grammar theories. The use of e.g. deep- and surface-structure as well as movement is not part of traditional descriptive grammars and therefore difficult to understand. The grammar and more specifically the ps-rules can be used for informative messages to the learner. Using a parsing method following Kato (1994) the following example could be explained in a way a learner would easily understand .

(9) * Dog barks.

(10) NP -> N {ntype $=_c$ mass | num = pl}

NP -> Det N ntype $=_c$ std

Here the explanation could be extracted with the help of the simplified example rules (10), that the noun-phrase containing only a noun should be either a mass-noun or plural in number or should include a determiner. The important point is that the term ‘noun-phrase’ is (maybe still) more common in language teaching than terms like for example ‘DP’ meaning ‘determiner-phrase’.

All grammatical information encoded in the f-structure and possibly in the ps-rules can be used in this way to present helpful instead of simple ‘wrong!’-messages to the learner.

The grammar so far includes the following grammatical constructions:

- Declarative sentences, imperative sentences
- Main verb-complements (direct object, indirect object, prepositional object, infintival complement, sentential complement)
- Topicalized constructions (subject-verb-inversion)
- Complex verbal groups (auxiliar-, modal-, 'Verbklammer'-construction, separable prefix)
- NPs with relativ sentences
- Simple adjectival phrases
- Attributives and attributive prepositional phrases

With these types of constructions the grammar covers almost all constructions learners entered in preliminary experiments. Mainly conjuctive and some subjunctive constructions the system could not handle during these experiments. Therefore it seems that the grammar does not need to be a *very* large grammar in a language learning scenario, since some construction will probably never be used by a learner.

5 Discussion and Conclusion

The parsing module in this CALL-system analyzes syntactically mildly deviating natural language input and is able to produce an error-feedback, which can guide the learner to a meaningful correction. The system therefore is suited for language learners because of immediate and precise feedback originating from the f-structures. Additionally the chart can be used to create a partial parse consisting of items covering large chunks of the input sentence.

In the mentioned preliminary user-study some 200 sentences were collected in two scenarios. The first was the graphical presentation of a car accident and the task for the learner to report this accident to the police ‘by telephone’. The second scenario consisted of simple questions presented to the learner about her/his name, hobbies, subjects taken at university, etc. The results show, that almost 30 % of the syntactically erroneous sentences lead to precise and meaningful error-messages. The 70 % of sentences not covered included mainly sentences with errors, which are not encoded in the f-structure (e.g. linearization) or are simply not included in the grammar at all (e.g. subjunctive sentences). About 5 % of the correct sentences were not covered by the grammar at that stage of development.

Some difficulties remain. Since the error-recognition takes place at the highest point in the derivation tree, one can not be sure to have found the minimal error.

(11) * [*Der* *Männer*]$_{NP}$ [*lachen*]$_{VP}$.

*	[*Der*	*Männer*]$_{NP}$	[*lachen*]$_{VP}$.
	sg-nom/pl-gen	pl	pl
	The	men	laugh

In the above example the analysis would give a mismatch in Case between the NP (genitive plural) and VP (subcategorising for nominative plural). But in fact *Männer* can also be nominative. It is therefore more plausible that the form for the nominative plural determiner was not known than that a genitive plural NP was constructed. For cases like this an error explanation should be based on an analysis of the tree as far down as possible, i.e. on the pre-terminal level.

Some features of the underlying grammar theory, which have proven useful to decribe linguistic phenomena have not been integrated yet. These include the concept of 'functional uncertainty' and also disjunctions, but the system does contain negation and conditional equations. While disjunctions can theoretically be replaced by listing the disjuncts, functional uncertainty is an inherent concept in the grammar theory.

Finally some work still has to be done to improve the efficiency of the parsing mechanism. The performance depends very much on the number of functions a lexeme can have and not so much on the number of ps-rules in the grammar. The higher the number of functions the more items are added especially to the chart containing active items. With all sensible features allowed to be relaxed the parsing-time needed was about 200 % longer than without using the error-recognition mechanism.

