EMERGING TECHNOLOGIES
FOCUSING ON FORM: TOOLS AND STRATEGIES

Robert Godwin-Jones

Virginia Commonwealth University

Using computers to help students practice and learn grammatical constructions goes back to the earliest days of computer-assisted language learning (CALL). With the coming of the Internet age, CALL began to focus more heavily on the new capabilities of group connectivity and computer-mediated communication. More recently, a gathering consensus has emerged that for adult learners, at least, an awareness of forms and rules is a vital component of online language learning (Skehan, 2003). Compared to the nature of earlier grammar-oriented applications, however, there is recognition today that a focus on form should not be an isolated, stand-alone activity but rather should be integrated into a communication-centered, networked language learning environment. Contemporaneously, it has become clear that grammar exercises need to require more than single word or phrase answers. The older exercise formats, such as multiple choice and fill in the blanks, should be supplemented by new and engaging interactions with real communicative goals. Something more than just canned feedback should accompany the exercises. Feedback should be informative, contextual, and, whenever possible, individualized. The expectation today is that programs will guide students to pay attention to forms and structures, to gain facility in their use, and to extend their language knowledge/usage to more complex constructions. Stated succinctly, grammar exercises need to be integrated, intelligent, and innovative.

Fortunately, technology developments and creative language professionals are moving us down this path. Recent trends in intelligent language tutors (ILT) are quite promising, despite the multiple challenges of natural language processing (NLP). Having over-promised and underachieved in the past, developers of ILT (also known as ICALL, intelligent CALL, or parser-based CALL) have mostly narrowed their ambitions and scope, resulting in actually deployed systems rather than just research prototypes. Advances in collecting and processing language corpora have helped in that process. At the same time, Web developments offer new approaches to exercise design and distribution.

INTEGRATING GRAMMAR INTO TASK-BASED ACTIVITIES

Much of the language instruction done today, either in the classroom or on the computer, revolves around specific tasks learners must carry out, either individually or in groups. Tasks lead learners to focus on communicating within concrete, realistic contexts, negotiating meaning towards reaching a specific goal, and gaining confidence in using the target language to achieve results. On a computer, that process can begin in the task planning stage. Building on the idea of pre-task vocabulary brainstorming, a possible partner activity could be a brief discussion of any constructions that learners anticipate using in the task, such as the subjunctive for hypothetical situations in Spanish. This would likely be most applicable to intermediate/advanced level students. The partner interaction could be done through a chat window integrated into the exercise environment, as suggested by Shaoqun Wu and Ian Witten (Wu &Witten, 2006). This provides the possibility of groups sharing ideas with one another. Integrating collaborative tools makes grammar exercises less isolating and more communicative. The kind of pre-task help suggested here is common in writing assignments (suggested sentence starters, possible sentence connectors, etc.), but is less frequent in the context of grammar exercises. Having some discussion of lexico-grammatical issues before the task begins may alert learners of their importance and possibly result in the use of more complex language.

Post-task activities can also be integrated into a task to encourage focus on form. If students know, for example, that a product of their efforts will be shared with others (through a forum, blog, or wiki), they are likely to be more deliberate in their use of language. The notion of posting students' work could also be extended from individual efforts to class exchanges or school collaborations. A necessary condition for greater focus on form is a level of comfort on the part of the students with the task content. If the topic is too unfamiliar or challenging, it is less likely that learners will focus on form. It is also unlikely in that case that they will consider using more complex language constructions. Whether it is a task to be evaluated by the teacher or processed by a computer, there needs to be recognition of efforts by students to stretch their skills beyond familiar collocations and simple sentence formations. This is one of the most challenging tasks for software, even if some form of artificial intelligence is used. It is easy to check for expected and typical responses; dealing with creativity is much harder.

