User Modeling and User-Adapted Interaction **14:** 317–350, 2004.
© 2004 *Kluwer Academic Publishers. Printed in the Netherlands.*

Empirical Derivation of a Sequence of User Stereotypes for Language Learning

LISA N. MICHAUD[1,†] and KATHLEEN F. McCOY[2]
[1]*Department of Mathematics and Computer Science, Wheaton College, Norton, MA 02766*
[2]*Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716*

(Received: 21 March 2002; accepted in revised form 12 Oct. 2003)

Abstract. The work described here pertains to ICICLE, an intelligent tutoring system for which we have designed a user model to supply data for intelligent natural language parse disambiguation. This model attempts to capture the user's mastery of various grammatical units and thus can be used to predict the grammar rules he or she is most likely using when producing language. Because ICICLE's user modeling component must infer the user's language mastery on the basis of limited writing samples, it makes use of an inferencing mechanism that will require knowledge of stereotypic acquisition sequences in the user population. We discuss in this paper the methodology of how we have applied an empirical investigation into user performance in order to derive the sequence of stereotypes that forms the basis of our modeling component's reasoning capabilities.

Key words. CALL, empirical analysis, ITS, NLP, parse disambiguation, student modeling, stereotypes

1. Introduction

We are currently developing the ICICLE system, a Computer Assisted Language Learning (CALL) system which instructs on English as a second language through the paradigm of a writing tutor (Michaud and McCoy, 2000; Michaud and McCoy, 2001; Michaud and McCoy, 2003; Michaud et al., 2000; Michaud et al., 2001). The name ICICLE represents 'Interactive Computer Identification and Correction of Language Errors.' The system's long-term goal is to employ natural language processing and generation to tutor students on grammatical components of their freeform written English. Of paramount importance of this goal is the correct analysis of the source and nature of user-generated language errors, and the production of tutorial feedback to student performance that is both correct and individualized to the student.

Our target audience for this work has been the native or near-native users of American Sign Language (ASL). Since ASL is a distinct and vastly different language from English (Baker and Cokely, 1980; Stokoe, 1976) and written English

[†] This work was completed while the primary author was a graduate student in the Department of Computer and Information Sciences at the University of Delaware.

(understandably) is a truly difficult language for many deaf individuals to master (Charrow, 1975; Charrow and Fletcher, 1974; Charrow and Wilbur, 1975; Quigley et al., 1977; Wilbur, 1977), we view the acquisition of written English skills to be a task in second language acquisition for these learners (Michaud and McCoy, 1998; Michaud et al., 2000). Because some deaf individuals achieve a high level of success with English while others struggle with basic elements of the language (Padden and Ramsey, 1998; Stewart, 2001; Swisher, 1989), and our system endeavors to accurately address the needs of as broad a spectrum of users as possible, user adaptivity through user modeling has been a primary focus of our system development efforts.

It has been said that a well-designed tutoring system actively undertakes two tasks: that of the diagnostician, discovering the nature and extent of the student's knowledge, and that of the strategist, planning a response (such as the communication of information) using its findings about the learner (Glaser et al., 1987; Spada, 1993). A user model typically serves as a repository for the information serving both of these two processes, representing what has been discovered about the learner and making that data available to drive the decisions of the adaptive system.

The diagnostician and strategist aspects of ICICLE enable it to process a sample of writing and interact with its user through a cycle of user input and system response. This cycle begins when a user submits a new piece of writing to review by the system[1]. The diagnosis module of the system performs an analysis on this writing, determining its grammatical errors. The user modeling component aids this determination by indicating the most likely diagnosis of student performance on the basis of what it has observed in the previous writing samples from this student. The 'strategist' part of the system then constructs a response in the form of tutorial feedback. This feedback is aimed toward making the student aware of the nature of the errors found in the writing and toward giving him or her the information needed to correct them. When the student makes those corrections and/or other revisions to the piece, it may be re-submitted for analysis and the cycle begins again.

Since the system is intended to be used by an individual over time and across many pieces of writing, these roles and the cycle they represent will be performed many times with any given user. At the same time, we expect the user to be changing as the learning process unfolds. We therefore envision a user model participating in this interactive cycle in several capacities. First, the analysis of user performance can feed data about the user's command of the language to the model. In this way, each iteration of the cycle provides more data to tune the model. Next, the model can serve up the data from previous interactions with the user to inform both the analysis and tutoring processes so that the most likely analysis of the user's text is selected and so that tutoring can be appropriately focused. Finally, the user model is dynamic and flexible, following the user as he or she learns. Since the user's mastery of the language is constantly changing, the model must attempt to provide

[1] The ICICLE system takes as input free-form, multi-sentence essays.

data which reflects the current moment. In some cases, this may mean that the model may need to predict aspects of the user's mastery which are not yet readily apparent in his or her language production, but which may be inferred from a combination of observed performance and knowledge about the typical learning patterns of the population. Because of this last characteristic of our user model, it is necessary for us to investigate a modeling architecture which allows for inferring data beyond what is directly observed.

1.1. AN INFERENCE-BASED USER MODEL FOR ICICLE

ICICLE's current implementation is a prototype application which uses a text parser to syntactically interpret potentially large samples of user-written text and provide simple feedback through 'canned' one-sentence comments on the errors it finds. The parser utilizes an augmented CFG grammar for Allen's TRAINS parser (a parser related to that which was presented in Allen (1995), version 4.0. Mal-rules, or 'buggy' rules, represent commonly-committed grammatical errors based on analysis of a corpus of writing samples from our user population (Suri, 1993; Suri and McCoy, 1993). With this augmentation, the system is able to recognize many morphosyntactic[2] errors. The coverage of this grammar was originally explored by Schneider and McCoy (1998) and is a focus of current evaluation.

One difficulty faced by the diagnosis process is that there are often multiple interpretations for a particular input sentence, some possibly indicating different errors. The current system selects between competing parses which span each utterance by choosing arbitrarily, with no selection criteria. Since determining the nature and cause of student errors is an integral step to deciding how to approach student instruction (Matz, 1982), the parser must be able to make principled decisions between these options.

To determine which of the parse possibilities best describes the student's actual performance, we have decided to augment our system's capabilities with a model of the student that indicates which grammar rules he or she is most likely to be using. These rules can be correct or incorrect, depending on the student's status in the acquisition of the grammatical concepts involved. We can then choose between structurally-differentiated parses by selecting the parse that uses the most likely rules.

We contend that one way to determine which rules a user is most likely to employ is to observe the user's correct (or incorrect) use of the various grammatical constructs. Essentially, we expect the user to use (Michaud and McCoy, 2001; McCoy et al., 1996):

- correct rules for grammatical constructs which he or she has successfully acquired
- both correct and incorrect (mal-) rules for those constructs currently being acquired (but not fully mastered at this time)

[2] Spanning both word morphology and sentence syntax.

– incorrect (mal-) rules for those structures which are beyond the user's current mastery of English

However, the system will have access to a finite amount of writing from the user, and it is not likely to cover all of the possible grammatical constructs about which the user modeling component desires to collect data[3]. Therefore, direct observation of the user through his or her interaction with ICICLE is going to reveal information about a large but *limited* (and most likely incomplete) number of grammatical constructs. Thus, these observations must be augmented. We contend that if we had a sequence of stereotypical acquisition states, we could extend the data we have on the user by adding information from the stereotypical state that the user's current acquisition status most closely resembles. Having a sequence of these states could only be possible if indeed there were a typical order in which grammatical constituents were acquired by second language learners. Support for the existence of such an order can be found in empirical studies on the acquisition of English (Bailey et al., 1974; Dulay and Burt, 1975; Gass, 1979; Larsen-Freeman, 1976; Krashen, 1982; Schwartz, 1998; Schwartz and Sprouse, 1996), where it is sometimes called a 'built-in syllabus' for second language (L2) acquisition (Corder, 1967; Higgins, 1995).

The intuition underlying our use of the stereotypical sequence of acquisition is that we can deduce through observation that certain grammatical constructions are acquired, being-acquired, or unacquired. We can then infer the acquisition status of a number of other grammatical constructs by selecting a stereotypical acquisition state which most closely matches our observations. This state would allow us to 'fill in the blanks' on the structures on which we have little or no data according to how well a typical learner at this stage has usually mastered these components of the grammar.

One 'tricky' aspect to inferring stereotypic information in our particular application is that we expect the user to change over time as he or she acquires the language. Therefore, while step i in the stereotypical acquisition sequence may best describe the user now, we expect that a step $> i$ will better explain their performance on a later writing sample. In order for this user model to work, we not only need to capture the stereotypical stages of second language acquisition and the status of all of the grammatical constructions at each stage, but we must be able to detect when the user has left stage i and has moved on to $i + 1$. Therefore, although our basic concept of stereotype is similar to the original concept introduced by Rich (1979) in that it represents a collection of users who typically share certain characteristics, our approach is unique because the system is not merely tuning a stereotype to fit a static individual, but is also dynamically switching the stereotype being used in order to follow the user along the path of acquisition.

[3] Recall from the Introduction that the input to ICICLE is free-form in nature, rather than solicited text which might be designed to explicitly steer the student into producing specific language structures.

In this paper, we concentrate on the development of the stereotypical stages of acquisition order. At each stage, we wish to determine which grammatical constructions are typically acquired, being acquired, or beyond the learner's current reach. We provide our method for empirically deriving this information from a corpus of writing samples representing written language from individuals at various stages of written English acquisition. We have proceeded from the assumption that an overall proficiency score assigned by experts in the evaluation of writing by learners of English as a Second Language correlates well with our concept of a 'level' of acquisition. We describe in the rest of this paper our efforts to distill from our corpus the desired information on sequential stereotypes in our domain so that we may implement a dynamic model which can supplement direct knowledge about the individual with information about the stereotypic learner.

1.2. SLALOM: AN ORGANIZATION ON GRAMMATICAL SPACE

Originally described in (McCoy et al., 1996), SLALOM (Steps of Language Acquisition in a Layered Organization Model) is the architecture we have proposed to capture these stereotypical stages of second language acquisition, and is one portion of the ICICLE user model. As introduced above, the overall user model design must capture the status of each of the grammatical structures of English as acquired, being-acquired, or unacquired. In this way, the model itself is essentially of an overlay design. The portion of the user model directly capturing observations of user performance is called MOGUL (Modeling Observed Grammar in the User's Language).