In this paper, I proposed a method for analysing free formed input in an CALL-environment that allows to provide feedback to the learner about errors made. The system is based on a well-founded grammar formalism and parsing technique. It provides feedback to the learner about a variety of errors without anticipating them in any way, including

- agreement
- auxiliary selection
- Case frames

The description of the error can be taken directly from the structure produced by the parser during the unification-process. By limiting the types of errors and scoring the error occurences a significant gain towards an efficient parsing process is achieved. The grammar theory LFG seems especially suited in an ICALL-system, because on the one hand it is computationally feasible and on the other hand uses concepts and terminology which correspond to the ones used in descriptive grammars.

References

Jan Alexandersson, Elisabeth Maier, and Norbert Reithinger. 1994. A robust and efficient three-layered dialog component for a speech-to-speech translation system. Verbmobil-report 50, DFKI, Saarbrücken.

Ted Briscoe, Ann Copestake, and Valeria de Paiva, editors. 1993. *Inheritance, Defaults and the Lexicon.* Cambridge University Press, Cambridge.

Miriam Butt, Stefanie Dipper, Anette Frank, and Tracy Holloway King. 1999. Writing large-scale parallel grammars for English, French and German. In *Proc. LFG99 Conference*, Manchester. CSLI Publications.

Bob Carpenter, 1993. *Skeptical and Credulous Default Unification with Applications to Templates and Inheritance*, pages 13–37. In Briscoe et al. (Briscoe et al., 1993).

J. Earley. 1970. An efficient context-free parsing algorithm. *Communications of the ACM*, 13.

Hans Jürgen Heringer. 1995. Aus fehlern lernen. CD-ROM für Win9x/NT, http://www.phil.uni-augsburg.de/phil2/faecher/germanis/daf/forschung/forschung.html.

Tsuneaki Kato. 1994. Yet another chart-based technique for parsing ill-formed input. In *Proc. 4th Conference on Applied Natural Language Processing (ANLP)*, pages 107–112, Stuttgart.

Jonas Kuhn and Christian Rohrer. 1997. Approaching ambiguity in real-life sentences - the application of an Optimality Theory-inspired constraint ranking in a large-scale LFG grammar. In *Proc. der Jahrestagung der Sektion Computerlinguistik der DGfS*, Heidelberg.

Wolfgang Menzel and Ingo Schröder. 1998. Constraint-based diagnosis for intelligent language tutoring systems. Fachbereich Informatik Report Nr. FBI-HH-B-208-98, Universität Hamburg.

Wolfgang Menzel. 1992. *Modellbasierte Fehlerdiagnose in Sprachlehrsystemen.* Niemeyer, Tübingen.

Sven Naumann and Hagen Langer. 1994. *Parsing.* Teubner, Stuttgart.

Kemal Oflazer. 1996. Error-tolerant finite-state recognition with applications to morphological analysis and spelling correction. *Computational Linguistics*, 22(1).

David Schneider and Kathleen F. McCoy. 1998. Recognizing syntactic errors in the writing of second language learners. In *Proc. 17th Int. Conference on Computational Linguistics (COLING)*, Montreal.

Andreas Schöter and Carl Vogel, editors. 1995. *Nonclassical Feature Systems*, volume 10 of *Edinburgh Working Papers in Cognitive Science.* Edinburgh University.

Christoph Schwarze. 1995. *Grammatik der italienischen Sprache.* Niemeyer, Tübingen.

Camilla Schwind. 1988. Sensitive parsing: Error analysis and explanation in an intelligent language tutoring system. In *Proc. 12th Int. Conference on Computational Linguistics (COLING)*, pages 608–613.

Carl Vogel and Robin Cooper, 1995. *Robust Chart Parsing with Mildly Inconsistent Feature Structures*, pages 197–216. Volume 10 of Schöter and Vogel (Schöter and Vogel, 1995).

Ralph M. Weischedel, Wilfried M. Voge, and Mark James. 1978. An artificial intelligence approach to language instruction. *Artificial Intelligence*, 10.