Post-task activities provide opportunities for students to expand and internalize what they have learned. There should be a record kept of task results and new information learned, either by the learner or automatically by the program. Note should be taken of remediation that needs to be done. Ideally, this information should be used to help determine the next steps in the student's language learning. As we shall see, this is a crucial aspect of best practices in the design of ILT's. Follow-ups to assigned tasks can have students write a short summary of what they learned (shared electronically) or construct sentences or dialogs using the practiced/learned structures and vocabulary in different contexts. Another follow-up could be a game or contest in which correct and rapid use of practiced forms is required. An additional post-task exercise could be the consultation of a concordance in which learners find and analyze occurrences of phrases or structures. This brings into play the use of language corpora, another important element in the creation of intelligent language tutors.

INTELLIGENT LANGUAGE TUTORS

For language learners to be successful, tasks should not be disconnected activities but integrated into a learning environment that tracks and analyzes their progress, offers help as needed, suggests and provides logical next steps, and gives appropriate feedback. An intelligent computer program, like a good teacher, should be able to treat students as distinct individuals, provide learner-specific guidance, and design a customized learning path. This is a tall order, and not just from a technical standpoint, as knowledge is required in fields such as learning psychology, discipline-specific pedagogy, and logic. This is on top of expertise in software engineering and human/computer interaction. To create an advanced language learning application, one needs to add to the mix of expertise listed above knowledge in areas such as second language acquisition, computational linguistics, formal grammar theory, and natural language processing. Creating an ILT is not a project to be undertaken lightly, and it is not surprising that, even with teams of experts, many projects never reach production status. Marina Dodigovic's recent monograph (2005) offers a glimpse into the many considerations and efforts that go into just the preparatory stage of designing an ILT. Similarly, Markus Dickinson and Joshua Herring (Ohio State University) show how complex the process can be, even in a restricted task such as analyzing Russian verb conjugation.

The ILT projects most likely to see the light of day are those that restrict the learner input to be processed and evaluated. This restriction might be through the exercise type, the length/complexity of the learner input, or aspects of the input which are excluded from analysis. For many areas of discrete grammar knowledge, little machine intelligence is required. To determine, for example, whether a German adjective ending is correct, it suffices to compare the student response to the known (single) correct answer. This kind of string matching is easy and fast but only provides responses of right or wrong to the learner. “Intelligence” lies in enabling the computer program to provide more nuanced and useful feedback to the learner. For German adjective endings, the system would need to know about variations based on gender, number, case, and context, and be able to evaluate a student error based on that information. Even such a small slice of language involves a large number of variables. Consider how much more complex a challenge it is to parse and evaluate even a short sentence. The system needs to analyze why a given response is incorrect in the given context, then provide options and help for learners to improve their response. It is little wonder that in the past ILT's were plagued by false positives (correct input marked incorrect) and missed incorrect constructions.

In recent years, a number of ILT's have become available which represent considerable advances over earlier systems in a number of ways, including processing responsiveness, evaluative accuracy, and meaningful feedback. Among recent ILT projects which have attracted generally positive receptions are E-Tutor (formally German Tutor) from Trude Heift, Robo-Sensei (formerly Banzai, for learning Japanese) from Noriko Nagata, Spanish for Business Professionals from L. Kirk Hagen, TAGARELA (for beginning Portuguese) from Luiz Amaral and Detmar Meurers, and imPRESSions from Joan-Tomàs Pujolà (for understanding English language media). Taken together as the current state of art in ILT's, there are a few best practices that can be gathered from these systems:

Reusability. The design of the project should whenever possible enable (and encourage) re-use. There is so much effort and expense involved in creating an ILT that building on previous projects makes more sense here than in any other area of CALL. One design factor which aids re-use is modularity. This involves separating the processing of student input into separate programs or modules which can be run sequentially (as in E-Tutor) or selectively on demand (as in TAGARELA). This approach to programming has allowed for creation of programs like Boltun, re-using TAGARELA for Russian, and a Greek ILT building on E-Tutor. The modular design of E-Tutor allows particular parts of the system, such as the punctuation checker, to be turned on or off as desired. This also allows updating of only one section of the system at a time, or adding custom modules in the future. It also builds in the option of sharing only particular modules. Of course, grammar models and analysis processes will vary according to the target language, which makes some parts of the system non-transferrable. There may be as well technical hurdles to moving modules from one system to another. These are typically complex systems, which may use a variety of languages, most often Prolog and Lisp for NLP and Java, Perl, or Python for Web delivery. This makes it important that possible compatibility be built in to a new system from the beginning. The ultimate re-use of these systems would be the ability for non-programmers to add or edit content. In that sense, the evolution of ILT's seems to point towards the need for creation of user-friendly authoring interfaces.

Flexible Feedback. Feedback should be tailored to a student's language level and based on what the system knows about the student's learning history. E-Tutor, for instance, bases feedback on the student's general level of German knowledge and incorporates a “student model” which tracks student work and calculates level adjustments (Heift, 2005). Knowledge of a student's language level can be important in processing errors correctly to determine whether the cause of an error is misspelling, carelessness, lack of knowledge, misapplication of a rule, interference from native language, etc. Additionally, E-Tutor features a “Report Manager” which allows students to review the work they have done, as well as to redo particular exercises. Generally, studies have shown advantages to providing rich meta-linguistic feedback (Heift, 2004; Murphy, 2007), although feedback that is too technical or uses unknown linguistic terminology is problematic. Providing feedback in the target language makes the system more widely deployable since students from a variety of native languages can use the same interface. Offering students choices in the extent and nature of the feedback would help to accommodate students with different learning styles. This is a feature of imPRESSions (Pujolà, 2001). Since there is no definitive answer to what feedback should optimally look like, it seems logical to give students some choice in the matter. ILT's generally give students feedback one error at a time, so as not to overwhelm them with information. However, for more advanced students, or those working independently, a range of options based on whatever information the system can provide might be optimal.

Learner Empowerment. Part of the rationale for offering students options is to encourage them to become independent learners. Whenever possible, users should be guided to find errors themselves, as is done in an experimental ILT for Arabic. Putting students into a more active role makes it more likely that form features will be noticed and retained. The imPRESSions system implements a delayed two-step feedback system that encourages students to reflect on the signaled errors before receiving an explanation. Important for learner autonomy as well is the availability of additional context-aware help in the form of links to grammar tutorials, dictionaries, and other resources. More and more students and adults are looking to learn or maintain languages independent of classroom instruction, often adding or improving language skills for career reasons. In this environment, catering to autonomous learning is more important than ever.

Beyond accuracy. The analysis of the learner's input should recognize and value accuracy but also complexity, fluency, and creativity. This is no trivial undertaking. But some systems do basic analysis for complexity such as comparing sentence lengths, calculating mean words per clause, examining syntactical complexity, or analyzing lexical sophistication (Gamper & Knapp, 2002). Fluency could be evaluated to some extent through timing of responses and spaced repetitions of prompt/responses. This would be more feasible with systems featuring voice input. Checking for possible use of appropriate idiomatic expressions or other desired collocations is also desirable, as this would provide an opportunity to praise students for risk-taking in the use of more complex language.

Integration. Focus-on-form activities should not be delivered in isolation but be part of an integrated, communication-oriented interface. Just as praising learners for good responses helps to build a positive experience from the start, so too, does embedding grammar exercises within meaningful tasks which are communicatively significant. The exercises might be part of a reading or multimedia unit with ready availability of social networking and collaborative tools. The “drill and kill” exercises of the past were often ineffective not only because they often invite mindless completion but also become students tend to get bored and lose focus.