Operating in concert with MOGUL, the SLALOM architecture is an organization on the space of grammatical concepts which enables us to capture the stereotypical stages of acquisition of those concepts. This allows us to augment the MOGUL markings we can obtain from observations with inferencing capabilities based on our concept of successive stereotypical states in order to derive data which has not been recorded directly in the model.

A simple representation of the SLALOM architecture can be seen in Figure 1, which is meant for illustrative purposes only. The architecture consists of hierarchies of grammatical concepts called Knowledge Units (KUs). The bottom of each hierarchy contains those that are generally acquired first, with successively acquired units stacked on top of the earliest-acquired ones. In the figure, we show possible hierarchies for several different types of grammatical forms: Morphology (completely filled in), VP Forms, Sentence Forms, and Relative Clauses (each of which shows just one representative KU). Dashed horizontal lines represent 'layers' in the hierarchies, indicating KUs that are acquired at roughly the same time. In this illustration, +s 3rd person singular and 's possessive morphology markings are (hypothetically) acquired at about the same time as the auxiliary 'be,' SV and SVO sentence forms, and Subject Relative Clauses.

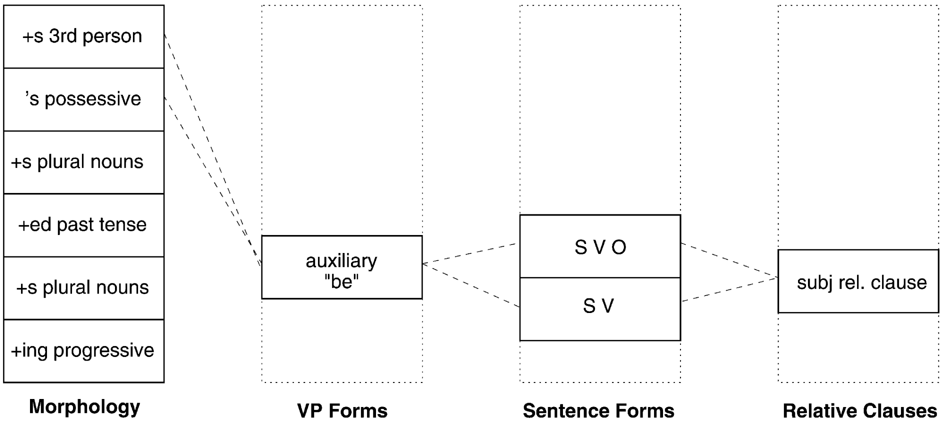

Figure 1. SLALOM: Steps of language acquisition in a layered organization model.

The KU units of the SLALOM architecture are abstract grammatical concepts, essentially 'bundles' of the grammatical rules in our parsing grammar[4]. Each KU represents a higher-level grammatical concept on which the MOGUL portion of the user modeling component stores the user's mastery[5]. For example, one KU mentioned earlier is Subject Relative Clauses. Bundled within this KU would be all of the rules from the parsing grammar which implement this concept. This includes not only those rules modeling correct execution of this type of relative clause, but also the mal-rules which realize the ways in which this structure is executed incorrectly by the learner population.

As introduced above, the KUs are connected to each other in SLALOM via two dimensions, represented vertically and horizontally in the figure. The first half of SLALOM's name, *Steps of Language Acquisition*, refers to how SLALOM captures the stereotypic linear order of acquisition. We have introduced how we represent this order graphically as occurring within constructed stacks or 'hierarchies'[6] of related KUs. As an example, we illustrate a Morphology hierarchy based on the findings of (Dulay and Burt, 1975)[7]. A given morphosyntactic KU is expected to be acquired subsequent to those below it, and prior to those above it, according to the natural order of a stereotypical learner from this particular L1 acquiring English[8]. The

[4] Section 3.1 discusses further how we have determined the relationship between grammatical concepts and our parsing rules.

[5] The term Knowledge Unit and the abbreviation KU are borrowed from (Desmarais et al., 1996).

[6] The term 'hierarchy' refers to the fact that certain KUs are learned before others.

[7] This is for example purposes only, since their morphology sequence is relatively simple. The sequences we actually use can be found in (Michaud, 2002).

[8] Although some research has shown high correlation between the acquisition orders of learners from different L1s, we wish to represent the most likely order possible and thus have restricted the model to representing learners from a specific L1, in our case ASL.

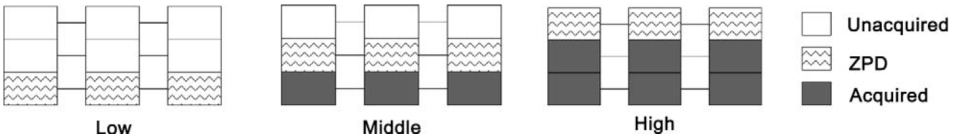

Figure 2. A simple view of stereotype progression in SLALOM.

figure's example represents the idea that '+ing progressive' is typically learned before '+s plural nouns,' which is typically learned before '+ed past tense.'

The hierarchies themselves are simply conceptual groupings of related structures which we have devised, so the order-of-acquisition relationships really exist at the global level rather than merely within the hierarchies. Therefore, we need to illustrate these relationships by coordinating the acquisition stages between the hierarchies. Dashed lateral connections in the figure represent the second dimension of relationship stored between the KUs in the model, namely that of concurrent acquisition[9]. We call these lateral groupings 'layers,' and the figure has one drawn in as an example[10]. This is the source of the *Layered Organization Model* part of SLALOM's name, referring to these layered groupings of KUs which essentially illustrate a progression through the acquisitional sequence representing those KUs being learned at certain stages of acquisition. In particular, we expect that students who are just beginning to learn written English will first struggle with those KUs in the 'first layer' and then will progress 'up' the hierarchies. A simple view of this progression can be seen in Figure 2. In reality, some KUs participate in more than one layer, which indicates the relative speed with which some concepts are acquired (i.e., KUs that span more than one layer take longer to master completely).

The uniqueness of the SLALOM concept is that it captures a sequence of stereotypes, not as a discrete set of characteristics as in canonical stereotype systems such as Grundy (Rich, 1979), or even a discrete sequence of stereotypic user mastery levels as in Mr. Collins (Bull et al., 1995), but rather implicitly within one interlinked architecture. Its layers each indicate the structures which may at one stage in the learning process form part of that stage's stereotypical ZPD. Those layers located 'below' in the model contain structures which are typically mastered before, while those layers 'above' are what is typically acquired later in the learning process, at a later stereotypical stage.

1.2.1. *Using SLALOM to Assist the Modeling of a User*

We describe here briefly how ICICLE's user modeling component will use the SLALOM architecture during the modeling process. Recall that each KU in the model represents two sets of grammar rules: those grammar rules which implement the

[9] This relationship may also be indicated within a hierarchy, not shown in this figure, to capture a partial ordering in which some structures in the same hierarchy are acquired together.
[10] The contents of this layer are not based on empirical findings and are for illustration only.

KU correctly and those mal-rules which represent errors a student may make in trying to use the KU before it is completely acquired. As ICICLE analyzes a piece of text written by a student, its user modeling component records the results of its analysis in the KUs which were used or attempted by the student in the text. A positive mark, or 'hit,' is recorded on a KU in the MOGUL component when rules that correctly implement a KU are used to parse the text; a negative mark, or 'miss,' is recorded when mal-rules associated with the KU are used. Since we expect the user to change over time, a user may at one point use only mal-rules for a particular KU, then move into a period of using both mal-rules and correct rules, and finally begin to use only correct rules. Thus the positive and negative marks on the KUs in MOGUL need to reflect current observations, and hits and misses may be retired after they pass outside of a certain window of the present.

From this information on the user's actual language production, we might conclude that a KU has been acquired if it has mostly positive marks (hits), or that it has not been acquired if it has mostly negative marks (misses). In addition, there may be some KUs that have more of an equal balance of hits and misses. This means that the user has sometimes implemented the KU successfully (using correct grammar rules) and sometimes incorrectly (using mal-rules). We argue that this latter set of KUs exhibiting significant variation corresponds to Vygotsky's *Zone of Proximal Development* (ZPD), essentially that subset of a cognitive skill which the learner is about to master (Vygotsky, 1986). Krashen's observation that at each step of language learning there is some set of grammar rules which the learner is 'due to acquire' (Krashen, 1982), and the fact that elements which are on the verge of being acquired vacillate between correct and incorrect applications (cf. (Ellis, 1994)), effectively reinforce the application of this concept to our domain. This notion of a ZPD along with our fully developed SLALOM model can be used to infer the status of KUs that do not have sufficient marks to make a judgment on the basis of actual observations.

Intuitively, at a specific moment in time, a learner's ZPD will be approximately one 'layer' in the SLALOM architecture, since the layers represent structures which are learned concurrently. As discussed above, the learner's current ZPD layer will progress over time, typically moving 'up' in the SLALOM architecture as learning progresses. In this way, the architecture approximates a sequence of stereotypical learner snapshots, indicating a typical progression in the acquisition of English.

The user modeling component of ICICLE can use the states in this progression to 'fill in the blanks' for the KUs in MOGUL with little or no direct data from the user. KUs which are at the same 'layer' as marked structures can inherit a similar marking; structures typically learned at earlier layers than acquired structures can be inferred as also acquired. This empowers the ICICLE system to make intelligent decisions on a user even when its knowledge of the individual is partial or even impoverished.

Empirical experimentation on a three-layered SLALOM prototype based on the linguistic acquisition sequences discovered in previous related work with deaf individuals (see below for references) has illustrated the potential of the SLALOM architecture working in concert with the MOGUL performance history to reflect a full image of a user who is part of a learning community and yet possesses individualistic learning traits; these results are discussed in (Michaud and McCoy, 2003). Given the exhibited strength of the model design even without data specific to our population, we have undertaken steps to further develop the SLALOM architecture to add to its differentiation power and refine its granularity.

1.2.2. *A Sequence of Acquisition*

The primary focus of the work described in this paper is to 'populate' the SLALOM architecture—to determine where the Knowledge Units of the model should be placed in our representation of stereotypic acquisition order. Recall from Section 1.1 that there is empirical support for such an order. However, although there is some support for the theory that this order may be universal regardless of first language (L1) (Dulay and Burt, 1975), at least in the case of morphology acquisition, few studies propose an order for the entire body of grammatical structures that we wish to cover in SLALOM and none addresses our learner group in particular[11]. Related efforts to establish orders of acquisition include that described by (Pienemann and Håkansson, 1999), who investigated several empirical studies on the acquisition of Swedish. Their work, like ours, covered both morphological and syntactic forms, and their review of the existing empirical work supported a basic outline of a prescribed order in which aspects of language are acquired.

