Multiple Learner Errors and Meaningful Feedback: A Challenge for ICALL Systems

Trude Heift
Simon Fraser University

ABSTRACT
This paper describes a web-based ICALL system for German that provides error-specific feedback suited to learner expertise. The main focus of the discussion is on the Domain Knowledge and the Filtering Module. The Domain Knowledge represents the knowledge of linguistic rules and vocabulary, and its goal is to parse sentences and phrases to produce sets of phrase descriptors. Phrase descriptors provide very detailed information on the types of errors and their location in the sentence. The Filtering Module is responsible for processing multiple learner errors. Motivated by pedagogical and linguistic design decisions, the Filtering Module ranks student errors by way of an Error Priority Queue. The Error Priority Queue is flexible: the grammar constraints can be reordered to reflect the desired emphasis of a particular exercise. In addition, a language instructor might choose not to report some errors. The paper concludes with a study that supports the need for a CALL system that addresses multiple errors by considering language teaching pedagogy.

KEYWORDS
Natural Language Processing (NLP)-CALL, Error Analysis, Feedback, German as a Second Language, Grammar Instruction

1. INTRODUCTION
While there has been a noticeable trend towards web-based applications in language learning, it has also been noted that very few of these applications follow basic instructional design principles (Felix, 2002; Bangs, 2002). In particular, meaningful feedback seems to be lacking in many (Bangs, 2002).

Meaningful feedback can be defined as any system response that provides a learning opportunity for students. For example, a linguistic explanation of an error will ideally enable students to eradicate the error (see also Pusack, 1983). In a CALL program, meaningful feedback can be produced in at least two ways: it can be manually encoded which, however, poses difficulties in terms of scalability, or it can be automatically generated by a sophisticated answer processing mechanism (e.g., by a parser and a grammar).
When designing a parser-based ICALL system, a number of pedagogical decisions need to be made to achieve meaningful feedback. A standard problem in parsing is that most sentences can be assigned more than one syntactic structure. For example, van Noord (1997) states that the *Alvey Tools Grammar* with 780 rules averages about 100 readings per sentence on sentences ranging in length between 13 and 30 words. The maximum was 2,736 readings for a 29 word sentence. Although typical language exercises do not contain 29 word sentences, an unconstrained grammar generally produces more parses than systems designed for only well formed input. As a result, Natural Language Processing (NLP) applications must incorporate techniques for selecting a preferred parse. Ideally, the techniques in an ICALL system are motivated by language teaching pedagogy to ensure meaningful feedback. (For examples, see Heift, 1998.)

A further pedagogical challenge for ICALL systems with respect to meaningful feedback is multiple errors made by students in one sentence. While it is desirable to construct an ICALL program capable of detecting and accurately explaining all errors, it does not follow that the system should display each and every error detected. In the absence of an error filtering mechanism, the sheer amount of feedback would overwhelm students. For example, in evaluating her own system Schwind (1990a, p. 577) reports that “(s)ometimes, however, the explanations were too long, especially when students accumulated errors.”

A previous study by Heift (2001), for instance, showed that approximately 40% of the sentences analyzed contained more than one error. However, a language instructor typically skips irrelevant errors and discusses the remaining ones one at a time. Example (1) shows a sentence with multiple errors.

(1a) *Heute meine Kindern haben gespielt mit der Hund.
(1b) Heute haben meine Kinder mit dem Hund gespielt.

‘Today my children were playing with the dog.’

The student who wrote this sentence made five errors:

1. word order: the finite verb *haben* needs to be in second position;
2. word order: the nonfinite verb *gespielt* needs to be in final position;
3. spelling error with the past participle *gespielt*;
4. wrong plural inflection for the subject *Kinder*; and
5. wrong case for the dative determiner *dem*.

From a pedagogical—and also motivational point of view, a system should not overwhelm students with instructional feedback referring to more than one error at a time. Schwind’s (1990a) solution to this problem is that multiple errors should be avoided from the outset. She suggests that sentence construction exercises should focus on specific grammatical phenomena such as prepositions or verb cases (see also Kenning & Kenning, 1990).

