A Description of BRIDGE, An Army Research Institute-Sponsored, NLP-Based, Foreign Language Tutor

Cathie E. Alderks

Follow this and additional works at: https://scholarsarchive.byu.edu/dlls

BYU ScholarsArchive Citation
Available at: https://scholarsarchive.byu.edu/dlls/vol19/iss1/4

This Article is brought to you for free and open access by the All Journals at BYU ScholarsArchive. It has been accepted for inclusion in Deseret Language and Linguistic Society Symposium by an authorized editor of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.
A Description of BRIDGE, An Army Research Institute-Sponsored, NLP-Based, Foreign Language Tutor

Cathie E. Alderks

Many students learn foreign languages in the classroom just to find themselves in situations once the class is finished that are not conducive to the retention of language skills. Many times their work or assignments have little to do with the acquired language. Time or other language learning opportunities may be limited such that the student is unable to maintain the acquired skills. Students may be able to obtain reading material for comprehension maintenance, but unless there are others available for practice, there is little opportunity for production. The use of a computer seemed to be the logical solution to these problems. However, few people have access to a large (mainframe) computer where many of the language learning programs reside. Therefore, for a computer to be an acceptable solution, the language software would, of necessity, need to be able to be run efficiently on a small (PC) computer.

The vast majority of computer assisted language learning (CALL) software emphasizes the skills involved with comprehension. Text and challenges/questions can be delivered in either written or spoken formats. However, students, in order to demonstrate their understanding must point out, move, choose, or match exactly (string match) an answer already predetermined to be correct by an instructor who programmed the correct answer into the system. Any deviation from what has been determined to be correct is considered incorrect. There is no opportunity for free expression. As anyone who has worked with a system requiring string matching knows, there is opportunity for frustration as no instructor can second guess everything a student will come up with, and a student can always concoct something seemingly obvious and correct that the instructor would never dream up. Recognizing these problems, and seeing the potential of parsers that have gained popularity in machine translation, the Army Research Institute (ARI) developed a unique product that built upon the existing CALL work and incorporated a natural language processor (NLP) to allow freely-typed expression by the student. Several other important features of the tutor include the detection and categorization of the type of error made by the student, extension to other languages, and a completely authorable interface that permits changing and/or adding new exercises without having to be knowledgeable of a specific programming language. Each of these attributes will be discussed in turn.

Incorporation of a Natural Language Processor

A broad goal of natural language processing is for a computer to understand language as a human understands language. That would include analyzing and interpreting free-form text and discourse. Technology isn’t there yet. A more modest and realistic goal includes parsing, that is, for the computer to analyze free-form text into basic grammatical structures of a particular language. NLP is commonly used in grammar checkers and other writing aids, machine translation, automated information retrieval, and message extraction.

One might ask the question of why we would want a parser in a foreign language tutor. We all know that there are many ways of expressing correct, or incorrect information. For example, if we were to ask the question in statement 1, there would be many ways of expressing the answer. Sev-

Cathie E. Alderks is a research psychologist with the Army Research Institute located in Alexandria, VA. She obtained her Ph.D. in psychology in 1982 from Brigham Young University. Her primary research interest is computer-assisted language learning for German, Arabic, and Spanish. Other research interests include leadership and cohesion and observational learning.
eral possibilities are listed in sentences 2-5 (answers 2-4 are grammatically correct; 5 is incorrect. 
(1) How did the dog get the bone?
(2) The boy gave the dog the bone.
(3) The dog got the bone from the boy.
(4) The bone was given to the dog by the boy.
(5) *The boy give to the dog the bone.

No matter how hard a teacher would try, a student would always come up with an answer that was correct but not anticipated. In string matching, an unanticipated, correct answer would be considered and counted as incorrect.

A parser could take the input sentence, analyze it, and then respond appropriately. Second guessing would become obsolete.

The BRIDGE tutor incorporates a parser that allows for the student to freely input sentences. Sentences are analyzed according to their grammatical structures. Unfortunately, at the present time, semantic correctness of a response is not checked.

