FAST – An Automatic Generation System for Grammar Tests

Chia-Yin Chen
Inst. of Info. Systems & Applications
National Tsing Hua University
101, Kuangfu Road,
Hsinchu, 300, Taiwan
G936727@oz.nthu.edu.tw

Hsien-Chin Liou
Dep. of Foreign Lang. & Lit.
National Tsing Hua University
101, Kuangfu Road,
Hsinchu, 300, Taiwan
hcliu@mx.nthu.edu.tw

Jason S. Chang
Dep. of Computer Science
National Tsing Hua University
101, Kuangfu Road,
Hsinchu, 300, Taiwan
jschang@cs.nthu.edu.tw

Abstract

This paper introduces a method for the semi-automatic generation of grammar test items by applying Natural Language Processing (NLP) techniques. Based on manually-designed patterns, sentences gathered from the Web are transformed into tests on grammaticality. The method involves representing test writing knowledge as test patterns, acquiring authentic sentences on the Web, and applying generation strategies to transform sentences into items. At runtime, sentences are converted into two types of TOEFL-style question: multiple-choice and error detection. We also describe a prototype system FAST (Free Assessment of Structural Tests). Evaluation on a set of generated questions indicates that the proposed method performs satisfactory quality. Our methodology provides a promising approach and offers significant potential for computer assisted language learning and assessment.

1 Introduction

Language testing, aimed to assess learners' language ability, is an essential part of language teaching and learning. Among all kinds of tests, grammar test is commonly used in every educational assessment and is included in well-established standardized tests like TOEFL (Test of English as Foreign Language).

Larsen-Freeman (1997) defines grammar is made of three dimensions: form, meaning, and use (See Figure 1). Hence, the goal of a grammar test is to test learners to use grammar accurately, meaningfully, and appropriately. Consider the possessive case of the personal noun in English. The possessive form comprises an apostrophe and the letter "*s*". For example, the possessive form of the personal noun "*Mary*" is "*Mary's*". The grammatical meaning of the possessive case can be (1) showing the ownership: "***Mary's book is on the table.***" (= a book that belongs to Mary); (2) indicating the relationship: "***Mary's sister is a student.***" (=the sister that Mary has).Therefore, a comprehensive grammar question needs to examine learners' grammatical knowledge from all three aspects (morphosyntax, semantics and pragmatics).
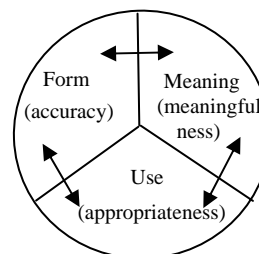

Figure 1: Three Dimensions of Grammar
(Larsen-Freeman, 1997)

The most common way of testing grammar is the multiple-choice test (Kathleen and Kenji, 1996). Multiple-choice test format on grammaticality consists of two kinds: one is the traditional multiple-choice test and the other is the error detection test. Figure 2 shows a typical example of traditional multiple-choice item. As for Figure 3, it shows a sample of error detection question.

Traditional multiple-choice is composed of three components, where we define the sentence with a gap as the stem, the correct choice to the gap as the key and the other incorrect choices as the distractors. For instance, in Figure 2, the

partially blanked sentence acts as the stem and the key "*more than*" is accompanied by three distractors of "*as much as*", "*up as*", and "*as many to*". On the other hand, error detection item consists of a partially underlined sentence (stem) where one choice of the underlined part represents the error (key) and the other underlined parts act as distractors to distract test takers. In Figure 3, the stem is "*Although maple trees are among the most colorful varieties in the fall, they lose its leaves sooner than oak trees.*" and "*its*" is the key with distractors "*among*", "*in the fall*", and "*sooner than.*"

In the Great Smoky Mountains, one can see _____ 150 different kinds of tress.
(A) more than
(B) as much as
(C) up as
(D) as many to

Figure 2: An example of multiple-choice.

Although maple trees are <u>among</u> the most colorful
                      (A)
varieties <u>in the fall</u>, they lose <u>its</u> leaves
      (B)   (C)
<u>sooner than</u> oak trees.
  (D)

Figure 3: An example of error detection.

Grammar tests are widely used to assess learners' grammatical competence, however, it is costly to manually design these questions. In recent years, some attempts (Coniam, 1997; Mitkov and Ha, 2003; Liu et al., 2005) have been made on the automatic generation of language testing. Nevertheless, no attempt has been made to generate English grammar tests. Additionally, previous research merely focuses on generating questions of traditional multiple-choice task, no attempt has been made for the generation of error detection test types.