While Schwind’s approach is probably inherent in many ICALL systems,
limiting the teaching domain is only a partial solution. Even a basic sentence in German, as illustrated in example (1), requires a number of rules and knowledge about the case system, prepositions, word order, and so on. Holland (1994), in her BRIDGE system, applies a different approach to address the problem of multiple errors. She displays only one error at a time and permits instructors to divide errors into primary, which are automatically displayed, and secondary, which are displayed only at students' request.

In a study regarding the volume of feedback for different kinds of learners, van der Linden (1993, p. 65) found that “feedback, in order to be consulted, has to be concise and precise. Long feedback (exceeding three lines) is not read and for that reason not useful.” She further states that displaying more than one feedback response at a time makes the correction process too complex for students (see also Brandl, 1995). Van der Linden's (1993) study makes three final recommendations:

1. Feedback needs to be accurate in order to be of any use to the student.
2. Displaying more than one error message at a time is not very useful because at some point they probably will not be read.
3. Explanations for a particular error should also be kept short.

With regard to feedback display, van der Linden's recommendations require a system submodule to sift all incoming errors. The errors have to be reported one at a time, and the error explanations should be brief. This approach provides students with enough information to correct the error, but not an overwhelming amount, and yet records detailed information within the student model for assessment and remediation. The question arises in which order the errors should be reported and, from a computational point of view, how this knowledge should be represented in an ICALL system.

The analysis described in this paper implements an Error Priority Queue which ranks student errors so as to display a single feedback message in case of multiple constraint violations. The ranking of student errors in the Error Priority Queue is, however, flexible: the grammar constraints can be reordered to reflect the desired emphasis of a particular exercise. In addition, a language instructor might choose not to report some errors. In such an instance, some grammar constraints display no feedback message at all, although the error is still recorded in the Student Model of the system.

The following section provides an overview of the architecture of our ICALL system for German. Section 3 discusses the Error Priority Queue which is used by the Filtering Module to rank student errors. Section 4 presents data on the use of our system that show the frequency and distribution of multiple errors. Finally, section 5 offers some concluding remarks.
2. SYSTEM OVERVIEW

The system consists of four major components: the Domain Knowledge, the Analysis Module, the Student Model, and the Filtering Module. Figure (1) illustrates how sentences are processed by the system.

Figure 1
System Overview

2.1 The Domain Knowledge

The Domain Knowledge represents the system’s knowledge of the language. It consists of a parser with a Head-driven Phrase Structure Grammar (HPSG). The goal of the Domain Knowledge is to parse sentences and phrases to produce sets of phrase descriptors. A phrase descriptor describes a particular grammatical constraint (e.g., subject-verb agreement), its presence or absence in the input sentence and the student’s performance on this constraint. In HPSG (Pollard & Sag, 1987, 1994), linguistic information is formally represented as feature structures. Feature structures specify values for various at-
tributes as partial descriptions of a linguistic sign. HPSG adopts a lexicalist approach in which syntactic information is described within each lexical entry. For example, the feature structure given in Figure 2 illustrates that the verb *geht* subcategorizes for a subject.

Figure 2
Partial Feature Structure for *geht*

The subject is minimally specified as a noun and its person and number features are structure-shared with the agreement features of the verb. Structure-sharing is indicated by multiple occurrences of a co-indexing box labeling the single value. Because syntactic information is expressed within each lexical entry, HPSG requires only a few generalized syntactic rules to specify how words and phrases combine into larger units.

A grammar is written as a set of constraints. To illustrate the concept, consider the simple sentences given in (2):

(2a) *Er gehst.
(2b) *Du gehst.

‘You are leaving.’