Detection and Categorization of Student Error Type

One of the requirements of the parser was that it detect the error, and then move on and give feedback about the error. The system had to allow errors to occur without bombing, but at the same time note them, categorize them, and provide feedback. There also had to be a check for simple "typos."

While the syntactic coverage is not complete when one considers the immensity of the German language, many constructions are covered. Among those are the simple locative, simple predicative, simple transitive, simple intransitive, conjunctions, simple questions such as yes/no or Wh-questions, imperatives, modals, inversion, and embedding. Five specific error types that the parser is designed to check include:

a. subject-verb agreement
b. subject not in the nominative case
c. verb-preposition agreement errors
d. preposition-noun agreement errors
e. modifier noun agreement errors.

There are several constructions that the parser simply doesn’t understand. These include:

a. Adjectives used as nouns, such as “das” used by itself to mean “this."
b. Relative clauses, such as in “Der Soldat”
c. Sentences that begin with a conjunction, such as “Und was ist das?” In general, the computer doesn’t do well with conjunctions even though they are covered and the student is advised to not use them.
d. Sentences with two or more verbs, such as “Der Soldat geht und arbeitet.”

In addition, the computer cannot detect the correctness of factual information. The tutor can give feedback only about form, not about the correctness of the factual information.

Extension to Additional Languages

Much time and effort goes into building a parser. It was desirable that the technology, time, and effort that went into the development of the German parser could be used for additional parsers for other languages.

The parser was constructed according to the principles of Government-Binding (Chomsky, 1981). Briefly, the underlying principles include the idea that a small number of abstract principles apply to many different constructions. Independent formulae are grouped into interacting modules. In turn, these modules are parameterised such that by modifying them to a small degree, patterns associated with a variety of languages can be generated. Put glibly, by switching a few switches, the basic constructions of another language can be handled.

In a test of this extendibility, a language unlike German was chosen: Arabic. Using those modules that captured the structural similarity across languages, a parser in Arabic was developed in less than 1/3 the time that was required for the German parser. This was all the more impressive because no comprehensive syntactic analysis of the Arabic language was available, whereas German had been extensively analyzed. It must be noted that development was still costly in that it took several months work by a computational linguist, but it did help to have the tools in place.

Examples of interacting grammar modules include phrase structure and case assignment. All languages have some sort of phrase structure. For example, German is Head-Final (the structure that determines the identity of a phrase generally appears on the final edge of the phrase), and English is Head Initial (the identifying structure appears at the beginning edge of the phrase). Arabic allows the English order, but also allows the verb to appear at the head of the sentence. The head of the category is that which determines that category’s type, the verb of a verb phrase, the noun of a noun phrase, etc. A language that has a verb appearing before its objects is head initial and will also have a noun appearing before its objects. Thus, individual language phrase structure rules for each category type are not needed as the general form for the...
module can be initialized as head-initial or head-final, depending on the requirements of the language—in this case head final for German (SOV) and head initial for Arabic and English (VSO and SVO). A general transformation allows the movement of the verb from the second position to the initial position for Arabic and from the final to the second position for German. In a simplified example, for the English sentence “Dogs eat meat,” the German sentence would be with an embedded clause “[I know that] dogs meat eat,” and the Arabic sentence would be “Eat dogs meat.” Figure 1 further illustrates this structure.

Figure 1. Phrase Structure Rules Generated by a Government-Binding Grammar Approach.

According to Government-Binding Theory, every lexical Noun phrase must receive a grammatical case. Examples of cases are nominative, accusative, dative, genitive. Case is assigned by predicates (verbs, prepositions, tense or agreement marker) to particular positions. For example, in the following sentences, the required pronoun marking is determined by whether the noun phrase is pre- or post-verbal.

(6) He(NOM) looked at him(ACC).
(7) *Him(ACC) looked at he(NOM).

