Dynamic Language Learning Tools

Lee Schwartz, Takako Aikawa, and Michel Pahud

Natural Language Processing & Learning Sciences and Technology, Microsoft Research, Redmond, WA, USA
leesc@microsoft.com, takakoa@microsoft.com, mpahud@microsoft.com

Abstract

In this paper, we present prototypes of individualized language learning tools. We show how Natural Language Processing (NLP) technology developed at Microsoft Research (MSR) facilitates the dynamic creation of language exercises and provides new user experiences in language learning, while raising the bar for educational technology. We demonstrate how our technology, with its rich linguistic resources, enables the dynamic creation of language exercises which are individualized and user-centric. We present some user scenarios to illustrate the flexibility and unlimited potential of our language learning tools. We show that with innovative ideas and vision, our NLP technology can shed new light on the way technology is used in language learning/teaching.

1. Introduction

Materials for learning and teaching languages abound on the web and in software packages. However, either they are static, and therefore dull, or they are active (generally in the form of games), but extremely limited. Materials available on the web and in software packages are not "tools" for language learning/teaching if they are hard to use both from the teacher's point of view and from the student's point of view. How can teachers incorporate canned materials into their curriculum? Do they have to rewrite their curriculum based on them? Teachers have neither the time nor the tolerance for that. What about students? Can they use limited and randomly selected materials in a systematic way to learn a language? Natural language is, after all, infinite; and learning language(s) involves a cumulative knowledge of grammar. Canned material gives users neither control nor flexibility. This is a primary reason why such educational materials may not attract educators and students. We argue that the language learning tools explored in this paper can overcome such issues and can serve as "dynamic", "unlimited", and "customisable" tools that users can adapt based on their needs and purposes.

2. A Brief Overview of NLPWin

Our NLP system, called "NLPWin" [1], is a natural language processing system that has been developed at MSR over the last 10 years. [1] NLPWin consists of the following major components:

- *Lexical Processing* includes tokenization, word segmentation, morphological analysis of words, identification of multi-word entries, and dictionary lookup.
- *Syntactic Processing* is the parsing of sentences to produce syntactic descriptions of the various segments therein.
- *Logical Form (LF)* specifies semantic relationships among the various segments of a sentence.
- *Generation* produces a sentence string from an LF.
- *Alignment* is a part example-based, part statistical process for extracting corresponding LF segments of two languages based on bilingual corpus data; it is used for our Machine Translation (MT) system [2].

It should be noted here that NLPWin can handle a variety of languages (e.g., Spanish, Japanese, French, German, etc.). Although in this paper we focus on language learning tools for English, Spanish, and Japanese, all of our tools are scalable to all NLPWin languages. We also have MT systems, along with large quantities of aligned corpus data. The tools presented in this paper utilize the rich linguistic resources of NLPWin.

3. Two Generators based on NLPWin

3.1. Content Question Generator

Our LF represents the semantic relationships of the arguments within a sentence. For instance, Figure 1 illustrates the LF of the sentence, "At school, John eats rice every day."

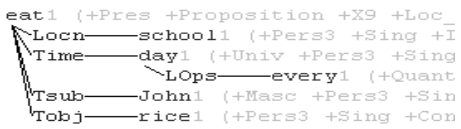


Figure 1: *An LF produced by NLPWin*

From LF, Generation can automatically produce wh-questions (i.e., "*who/what/when/where/why/how*" questions) for a given text with increasing levels of complexity. From the above input, for instance, we can generate the following questions-answer (QA) pairs:

1 NLPWin is used in M icrosoft applications such as spell checking, grammar checking, search, and machine translation (MT).

```
========================================
<Q>At school, what does John eat every day?</Q>
<A>Rice</A>
========================================
<Q>At school, who eats rice every day?</Q>
<A>John</A>
========================================
<Q>Where does John eat rice every day?</Q>
<A>At school</A>
```

Figure 2: *QAs generated from the LF in Figure 1.*

3.2. Grammar Exercise Generator

As mentioned in Section 2, NLPWin contains rich lexical information. Our generation component can utilize lexical information to create grammar-based exercises automatically. For instance, from the simple sentence, "*The man eats rice.*", we can generate numerous variations of that sentence, including those below. We can also present the parts of speech of all the words in a given sentence.