1.3. EMPIRICALLY DEVELOPING THE SEQUENCE

In the rest of this paper we will describe how we have worked toward the empirical derivation of a stereotypic acquisition order for our own user population, with the first language (L1) of American Sign Language. We sought to identify: (1) which aspects of English are mastered in what order, and (2) what groups of items are learned around the same time. This information will enable us to complete the implementation of the SLALOM architecture.

Our empirical work centers on a thorough examination of a corpus of 106 samples of writing by deaf individuals at various levels of English proficiency. The collection of these samples is described in (Suri, 1993; Suri and McCoy, 1993). The total length of this corpus is 1,793 sentences[12].

[11] Although the studies by Quigley et al.—summarized in Quigley et al. (1977)—did perform a fairly extensive analysis of deaf acquisition of English, the studies predate the acknowledgment of ASL as a language and therefore do not take into account the Ll background of the learners at all. We used this data in a prototype SLALOM implementation, but since the language background of the American deaf population is very heterogeneous and we wish to capture specifically characteristics of those with the Ll of ASL, their results are less salient to the ICICLE project than our own studies explicitly focusing on ASL natives.

[12] The average sample length is 16 sentences; there is a great deal of variation in length from sample to sample.

The corpus has been divided into groups on the basis of an overall assessment of the writing, described later in this paper. We hypothesize that writing samples receiving low assessment scores are representative of an individual who has not yet acquired very many of the KUs in our model; those with higher scores, on the other hand, are further along in the acquisition process. Thus, our methodology has been:

1. To analyze each piece of writing to get an indication of the writer's mastery of various grammatical structures (KUs). This analysis is described below.
2. To divide the samples into groups representing low-to-high acquisition states of English, using a method independent of the analysis in the previous step.
3. To verify that the samples within a particular group do indeed all capture similar language mastery characteristics.
4. To determine the set of *acquired/unacquired/ZPD* markings on each of the KUs in our user model for each group, using a statistical analysis of the grammatical performance of writers in that group.

Our initial analysis concentrated on user errors.

2. Profiling User Errors

Both Corder (1967) and Glaser et al. (1987) have made statements that the clue to characterizing the proficiency level of a learner is to study the systematic errors he or she commits. In our first efforts to establish profiles of how learners perform at different levels of acquisition, we concentrated on learner errors in an attempt to characterize the 'error profile' of each level represented by our corpus. It was our goal to discover if a progression of such profiles could be determined as representative of the marching frontier of acquisition, understanding that as a structure becomes the focus of the linguistic Hypothesis Testing which occurs at this frontier, the number of errors made on that structure may increase, and that number would drastically decrease once the structure is acquired. Therefore, data indicating errors made by beginning learners which are not made by higher-level learners would illustrate those linguistic structures which are mastered after the lowest levels; conversely, errors which are not made until higher levels may indicate structures which are avoided by lower-level learners until they become the focus of Hypothesis Testing at the level where the errors first appear.

2.1. PREPARATION OF THE CORPUS

Before an analysis of the writing samples could be done, the samples needed to be tagged by hand to mark the errors they contained. An initial analysis of the corpus had been performed earlier in the ICICLE project. This preliminary study of our sample corpus yielded a taxonomy of language errors which are typical of this user population and which provided our initial directions in developing the mal-rules to add to our grammar (McCoy et al., 1996; Suri and McCoy, 1993). During this analysis the samples were annotated according to the errors occurring in each sentence. Our current study uses this first analysis as a starting point.

2.1.1. *Developing a Set of Error Codes*

While our initial annotation of the corpus was sufficient for its intended purpose in our early explorations of the students' errors, it proved to be too informal for the task of developing the stereotypical acquisition sequence. The original tagging of the corpus of writing samples had been performed by multiple individuals using a list of error codes that had a common source with the taxonomy mentioned above; but these error codes were frequently ill-defined and often expanded in an ad hoc fashion as new shorthand codes were added to handle errors which were not satisfactorily covered in a previous iteration. As a result, the initial collection of tagged samples was largely inconsistent, with different coders adopting different 'styles' according to their interpretation of the meanings of the tags. Some coders even disagreed with their own tags when they were later reviewed.

These differences and disagreements were further complicated by other challenges inherent in the coding process, such as the difficulty of determining the writer's intent in ambiguous sentences, different errors whose surface realizations were identical and thus could be confused, and the interactions of multiple errors in a sentence (Suri, 1993). In order to standardize the tagging process, we decided to evaluate and revise the current list of error codes and develop a *coding manual* to contain formal definitions and explanations of each error code.

The original list of error codes was first examined and compared against the corpus. Several codes were discarded (if unsupported by instances in the corpus) or subsumed into others (when a more general definition was preferred), while new codes were generated to cover errors that had not yet been addressed. The coding manual which resulted from these revisions now contains 68 error codes, some of which are shown in Figure 3. While some of these codes are fairly straightforward, such as subject/verb agreement errors (`sv`), others required detailed explanations in the manual. The term 'dummy subject' was borrowed from our grammaticist's rule descriptions to indicate non-referential subjects such as extraposed *it* (It is nice to see you) and existential *there* (There is a rabbit on your chair). Some of the codes refer to errors which are fairly unique to our user population—for instance, the usage of *here* or *there* as a pronoun referring to a place rather than an adverb (I like here/Here is nice).

`id`	Incorrect Determiner	`mds`	Missing Dummy Subject
`md`	Missing Determiner	`ids`	Incorrect Dummy Subject
`sv`	Subject/Verb agreement	`mo`	Missing Object
`ht`	Here/There as a Pronoun	`bh`	Be/Have Confusion
`nf`	Noun Formation	`ii`	Incorrect Intensifier

Figure 3. Example of codes from our error coding mannual.

Many of these codes focus on syntactic errors that the system could be expected to recognize with its sentence-level interpretations. Each mal-rule in the parsing grammar is represented by an error code in the manual. Other error codes capture semantic or discourse-level errors that the system is not currently capable of recognizing. We have included these error codes in the manual for completeness in order to aid the development of future versions of the system.

2.1.2. *Tagging the Corpus*

The process of applying these error codes proceeds as follows. The human coder reads through each writing sample one sentence at a time[13]:

(1) *So two of us called cab and brought us home.

The coder must decide how the sentence should be interpreted, and if there are errors, must determine what the 'corrected' version—i.e., in Standard English—would be. This version of the sentence is then recorded. (Emphasis has been placed on the inserted words in these examples only; the actual coding is performed in a plain text file and there is no emphasis on the corrected portions.)

(2) So two of us called a cab and it brought us home.

Given this interpretation, the error codes from the manual are then listed before the correction to indicate what errors were found by the coder, in the order in which they occurred in the sentence. In the case of example (1–2), the errors the coder found were *missing determiner* (`md`) and *missing subject* (`ms`):

(3) (`md ms`) So two of us called a cab and it brought us home.

Essentially, the coders iterated through the sentences and processed each in a linear fashion from the beginning of the sentence to the end, asking two questions at each step through the sentence's syntax:

1. Is there an error at this point in the sentence?
2. If yes, what is the error according to the manual?

Because of the variability introduced by human interpretation of the underlying meaning behind complicated, very 'buggy' sentences, it turned out that the answer to Question #1 introduced more problems than we expected. This is discussed further below.

2.1.3. *Testing Inter-Coder Reliability*

After the completion of our formal coding manual, we sought to illustrate that the explanations enabled multiple coders to approach the task described above in a consistent, reproduceable fashion. We randomly selected 20% of our corpus (23 writing samples in all) and distributed them to two coders to be tagged.

[13] Examples in this paper have been taken from our actual annotated corpus.

Once the two coders had both marked the same test samples, we needed a method for determining whether or not they had separately identified the same errors for each sentence. This would verify that our coding manual had standardized the coding practice sufficiently to allow the task to be split up between multiple individuals without fear that individual variation would make the results incomparable. Determining the level of coder agreement, however, was complicated by the fact that many sentences were far more complicated than (1), so there were often several different errors identified in a single sentence—and in many cases only some of those errors were the same between the two lists of codes, while others were different.

A further complication arises from the flexibility in the coding task. Recall that at each step in the sentence, a coder has the freedom to judge *whether or not an error has occurred.* Even when the judges agree on an interpretation of a sentence, it is possible to disagree on whether a phrase contains an error. This can result in error code sequences of differing lengths. Take for example the following sentence:

(4) *Those who argue that it will have less hazing incidents here on campus if the abolishment of fraternities and sororities are done.

Each coder reviewed this sentence, determined corrections to be made, and wrote a corrected version of the sentence, recording the sequence of error codes corresponding to the errors they each corrected:

(5) [*Coder 1*] (`mds ids bh ii sv`) There are those who argue that there will be fewer hazing incidents here on campus if the abolishment of fraternities and sororities is done.

(6) [*Coder 2*] (`mds ids sv`) There are those who argue that there will be less hazing incidents here on campus if the abolishment of fraternities and sororities is done.

One of the differences between these two interpretations is introduced via the 'gray areas' of grammaticality where even native speakers' judgments may differ, as in the case where in (5) Coder 1 marked that it should be 'fewer . . . incidents' while in (6) Coder 2 found 'less . . . incidents' to be acceptable. Another reason was human error, as when Coder 1 marked a *be/have error* (`bh`) in (5), when the error was only created when the *it* was changed to *there*, and the coding manual instructions specify that errors which are created only by other corrections to the sentence should not be marked.

Our problem was how to determine the agreement between two coders in light of the differences in code sequence length like those between (5) and (6). It is clear from this example that a simple one-for-one comparison of error code sequences should not be performed. A simple comparison would align the two sequences in this fashion:

```
(7) (mds|ids|bh |ii|sv)
    (mds|ids|sv)|  |
```

However, inspection of the original corrections provided by the coders indicates that the *be/have error* (bh) marked by Coder 1 in (5) is not a disagreement with the *subject/verb disagreement* (sv) marked by Coder 2 in (6). Rather, the correspondence between the two sequences is:

(8)	(mds	ids	bh	ii	sv)
	(mds	ids			sv)

We therefore needed to devise a method for comparing the coding judgments of two individuals that took into account these unequal length strings where 'gaps' in one string or another—essentially disagreements on *whether an error had occurred here*—were accommodated.