In (2a), the constraints between the subject *er* and the verb *gehst* require agreement in number, person, and that the subject is the nominative case. Any of the three constraints could block parsing if not successfully met. Figure 2 shows that the subject of the verb *gehst* must be a 2nd person, singular, nominative, noun. The element, *Er*, given in Figure 3, is inflected for 3rd person, singular, nominative; thus, *Er* gehst will fail to parse.
However, a constraint can be relaxed by changing its structure so that it records whether or not agreement is present rather than enforcing agreement between two constituents. To achieve this end, the subject *er*, given in Figure 4, is no longer marked as [per 3rd], [num sg], [case nom]. Instead the feature per, for example, specifies all possible person features, that is, 1st, 2nd, and 3rd. For *er*, the value for 1st and 2nd is error, while for 3rd it is correct. Conceptually, the feature structure states that it is incorrect to use *er* as a 1st and 2nd person pronoun, but correct for 3rd person.

The verb *gehst*, given in Figure 5, no longer subcategorizes for a subject marked [per 2nd]. Instead, the verb *gehst* will inherit the value of 2nd from its subject during parsing. For the subject *er*, the value for 2nd is an error indicating that the constraint on person agreement has been violated but, importantly, allowing the parse to succeed.
For the correct sentence *Du gehst*, given in example (2b), the value for 2nd would be correct because *du* is a second person, nominative pronoun, as illustrated in Figure 6. During parsing, *gehst* will inherit the feature value correct indicating that the constraint on person agreement has been met.
Multiple Learner Errors and Meaningful Feedback

In addition to the HPSG features, the grammar uses a feature descriptor representing the description of the phrase that the parser builds up. During parsing, the values of the features of descriptor become specified. For example, the phrase descriptor vp_per records the constraint on person agreement. For the sentence *Er gehst, vp_per will inherit its value from the feature 2nd, given in the lexical entry of gehst in Figure 5. The final result is the phrase descriptor [main_clause [vp_per [2nd error]]], indicating that the required grammatical constraint on person agreement has not been met.

For a sentence that contains multiple errors, each phrase descriptor indicates precisely where the errors occurred. For example, the sentence given in (3a) not only contains an agreement error in person but also a case error with both the direct and indirect objects.

(3a) *Er gibst den Mann der Buch.
(3b) Du gibst dem Mann das Buch.
‘You are giving the man the book.’

The responsibility of the head of a phrase is to collect the phrase descriptors of its complements. To achieve this objective, the phrase descriptors are percolated up the syntactic tree via the Descriptor Principle which states that the descriptor features of the mother node are the descriptor features of the head daughter. As a result, the phrase descriptors for the incorrect sentence in (3a) are:

\[
\begin{align*}
\text{[main\_clause [vp\_per [2nd error]]]} \\
\text{[main\_clause [indir\_obj [det denerror]]]} \\
\text{[main\_clause [dir\_obj [det dererror]]]}
\end{align*}
\]

The method described here for analyzing constraints has two advantages. First, the technique is very general in that it can be applied to a variety of grammatical phenomena. The phrase descriptors record whether a grammatical phenomenon is present in the input and, if so, whether it is correctly or incorrectly formed. Phrase descriptors provide very detailed information indicating precisely where an error in the sentence occurred. Second, errors are not treated differently from well-formed input. Neither the parsing nor the grammar formalism needs to be altered, and no additional rules are required.2

2.2 The Analysis Module

The Analysis Module takes a phrase descriptor as input and generates sets of possible responses to the learner’s input that the instructional system can use when interacting with students. A response is a pair that contains a message the system uses to inform students if a phrase descriptor indicates there has been an error and a Student Model update. The Student Model update contains the name
of the grammar constraint in the Student Model along with an instruction to increment or decrement the corresponding cumulative total.

The Analysis Module generates sets of instructional feedback of increasing abstraction. For example, consider the ungrammatical sentence in (4a).

\( \text{(4a)} \) *Der Mann dankt das Mädchen.
\( \text{(4b)} \) Der Mann dankt dem Mädchen.

‘The man thanks the girl.’

Inexperienced students should be informed that *Mädchen* is a neuter noun, that the verb *danken* is a dative verb, and that the determiner *das* is incorrect. Students who have mastered case assignment (as indicated by the Student Model) may be informed only that the case of the object is incorrect.