In this paper, we present a novel approach to generate grammar tests of traditional multiple-choice and error detection types. First, by analyzing syntactic structure of English sentences, we constitute a number of patterns for the development of structural tests. For example, a verb-related pattern requiring an infinitive as the complement (e.g., the verb "*tend*") can be formed from the sentence "*The weather tends to improve in May.*" For each pattern, distractors are created for the completion of each grammar question. As in the case of foregoing sentence, wrong alternatives are constructed by changing the verb "*improve*" into different forms: "*to improving*", "*improve*", and "*improving.*" Then, we collect authentic sentences from the Web as the source of the tests. Finally, by applying different generation strategies, grammar tests in two test formats are produced. A complete grammar question is generated as shown in Figure 4. Intuitively, based on certain surface pattern (See Figure 5), computer is able to compose a grammar question presented in Figure 4. We have implemented a prototype system *FAST* and the experiment results have shown that about 70 test patterns can be successfully written to convert authentic Web-based texts into grammar tests.

I intend _____ you that we cannot approve your application.
(A) to inform
(B) to informing
(C) informing
(D) inform

Figure 4: An example of generated question.

---

* X/INFINITIVE * CLAUSE.
→
* _____ * CLAUSE.
(A) X/INFINITIVE
(B) X/to VBG
(C) X/VBG
(D) X/VB

Figure 5: An example of surface pattern.

2 Related Work

Since the mid 1980s, item generation for test development has been an area of active research. In our work, we address an aspect of CAIG (computer-assisted item generation) centering on the semi-automatic construction of grammar tests.

Recently, NLP (Natural Language Processing) has been applied in CAIG to generate tests in multiple-choice format. Mitkov and Ha (2003) established a system which generates reading comprehension tests in a semi-automatic way by using an NLP-based approach to extract key concepts of sentences and obtain semantically alternative terms from WordNet.

Coniam (1997) described a process to compose vocabulary test items relying on corpus word frequency data. Recently, Gao (2000) presented a system named AWETS that semi-automatically constructs vocabulary tests based on word frequency and part-of-speech information. Most recently, Hoshino and Nakagawa (2005) established a real-time system which automatically generates vocabulary questions by utilizing machine learning techniques. Brown, Frishkoff, and Eskenazi (2005) also introduced a method on the automatic generation of 6 types of vocabulary questions by employing data from WordNet.

Liu, Wang, Gao, and Huang (2005) proposed ways of the automatic composing of English cloze items by applying word sense disambiguation method to choose target words of certain sense and collocation-based approach to select distractors.

Previous work emphasizes the automatic generation of reading comprehension, vocabulary, and cloze questions. In contrast, we present a system that allows grammar test writers to represent common patterns of test items and distractors. With these patterns, the system automatically gathers authentic sentences and generates grammar test items.

3 The *FAST* System

The question generation process of the *FAST* system includes manual design of test patterns (including construct pattern and distractor generation pattern), extracting sentences from the Web, and semi-automatic generation of test items by matching sentences against patterns. In the rest of this section, we will thoroughly describe the generation procedure.

3.1 Question Generation Algorithm

Input: *P* = common patterns for grammar test items, URL = a Web site for gathering sentences
Output: *T,* a set of grammar test items *g*

1. Crawl the site URL for webpages
2. Clean up HTML tags. Get sentences S therein that are self-contained.
3. Tag each word in *S* with part of speech (POS) and base phrase (or chunk). (See Figure 6 for the example of the tagging sentence "*A nuclear weapon is a weapon that derives its and or fusion.*")

Lemmatization: a nuclear weapon be a weapon that derive its energy from the nuclear reaction of fission and or fusion.
POS: a/at nuclear/jj weapon/nn be/bez a/at weapon/nn that/wps derive/vbz its/pp$ energy/nn from/in the/at nuclear/jj reaction/nns of/in fission/nn and/cc or/cc fusion/nn ./.
Chunk: a/B-NP nuclear/I-NP weapon/I-NP be/B-VP a/B-NP weapon/I-NP that/B-NP derive/B-VP its/B-NP energy/I-NP from/B-PP the/B-NP nuclear/I-NP reaction/I-NP of/B-PP fission/B-NP and/O or/B-UCP fusion/B-NP ./O

Figure 6: Lemmatization, POS tagging and chunking of a sentence.

4. Match *P* against *S* to get a set of candidate sentences *D*.
5. Convert each sentence *d* in *D* into a grammar test item *g*.

3.2 Writing Test Patterns

Grammar tests usually include a set of patterns covering different grammatical categories. These patterns are easily to conceptualize and to write down. In the first step of the creation process, we design test patterns.

A construct pattern can be observed through analyzing sentences of similar structural features. Sentences "*My friends enjoy traveling by plane.*" and "*I enjoy surfing on the Internet.*" are analyzed as an illustration. Two sentences share identical syntactic structure {* enjoy X/Gerund *}, indicating the grammatical rule for the verb "*enjoy*" needing a gerund as the complement. Similar surface patterns can be found when replacing "*enjoy*" by verbs such as "*admit*" and "*finish*" (e.g., {* admit X/Gerund *} and {* finish X/Gerund *}). These two generalize these surface patterns, we write a construct pattern {* VB VBG *} in terms of POS tags produced by a POS tagger. Thus, a construct pattern characterizing that some verbs require a gerund in the complement is contrived.

