Chen, W. et al. (Eds.) (2017). Proceedings of the 25th International Conference on Computers in Education. New Zealand: Asia-Pacific Society for Computers in Education

Automatic Question Generation System for English Exercise for Secondary Students

Tasanawan SOONKLANG*, Sunee PONGPINIGPINYO, Weenawadee MUANGON & Sirak KAEWJAMNONG
Department of Computing, Faculty of Science, Silpakorn University, Thailand
*soonklang_t@su.ac.th

Abstract: Automatic Question Generation (AQG) is a research trend that assists teachers to create efficiency assessments. In this paper, we propose a web-based system as a tool to generate English exercises for secondary school students automatically. The system applies NLTK library tags function words. The source texts are selected based on their grammar. The AQG module can generate exercises in ten topics; 1) Noun, 2) Pronoun, 3) Verb, 4) Adverb, 5) Adjective, 6) Comparison, 7) Conjunction, 8) Article, 9) Preposition, and 10) twelve Verb Tenses. There are four types of generated exercises; 1) complete the missing blank, 2) choose the correct answer from two choices, 3) true or false questions, and 4) error correction. The evaluation results show that our proposed system performs effectively with 97.36% F-measure. From the user experience point of view, they are well satisfied with ease of use of the system, including the capability and accuracy to generate a variety of exercises.

Keywords: Automatic Question Generation System, English Exercise, Secondary Students

1. Introduction

English language is essential for communication in non-native English countries. Therefore, it is necessary for people, especially high school students, to comprehend English by practicing. Thus, appropriate exercises are important. In their English class, students can build upon their background knowledge, clear up their confusion and improve reading comprehension by practicing a number of English exercises. The effective questioning strategies can be done using many methods. These exercises are normally created by teachers. With the Automatic Question Generation (AQG) system, exercises can be generated automatically via the computer which can produce a number of questions faster than human experts without losing the assessment quality (Pino et al 2008).

The AQG system generates reasonable questions from an input, which can be structured (for example, a database) or unstructured (for example, a text) (Susanti et al 2015). In this paper, the proposed AQG system is a web-based tool that can generate four types of questions from secondary school English textbooks. The AOG is designed for both teachers and students. The questions are automatically generated and cover ten English grammar lessons for seventh to ninth grade students in Thailand. The ten topics are 1) Noun, 2) Pronoun, 3) Verb, 4) Adverb, 5) Adjective, 6) Comparison, 7) Conjunction, 8) Article, 9) Preposition, and 10) twelve Verb Tenses. The vocabulary test, however, is not included in any topic. The question types are 1) Complete the missing blank whereby an English sentence has a gap to be filled. 2) Choose a correct answer from two choices 3) True or false questions with correction. Rather than just state their answer to be, for example, the false question, students have to correct the false question as well. 4) Error correction whereby the student must amend the incorrect word in the sentence.

The proposed AQG system is capable of checking answers, providing solutions and calculating scores to evaluate the English efficiency of students. As the text is an input, natural language processing performs an important role in our proposed AQG system.

This paper is organized as follows. In Section 2, the current scientific research in QA and QG is introduced. Section 3 describes the proposed QA system. In Section 4, the proposed framework is evaluated and explained with illustrative examples. Finally, the conclusion and future work are encompassed in Section 5.

2. Related Work

In this section, we review approaches to develop question generation on English grammar and vocabulary domain. Several researchers deployed the AQG technology as an educational application. In English education, teachers create exercises for students to learn grammatical knowledge. Chen et al (2008) presented an internet-based system that helps teachers to make cloze tests from online news articles. This system is the assisting tool for multiple-choice and fill-in-the-blank questions which allows teachers to choose the distractors from a system's suggestion or to create them themselves.

Kunichika, et al (2001) proposed RevUP for gap-fill AQG. This system selects important sentences from texts by using sentence-ranking method from a collection of human annotations to select a gap-phrase from each sentence. Amazon Mechanical Turk is used as data for classification to predict the relevant gaps. Moreover, they use the semantic technique to choose distractors similar to the gap-phrase. Becker et al (2012) applied dictionaries of synonyms and antonyms to generate questions for grammar and reading comprehension assessment. This generates five types of questions: 1) asking about the content of one sentence, 2) antonyms and synonyms, 3) modifiers appearing in plural sentences, 4) asking about the contents represented by plural sentences with relative pronouns, and 5) asking about time and space relationship. This research used syntactic and semantic techniques to extract information from an original text. They use syntactic trees to label parts of speech and modify word relationship. Moreover, Becker et al's system employs the feature structures to extract grammar. Flesch (2016) proposed generating questions from text, which helps children in grade 1-3 to understand contents. The system generates questions from the situation model of text, which is constructed by schema-building rules. The situation model is an intelligent method to be used with the appropriate mental age required to transform the sentence into a question.