In the correct sentence, sentence 6, the subject “He” is in the nominative case and the object of the preposition “him” is in the accusative case. The incorrect example, sentence 7, has the subject incorrectly in the accusative case and the object of the preposition incorrectly in the nominative case. Within the parser, modules that assign case can be switched as appropriate according to the language in question.

A Completely Authorable Interface

One of the goals of the project was to design a computerized tutor with a completely authorable interface such that an instructor could change or add lessons without having to learn or know any programming languages. In other words, the requirement was that anyone without specific computer knowledge, capabilities, or skills would easily be able to produce new lessons.

Additionally, an authorable interface is desirable for theoretical reasons. As everything is not known in second language acquisition, the ability to author lessons to perform research that will determine ideal instructional design and investigate the manner in which particular languages are learned is important. Examples of research questions might be: Does stopping the student for errors aid in language learning, or is it best to allow certain errors to go uncorrected or unsignalled? Are certain constructions always learned before other constructions?

The present configuration allows two categories of challenge: oral and text-based. Within each category there are several types of exercise: fill-in-the-blank, complete sentence, graphic/location (map) identification, multiple choice, and (within text-based only), sorting. In the oral exercises, the challenge is spoken (pre-recorded) and the answer is typed into the keyboard. In the text-based exercises, all challenges are presented via the monitor with no sound being produced. Unfortunately, speech recognition technology is not yet available to allow responding through speech. Responses for the complete sentence and the fill-in-the-blank
exercises are sent to the parser for evaluation. As the other types do not require grammatical analysis, they are not sent to the parser; in these cases only are the answers predetermined.

There are several ways in which the interface is authorable. To best demonstrate these, first a mini-lesson will be described; then, the authoring interface for that mini-lesson will be described. The mini-lesson will consist of three exercises. The first will be a multiple-choice exercise representative of those exercises that can tap factual information and do not go to the parser. The second will be a graphic/map exercise. The third will be a complete sentence exercise where the response is sent to the parser for evaluation.

First the student logs into the computer. For the sake of this example the student will be called Joe. If Joe is unfamiliar with a computer, he may choose a short tutorial that will familiarize him with basic commands and procedures such as how to use a mouse. If Joe has at least elementary computer literacy, he may choose to move into a lesson. He records his name or other identifying code at a prompt. A select screen then comes up on the computer (Figure 2) and Joe moves the mouse to highlight a desired lesson, in this case a lesson called “paper.seq.” When he clicks on “OK,” the lesson is activated. If he were to change his mind, he could click on “Cancel” and then choose a different lesson.

The first exercise of the lesson then comes up on the screen. (See Figure 3.) Depending on whether the question refers to a text or map/graphic, Joe may go to the menu bar and choose the text (Figure 4) or map/graphic for examination. These materials, along with the other aids identified in the menu bar are available throughout the lesson. A click on the text or map/graphic returns Joe back to the exercise. In our example, Joe chooses “die Bundesstrasse 254.” This answer is incorrect and a feedback window (Figure 5) comes on the screen. Joe may try again, stop the lesson, or browse. Browse allows Joe to check various helps. By clicking on “Explanation” on the menu bar (Figure 6), Joe may


Figure 4. The text as brought down by a click on the “text” button on the menu bar. Any text may be put in by the instructor.

Figure 5. The incorrect feedback window that appears for an incorrect response.
A Description of BRIDGE, An Army Research Institute-Sponsored, NLP-Based, Foreign Language Tutor

Figure 6. An incorrect choice with possible hint and explanation choices to aid the student in obtaining the correct response.

Figure 7. A hint specific to the exercise at hand.
elicit various helps, hints, explanations, or add comments about the lesson. Joe chooses “See Hint Text” (Figure 7). After pulling down the text, reading the text with emphasis given to the hint, Joe chooses “Lauterbach” (Figure 8) and the feedback of “Correct” (Figure 9) comes on the screen. At this point, Joe may “Browse” some more if he needs to, stop the lesson, or move on to the next exercise. Now, Joe wants to check a word in the dictionary. He clicks “Dictionary” on the menu bar. A card file dictionary comes up (Figure 10). Joe may search through the dictionary at will by flipping through the cards or by searching for a specific word. As with typical “Windows” operation, a click in the upper left hand corner allows for Joe to close the dictionary window.