Distractor generation pattern is dependent on each designed construct pattern and therefore needs to design separately. Distractors are usually composed of words in the construct pattern with some modifications: changing part of speech, adding, deleting, replacing, or reordering of words. By way of example, in the sentence "*Strauss finished writing two of his published compositions before his tenth birthday.*", "*writing*" is the pivot word according to the construct pattern {* VBD VBG *}. Distractors for this question are: "*write*", "*written*", and "*wrote*". Similar to the way for the construct pattern devise, we use POS tags to represent distractor generation pattern: {VB}, {VBN}, and {VBD}. We define a notation scheme for the distractor designing. The symbol $0 designates the changing of the pivot word in the construct pattern while $9 and $1 are the words proceeding and following the pivot word, respectively. Hence, distractors for the abovementioned question are {$0 VB}, {$0 VBN}, and {$0 VBD}

3.3 Web Crawl for Candidate Sentences

As the second step, we extract authentic materials from the Web for the use of question stems. We collect a large number of sentences from websites containing texts of learned genres (e.g., textbook, encyclopedia).

3.4 Test Strategy

The generation strategies of multiple-choice and error detection questions are different. The generation strategy of traditional multiple-choice questions involves three steps. The first step is to empty words involved in the construct pattern. Then, according to the distractor generation pattern, three erroneous statements are produced. Finally, option identifiers (e.g., A, B, C, D) are randomly assigned to each alternative.

The test strategy for error detection questions is involved with: (1) locating the target point, (2) replacing the construct by selecting wrong statements produced based on distractor generation pattern, (3) grouping words of same chunk type to phrase chunk (e.g., "the/B-NP nickname/I-NP" becomes "the nickname/NP") and randomly choosing three phrase chunks to act as distractors, and (4) assigning options based on position order information.

4 Experiment and Evaluation Results

In the experiment, we first constructed test patterns by adapting a number of grammatical rules organized and classified in "How to Prepare for the TOEFL", a book written by Sharpe (2004). We designed 69 test patterns covering nine grammatical categories. Then, the system extracted articles from two websites, Wikipedia (an online encyclopedia) and VOA (Voice of American). Concerning about the readability issue (Dale-Chall, 1995) and the self-contained characteristic of grammar question stems, we extracted the first sentence of each article and selected sentences based on the readability distribution of simulated TOEFL tests. Finally, the system matched the tagged sentences against the test patterns. With the assistance of the computer, 3,872 sentences are transformed into 25,906 traditional multiple-choice questions while 2,780 sentences are converted into 24,221 error detection questions.

A large amount of verb-related grammar questions were blindly evaluated by seven professor/students from the TESOL program. From a total of 1,359 multiple-choice questions, 77% were regarded as 'worthy' (i.e., can be direct use or only needed minor revision) while 80% among 1,908 error detection tasks were deemed to be 'worthy'. The evaluation results indicate a satisfactory performance of the proposed method.

5 Conclusion

We present a method for the semi-automatic generation of grammar tests in two test formats by using authentic materials from the Web. At runtime, a given sentence sharing classified construct patterns is generated into tests on grammaticality. Experimental results assess the facility and appropriateness of the introduced method and indicate that this novel approach does pave a new way of CAIG.

References

Coniam, D. (1997). A Preliminary Inquiry into Using Corpus Word Frequency Data in the Automatic Generation of English Cloze Tests. CALICO Journal, No 2-4, pp. 15- 33.

Gao, Z.M. (2000). AWETS: An Automatic Web-Based English Testing System. In Proceedings of the 8th Conference on Computers in Education/International Conference on Computer-Assisted Instruction ICCE/ICCAI, 2000, Vol. 1, pp. 28-634.

Hoshino, A. & Nakagawa, H. (2005). A Real-Time Multiple-Choice Question Generation for Language Testing-A Preliminary Study-. In Proceedings of the Second Workshop on Building Educational Applications Using NLP, pp. 1-8, Ann Arbor, Michigan, 2005.

Larsen-Freeman, D. (1997). Grammar and its teaching: Challenging the myths (ERIC Digest). Washington, DC: ERIC Clearinghouse on languages and Linguistics, Center for Applied Linguistics. Retrieved July 13, 2005, from http://www.vtaide.com/png/ERIC/Grammar.htm

Liu, C.L., Wang, C.H., Gao, Z.M., & Huang, S.M. (2005). Applications of Lexical Information for Algorithmically Composing Multiple-Choice Cloze Items, In Proceedings of the Second Workshop on Building Educational Applications Using NLP, pp. 1-8, Ann Arbor, Michigan, 2005.

Mitkov, R. & Ha, L.A. (2003). Computer-Aided Generation of Multiple-Choice Tests. In Proceedings of the HLT-NAACL 2003 Workshop on Building Educational Applications Using Natural Language Processing, Edmonton, Canada, May, pp. 17 – 22.

Sharpe, P.J. (2004). How to Prepare for the TOEFL. Barrons' Educational Series, Inc.

Chall, J.S. & Dale, E. (1995). Readability Revisited: The New Dale-Chall Readability Formula. Cambridge, MA:Brookline Books.