Brown et al (2005) proposed the REAP system that generates many types of vocabulary questions, including definition, synonym, antonym, hypernym, hyponym, and cloze questions, by using WordNet (Fellbaum, 1998). Gap-fill questions are generated with three stages: sentence selection, key selection and distractor selection. Hoshino and Nakagawa (2007) presented gap-fill questions by choosing the informative sentences from the documents. Moreover, they applied syntactic and lexical features in the process of distractor selection. Narendra et al (2013) presented automatically generating English vocabulary tests. The method consists of four components: target word, reading passage, correct answer and distractor. The target words are taken from web texts and used in reading passages. Then both correct answers and distractors are generated from WordNet lexical dictionary. Kumar et al (2015) presented an automatic cloze question generation system that generates a list of important cloze questions from English articles. They proposed a semi-structured approach. Firstly, for example, knowledge from a Cricket portal is extracted as the summarized results. Then, the top-ten-ranking sentences are selected. Lastly, the meaningful distractors are chosen from the knowledge base.

3. Methodology

3.1 System Architecture

This web-based application is designed for both teachers and students. Teachers can use the system to create the English grammar exercises for students in secondary school. The questions can be generated automatically from source English texts. The topics covered ten English grammar lessons for seventh grade to ninth grade in Thailand. Our system supports four types of question. Table 1 describes lists of English topics and our generated question types.

Table 1: Lists of English topics and our generated question types.

Topics	Types of Questions			
	Fill in the blank	**Either/Or choices**	**True/False**	**Error Correction**
Article	✓			
Preposition	✓			
Verb Tense	✓			
Noun	✓	✓	✓	✓
Pronoun	✓	✓	✓	✓
Verb	✓	✓	✓	✓
Adverb		✓	✓	✓
Adjective		✓	✓	✓
Comparison		✓	✓	✓
Conjunction		✓	✓	✓

Since teachers can select text from any source as an input, such as news, documents, reading passages and so on, our system will firstly analyze the suitability of the input text for the students in terms of their reading level. We use the Flesch Reading Ease test (Flesch, 2016) for calculating the readability scores. The formula is as follows:

$$ReadingEase = 206.835 - 1.015\left(\frac{TotalWords}{TotalSentences}\right) - 84.6\left(\frac{TotalSyllables}{TotalWords}\right) \quad (1)$$

We also used CMU Pronouncing Dictionary (CMUdict, 2016) to provide words and their phonemes to compute the total syllables by counting the vowel phonemes of all words. This test rates text on a 100-point scale. The high score means the document is easier to understand. The appropriate score for our students' level is between 60 and 69.

After choosing the proper text, the teacher will select the topics and the question types to build the exercises. Then, the system generates all possible questions corresponding to the selected topics. It is possible that one sentence can create more than one question for the same and different topics. For example, for the input sentence "*He sat down and ate breakfast*", the system will generate two questions of verb tense topic. The first one is "*He ____ down and ate breakfast.* [sit]" and the second one is "*He sat down and ____ breakfast.* [eat]" Therefore, the questions will be approved and marked by the teacher to form the final exercises.

3.2 System Architecture

There are several steps to generate questions and answers. Firstly, source input text is tokenized into sentences and words. Secondly, part of speech (POS) tag is labeled to each word. After that, the sentence with the tags for the target topic will be searched. Then, a question can be created with its correct answer. Finally, incorrect choices can also be generated, if necessary. Figure 1 describes the processing steps to generate questions and answers.

In our implementation, we used NLTK library for tokenizing and tagging. For example, the sentence "*The best time to fish is early or late in the day*" is transformed with tags as "The<DT>, best<NN>, time<NN>, to<TO>, fish<NN>, is<MD>, early<JJ>, or<CC>, late<JJ>, in<IN>, the<DT>, day<NN>". The Penn Treebank tag set is used for labeling tags.