Input: The man eats rice.
Present Passive => Rice is eaten by the man.
Present Perfective=> The man has eaten rice.
Present Progressive=> The man is eating rice.
Past => The man ate rice.
etc.

Figure 3: *Verb Inflection Exercise*

Input: The man eats rice.
Plural Subject => The men eat rice.
Plural Object => The man eats rice.

Figure 4: *Noun Inflection Exercise*

Word:	Part-of-speech
rice =>	NOUN
eat =>	VERB
man =>	NOUN
the =>	ADJ

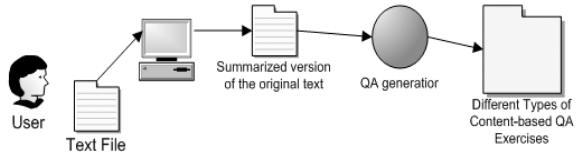
Figure 5: *Part-of-speech Exercise*

4. Dynamic Language Learning Tools

In this section we present prototypes of language learning tools that we created using Content Question Generator and Grammar Exercise Generator described in Section 3. Unlike most language learning materials available on the Web or off the shelf, our tools can take any materials that users want to use and process them on the fly. Nothing needs to be canned; everything is automatic. In this respect, our tools are user-centric. Furthermore, as mentioned in Section 2, all of our tools are scalable to other languages.

4.1 Content QA Maker

This tool automatically creates Wh-questions and provides the answers for these questions by processing input file(s). Figure 6 describes the architecture of Content QA Maker; Figure 7 provides snapshots of the tool. [2]



2. For summarization in Figure 7, we used the summarization function (AutoSummarize-50%) in MS Word.

Figure 6: *Content QA Maker architecture*

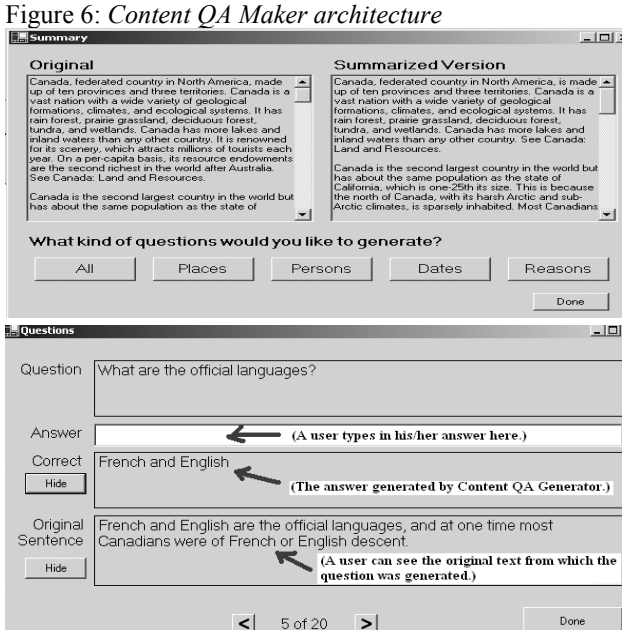


Figure 7: *User Interface of Content QA Maker*

4.2 Grammar Exercise Maker

Using Grammar Exercise Generator described in Section 3.2, we created the tools depicted in Figures 8-9. The Noun-Conjugation Quiz in Figure 8 asks a user to change the number of a nominal argument in a sentence and modify other elements of the sentence, as needed. The Part-of-Speech Quiz in Figure 9 asks for the parts of speech of the words in a given sentence.

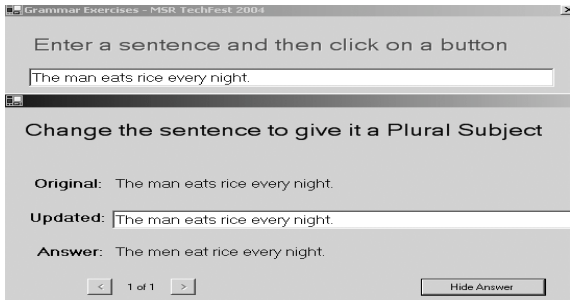


Figure 8: *Noun-Conjugation Quiz*

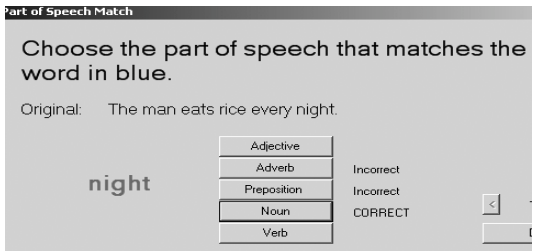


Figure 9: *Part-of-speech Quiz*

4.3 Easy Editor

Our Content Question Generator processes input sentences and spits out wh-questions and their answers. Admittedly, the Generator sometimes produces uninteresting questions, or, if the sentence has been misanalyzed, ungrammatical questions. Therefore, we