Note that we could have chosen to have coders indicate the part of the sentence to which each error code pertained, for example by underlining as is seen in Examples (5–6). Intuitively, this would have provided more information about which codes corresponded to each other between the two sets. However, in reality the following difficulties remain:

- How should the coder define the span of an error code to underline? Does it apply to just the word(s) which must be changed, or to the entire constituent whose grammaticality is affected by the error? Different coder perceptions could result in overlapping but unequal boundaries defining the affected segment of the sentence, complicating the matching process[14].
- When the two coders make different changes to the sentence to 'correct' the same error—modifying one or more different words to render the sentence in Standard English[15]—the error codes which refer to the same problem in each string may be reflected in non-overlapping spans, which would not be considered marking the same error if these spans were used to determine which error codes correspond.
- There are cases when multiple errors occur within the same boundaries. In this case, it is still possible for one coder to record more error codes than the other, in which case we are still faced with the necessity of determining an alignment as found in (8) on page 16.

Because of these considerations, we felt that using location information in our coding process would not simplify our inter-coder comparisons.

2.1.4. *An Adapted Algorithm for String Comparison*

Our comparison task, therefore, was faced with two sequences of error codes, often of different lengths, usually containing between three and ten error codes apiece.

[14] In a similar coding task by Carletta et al. (1997) this difficulty was also encountered and had the authors were forced to discard any coded instance where there was disagreement on the beginning and end of the span involved.

[15] This is often the case when sentences are extremely ungrammatical and the coders are forced to make larger intuitive leaps toward a reconstruction.

We desired to determine the level of agreement between them, but the differing lengths required that we devise a method to posit gaps in order to line up the sequences in the way that most likely reflected the actual correspondence between the two coders' decision processes.

To address this task, we borrowed a page from bioinformatics research, which has developed many algorithms to compare two strings of DNA in order to determine their similarity. These tasks treat the DNA strands as strings of nucleotide characters and allow for the fact that a pair of strings may have some nucleotides which match, but sometimes there will be 'substitutions' (one string will be GC$\underline{\text{A}}$ when the other is GC$\underline{\text{T}}$) or 'gaps' (one string has a nucleotide which has no correspondent in the other).

Because of the similarity between our problem and this DNA matching process, we decided to apply the Smith-Waterman algorithm (Smith and Waterman, 1981) to our task. This algorithm attempts to determine the correspondence between two DNA strings by searching for an alignment minimizing the number of 'mutation events' required to convert one string into another. This is accomplished by computing a matrix of alignment scores where nucleotide matches are rewarded with higher scores and substitutions or gaps are penalized by depressed scores. It is described in (Nicholas et al., 1998) as a recursive equation which for global string alignment specifies the value of each location in the alignment matrix SW as:

$$\mathrm{SW}_{i,j} = \max\left\{\begin{array}{l} \mathrm{SW}_{i-1,j-1} + s(a_i, b_j) \\ \mathrm{SW}_{i-k,j} + g_j \\ \mathrm{SW}_{i,j-k} + g_i \end{array}\right\} \tag{i}$$

Each location $\mathrm{SW}_{i,j}$ in the matrix contains the Smith-Waterman score for the partial alignment ending at residue i of sequence a and residue j of sequence b. This is calculated by seeking the best extension of a previous partial alignment (whose score can be found in a part of the matrix previously calculated, the first term in each equation). These extensions involve either matching the next residues a_i and b_j from each sequence (whose level of similarity, or match, is scored by the function $s(a_i, b_j)$) or introducing gaps in one string or another, the penalty for which is given by the terms g_j and g_i.

Table I illustrates the tuned values of the penalties and rewards in the algorithm in order to intuitively reflect the relative importance of matches, mismatches, and gaps in our particular problem[16]. We wished to match identical codes when possible, so the reward for a match was fairly high, while gaps were penalized but not severely. Using these values, we implemented the algorithm in a C program and set it up to automatically process the pairs of ASCII files containing coder-assigned tags for the 23 test compositions. An example output using the codes from example sentences (5) and (6) can be seen in Figure 4. The program marked gaps with an asterisk for each 'empty' code.

[16] These values were derived through trial and error.

Table I. Scoring values in our smith-waterman implementation

Term	Points
Match reward $s(a_i, b_j)$ where $a_i = b_j$	+5
Mismatch penalty $s(a_i, b_j)$ where $i \neq b_j$	0
Start gap penalty g_0	0
Extending gap penalty g_1 such that $g_j = g_i = g_0 + k * g_1$	−1

```
mds     ids     bh      ii      sv
mds     ids     *       *       sv
```

Figure 4. Smith-Waterman alignment program output.

Tested on a randomly selected 10% of the sentences in the overlapped portion of the corpus, this program produced alignments consistent with human judgment 31 of 36 times[17]. This is a very satisfactory performance result, particularly when one considers that a human judge can take into account the corrected sentences provided by each coder in order to identify each coded instance, while the automated algorithm only has access to the error code sequences.

2.1.5. *Reliability Results*

The output of our alignment program provided us with correspondences between the two sets of codes for each sentence in the corpus. In the alignments that it found, there were several possibilities for each paired location in the two strings:

1. The paired codes could be identical, which indicated the coders agreed both that there was an error in this location and what the error was.
2. The paired codes could be different, showing that although both coders had found an error here, they disagreed on how that error should be tagged.
3. One error code in one sequence could be paired with a gap in the other, in which case one of the coders had identified an error that had been ‘ignored’ by the other[18].

Of these possibilities, the last was the largest source of disagreement between our coders. Notice that this ‘gap’ case indicates a difference in the grammaticality judgments between the coders, a difference in their semantic or discourse interpretations of the sentence, or human error. The instances of gaps were evenly distributed

[17] Of the five that were misaligned, four were sentences which involved serious enough errors that the coders diverged significantly in their ‘correct’ reconstructions, leading to long and greatly different strings of error codes. One of these four involved a series of error codes so long that it was exceedingly difficult to determine the correspondence by hand.

[18] While this appears to be the primary cause of gaps, they can also be introduced through coder error is in the case of the additional `bh` code in (5).

Table II. Agreement statistics on overlapped coding

432	Errors marked by both coders
265	Agreed errors
61%	$P(A)$ (bare agreement) across all instances

between the two coders[19]. As discussed in Section 2.1.3, our goal in testing the coding manual was to determine if the coders assigned the same code in instances where they both intended to mark an error. Since both coders were equally qualified native speaker judges and either human interpretation could be correct, the differences in grammaticality judgments or interpretations could not be a factor in determining whether or not the coding manual was sound. Our statistics of manual reliability, therefore, do not count these 'gaps' as disagreements[20].

Table II shows our agreement statistics on the 'non-gap' cases[21]. There were 432 of these cases. The $P(A)$ statistic reflects the percentage of these cases where both coders assigned the same error code tag.

The figure of 61% agreement is fairly satisfactory if one takes into account the use of the high number—68—of categories (error codes) in our coding task, and it is sufficient for the error analysis tasks described in this paper because of the acceptability of either coder's analysis of the errors occurring in a sentence.

Recall, however, that our eventual goal in this work is to determine a sequence of language acquisition over linguistic elements that our parser can recognize. We would therefore prefer to have higher agreement and reliability over those errors the parser would be able to identify. Upon further inspection of our codes, we found that only 47% (33) of the error codes fell into this category, having been implemented as mal-rules in our parsing grammar. Many of the remaining half had been excluded because they were based on discourse-level information currently unavailable to the parser, which focuses on sentence-level syntax and is unable to track other errors such as inconsistent use of tense, person and number of referential pronouns, or semantic distinctions between content words. Since we intended to extend our work by comparing human choices against parse selections (see Section 3.3), and the parses would only contain those errors implemented in the grammar, we reran our agreement analysis looking only at those instances where both coders had used error codes from that grammar. The results are shown in Table III.

Carletta (1996) points out that agreement statistics on this type of task should also take into account how much of the agreement between coders is due purely to

[19] During the alignment process, Coder 1's code sequences had 112 gaps introduced into them, while Coder 2's sequences had 106 gaps inserted.

[20] We are therefore examining only 432 of the 794 total instances of error codes in the corpus where at least one coder indicated an error. In addition to the gaps that were inserted by the alignment program, instances involving the code 'none,' meaning no error was found in the sentence, were also not considered because 'none' is equivalent to intentionally positing a gap.

[21] We earlier reported in Michaud and McCoy (2001) and Michaud et al. (2001) a $P(A)$ of .81 for these cases; this figure was found to be incorrect because of a programming error and has been discarded in favor of the results discussed here.

Table III. Agreement statistics when both codes were implemented in the parsing grammar

166	Both marked errors are implemented
140	Agreed implemented instances
84%	$P(A)$ for both implemented
82%	*Kappa value* for both-implemented cases

chance. Therefore, we also calculated her proposed *Kappa Statistic* to factor out this chance agreement. She defines Kappa as:

$$K = \frac{P(A) - P(E)}{1 - P(E)} \tag{ii}$$

The term $P(A)$ refers to the level of actual agreement between the coders (84% in this case), while $P(E)$ refers to the probability that the two coders agreed by chance. This was calculated by looking at the frequency $f()$ with which the two coders used each error code e_i across the coding instances being examined. The function $f()$ for a specific coder j over the N instances being examined is given as:

$$f_j(e_i) = \frac{\text{count}_j(e_i)}{N} \tag{iii}$$

Note that $\text{count}_j(e_i)$ equals the number of times the coder j marked the code e_i in these instances. The denominator indicates how many of these instances are being examined.

Given this definition of the frequency function $f_j()$, the chance agreement is calculated as:

$$P(E) = \sum_{i=1}^{33} f_1(e_i) * f_2(e_i) = 0.05992 \tag{iv}$$

As was reflected in Table III, the Kappa value we obtained with this calculation was .82, which satisfies Carletta's standards for strong conclusions ($K \geqslant .8$).

Confident that our coders were providing our study with similar 'human intuitions' on all codes and strong agreement on those implemented in the parsing grammar, we determined that our coding manual was reliable enough that we could divide our remaining samples between the coders and have one coder code each of the remaining samples, while retaining a reasonable belief that the results we obtained would reflect a consistent and reliable view of the errors committed by the writers.

Subsequent to our satisfactory evaluation of the coding manual, the remaining 80% of the corpus was divided between the two coders and all of the sentences were tagged with error codes. Since the test portion had been double-coded and the judgments were similar but not identical, a random-choice program selected which coder's interpretation we would use for each of those samples to complete our fully-annotated corpus.