Joe moves onto the next exercise. He is to identify the location of the place written in the upper left hand corner of the map (Figure 11). By moving the mouse, the cursor is positioned to the desired location, in this case “auf einem berg,” “on a mountain.” A click seals the choice. Joe chooses correctly and the “Correct” feedback window (Figure 9) comes over the screen. If Joe had chosen incorrectly, he would have received the “incorrect” feedback and had another opportunity to try again. As only two tries are allowed per exercise, if he had failed the second try, he would have been
Figure 10. The card file dictionary that is available to the student throughout the lesson.

Figure 11. An example of the map/graphic exercise. The student is to find and then click on the location written in the top left hand corner.
Figure 12. An example of a complete sentence exercise complete with an incorrect response.

Figure 13. Feedback window for a complete sentence exercise that contains secondary errors.

Figure 14. A feedback screen explaining a modifier/noun error showing the words involved in the error.
SECONDARY EXPLANATION ERROR: (2/2)

Subject + Verb Agreement Error: wir, seh+e

In an incorrect answer, modifier does not agree with its noun in gender, number, and case (modifier noun agreement error).

Figure 15. A feedback screen explaining a subject/verb agreement error showing the words involved in the error.

Figure 16. A complete sentence question with a correct response.

Figure 17. The lesson is completed.

shown the correct location and then moved onto the next exercise.

The third exercise is a text-based, complete sentence query. The question comes across the screen “What do we see when we follow along the river Lauterbach to the north” (Figure 12). Joe types in “We see the city Lauterbach” and the sentence is sent to the parser for evaluation.

Errors are classified as either primary or secondary depending upon the lesson objectives and may be changed according to the emphasis as determined by the instructor. For example, in an exer-
cise involving location and direction, case errors following prepositions may be determined to be the main objective, therefore, only these specific errors "count" in determining "correctness." Others, considered secondary, are de-emphasized and will be noted, but will not stop the progression of the lesson. These secondary errors may be investigated should the student desire, but it is not required. Therefore, the lesson progression is not stopped for every little error. Any number of captured errors may be classified as primary or secondary, as the instructor desires. Stated simply, if a primary error is made, the progression of the lesson is stopped and the student must check the error. If a secondary error is made, it is recorded and noted, but the progression of the lesson is not stopped. The student may look at error messages if he desires, but it is not required.

In this case, Joe made two secondary errors (Figure 13). His response is considered correct as the structures picked to be important to the actual lesson objectives for this particular lesson are correct. He may "Browse" and look at his secondary errors if he desires, but that is his choice. In this case, Joe decides to look at his errors. Each error message comes up separately (Figures 14 and 15) with explanations. Joe then returns to the question and tries again (Figure 16). This time, he is correct and the correct window comes on his screen (Figure 9). As Joe has finished the lesson, he then exits (Figure 17).

At any time, Joe may check his scores by clicking "Progress" from the menu bar. Figures 18 and 19 show his progress. For this example, the example lesson was repeated three times. Figure 18 lists the number of exercises in the lesson (9) and the number that remain to be completed. Unadjusted Scores records exactly what happened on each of the trials for each lesson. If the first try was incorrect and the second was correct, one incorrect and one correct try would be recorded. Adjusted Scores are obtained by the final result of the exercise. If a question were correct on either of the tries, it would be considered correct. Only those questions that were never correctly answered are considered as wrong. Figure 19 illustrates what happened on each trial. Listed are the exercise number, fixed or random sequence of exercises, correct or incorrect, the time to respond, the primary and secondary lesson objectives, and any other pertinent information. For example, in trial #5, a word was typed in that was not found in the lexicon (in this case, a "typo"); that is noted in the data.