The tendency towards modularity in the design of ILT's might permit more flexible integration of other online tools and services. In particular, enabling collaboration among students using the same ILT has the potential for students to learn from one another and become more fully engaged in the process. Another desirable enhancement to ILT's would be an improved and more contemporary user interface. A number of ILT's use a Java applet front end with unappealingly generic user controls and text display. Other interfaces use graphics that seem more appropriate in applications for children. In some cases, multiple windows result in a cluttered look that may be confusing for users. Few, if any, ILT's use graphics or animations in feedback; nor do they take advantage of audio or video. These would be especially useful features for elementary level learners, whose L2 reading skills preclude lengthy written feedback. ILT projects typically involve a team of linguists and programmers but not always an instructional designer. Design should not be an afterthought, but an aspect of the system taken seriously from the outset, just as much as the parsing algorithms and programming logic. Students today have considerably higher expectations in this area.

Student reception of the software is likely to be a crucial factor for teachers in deciding whether to use an ILT. Another important consideration for teachers is whether the program can be successfully used in conjunction with the textbook currently being used. Noriko Nagata lists on her Web site the textbooks most frequently used in teaching Japanese in the U.S. and how each can be used with Robo-Sensei. In fact, the organization of Robo-Sensei, by cultural units, makes it easier to envision how the program could be used in virtually any Japanese language learning environment. Here, too, modularity, in this case in content organization, can be a significant benefit. The highlighting of cultural aspects of language learning is another example of a helpful integration of an ILT into a comprehensive online environment.

LANGUAGE CORPORA AND LEARNING GRAMMAR

Most ILT's make use of two different kinds of language corpora: a collection of texts from native speakers, presumably using standard language, and a collection of learner language. Outside of their use in ILT's, language corpora have not been widely used in language learning. There have even been questions raised in recent years about whether such collections of texts can truly be considered examples of "authentic" language use, since the texts do not appear in the context in which they were actually used (Kaltenböck & Mehlmauer-Larcher, 2005). Those intrepid souls who do use corpora in language learning, usually through concordances, do not always have a positive experience to recount, as in a recent report from Sweden which concluded, "Judging from the results of our project it is not obvious that corpora facilitate students' understanding of grammatical principles" (Vannestål & Lindquist, 2007, p. 344). While such forthrightness in an academic report is refreshing, it should not be taken as a wholesale repudiation of corpora use in a language learning environment. In fact, the authors found, as have others using corpora, that some students have very positive experiences.

The consensus seems to point to the benefits of implementing corpora only at intermediate and above levels, and also of providing students with ample pre-task background and training (Chambers & O'Sullivan, 2004). Students beyond the elementary level can work with corpora deductively or inductively: students can be given an example of a structure and be asked to find additional occurrences or could be given sample constructions and assigned to come up with the rules governing their use. There is some evidence that small, more homogeneous corpora work better in learning environments (Braun, 2005; Kaltenböck & Mehlmauer-Larcher, 2005). There is always the danger in searching a large corpus of an overload of information. Having a smaller, carefully selected corpus makes it more feasible for a student to be able to expand from examining a single utterance to viewing that string in the context of the entire text. This aids in promoting one of the principal benefits of using a corpus, namely discovery learning.

Part of the negative experiences some learners have had using corpora has to do with presentation (Vannestål & Lindquist, 2007). The use of corpora can be confusing and frustrating to learners in the form in which they are traditionally presented, the KWIC (key word in context) format. This shows a search term in a line-by-line report with the term surrounded by the text immediately preceding and following it. Because only snippets of text are given, some users may have difficulty in understanding the context in which the term appears. Rather than direct access to the corpora through a KWIC search, some programs provide indirect access by using the corpora as a source from which to draw examples and questions. An experiment in New Zealand uses the Greenstone digital library software to create a repository of texts that forms the basis of a set of language learning activities. The program is designed to teach Business English, and the texts for that purpose were retrieved from a surprising source, Wikipedia. However one may judge the reliability of Wikipedia, it certainly represents actual current language use of a certain genre. There is a tool for automatic retrieval of related Wikipedia articles, called the Wikipedia Miner. Metadata, to be used in automatic generation of exercises for the project, was extracted for each article using the OpenNLP, a set of tools for natural language processing. This clearly does not result in anything like a traditional, richly tagged language corpus. But it serves its purpose, and one could extend such an approach to other Web text collections.