provide filtering tools for QAs. Our Easy Editor, illustrated in Figure 10, is such a tool.

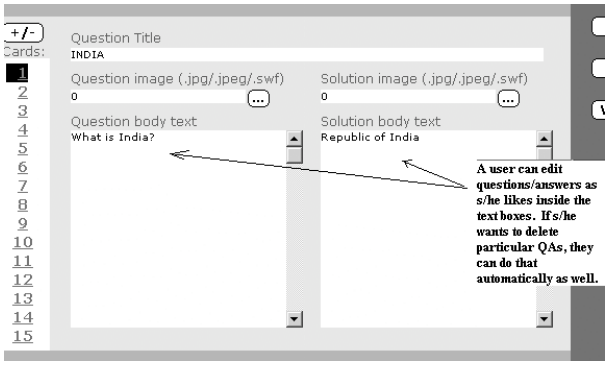


Figure 10: *Easy Editor*

Within the environment of Easy Editor, a user can freely edit or delete QAs created by the Content Question Generator. In this way, humans can correct, delete, or make more relevant the output of our Content Question Generator, thereby having more control over the quality of the quiz materials.

4.4. Flashcard Game

Flashcard Game is a UI that uses the output of the Content Question Generator to create flashcard games, such as that illustrated in Figure 11.

Figure 11: *Flashcard Game*

4.5. NLP-based Game Generator

NLP-based Game Generator is an authoring tool that incorporates graphic objects, NLPWin (i.e., Content Question Generator and Grammar Exercise Generator), and game elements. The architecture of this tool, which generates an NLP-based game from an input statement, is schematized in Figure 12. Snapshots of Language Learning Game are provided in Figure 13.

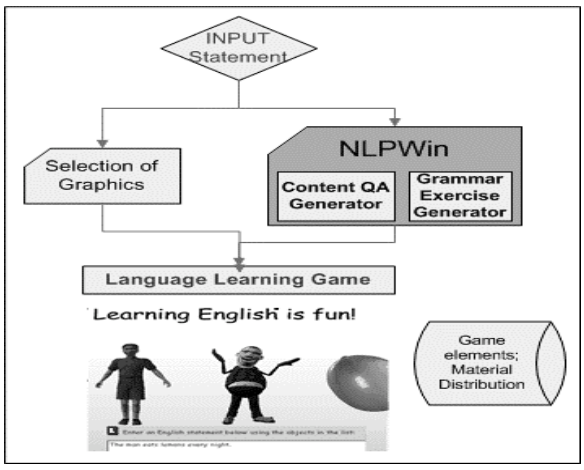


Figure 12: *Overview of NLP-based Game Generator*

Figure 13: *NLP-based GameGenerator*

The games produced by this tool contain multiple-choice questions as well as fill-in-the-blanks grammar exercises. Regarding the multiple-choice questions, it is important to note that the tool selects distracters based on the semantic categories of the arguments of a given sentence. For instance, if a user types in the sentence, "*the man eats rice*", NLPWin spits out the semantic categories of 'human' and 'food' for the subject "man" and the object "lemon", respectively, and passes this information over to the NLP-based Game Generator so that the Game Generator can select "appropriate" distracters for particular questions based on semantic categories of objects in question.

4.6. Context-sensitive Bilingual Word Lookup Tool

The Context-sensitive Bilingual Word Lookup tool is a bilingual dictionary lookup tool that provides contexts as well as meanings of words; it is based on our MT/aligned corpus data. Figure 14 shows the user interface of the Context-sensitive Bilingual Word Lookup tool.