2.2. DETERMINING LEVELS OF PROFICIENCY

Our overall goal was to determine an error profile for writers at each of various levels of proficiency, illustrating how the errors committed by a writer tend to change as his or her mastery of the language develops. In order to do this, we had to divide the writers represented by the samples we had obtained into groups representing different proficiency levels. The proficiency levels had to be determined independently of the identification of errors, in order for us to be able to reliably investigate any relationship that might exist between these errors and the level of the writers.

It was therefore important that these judgments of proficiency level be uninfluenced by the coding process previously described. In order to obtain independent, holistic ratings representing levels of proficiency, we chose to have the proficiency levels be determined by judges experienced in the assessment of proficiency in English as a Second Language. For this phase of our work, we collaborated with four instructors at the University of Delaware's English Language Institute (ELI), a program which provides foreign students with English language courses. This program serves around 1800 students each year, and in 2001 became one of only 24 English programs in the country to achieve accreditation by the Commission on English Language Program Accreditation (CEA). The instructors have a high level of expertise in language assessment and were very willing to assist us with our task.

The proficiency-scoring system which they applied to our writers is from the national Test of Written English (TWE), a free-form essay-scoring test which ELI uses for placing new students into the mandatory English classes which foreign students at the University must pass in order to assume certain Teaching Assistant duties. The TWE scores range from 1 to 6 and they are meant to represent a holistic judgment of the student's overall level of English mastery.

In order to obtain reliable scores on the samples of our corpus, the four ELI volunteer judges were each assigned a random portion of the samples so that each sample was read by two judges. If the judges disagreed on the rating, a third arbitrated so that each final score represented either a consensus or a majority. Figure 5 illustrates the distribution of scores among the 106 samples in our corpus.

An unexpected result of the TWE scoring of the writing samples was that they were not well distributed; in fact, as shown in the figure, 95% of our samples were concentrated in only three levels. One reason for this is that one of our largest sources of samples was from an English language entrance exam at a school for deaf students. The samples we obtained from this source were those that were on the borderline of passing this entrance requirement. These samples, therefore, are all of a similar level of proficiency.

This may have affected some of the results we later obtained while trying to find significant distinctions between samples of different levels. In future efforts to obtain samples, it is clear that we will want to focus on obtaining ones which represent a broader spectrum of proficiency.

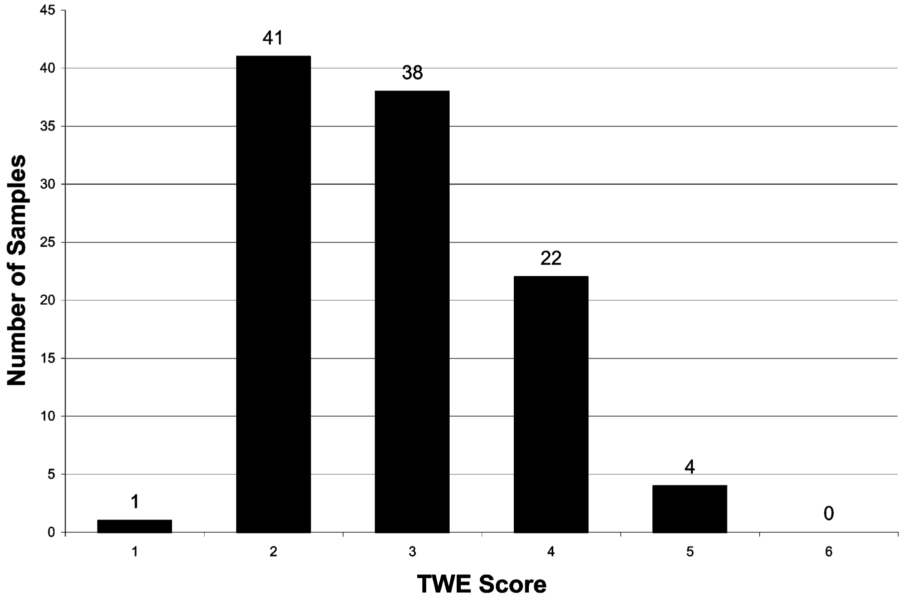

Figure 5. Distribution of graded samples among the TWE scores.

2.3. INITIAL EXPLORATION: STATISTICAL CLUSTERING

Following the stratification of our samples into these levels of competence, we had the following data prepared for each:

- The writer's level of English proficiency as judged on the TWE scale.
- The number of times each of the 68 error codes had been marked in the sample.

These data were prepared for an initial exploration to confirm that the profiles of errors committed by the learners at different levels of ability were distinguishing characteristics of those groups[22]. Specifically, we wished to determine if learners in the same proficiency group committed similar errors, while those in different groups committed different kinds of errors.

We decided to approach this question by first clustering the samples together based on the errors that had been identified. Statistical clustering algorithms had previously been applied to identifying groups of similar users (essentially, stereotypes) by Milne et al. (1996). If statistically-derived clusters based on the error profiles divided the samples in a similar fashion as did the ELI scorers, this would confirm a correspondence between what errors were committed and what score the sample was given. This would in turn be very strong evidence of the existence of the stereotypes we sought, since it would verify that students placed together

[22] If they were not, then our study would not support the notion of a stereotypical acquisition order. This would mean that it would not be possible to derive stereotypes in the manner we suggest, or even possibly put into question whether the stereotypes actually exist.

in the same group by their TWE score had committed similar errors, and that those errors were different from those made by students in other groups.

Since our samples varied greatly in length, ranging from 2 to 58 sentences long, we normalized the error counts by dividing them by the number of sentences in the sample to obtain an *average count per sentence* figure. Each sample was thus represented by a vector where each element in the vector represented an error code and the number represented the average number of times that code occurred per sentence in that sample. Using the statistical application SAS, we applied clustering algorithms to our data, instructing the program to form groups or clusters of writing samples which were minimally 'distant' (different) from each other, and so should represent samples which have the same errors in approximately the same magnitude. We report here the results of applying Ward's Minimum-Variance method (Ward, 1963) to our error vector data.

This clustering algorithm begins by assuming that each 'observation,' or vector of data (in our case, the set of 68 normalized error counts), forms a cluster C_K where the number of observations $N_K = 1$. It then recursively iterates through the clusters and joins together those which are closest according to a calculation of distance. At the heart of the distance calculation when considering clusters C_K and C_L are the mean vectors $\bar{x}_K$ and $\bar{x}_L$, the means of the values in the cluster so far. Of the many different clustering algorithms applied to this set of data, while several obtained similar results, we choose the results from Ward's because this distance metric most closely represented how we wanted to measure similarity between sets of error code occurrence figures.

Since we wanted to compare the clusters to the six TWE scores, we also requested that the statistical program 'stop' the clustering process at some point where the granularity of the clusters approximated that of the TWE score groups, for a total of five final clusters. Figure 6 illustrates the relationships between the TWE scores 1-6 and the five clusters we obtained. The rows each represent one TWE score grouping, and the numbers in that row indicate the number of samples from that group that were placed in each of the five clusters. These results were first discussed in (Michaud et al., 2001).

Despite our sparse data problem with only 5% of the samples occurring at TWE levels 1, 5, and 6, there is a clear trend with lower and higher proficiency levels showing a preference to different clusters, overlapping in Cluster 2. We concluded from this that we had obtained some evidence to support our belief that the error sets committed by learners at different levels underwent an overlapping but changing progression from level to level. This gave us confidence to go on to the next step, which was to determine whether we could establish the nature of the relationship between the error code annotations and the TWE scores. We wanted to determine whether or not the errors committed by a learner would help us identify his or her TWE score (as a marker of where the learner was in the acquisition process). By identifying the different errors committed by different score levels, we hoped to take a step toward identifying the stereotypical sequence of acquisition.

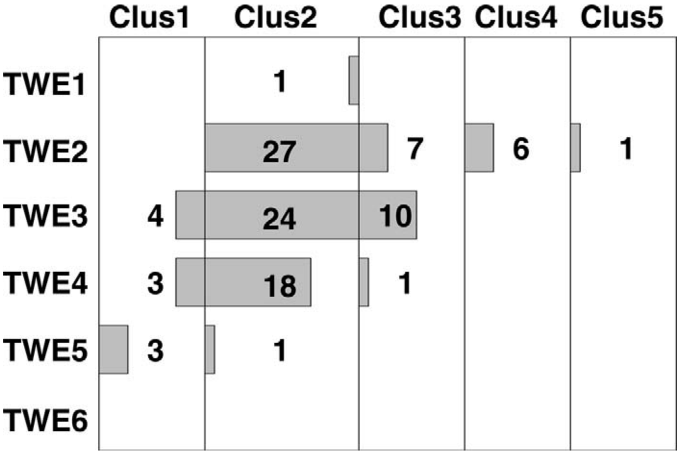

Figure 6. Distribution of levels of proficiency across Ward clusters.

2.4. IDENTIFYING THE ERROR PROFILES

Essentially, our next investigative step sought to identify what precisely the ‘error profiles’ were for each of the levels represented by our corpus—specifically, which errors distinguished each proficiency group by occurring significantly more often in that group than in any other. With this data, we could potentially identify what other errors we could expect a user to commit, given the set of errors they have committed and the similarity of that set with one of these profiles.

Because we could not depend on the tiny number of samples in TWE levels 1 and 5 to independently give us reliable results, we concentrated at this stage on a ‘collapsed’ TWE score which we renamed *low, middle*, and *high.* In this modified score, the levels with poor representation were collapsed with the central three. This is illustrated in Table IV.