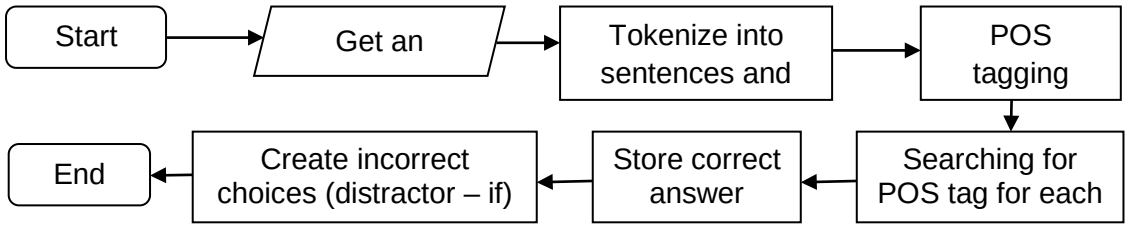

Figure 1. Processing steps to generate questions and answers

In the searching process, we put "#" to cover the word with a target tag in the sentence. For example, if we need a sentence with a coordinating conjunction, the example sentence will be "*The best time to fish is early #or# late in the day*." Then, we store "*or*" as the answer. After that, we create an incorrect choice. There are different techniques based on each topic. In our example for a coordinating conjunction, we randomly select a distractor from the list of vocabularies with a conjunction tag <CC> and a different meaning from the correct word. In this case, the word "*and*" is given as a distractor.

Here is the example question of all four question types from the input sentences.

Topic: Verb tense (past tense)
Input sentence: He sat down and ate breakfast.
Fill in the blank: He ___ down and ate breakfast.

Topic: adjective
Input sentence: It was late at night.
Either/Or choice: It was _____ (*late/lately*) at night.
True or false: It was *lately* at night.
Error correction: It was ~~*lately*~~ _____ at night.

Topic: Pronoun
Input sentence: They are both considered stable production releases.
Error correction: ~~*Theirs*~~ _____ are both considered stable production releases.

4. Evaluation

4.1 Evaluation Results

To evaluate the English exercises that are created from our system, we used 30 reading passages from English textbooks taught in Thailand secondary schools as input text. We measured the performance of the system by using precision, recall and f-measure. Precision is a measure of how precise is the system in generating candidate questions that correctly match with the selected topics. Recall is a measure of how many truly relevant sentences of the chosen topics are returned. F-Measure is the harmonic mean of precision and recall. Let *A* is the number of sentences that the system selects and matches correctly with the topics. Let *B* is the number of sentences that the system selects but which are mismatched with the topics. Let *C* is the number of sentences in a lesson which match the chosen topics, but they are not selected by the system. The precision, recall and f-measure are defined as follows.

$$Precision = \frac{A}{A+B} \tag{2}$$

$$Recall = \frac{A}{A+C} \tag{3}$$

$$F-Measure = \frac{2*Precision*Recall}{(Precision+Recall)} \tag{4}$$

The percentage of precision, recall and f-measure of our experiments of each English topic and the average percentage of the system performance are show in the table 2.

Table 2: Percentage of precision, recall and f-measure of all topics.

Topics	Percentage		
	Precision	**Recall**	**F-Measure**
Article	100.00	100.00	100.00
Preposition	99.67	100.00	99.83
Verb tenses	97.95	75.85	84.63
Noun	98.78	99.13	98.95
Pronoun	100.00	100.00	100.00
Verb.	97.11	95.00	96.04
Adverb	99.05	100.00	99.52
Adjective	99.05	100.00	99.52
Comparison	98.89	100.00	99.49
Conjunction	100.00	91.67	95.65
Average	**99.05**	**96.17**	**97.36**

The user experience of the system is evaluated by ten users who generated exercises and then tested these exercises with ten secondary school students. The result shows that they are well satisfied with the ease of use of the system, the variety of question types, and the capability to select the proper sentences to generate questions.