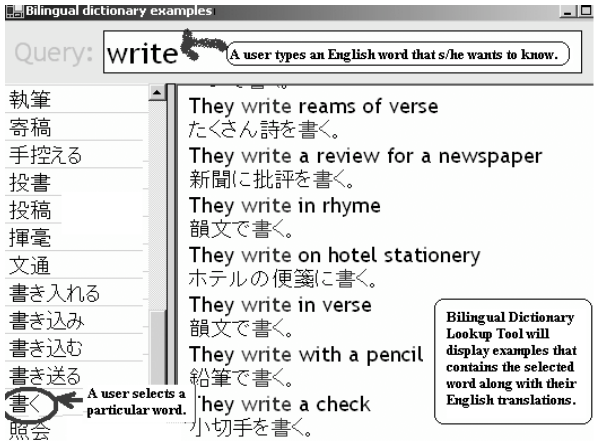


Figure 14: *Context-sensitive Bilingual Word Lookup*

5. User Scenarios

This section provides user scenarios for the language learning tools described in Section 4 that illustrate the flexibility and customizable nature of these tools.

5.1 School Settings

Scenario A: Social Studies Class

A teacher assigns students to read (pre-selected) articles on the web and wants to give them comprehension quizzes to make sure they have understood the materials. Using Content QA Maker/Easy Editor, s/he can create such quizzes easily. Or, as another possibility, students might be asked to submit test

questions to the teacher, in which case they can have Content QA Maker generate questions and they can pool questions with their classmates, selecting the ones they find most relevant.

Content QA Maker is not limited to use by and for the teacher, however. Students can create self-checking quizzes to make sure they really understand whatever they read.

Scenario B: (English/Foreign) Language Class

In a foreign language class (e.g., Spanish class), a teacher assigns students to do inflection exercises (verb tense, noun, adjective, etc.). Such exercises are mechanical, but they are fundamental to learning a foreign language. Our Grammar Exercise Generator can generate such mechanical exercises automatically based on the sentences that a teacher wants to use for his/her class and it provides the correct answers. Or, the student can enter any sentence he wants and find related forms of the sentence.

5.2. Settings outside of school

(Scenario A: Employee Training)

Companies are creating online courses/materials for employee training. Our Content QA Maker is suitable for such environments; it can generate question-answer pairs related to online course materials so that users can create quizzes or test themselves. Or, individuals can just use this tool to quiz themselves. For instance, let's suppose that one is going to have a job interview. S/he would go to some web sites on the company s/he is going to have an interview with for the preparation of his/her interview. S/he can use our tool on such occasions.

6. Challenges

Before making our concluding remarks, we would like to address a couple of issues that we have struggled with in creating these tools. First, limiting/selecting questions created by Content Question Generator is difficult. As mentioned earlier, the determination of what a good question is varies with the users' needs and purposes. What constitutes a good question is something we'd like to investigate from the point-of-view of NLP research. Second, we can create a quiz and provide the answers, but we cannot grade/mark the quiz. Currently, what we can do is simply to present pairs of QAs generated by Content Question Generator. The question of how to grade/mark answer(s) given by a user is difficult to resolve. If the answer to a question is "George Washington", for example, and the user answers "the first president of the U.S.", it would be no small matter to determine that s/he is right. The more freedom we give the user to enter input, the more difficult it is for us to determine the full range of correct answers for any questions we might ask about that input. We will leave these two issues for future research.

7. Concluding Remarks

To conclude, we have presented dynamic language learning tools based on our NLP technology. As mentioned at the outset of the paper, materials for learning and teaching languages are everywhere, yet they are generally canned. Educators and students have no control over what materials to use. Our tools, by contrast, can help them use the material they want in the way that they want. The tools and user scenarios explored in this paper are by no means all of those possible. The imagination is the limit.

Acknowledgements

We would like to express our gratitude to developers in our group; Chris Quirk and Gary Kacmarcik for providing technical support and developing UIs. Also, we are grateful to Manish Mangrulkar for providing Easy Editor/Flashcard Game UIs.

References

[1] Heidorn, G. E. (2000). Intelligence Writing Assistance. In Dale R., Moisl H., and Somers H. (eds*.), A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text.* Marcel Dekker, New York, 1998 (published in August 2000), 181-207.

[2] Menezes A. and Richardson S. (2001). A best-first alignment algorithm for automatic extraction of transfer mappings from bilingual corpora. To appear in *Proceedings of the ACL 2001*, Toulouse, France.