For this step of our analysis, our tool of choice was Multivariate Analysis of Variance (MANOVA). This was selected because what we wished to test was the relationship between the number of times a user committed each error and the TWE score the user had been assigned. Therefore, one dependent variable was the collapsed TWE score, and our analysis tested whether it could significantly ‘predict’ the number of times a user committed a given error. Once again we wanted to compensate for the variable lengths of the essays, but we decided that the normalized values that divided error counts by the number of sentences were a somewhat simplistic method of accounting for length variation. Since the Analysis of Variance

Table IV. Collapsed levels of proficiency for MANOVA analysis

TWE score	1	2	3	4	5	6
Collapsed TWE	Low		Mid	High		

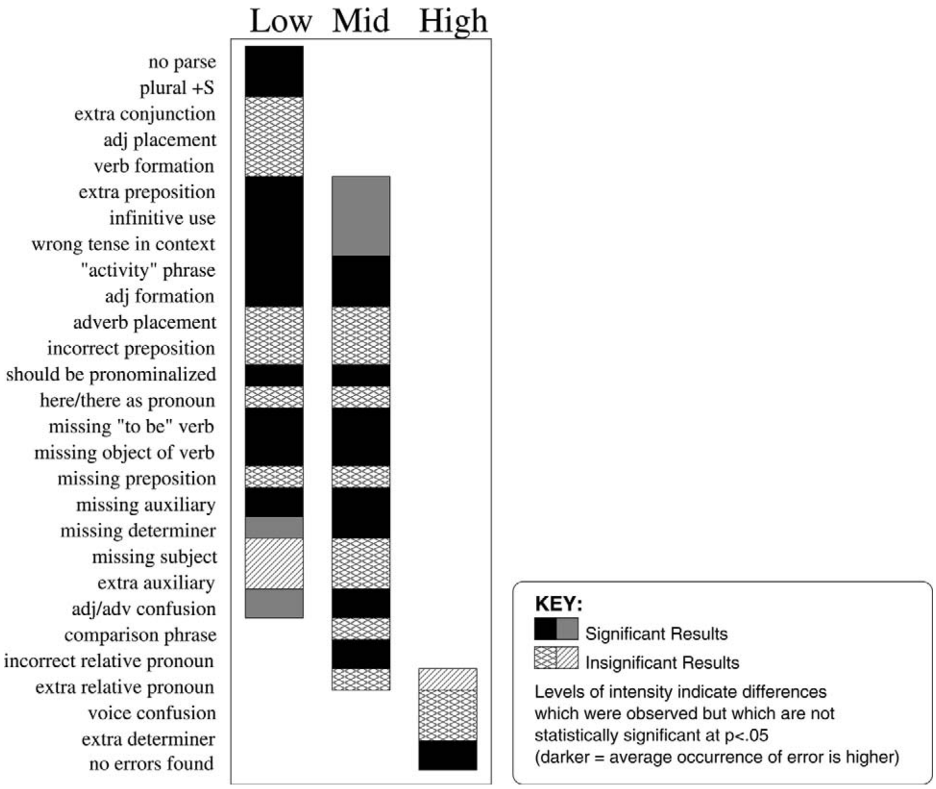

Figure 7. Illustrating the errors each level is most likely to commit.

test allows easily for multiple dependent variables, we used the length of the sample as a second dependent variable in our tests[23]. The results we obtained were previously reported in (Michaud and McCoy, 2001).

These results indicated that many of our error codes did occur at different magnitudes between *Low*, *Middle*, and *High* samples. This is illustrated in Figure 7, which shows the results on a subset of the 47 error codes for which we obtained discernible results.

In the figure, a bar indicates that this level of proficiency committed this type of error more frequently than the others. If two of the three levels are both marked, it means that they both committed the error more frequently than the third, but the difference between those two levels was unremarkable. Solid shading indicates results which were statistically significant (with an omnibus test yielding a significance level of $p < .05$), and intensity differences (e.g., black for *extra preposition* in the low level, but grey in the middle level) indicate a smaller difference that

[23] We did analyses in parallel using both word count and sentence count as measures of sample length, getting nearly identical results both times. The results displayed in this work reflect the analysis done with the length measured in sentences.

was not significant. In the example we just mentioned, the low-level writers committed more *extra preposition* errors than the high-level writers with a significance level of $p = 0.0082$, and the mid-level writers also committed more of these errors than the high-level writers with a significance of $p = .0083$. The comparison of the low and middle levels to each other, on the other hand, showed that the low-level learners committed more of this error, but that the result was strongly insignificant at $p = .5831$.

The cross-hatched and diagonal-striped results in the figure indicate results which did not satisfy the cutoff of $p < .05$ for significance, but were considered both interesting and close enough to significance to be worth noting. The diagonal stripes have 'less intensity' and thus indicate the same relationship to the cross-hatched bars as the gray does to the black—a difference in the data which indicates a lower occurrence of the error which is not significantly distinguished (e.g., high-level learners committed *extra relative pronoun* errors less often than mid-level learners, and both high- and mid-level learners committed it more often than the low-level learners), but, again, not to a significant extent.

Notice that the overall shape of the figure supports the notion of an order of acquisition of structures because one can see a 'progression' of errors from level to level. Very strongly supportive of this intuition are the first and last errors in the figure: 'no parse,' indicating that the coder was unable to understand the intent of the sentence, occurs statistically more often at the lowest level than at the other two levels, while 'no errors found' was significantly most prevalent at the highest level. These two were our strongest results.

Other data which is more relevant to our goals also presents itself. The lowest level exhibited higher numbers of errors on such elementary language skills as putting plural markers on nouns, placing adjectives before the noun they modify, and using conjunctions to concatenate clauses correctly. Both the low and middle levels struggled with many issues regarding forming tenses, and also exhibited 'ASLisms' in their English, such as the dropping of constituents which are either not explicitly realized in ASL (such as determiners, prepositions, verb subjects and objects which are established discourse entities in focus, and the verb 'TO BE'), or the treatment of certain discourse entities as they would be in ASL (e.g., using 'here' as if it were a pronoun). While beginning learners struggled with more fundamental problems with subordinate clauses such as missing gaps (failing to leave a gap for the relativized constituent), the more advanced learners struggled with using the correct relative pronouns to connect those clauses to their matrix sentence. Where the lower two levels committed more errors with missing determiners, the highest level among our writers had learned the necessity of determiners in English but was overgeneralizing the rule and using them where they were not appropriate. Finally, the upper level learners were beginning to experiment with more complex verb constructions such as the passive voice. All of this begins to draw a picture of the sequence in which these structures are mastered across these levels.

2.5. DISCUSSION

While Figure 7 is meant to illustrate how the three different levels committed different sets of errors, it is clear that this picture is incomplete. The low and middle levels are insufficiently distinguished from each other, and there were very few errors committed most often by the highest level. Most importantly, many of the distinctions between levels were not achieved to a significant degree.

One of the reasons for these problems is the fact that our samples are concentrated in only three levels in the center of the TWE spectrum. We hope to address this in the future by acquiring additional samples. Another problem which additional samples will help to solve is the sparseness of data on error occurrence. Across our 106 samples and 68 error codes, only 30 codes occur more than 25 times in the corpus, and only 21 codes occur more than 50 times. Most of our insignificant differences come from error codes with very low frequency, sometimes occurring as infrequently as 7 times.

What we have established is promising, however, in that it does show statistically significant data spanning nearly every syntactic category—noun phrases, verb phrases, and others are all represented in our results. As an initial step toward characterizing learners at each level, we had made progress; however, it was clear to us that we needed to expand our investigation beyond errors if we were to establish the partial orders of acquisition on which to base the SLALOM architecture.

3. Profiling Overall Performance: Ongoing and Future Work

While the above experiments are an excellent start, in order to truly determine the order in which structures are mastered, one has to look beyond the story told by the errors made by the learners. After all, if learners *A* and *B* both commit an error 10 times, the error count alone appears to indicate that their mastery of the structure is about the same. However, if one were to find out that learner *B* successfully executed that structure 20 times in addition to those errors while *A* had no successful executions, then it appears that the mastery of the structure is actually quite different for these two learners. For *A* the structure is clearly *unacquired*, but for *B* it is either in the ZPD or *acquired*, depending on what standards of performance one is using.

Therefore, our next step in developing the SLALOM architecture has been to obtain the success/attempt ratios which would give us a much clearer picture of how well each learner was able to perform on each structure. We call these 'performance profiles.' Our goal is to re-apply the MANOVA analysis process, this time looking not just at the relationship between how often the error occurs and the level of the sample, but at the learner's overall ratios of performance given both failures *and* successes in language construction usage.

3.1. ABSTRACTING FROM RULES TO KUS

Recall from Section 1.2 that our target user model does not deal solely with the user's performance on the level of rules, but also at the level of rule-abstractions which we call Knowledge Units, or KUs. These KUs represent broader grammatical constructions and are realized by both positive and negative rules in the parsing grammar, representing ways this structure can be executed both successfully and unsuccessfully in the English language. What we desire to accomplish at this stage is to analyze our users' performance on each of these KUs; i.e., for each of these grammatical concepts, we wish to determine the users' rate of success and how those rates differ between the levels of proficiency.

We have therefore developed a database which associates each specific rule and mal-rule in the grammar with any KUs which represent the abstract grammatical concepts realized by the rule[24]. These KUs are the building blocks of our user model, each representing a grammatical concept the user may or may not have mastered at the current time. Intuitively, occurrences of the mal-rules associated with a KU in the writing of a user are indications that the KU has not been mastered, while occurrences of the correct rules from the KU are positive indications of the user's mastery of that KU. In this way, the rule/KU relationship indicates overall mastery by showing that out of n times that the structure represented by that KU was attempted (the total count of executions of all rules which participate in that KU, it was successfully executed some m times (counting just the correct rules).

3.2. OBTAINING THE PERFORMANCE DATA

Having established this correspondence between grammatical rules and the broader grammatical constructs which they implement, our next step is to determine from our samples precisely which grammatical rules (and mal-rules) are executed by students in each of our revised TWE groups. From this data, we would be able to determine whether for a group of users a specific KU is typically *acquired* (indicated by students consistently using the correct grammar rules to implement the KU construct), *ZPD* (with students showing variation between both correct and incorrect rules) or *unacquired* (shown by students consistently using the incorrect mal-rules for that KU). A MANOVA analysis on these results would reveal whether or not significant differences existed between the groups in terms of KU acquisition status, indicating which KU are mastered at each level.

In order to generate this type of analysis, we essentially need complete syntactic data on every sentence in our corpus, which would provide us with the full image of every structure each writer used correctly or incorrectly. The ideal source of this data would be the output of the ICICLE system's parser given the entire corpus

[24] In the case of some very 'flat' rules, several grammatical concepts are involved, hence the inclusion of some rules in multiple KUs. Since each mal-rule is specifically designed to model a specific error, the malrules by contrast typically occur in only one KU each.

as input. From the parse trees it produces, we could develop performance statistics by counting structures used both correctly and incorrectly throughout each tree. Unfortunately, even for those sentences that the current ICICLE prototype can parse successfully, the parser most often comes up with multiple interpretations. Recall that we discussed earlier how distinguishing between these possibilities has been one of the major motivations for creating the user model. Therefore, without that model yet implemented, at this time ICICLE does not have the capability of intelligently selecting a single representative parse tree for each sentence.