2.3 The Student Model

The Student Model dynamically evolves based on students’ performance. The information in the model is used for two main functions: (a) modulation of instructional feedback and (b) assessment and remediation.

The Student Model keeps track of individual students’ performance on a variety of grammatical phenomena (e.g., agreement, modals, and verb particles) based on the information obtained from the phrase descriptors. Phrase descriptors correspond to structures in the Student Model and are the interface medium between the Student Model and the grammar of the system. The Student Model passes instructional feedback suited to learner expertise to the Filtering Module.

For each grammar constraint, the Student Model keeps a counter for each student with a score for each grammar skill. This score ranges from 0 to n, where we have set n to 30. The score increases when students provide evidence of a successful use of that grammar skill and decreases when they provide evidence of an unsuccessful use of that grammar skill. The amount by which the student score increases or decreases can vary depending on the current value of the score. Initially, we set all scores to an intermediate level, but pretesting can determine individual differences from the outset.

For the purposes of modulating instructional feedback, we have identified three categories of scores. Scores from 0-10 are assigned to the novice category, 11-20 to the intermediate category, and 21-30 to the expert category. When students make an error on a particular grammar skill, the message they receive depends on their score for that skill. If they are ranked as a novice, they will receive a more informative message than if they are ranked as an expert. Since the score for each grammar skill is independent of the score for the other grammar skills, students may be expert at subject-verb agreement but novice at forming the passive—and receive the appropriate message.

The score information is also used for a variety of remediation and assess-
Multiple Learner Errors and Meaningful Feedback

By comparing the Student Model at the beginning and end of a session, we can provide a summary of the mistakes that students made during that session. In our current system, the mistakes are summarized into general categories such as “Verb Tenses,” “Pronouns,” and so forth. These groups are set by means of a parameter file. Similarly, we can also identify the grammar skills where students were correct and provide a “positive” of what they did right. At present we show a list of the errors at the end of each exercise set.

One can also examine the Student Model overall and identify students’ current strengths and weaknesses. We identify students’ strengths as the five highest scoring grammar skills that have a score greater than 15 (half of the total scale). We identify students’ weaknesses as the 5 lowest scoring grammar skills that have a score less than 15. Students can access this information.

Finally, the Student Model information can also be used to provide exercises to students which focus on their areas of weaknesses. Instead of repeating the same exercise in which they made the mistake, the system has the capacity to identify other examples requiring the same grammar skill. This feature avoids the problem of students rote learning the solution to a particular example without actually learning the general solution.

2.4 The Filtering Module

The Filtering Module determines the order of the instructional feedback displayed to students. The system displays one message at a time so as not to overwhelm them with multiple error messages.

The grammar constraints produced by the phrase descriptors are hierarchically organized. An Error Priority Queue determines the order in which instructional feedback is displayed by considering the frequency and importance of an error in a given sentence. After students make the required correction, the whole evaluation process is repeated.

3. The Error Priority Queue

The Student Model maintains grammar constraints and selects instructional feedback suited to students’ expertise. In case of multiple errors, the Error Priority Queue determines the order in which instructional feedback messages are displayed. It ranks instructional feedback with respect to the frequency and importance of an error within a given sentence.

The Error Priority Queue for the grammar constraints of a main clause is partially given in Figure 7. The names of the grammar constraints generated and maintained by the Analysis Module and Student Model, respectively are given in parentheses.
The grammar constraints, given in the Error Priority Queue are grouped according to grammatical phenomena. For example, the group Prepositional Phrases in a Main Clause contains all constraints relevant to prepositional phrases. Each member of a group in the Error Priority Queue refers to a node in the Student Model.

The groups in the Error Priority Queue are sorted according to the frequency and importance of an error within a sentence. If students made multiple errors, the system ranks instructional feedback messages according to the order specified and displays them one at a time.