The authoring screens are extremely easy to use. By choosing to enter the author mode of the tutor, an instructor may add new exercises, modify existing exercises, or sequence exercises into lessons. Upon entering the author mode of the tutor, an interface screen appears (Figure 20). The instructor may choose to look at exercises (View/Add Exercise) or add to/construct a lesson sequence (Add to Sequence). The choice is made by clicking on the appropriate box. (An option under "Ac-
A Description of BRIDGE, An Army Research Institute-Sponsored, NLP-Based, Foreign Language Tutor

Student Scores

1. tcs019, Fixed Sequence, C, 204, (220,230,20), (-), (200,210,240), (-), OKAY
2. tmap004, Fixed Sequence, W, 1146, (-), (500), (-), (-).
3. tmap004, Fixed Sequence, C, 2, (500), (-), (-), (-).
4. tmc105, Fixed Sequence, C, 85, (20), (-), (-), (-).
5. tcs019, Fixed Sequence, WORD NOT FOUND (die)
6. tcs019, Fixed Sequence, C, 201, (220,230,20), (-), (210,240), (200), OKAY
7. tmap004, Fixed Sequence, W, 63, (-), (500), (-), (-).
8. tmap004, Fixed Sequence, W, 79, (-), (500), (-), (-).
9. tmc105, Fixed Sequence, W, 43, (-), (20), (-), (-).

Extra Practice Scores

Figure 19. Student scores by trial. Recorded are the exercise, type of sequence, correct or incorrect, objectives, and additional comments.

Foreign Language Tutor Authoring Interface

○ View/Add Exercise × Add to Sequence

Append an Existing Sequence

Exercise Sequence
1. tcs019
2. tmap004
3. tmc105

Figure 20. The authoring interface.
Proceedings of the 1993 Deseret Language and Linguistics Society

The types of lesson according to code are listed in Table 1.

Table 1. Exercise types with their corresponding codes.

<table>
<thead>
<tr>
<th>Code</th>
<th>Type of Lesson</th>
</tr>
</thead>
<tbody>
<tr>
<td>omc</td>
<td>oral multiple choice</td>
</tr>
<tr>
<td>omap</td>
<td>oral map/graphic</td>
</tr>
<tr>
<td>ofib</td>
<td>oral fill-in-the-blank</td>
</tr>
<tr>
<td>ocs</td>
<td>oral complete sentence</td>
</tr>
<tr>
<td>tmc</td>
<td>text-based multiple choice</td>
</tr>
<tr>
<td>tmap</td>
<td>text-based map/graphic</td>
</tr>
<tr>
<td>tfib</td>
<td>text-based fill-in-the-blank</td>
</tr>
<tr>
<td>tcs</td>
<td>text-based complete sentence</td>
</tr>
<tr>
<td>tsort</td>
<td>text-based sorting</td>
</tr>
</tbody>
</table>

To sequence exercises, an instructor need only to click on the desired exercise name and it appears in the exercise sequence box. Once saved and named, the sequence is available for use. If a mistake is made, a click on the exercise in the sequence box removes it.

New exercises may be constructed by simply pulling up a template (by “Add Exercise” on the menu bar or by clicking on an exercise after checking the “View/Add Exercise” option). The exercise template for a multiple choice exercise is located in Figures 21 and 22. The “ID” is the exercise name. The numbers by “Primary GF, Secondary GF (learning objectives), Specific Directions, and General Directions” are codes of lists of directions that are called up as needed. A sample of these lists is shown in Figure 23 and is for the Specific Directions. Notice, the directions are in both English and the target language. The choice can be made by the instructor as to which will appear and all lists are completely authorable.

To change or alter a multiple choice question, one must merely have the template on the screen. Under “Challenge” a new challenge must be written. Correct response(s) are then entered under “Correct.” Incorrect response(s) or distractors are entered under “Wrong.” The number of correct answers to appear each time the question is posed is entered under “Num” for the correct responses. The number of distractors to appear each time the question is posed is entered under “Num” for the “Wrong” responses. Each time the question is used, the choices are randomized for presentation. If more than the desired number of choices are listed on the template, the computer randomly selects among the choices to provide for the desired number of possibilities.