Another option unavailable to us was hand-tagging the corpus with the additional data. Marking both the errors *and* every correct grammatical constituent in a sentence would be an overwhelming task for a human coder. An alternative would be to have ICICLE parse the sentences and return all interpretations it could find, and have a human inspect each of these parses trees and select the ‘correct’ one. However, this iteration through the 1793-sentence corpus would also be exceedingly slow and tedious. In most cases, the parser produces multiple parse trees for each of the several hundred sentences and it is very difficult to distinguish them via visual inspection. Clearly, the most desirable approach would be a fully-automated process, but the challenge has been to devise one that could select the right parse tree, something that still seemed to require the assistance of ‘human intuition.’

Therefore, we developed a method of using the human intuition that had already been provided to us for each sentence—namely, the error codes—to enable the ICICLE parser to identify the parse trees closest to a human’s interpretation. To do this, we essentially needed to develop a way to compare the competing parse trees the ICICLE prototype could produce against the interpretation which had been implicitly recorded during the error coding process. The rest of this section details how we were able to convert the hierarchical parse trees into a form which could be compared against the error code sequences to find the closest match.

3.3. SELECTING PARSE TREES

We determined that the best way to use the existing corpus to provide the system with the judgments it needed was to perform the following:

1. Obtain a log of all competing parse trees for each sentence in the corpus.
2. Since each mal-rule has a corresponding error code in our manual, extract from each parse tree the ordered sequence of error codes corresponding to the mal-rules in the tree, e.g. (`mds ids sv`) as in Example (6).
3. Using the Smith-Waterman alignment algorithm, compare the competing sequences generated by the parser against the sequence assigned by a human coder.
4. Select a parse with the closest match to the human-assigned codes.
5. Derive performance statistics from the selected parse.

We began this process by modifying the existing ICICLE user interface to create a spin-off application we called ‘Treefile,’ whose primary function was to take a set of sample files as input, run ICICLE’s parser on each in turn, and log all parse trees spanning the input to text files[25]. The output from this is then run through a parse disambiguation process based on the steps listed above.

We had previously determined the correspondence between the mal-rules of the system's parsing grammar and the error codes that we used in our coding process. Recall that our investigation of the reliability of the coding manual also included very strong results specifically for those error codes which were represented in the parsing grammar. Since no error code which is not represented in the grammar could be produced as a translation from a system-generated parse, the high reliability of this smaller set is highly relevant to this task.

Our disambiguation program therefore has as its task the comparison of a set of hierarchical parse trees on the one hand—each node of each tree representing a grammar rule which may be a mal-rule—and a single, linear sequence of error codes on the other hand, ordered according to the moment in the sentence where the coder encountered the error. Although the conversion of a mal-rule to its corresponding error code is straightforward, the comparison of a hierarchical tree to a linear sequence is not. If the mal-rules were placed only at the leaves, a simple depth-first traversal would extract them in the proper order. However, in this situation they are located throughout the tree, even possibly at the root node itself.

A depth-first traversal is still desirable in order to encounter the constituents of the sentence in order; what remains to be determined is how to process the nodes in a hierarchical fashion that comes as close to possible to the order in which the human coder would indicate the errors as they are found. The choice is between adding a mal-rule corresponding to a root of a subtree prior to adding those representing its children or subsequent to them[26].

In our current implementation, the root node is processed before its children. This is partly motivated by one of our most common mal-rules, the one which represents subject-verb disagreement. This mal-rule occurs at the unification of the subject with the 'verb phrase' which involves the verb plus any complement. The spans of these two constituents are indicated in example (9):

(9) $\ast$ $\boxed{\text{It}}$ $\boxed{\text{make me so excited to meet lot of Deaf students}}$.

As shown in Figure 8, this unification occurs at or near the root of the entire parse, and it spans the entire clause. Therefore, the algorithm traversing the parse tree will encounter a mal-rule which it translates to '`sv`' (shown next to the tree node) as an ancestor node to any other errors in this sentence. There is in fact another error in this sentence, contained within the verb complement. The human coder who reviewed this sentence annotated it as follows:

(10) ∗It make me so excited to meet lot of Deaf students.
(`sv pl`) It makes me so excited to meet lots of Deaf students.

[25] Thanks are due to Greg Silber for creating this application.

[26] Although a third option exists—to insert the parent in the midst of its children—it would be difficult to determine this placement when the number of children exceeded two.

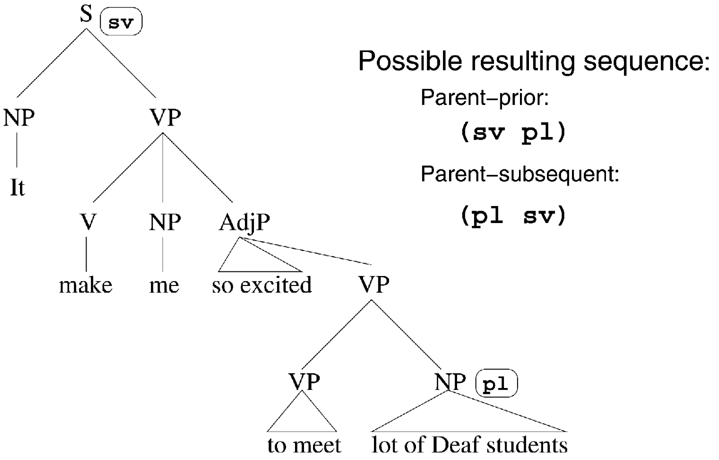

Figure 8. A parse tree and the possible resultant error code sequences.

If the algorithm was reading through a parse tree containing an identical interpretation, and it placed the error code of a parent node after those of its children, it would derive the sequence (pl sv). On the other hand, if it placed the parent before its children, it would derive the sequence (sv pl), and our alignment algorithm would be able to more easily identify this tree as containing the same interpretation as was chosen by our human coder.

It is obvious that the parent-prior choice would not have been as successful if the child errors had been in the subject rather than the verb complement. In that case, they should have been ordered before, rather than after, their parent. But for the sv case in general, our corpus tends to have longer, more complicated verb complements than subjects. There is more opportunity for error there and, in general, the parent-prior technique has enjoyed a high level of success.

Following the acquisition of the system-generated error code sequences to be matched against the human-generated sequence, the modified Smith-Waterman algorithm we described in this paper is then run on all competing sequences, comparing them against the one representing human intuition. Since this matching algorithm computes an alignment score in the last position of the alignment matrix to represent the fitness of the total alignment of both strings, we use this score to select a parse whose error-code alignment is the best match with the sequence generated by the human coder.

At this time, the ICICLE parser is undergoing revision to improve its grammatical coverage in order to better address the task of parsing the entire corpus. When it is completed, we will generate the statistics of user performance at different levels of competence according to the disambiguation process described above.

3.4. FROM DATA TO SLALOM

This data will provide us with the inference knowledge we require in order to complete the sequences represented by the SLALOM architecture. By establishing which grammatical constructions are mastered to which degree at which levels of proficiency, we will have identified the order in which structures are typically acquired, and those structures which are being acquired concurrently by learners at the same level. With this information, we will be able to build the links in the inference network that will enable SLALOM to provide data on structures previously unseen in a user's language production. This will essentially result in a probabilistic model of language development based on our observations of the users at each level of the acquisition ladder.

4. Conclusion

This paper has addressed the application of an empirical methodology toward the acquisition of stereotypical data on a user population. Although our specific domain is the learning of written English by deaf users of American Sign Language, we wish to note that this method could be easily generalized to any domain in which one wanted to establish user stereotypes.

The steps of our methodology that we have described in this paper are summarized below:

1. Identify a sample set of users from which to collect information about the general user population. In our case, we collected writing samples from several colleges of the deaf.
2. Classify these users into the groups which are representative of the stereotypes desired. We applied the expert judgments of certified English instructors to our classification task.
3. Decide what data needs to be collected from these users in order to give meaningful support to the user stereotypes. In our case, we wished to determine the users' levels of knowledge on the KUs in the SLALOM architecture. We did a first pass on data collection by marking just the errors on those KUs, looking for beginner errors which 'disappeared' from the performance of more experienced learners, indicating KUs which had been mastered after the early stages, or errors which did not appear at all until the KUs involved with those structures first came under the focus of the ZPD.
4. Collect the data through a verifiable process. In our case, we used a coding manual to standardize the error marking process, and we verified its reliability.
5. Apply statistical analysis to the collected data to determine the associations between the characteristics displayed in the data and the classifications of the users. We are accomplishing this through the application of MANOVA analyses, first to the errors marked in the samples, and then to the KU performance ratios.

This method is general enough to be applied to any domain, whether or not the stereotypes are being derived to represent a progression of knowledge. In a general sense, the steps outlined above can be used to procure information on what characteristics of the sample users are unique to each stereotype grouping. This information can then enable a system to do intelligent stereotype-selection for a new user, and enable inferencing behavior to fill in the data the stereotype can provide.

In the specific sense of knowledge acquisition stereotypes, the initial classification of the users into groups imposes an ordered sequence on the stereotypes from beginner to advanced. A system such as ICICLE can then attempt to track a user as he or she progresses through this order, using the stereotypes again to fill in data that has not been directly provided through observation of the student. We provided the SLALOM architecture as a way to implicitly encode such a sequence within an inference-capable overlay user model.

Finally, recall that we addressed in Section 1.2.1 the notion of user model data reflecting only the 'current' moment in time. Since a distinctive quality of this kind of stereotype is that we expect the user to change over time, user data should probably have a decay feature so that 'old' performance data is retired in order for performance statistics to reflect the current time only. We are presently investigating methods of maintaining a 'sliding window' within the user model that allows us to look only at data which is relevant to the user's current status in his or her progression toward language mastery.

Acknowledgments

This work has been supported by NSF Grants #GER-9354869 and #IIS-9978021. We would like to thank the readers at the English Language Institute for their expert judgments and Dr. H. Lawrence Hotchkiss at Research Data Management Services at the University of Delaware for his help with statistically analyzing our data. We would also like to thank the other members of the ICICLE group, including Matt Huenerfauth, Jill Janofsky, Chris Pennington, Litza Stark (one of our coders), and Greg Silber.

References

Allen, J.: 1995, *Natural Language Understanding.* California: Benjamin/Cummings, second edition.