Corpora can be annotated in a very detailed or fairly cursory manner. In either case, much of that work has traditionally been done by hand, in a process that it similar to creating HTML, with words, clauses, and sentences sometimes having multiple identifying tags. New tools like OpenNLP and Callisto semi-automate that process. Other tools that are frequently used include TNT, GATE, and WordSmith Tools. Whether a language teacher can use a corpus or not depends ultimately on whether there is one available for the target language. While there are a number of corpora freely available for English, that number decreases significantly for other languages, especially less commonly taught ones. Fortunately, the availability of easy-to-use annotation tools makes it more likely that more text collections will be gathered and tagged.

One of the recent trends has been a greater interest in the creation and pedagogical use of learner corpora. These are collections of learner texts, which are especially useful if encoded with error annotations. Learner corpora enable detection of key aspects of learner language by identifying typical error patterns. Learner corpora are being used in the creation of learner dictionaries and reference works, which list common errors associated with particular words, expressions, or constructions. There are also on-line tutorials that are built around learner corpora. The iWrite program from Iowa State University features a collection of annotated student essays encoded in XML. Students are able to generate pages which list both examples of errors and corrected versions of the sentences. They are also able to go directly from such a page to view the entire essay with the errors highlighted. Students also have the option of generating worksheets in which errors of a particular kind are highlighted. These are designed to be used within work groups. The eXXELant program, based on the Frida collection of French learner texts, similarly provides access to malformed and corrected versions of lexico-grammatical constructions.

Collecting texts for a learner corpus is not easy for legal and practical reasons. However, they do offer tremendous potential for language learning. One could envision students creating the equivalent of a personal corpus, which collects, like a portfolio, the texts they have written (or collected) and which can then be searched for recall of particular vocabulary or constructions. Student and/or teacher notes could also be included. Such a set-up might enable students to more easily relate new material and structures to those already learned.

OUTLOOK: AJAX ON THE RISE

Five years ago (LLT, 2004), I wrote about dynamic Web page creation, which was in turn an update of a column from 10 years ago (LLT, 1998) on a similar topic. In 2004, XML had come into its own. “Remote scripting” (updating pages without reloads) and “Web services” (XML-based machine-to-machine communication) seemed to hold considerable promise for educational uses. Today remote scripting goes by the moniker of AJAX (asynchronous javascript and XML) and is one of the hottest technologies on the Web. This is true as well of Web services, which are fast becoming the backbone of how the Web operates. Yet language learning applications using either are few and far between. One simple example of an AJAX application is learnhanzi. This is code (consisting of HTML, CSS, and javascript) which displays a set of character flashcards on a page from a server database of over 7000 characters. The characters are loaded on demand but enough are buffered so that the user experiences no delay progressing from one character to the next. The character buffer is garbage-collected once the user progresses, so as not to create memory issues in the browser. Feedback is given when the “Eval” button is pressed.

This is technology that could be quite usefully deployed in both ILT's and corpora-based applications. An advantage of AJAX is the ability to combine CSS and javascript to build quite sophisticated user interfaces. This allows a more quickly understood, easily navigable, and less cluttered user interface than is typically the case in ILT's and concordances. The use of AJAX would potentially solve, for example, the problem Joan-Tomàs Pujolà encountered in imPRESSions where users had to scroll back and forth to consult different sections of a page (Pujolà, 2001). Using CSS and javascript, different sections of the page could be collapsed until needed by the user. The fact that an AJAX application can draw down resources from a server behind the scenes means that database access is freely available without the necessity of waiting for a page to load. For some aspects of server-client communication in ILT's, as well as corpora-based applications, an AJAX interface could help reduce the latency that sometimes occurs.