The Error Priority Queue shown in Figure 7 reflects the default setting for the importance of an error in a given exercise. For example, grammar constraints in the group Word Order in a Main Clause refer to errors in linear precedence. In the default setting, they are reported first since word order is one of the fundamental concepts of a language and likely to have high priority in most exercises.

The ordering of the groups of grammar constraints can, however, be altered to reflect the pedagogical practices of a particular language instructor. For ex-
ample, an instructor might want to center exercises around dative case assignment, in which case the grammar constraints could be reordered so that errors of indirect objects are reported first. In addition, a language instructor might choose to suppress some errors, those not relevant to a specific exercise, in order not to distract the student from the main task. Suppressing certain errors does not affect their contribution to the Student Model on the rationale that behind-the-scenes information should be as detailed as possible.

4. THE SYSTEM IN USE

In using the system with our students, we were interested in the frequency and types of multiple errors that occurred. For data collection, we implemented a tracking system to collect detailed information on the student-computer interaction (see Heift & Nicholson, 2000).

Over the course of the semester, 33 students in two introductory courses of German completed six chapters with a total of 120 exercises. Each chapter contained a variety of tasks ranging from typing in words or sentences and clicking or dragging objects to listening to words and phrases. The grammatical structures present in the exercises were: (a) gender and number agreement of noun phrases, (b) subject-verb agreement, (c) present tense of regular and irregular verbs, (d) accusative and dative objects/prepositions, (e) two-way prepositions, (f) present perfect, (g) auxiliaries, (h) word order of finite and nonfinite verbs, and (i) modal verbs. The linguistic structures had all been practiced in communicative class activities prior to the computer sessions.

A total of 1,004 sentences containing 1,387 errors were submitted and analyzed. Table 1 shows that approximately 30% of the 1004 incorrect sentences contained multiple errors at initial submission. Further analysis indicated that, if multiple errors occurred, two errors per submission occurred most frequently (one submission contained five errors).

Table 1
Error Distribution

<table>
<thead>
<tr>
<th>Number of Submissions</th>
<th>Number of Errors</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>720</td>
<td>1</td>
<td>71.7</td>
</tr>
<tr>
<td>198</td>
<td>2</td>
<td>19.7</td>
</tr>
<tr>
<td>74</td>
<td>3</td>
<td>7.4</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>1.1</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>0.1</td>
</tr>
</tbody>
</table>

We assume that, in a less constrained practice environment, the percentage of multiple errors would be even higher. To some extent, the limited number of multiple errors may be due to the fact that students had previously practiced the grammatical structures and vocabulary in communicative activities in class.
Additionally, each chapter gradually introduced at most two new grammatical concepts.

In determining the appropriateness of the ranking of grammar constraints in our Error Priority Queue, we considered the distribution of total number of errors that occurred. The error breakdown in Table 2 shows that 409 (29.5%) of the 1,387 errors were due to spelling mistakes, while the remaining 978 (70.5%) errors occurred in grammatical constructions.3

Table 2
Error Breakdown

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Number of Errors</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spelling</td>
<td>409</td>
<td>29.5</td>
</tr>
<tr>
<td>Grammar</td>
<td>978</td>
<td>70.5</td>
</tr>
<tr>
<td>Total</td>
<td>1,387</td>
<td>100.0</td>
</tr>
</tbody>
</table>