The second page of the template (Figure 22) allows one to choose a map file or a text file to correspond with the exercise. Any text file may be used. It is limited only by the instructor’s desires and the computer’s memory. Any bitmap may be used for the map. It need not be a map but can be any graphic.

A template for a map/graphic exercise is shown in Figure 24. Again, all an instructor must do is to type in a challenge, add a short description, and identify a map file name (bitmap). To set the correct answer, a click on “Set Correct Answer” will bring up the graphic. By moving the cursor with the mouse, a location is identified. A click will set the location (see Figure 25). The coordinates (“correct button location” on Figure 24) will automatically change as the mouse is moved to the new correct location.

![Figure 21. An example of a multiple choice template, first screen.](image-url)
A Description of BRIDGE, An Army Research Institute-Sponsored, NLP-Based, Foreign Language Tutor

<table>
<thead>
<tr>
<th>Actions</th>
<th>Add Exercise</th>
<th>Default Templates</th>
<th>Exercise View</th>
<th>Comments</th>
<th>Instructor_Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID: tmc105</td>
<td>Set(s): 2</td>
<td>Primary GF: 20</td>
<td>Secondary GF:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specific Directions: 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>General Directions: 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Challenge: Was liegt nördlich des Hahntig?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maps:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TextFiles:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 22. An example of the second screen of a multiple-choice template.

**Actions**

**Specific Directions**

**Add New Item**

**English:**

0. "**Unknown Specific Direction**"
1. "Read the sentence in the top left of the screen. Click on the number of the word or phrase that fills in the blank."
2. "Listen to the sentence. Click on the number of the word or phrase that fills in the blank."
3. "Below is a list of words or phrases. For each word, press the ← to move it into the left column."
4. "In the top left corner of the screen is a location. Click on the location on the map."

**Foreign:**

0. "**Unknown Specific Direction**"
3. "Es folgt ein Liste von Wörtern oder Phrasen. Für jede, klicken Sie, ← um sie in die linke Ecke links oben."
4. "In der Ecke links oben ist eine Ortsangabe. Klicken Sie diese Ortsangabe."

Figure 23. Screen for Specific Directions from the Authoring portion of the tutor.
### Figure 24. The authoring template of a map/graphic exercise.

<table>
<thead>
<tr>
<th>Actions</th>
<th>Add Exercise</th>
<th>Default Templates</th>
<th>Exercise View</th>
<th>Comments</th>
<th>Instructor_Help</th>
</tr>
</thead>
</table>

- **ID:** map004
- **Set(s):**

**Primary GF:** 500

**Secondary GF:**

**Specific Directions:**

**General Directions:**

**Challenge:** auf einem Berg

#### Set Correct Answer

- **Correct Button Location:** 3024,1812
- **Description:** The area on and above Hainlick hill, but below 285.
- **Map File Name:** newgerm.bmp

### Figure 25. Setting the coordinates for a map/graphic exercise.

Note: For a full understanding of the content, please refer to the original document. This description provides a simplified version for readability.
As no correct answers must be recorded for those questions whose answers go to the parser, all that must be entered to construct new exercises for complete sentence or fill-in-the-blank queries is the challenge (see Figure 26).

Summary

The authorability of the tutor provides many opportunities for research in second language acquisition as well as providing excellent opportunities for language maintenance and learning. Flexibility of tailoring lessons for specific purposes is an important advantage of the system, especially considering the ability to use a commonly available PC. As notebook computers become more readily available and the price comes down, the possibility of taking computerized language training, maintenance, and practice anywhere (on the plane, to remote locations, etc.) will aid in the effectiveness of language learning. The ability to freely input self-generated statements allows for practice in production skills with immediate correction if needed and desired. Students are able to “try out” constructions with which they may feel “shaky” to gain confidence and further increased proficiency.

References


End Note

1 The views expressed in this paper are those of the author and do not necessarily reflect the views of the U.S. Army Research Institute or the Department of the Army.