An example of the potential of AJAX for language learning applications can be seen in the recently released Google AJAX Language API. This is a set of functions which can be integrated into any Web page and feature language (machine) translation, language detection (evaluates a text and returns its probable language), and transliteration (converting from Latin input to other alphabets). Currently 85 languages and 29 translation pairs are supported. Google supplies an example of language detection and a simple Spanish vocabulary exercise. These are very basic examples, but the developers' guide gives a sense of some of the directions one could pursue with such a tool. It also demonstrates how easy it is to include the functionality of the API's by simply copying some CSS and HTML code and inserting references to a javascript library. This gives a possibility of integrating AJAX applications into any Web page. AJAX applications also run on many mobile phones, making it a better choice for mobile delivery than Flash or Java, which are less widely supported.

To help developers of ILT's be as responsive as possible both to technology advances and user needs, it would be helpful to see more objective evaluations of those systems. Most of the detailed information and studies that are available comes from the developers themselves. It would be useful to have third parties conduct studies of the effectiveness of ILT's in comparison with other language learning options. Needed as well are studies of the usefulness of combining ILT's with other tools and services. There are quite a few recent studies of corpora use, which are very helpful in understanding the pros and cons of using corpora in language learning. This kind of balanced assessment would be welcome for ILT's. Informed reviews might help to build a consensus about optimal structure and functionality of ILT's, which could aid in sharing and interchangeability.

REFERENCES

Braun, S. (2005). From pedagogically relevant corpora to authentic language learning contents. *ReCALL*, *17*(1), 47-64.

Chambers, A., & O'Sullivan, I. (2004). Corpus consultation and advanced learners' writing skills in French. *ReCALL*, *16*(1), 158-172.

Dodigovic, M. (2005). *Artificial intelligence in second language learning.* Clevedon, UK: Multilingual Matters Limited.

Gamper, J., & Knapp, J. (2002). A review of ICALL systems. *Computer Assisted Language Learning*, *15*(4), 329-342.

Heift, T. (2004). Corrective feedback and learner uptake in CALL. *ReCALL*, *16*(2), 416 - 431.

Heift, T. (2005). Inspectable learner reports for web-based language learning. *ReCALL*, *17*(1), 32-46.

Kaltenböck, G., & Mehlmauer-Larcher, B. (2005). Computer corpora and the language classroom: On the potential and limitations of computer corpora in language teaching. *ReCALL*, *17*(1), 65-84.

Murphy, P. (2007). Reading comprehension exercises online: The effects of feedback, proficiency and interaction. *Language Learning & Technology*, *11*(3), 107-129.

Nagata, N. (2002). BANZAI: An application of natural language processing to web-based language learning. *CALICO Journal*, *19*(3), 583-599.

Pujolà, J-T. (2001). Did CALL feedback feed back? Researching learners' use of feedback. *ReCALL*, *13*(1), 79 - 98.

Skehan, P. (2003). Focus on form, Tass, and technology. *Computer Assisted Language Learning, 16*(5), 391-411.

Vanneståla, M. & Lindquist, H. (2007). Learning English grammar with a corpus: Experimenting with concordancing in a university grammar course. *ReCALL*, *19*, 329-350.

Wu, S., & Witten, I.H. (2006) Towards a digital library for language learning. *Proceedings of the European Conference on Digital Libraries* (ECDL 2006), 341-352.

RESOURCE LIST

Intelligent Language Tutors & Related Projects

- E-Tutor
- Spanish for Business Professionals
- TAGARELA
- imPRESSions
- Freetext
- Satzfee
- Robo-Sensei :: home
- ELISA - English Language Interview Corpus as a Second-Language Learning Application
- ELDIT - Elektronisches Lern(er)wörterbuch Deutsch Italienisch

Corpora Tools

- Callisto
- TNT
- WordSmith Tools
- GATE, A General Architecture for Text Engineering
- Glosser-WeB