Breakdown of Grammar Errors

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Number of Errors</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct Objects (gender, number, case)</td>
<td>280</td>
<td>28.6</td>
</tr>
<tr>
<td>Subject-Verb Agreement (person, number)</td>
<td>182</td>
<td>18.6</td>
</tr>
<tr>
<td>Prepositional Phrases: Two-way (gender, number, case)</td>
<td>171</td>
<td>17.5</td>
</tr>
<tr>
<td>Subjects (gender, number, case)</td>
<td>130</td>
<td>13.3</td>
</tr>
<tr>
<td>Indirect Objects (gender, number, case)</td>
<td>86</td>
<td>8.8</td>
</tr>
<tr>
<td>Prepositional Phrases: Accusative (gender, number, case)</td>
<td>42</td>
<td>4.3</td>
</tr>
<tr>
<td>Missing Words/Constituents</td>
<td>36</td>
<td>3.7</td>
</tr>
<tr>
<td>Prepositional Phrases: Dative (gender, number, case)</td>
<td>24</td>
<td>2.5</td>
</tr>
<tr>
<td>Extra Words</td>
<td>11</td>
<td>1.1</td>
</tr>
<tr>
<td>Auxiliaries</td>
<td>7</td>
<td>0.7</td>
</tr>
<tr>
<td>Word Order</td>
<td>6</td>
<td>0.6</td>
</tr>
<tr>
<td>Verb Complements (infinitive vs. past participle)</td>
<td>3</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Most errors occurred with direct objects (28.6%) and subject-verb agreement (18.6%). However, these were the most frequent constructions contained in the 120 exercises of the study. In contrast, only chapters 5 and 6 (40 exercises in total) focused on the present perfect and modals. These constructions were not contained in any of the previous chapters, thus there were fewer opportunities for errors with these grammar topics than, for example, with subject-verb agreement. However, the fact that errors with two-way prepositions rank third in the list confirms an instructor’s assumption that this structure poses considerable problems for German language students. Only chapter 4 (20 exercises in total) focused on two-way prepositions.

We further examined the data with respect to multiple errors that occurred with each submission. Besides the co-occurrence of spelling and grammar mistakes, we found that a combination of errors relating to direct objects, subject-verb agreement, and prepositional phrases co-occurred most frequently. This
finding is not surprising given the error frequency presented in Table 2. Unlike our initial assumption, however, errors with subject-verb agreement were more frequent than mistakes with the subject, and we changed the ranking of grammar constraints in our Error Priority Queue accordingly. Finally, although the data showed that errors in word order were not as frequent as initially anticipated we nonetheless feel that these errors should be addressed first and thus given highest priority in the Error Priority Queue. After all, correct word order might assist the student in identifying phrasal constituents.

5. CONCLUSION
Multiple errors pose pedagogical as well as linguistic challenges to any ICALL system. To address them effectively, computational algorithms have to be developed that take language teaching pedagogy into account. In this paper we discussed our phrase descriptors that indicate precisely the types of errors that occurred in sentences. We further described the filtering mechanism of our ICALL system that ranks learner errors in case of multiple errors. The Filtering Module uses an Error Priority Queue that can be adjusted to instructors’ and learners’ needs. We presented data on the use of our system that suggested changes to the initial ranking of the grammar constraints of our Error Priority Queue. However, the data of our study were limited to grammatical constructions commonly found in an introductory course for German. Subsequent studies that incorporate a wider range of grammar topics will certainly provide more insight in the study of multiple errors.

NOTES

1 The error types for determiners are more fine grained indicating the incorrect article that has been used. For example, den error indicates that students incorrectly used the determiner den for the indirect object. For a detailed analysis, see Heift (1998).

2 For a detailed analysis of errors in linear precedence, see Heift (1998). For alternative approaches to error detection, see Reuer (this volume), L’haire and Vandeventer (this volume), Menzel and Schröder (1998), Schneider and McCoy (1998).

3 In addition to the grammar and parser, our system includes a spellchecker which ensures that the words contained in the student input are correctly spelled before the sentence is sent to the parser (see Heift & Nicholson, 2001).

REFERENCES


Multiple Learner Errors and Meaningful Feedback


AUTHOR'S BIODATA

Dr. Trude Heift is Assistant Professor in the Linguistics Department and the Director of the Language Learning Centre at Simon Fraser University. Her research areas are CALL, Computational, and Applied Linguistics. Her main interests are in ICALL, human-computer interaction, learner strategies, student modeling, and error analysis.

AUTHOR'S ADDRESS

Dr. Trude Heift  
Linguistics Department  
Simon Fraser University  
Burnaby, B.C.  
Canada V5A 1S6  
Tel: 604/291-3369  
Fax: 604/291-5659  
Email: heift@sfu.ca