Bailey, N., Madden, C. and Krashen, S. D.: 1974, Is there a 'natural sequence' in adult second language learning?. *Language Learning* **24**(2), 235–243.

Baker, C. and Cokely, D.: 1980, *American Sign Language: A Teacher's Resource Text on Grammar and Culture.* Silver Spring, MD: TJ Publishers.

Bull, S., Brna, P. and Pain, H.: 1995, Extending the scope of the student model. *User Modeling and User-Adapted Interaction* **5**(1), 45–65.

Carletta, J.: 1996, Assessing Agreement on Classification Tasks: The Kappa Statistic. *Computational Linguistics* **22**(2), 249–254.

Carletta, J., Isard, A., Isard, S., Kowto, J. C., Doherty-Sneddon, G. and Anderson, A. H.: 1997, The Reliability of a Dialogue Structure Coding Scheme. *Computational Linguistics* **23**(1), 13–31.

Charrow, V. R.: 1975, A psycholinguistic analysis of Deaf English. *Sign Language Studies* **7**, 139–150.

Charrow, V. R. and Fletcher, J. D.: 1974, English as the Second Language of Deaf Children. *Developmental Psychology* **10**(4), 463–470.

Charrow, V. R. and Wilbur, R. B.: 1975, The Deaf Child as a Linguistic Minority. *Theory into Practice* **14**(5), 353–359.

Corder, S. P.: 1967, The Significance of Learners' Errors. *International Review of Applied Linguistics* **5**(4), 161–170.

Desmarais, M. C., Maluf A. and Jiu, J.: 1996, User-expertise modeling with empirically derived probabilistic implication networks. *User modeling and user-adapted interaction* **5**(3/4), 283–315.

Dulay, H. C. and Burt, M. K.: 1975, Natural Sequences in Child Second Language Acquisition. *Language Learning* **24**(1).

Ellis, R.: 1994, *The Study of Second Language Acquisition.* New York: Oxford University Press.

Gass, S.: 1979, Language Transfer and Universal Grammatical Relations. *Language Learning* **29**(2), 327–344.

Glaser, R., Lesgold, A. and Lajoie, S.: 1987, Toward a cognitive theory for the measurement of achievement. In: R. R. Ronning, J. A. Glover, J. C. Conoley, and J. C. Witt (eds.): *The Influence of Cognitive Psychology on Testing*, Vol. 3 of *Buros-Nebraska Symposium on Measurement and Testing.* New Jersey: Lawrence Erlbaum Associates, Chapt. 3, pp. 41–85.

Higgins, J.: 1995, *Computers and English Language Learning.* Norwood, New Jersey: Ablex Publishing Corporation.

Krashen, S. D.: 1982, *Principles and Practice in Second Language Acquisition.* New York: Pergamon Press.

Larsen-Freeman, D. E.: 1976, An explanation for the Morpheme Acquisition Order of Second Language Learners. *Language Learning* **25**(1), 125–135.

Matz, M.: 1982, Towards a process model for high school algebra errors. In: D. Sleeman and J. Brown (eds.): *Intelligent Tutoring Systems*, Computers and People Series. Academic Press, Chapt. 2, pp. 25–50.

McCoy, K. F., Pennington, C. A. and Suri, L. Z.: 1996, English Error Correction: A Syntactic User Model Based on Principled Mal-rule Scoring. In: *Proceedings of the Fifth International Conference on User Modeling.* Kailua-Kona, Hawaii, pp. 59–66.

Michaud, L. N.: 2002, Modeling User Interlanguage in a Second Language Tutoring System for Deaf Users of American Sign Language. Ph.D. thesis, Dept. of Computer and Information Sciences, University of Delaware. Tech. Report # 2002–08.

Michaud, L. N. and McCoy, K. F.: 1998, Planning Text in a System for Teaching English as a Second Language to Deaf Learners. In: *Proceedings of Integrating Artificial Intelligence and Assistive Technology, an AAAI '98 Workshop.* Madison, Wisconsin.

Michaud, L. N. and McCoy, K. F.: 2000, Supporting Intelligent Tutoring in CALL By Modeling the User's Grammar. In: *Proceedings of the Thirteenth International Florida Artificial Intelligence Research Society Conference (FLAIRS-2000)*. Orlando, Florida, pp. 50–54.

Michaud, L. N. and McCoy, K. F.: 2001, Error Profiling: Toward a Model of English Acquisition for Deaf Learners. In: *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics.* Toulouse, France, pp. 386–393.

Michaud, L. N. and McCoy, K. F.: 2003, Evaluating a Model to Disambiguate Natural Language. In: P. Brusilovsky, A. Corbett, and F. de Rosis (eds.): *Proceedings of the 9th International Conference on User Modeling*, Vol. 2702 of *Lecture Notes in Artificial Intelligence.* Johnstown, PA, pp. 96–105.

Michaud, L. N., McCoy, K. F. and Pennington, C. A.: 2000, An Intelligent Tutoring System for Deaf Learners of Written English. In: *Proceedings of the Fourth International ACM SIGCAPH Conference on Assistive Technologies (ASSETS 2000).* Washington, D.C.

Michaud, L. N., McCoy, K. F. and Stark, L. A.: 2001, Modeling the Acquisition of English: An Intelligent CALL Approach. In: M. Bauer, P. J. Gmytrasiewicz, and J. Vassileva (eds.): *Proceedings of the 8th International Conference on User Modeling*, Vol. 2109 of *Lecture Notes in Artificial Intelligence.* Sonthofen, Germany, pp. 14–23.

Milne, S., Shiu, E. and Cook, J.: 1996, Development of a Model of User Attributes and its Implementation with an Adaptive Tutoring System. *User modeling and user-adapted interaction* **6**(4), 303–335.

Nicholas, Hugh, B., J., Deerfield, I. David, W. and Ropelewski, A. J.: 1998, Sequence Analysis Tutorials: A Tutorial on Searching Sequence Databases and Sequence Scoring Methods. Biomedical Supercomputing Initiative of the Pittsburgh Supercomputing Center (PSC), Carnegie Mellon University and University of Pittsburgh, http://www.psc.edu/biomed/training/tutorials/sequence/db/.

Padden, C. and Ramsey, C.: 1998, Reading Ability in Signing Deaf Children. *Topics in Language Disorders* **18**(4), 30–46.

Pienemann, M. and Håkansson, G.: 1999, A Unified Approach Toward the Development of Swedish as L2: A Processability Account. *Studies in Second Language Acquisition* **21**, 383–420.

Quigley, S. P., Power, D. J. and Steinkamp, M. W.: 1977, The Language Structure of Deaf Children. *The Volta Review* **79**(2), 73–84.

Rich, E.: 1979, User Modeling via Stereotypes. *Cognitive Science* **3**, 329–354.

Schneider, D. and McCoy, K. F.: 1998, Recognizing Syntactic Errors in the Writing of Second Language Learners. In: *Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and the Seventeenth International Conference on Computational Linguistics*, Vol. 2. Université de Montréal, Montréal, Québec, Canada, pp. 1198–1204.

Schwartz, B. D.: 1998, On Two Hypotheses of Transfer in L2A: Minimal Trees and Absolute L1 Influence'. In: S. Flynn, G. Martohardjono, and W. O'Neil (eds.): *The Generative Study of Second Language Acquisition.* Mahwah, NJ: Lawrence Erlbaum, Chapt. 3, pp. 35–59.

Schwartz, B. D. and Sprouse, R. A.: 1996, L2 Cognitive States and the Full Transfer/Full Access model. *Second Language Research* **12**(1), 40–72.

Smith, T. F. and Waterman, M. S.: 1981, The Identification of Common Molecular Subsequences. *Journal of Molecular Biology* **147**(1), 195–197.

Spada, H.: 1993, How the Role of Cognitive Modeling for Computerized Instruction is Changing. In: P. Brna, S. Ohlsson, and H. Pain (eds.): *Proceedings of AI-ED'93, World Conference on Artificial Intelligence in Education.* Edinburgh, Scotland, pp. 21–25. Invited talk.

Stewart, D. A.: 2001, Pearls of Wisdom: What Stokoe told us About Teaching Deaf Children. *Sign Language Studies* **1**(4), 344–361.

Stokoe, W. C.: 1976, The Study and Use of Sign Language. *Sign Language Studies* **10**, 1–36.

Suri, L. Z.: 1993, Extending Focusing Frameworks to Process Complex Sentences and to Correct the Written English of Proficient Signers of American Sign Language. Ph.D. thesis, Department of Computer and Information Sciences, University of Delaware. Technical Report TR-94–21.

Suri, L. Z. and McCoy, K. F.: 1993, A Methodology for Developing an Error Taxonomy for a Computer Assisted Language Learning Tool for Second Language Learners. Technical Report TR-93–16, Department of Computer and Information Sciences, University of Delaware.

Swisher, M. V.: 1989, The Language-Learning Situation of Deaf Students. *TESOL Quarterly* **23**(2), 239–257.

Vygotsky, L. S.: 1986, *Thought and Language.* Cambridge, Massachusetts: The MIT Press. Translation revised and edited by Alex Kozulin; originally published in 1934.

Ward, Joe. H., J.: 1963, Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association* **58**, 236–244.

Wilbur, R. B.: 1977, An Explanation of Deaf Children's Difficulty with Certain Syntactic Structures of English. *The Volta Review* **79**(2), 85–92.

Authors' vitae

Dr. Lisa N. Michaud is an Assistant Professor of Computer Science at Wheaton College in Norton, Massachusetts. She earned her B.A. in Computer Science and English Literature from Williams College, and completed an M.S. and a Ph.D. in Computer and Information Sciences at the University of Delaware. Her research interests center on the application of User Modeling to tutoring environments, especially those involving the acquisition of linguistic skills. In addition to continuing her collaboration with the ICICLE project, she is working on implementing user modeling in the King Alfred system, an application used by Michael Drout of Wheaton's English Department to teach the syntax of Anglo-Saxon English to undergraduates.

Dr. Kathleen F. McCoy is a Professor of Computer and Information Sciences and the Director of the Center for Applied Science and Engineering in Rehabilitation at the University of Delaware. She received her B.S. in Computer and Information Sciences from the University of Delaware, and her M.S. and her Ph.D. in Computer and Information Science from the University of Pennsylvania. Her research interests include Natural Language Processing (Computational Linguistics) and its subfields Natural Language Generation and discourse, User Modeling, and Intelligent Tutoring Systems, with emphasis on applications for people with disabilities.