4.2 Result Analysis

The above results demonstrate satisfactory performance. However, the verb topic has the lowest value in precision score. Also, the twelve verb tenses category has the lowest percentage of recall. After carefully analyzing the results, we found that the lowest precision arises from faulty NLTK tagging. For example, in our case the words that suffix with –ing have to be a noun but NLTK tagged these as a verb. Moreover, some selected sentences do not match with the correct syntax, such as the sentence: “They are fun to play.” The NLTK library tags “fun” as <VBN> refers to the verb (past participle) but it actually serves as an adjective. In another example, “We never hear police sirens or fire sirens”, the extracting system tagged the grammar of the sentence as follows.

```
We<PRP>, never<BB>, hear<JJ>, police<NNS>, sirens<NNS>, or<CC>, fire<VB>,
sirens<NNS>
```

The false result of NLTK tagging function is the word “*hear*” <JJ> as the adjective, but in fact "*hear*" is a verb. Thus this sentence is not selected to generate question. This is the reason that the scores of precision, recall and f-measure were decreased.

Because the NLTK tagging function provided the wrong tag for sentences and made the sentences inappropriate to the verb topic. In order to solve this problem, we analyzed all the mistakes of NLTK including the structure of the sentence, position of the word and so on. It was noticed that the tag type were often wrong in respect of the received results. For example, NLTK tags possessive pronouns as the plural noun <NNS>. Thus, when the user selects the topic for pronoun grammar to generate questions, these sentences are not used. In this case, we modified this by changing the tag to personal pronoun <PRP>.

5. Conclusions

The proposed system AQG was used to generate English Exercise for secondary school students. It is a Web-based application. The developed system applies NLTK library tags function words, and source sentences are selected based on grammar. The experiment we conducted generated exercises from 30 reading passages which showed that our system performed well with 97.36% F-Measure. Word tagging enhances the system to select correct word function and structure of sentences. However, one drawback of using the NLTK library is that this library tool does not tag correct words completely. Therefore, we plan to apply rules and templates to enhance our system performance. In addition, we also plan to develop the system to cover lessons for higher level students.

References

Agarwal, M., and Mannem, P. (2011). *Automatic gapfill question generation from text books*. In Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications, pp. 56-64.

Becker L., Basu, S., Vanderwende L. (2012). *Mind the gap: learning to choose gaps for question generation*. In Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, 2012, pp. 742–751

Brown, J., Frishkoff, G., Eskenazi, M. (2005). *Automatic Question Generation for Vocabulary Assessment.* In Proceeding of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, Association for Computational Linguistics 2005, pp. 819-826.

Chen, W., Aist , G., and Mostow J. (2009). *Generating Questions Automatically from Informational Text*. In S. Craig & S. Dicheva (Ed.), Proceedings of the 2nd Workshop on Question Generation. AIED 2009, pp.17-24

Fellbaum, C. (1998). WordNet. An electronic lexical database. Cambridge, MA: MIT Press London.

Flesch, R. (2016). *How to Write Plain English*, http://pages.stern.nyu.edu/~wstarbuc/Writing/Flesch.htm, Retrieved 10 June 2016.

Hoshino, A., Nakagawa, H. (2007). *Assisting cloze test making with a web application*. In Proceedings of Society for Information Technology and Teacher Education International Conference. Chesapeake, VA, 2007. AACE., pp. 2807- 2814

Kumar, G., Banchs R. E.and D'Haro, L. (2015). RevUP: *Automatic Gap-Fill Question Generation from Educational Texts*. In Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications, 2015, pages 154–161.

Kunichika, H., Katayama, T., Hirashima,T. and Takeuchi, A. (2001). *.Automated question generation methods for intelligent English learning systems and its evaluation*. In Proceedings of the International Conference on Computers in Education ICCE2004, 2001, pp. 1117–1124

Narendra, A., Agarwal, M. & shah, R. (2013), *Automatic Cloze-Questions Generation*., Proceedings of Recent Advances in Natural Language Processing, September 2013, pages 511–515, Hissar, Bulgaria, 7-13.

Pino, J., Heilman, M., and Eskenazi, M., (2008). *A selection strategy to improve cloze question quality*. In Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains. 9th International Conference on Intelligent Tutoring Systems, Montreal, Canada, pp. 22-32.

Susanti, Y., Iida, R., Tokunaga, T. (2015). *Automatic generation of English vocabulary tests*. In Proceedings of the 6th International Conference on Computer Supported Education (CSEDU 2015), pp.77-87, 2015.

The CMU Pronouncing Dictionary. (2016). http://www.speech.cs.cmu.edu/cgi-bin/cmudict, (Retrieved 20 July 2016).

Xuchen Y., Gosse B., Zhang Y. (2012). *Semantics-based Question Generation and Implementation*, Dialogue and Discourse 3(2) (2012), pp 11-42.