ASEE: An Automated Question Answering System for World History Exams

Tao-Hsing Chang
National Kaohsiung University
of Applied Sciences
415, Jiangong Rd., Sanmin Dist.
Kaohsiung 80778, Taiwan
changth@gm.kuas.edu.tw

Yu-Sheng Tsai
National Kaohsiung University
of Applied Sciences
415, Jiangong Rd., Sanmin Dist.,
Kaohsiung 80778, Taiwan
1102108158@gm.kuas.edu.tw

ABSTRACT
This study designed a system called ASEE, which can answer the multiple-choice items provided by the QALab-2 task in NTCIR-12 conference. This system adopts Wikipedia as its knowledge source, using the Stanford Parser to analyze the linguistic features of the items and retrieve key words; it then determines the probability of each option as the correct answer through an algorithm and finally selects the best one. Experimental results shows that the system can correctly answer 21 of 36 questions, which originated from World History B of the National Center Test for University Admissions in Japan in 2011.

Team Name
KUAS

Subtasks
English

Keywords
Question Answering, University Entrance Examination, World History, Multiple-Choice Question

1. INTRODUCTION
Automated question answering systems for entrance exams, a real-world complex Question Answering (QA) technology, are a challenge to develop in a way that provides accurate results, therefore a viable solution has yet to be proposed. These systems analyze items, compare the analyzed results with the known knowledge for inference and judgment, and eventually propose answers to the items. This field is closely related to the area of textual entailment inference and can share findings with various fields of social science and computer science, such as item difficulty analysis, automated item generation, and automated construction for conceptual networks. Owing to the rapid development of natural language processing (NLP) and information retrieval, which has provided many tools and methods that are needed by QA techniques, the development of the systems has become a hot research topic in recent years.

Given that there are various types of exam items, the automatic question answer system developed for different types of items vary accordingly. Gronlund [1] classified item types into four major groups: (1) selected response, such as multiple-choice items or true–false items; (2) supply response, such as short answer questions, fill-in-the-blanks, or essay; (3) restricted performance, which refers to highly structured tasks such as measurement of humidity; and (4) extended performance, which refers to assessments that require more understanding and judgment, such as problems that must be solved with computers. Among all types of items, multiple-choice is the most common and easy to analyze, because it provides sufficient information on the subject, and the correct answer is restricted to one within four given choices.

Our previous studies [2][3] on automated essay scoring showed that models constructed by using multiple linguistic features at the sentence level had performed well in predicting the quality of an essays. The tool [4] of NLP can analyze and determine the part of speech for each word and the syntactic structure of the sentences. Machine learning approaches or rule-based algorithms, which employ the n-gram model of these structural components, are well suited to distinguish the differences between texts. Assuming that each option of an item is considered a different text and the correctness of an option is regarded as the quality of the option, the text difference identification models mentioned in the earlier paragraph may be applied to answer questions.

Based on the aforementioned observations, this study designed a system called ASEE, which can answer the multiple-choice items provided by the QALab-2 task in NTCIR-12 conference. The task collected many items from the National Center for University Admissions on the subject of World History. This system adopts Wikipedia as its knowledge source, using the Stanford Parser to analyze the language structures of the items and retrieve key words; it then determines the probability of each option as the correct answer through algorithms and finally selects the best one.

This paper is organized as follows: the second section reviews previous studies and research related to this subject. Section 3 is a detailed illustration of the four-stage method proposed by our study. The fourth section demonstrates the tested data of our study. The last section discusses the advantages and limitations of the proposed method as well as suggestions for future studies.

2. RELATED WORKS
Utilizing textual entailment inference methods to work directly with automatic question answering for entrance exams is quite an intuitive approach. The NUL system [5] is one example of using the fact validation (FV) problem-solving method, which considers the relation between question–answer pairs as a textual entailment relation. This system employs a question-converting module to convert the question to the t1 sentence of the FV task and its answer as the t2 sentence of the FV task. A previously developed FV task

module is then applied to compute the entailment relation between each question–answer pair. Finally, an answer-conversion module is used to determine the best option based on these relationships. This structure is rather dependent on whether the question conversion module can effectively convert the questions into normalized formats that can be processed by the FV task module. Moreover, the FV task module should achieve reasonably high accuracy.

Kano (2014) [6] proposed a viewpoint targeting the features of an automatic question-answering system employed in history exams: if the input option is in line with the history, then there should be a dense distribution of the keywords in certain extracts of the textbooks. Therefore, his study attempted to use a simple, keyword-based technique to solve the problem. The method consisted of three steps: keyword extraction, keyword weighting, and textbook search and scoring. The correct answer was then identified based on the highest score. This viewpoint is quite reasonable; however, the reason why this simple method is effective lies in the high correlation between the textbooks and the questions. Hence, when applied to a situation with a non-textbook knowledge source, the original method must be modified.

Kimura et al. [7] also developed a method aiming at the characteristics of historical knowledge. Because the questions with which they were working were history related, their method selected the best option based on identification and comparison of historical periods. The study first developed a database of Time Period Beginning points and Time Period End points of the world's historical events. A date identification system was then designed, based on this database, to check the historical period of a question and its options. The option with the highest matching rate of overlapping terms with the question was selected as the best answer. This method can be effectively used to handle chronological questions; however, it cannot be used for other types of questions.

Some systems categorize items into several types and process different types of items with different models and knowledge sources. For example, the FLL system [8] classifies items into various classes and uses a three-solution model to determine the correct answers based on the degree of features of each item. The Forst system [9] presents a model that provides different methods based on the characteristics of different types of items. This system applies five dedicated modules and one common module with basic elements to process 18 item format types. Each model can select different knowledge sources according to the characteristics of the item type and use the similarity between the options and the knowledge source as its reference to identify the answers.

3. METHODOLOGY

Our method can be divided into four stages. The first stage defines the item classification. Given that the item types affect the question-answering strategies, the study developed a simple module to identify the types of items. Next, in stage two, a searching Wikipedia module is used to search relevant Wikipedia articles to compute the correctness of each option. The third stage employed an evaluation formula to compute the validity of each option, according to the Wikipedia article located in stage two. The fourth stage involved an algorithm that compared the validity of different options to find the most likely answer. The following sections will describe the four-stage approach in detail.

3.1 Identifyng Item Types

Figure 1 shows a typical multiple-choice item. Usually, an item contains three components: scenario, stem, and options. A scenario provides instructions related to background knowledge or a description of the situation. A stem describes the question to be answered by respondents. The options are the selectable range of answers from which the respondents can choose. Moreover, there is sometimes underlined text, or text with boldface in scenarios used to indicate key contents, which are called emphasis.

《scenario》

Throughout history, (7) media have been used as a means of spreading (8) political propaganda. For example, during the period before it made Morocco a protectorate, France published a newspaper in Arabic in the town of Tangier, beside the Strait of Gibraltar, as a means of explaining to Moroccanintellectuals its mission and stance in terms of bringing "civilization" to Morocco. On the other hand, amid (9) moves by the Great Powers to expand their influence, forces demanding that the rulers of their countries promote various reforms also printed and distributed pamphlets, which were used as a means of securing support among intellectuals.

《stem》

From (1)-(4) below, choose the one sentence that correctly describes history in relation to the underlined portion (7).

《options》

(1) Morse invented the telegraph.

(2) Arkwright invented wireless telegraphy.

(3) Radio broadcasts began in the United States of America in the latter half of the 19th century.

(4) The internet became prevalent during the first half of the 20th century.

Figure 1: An Example to illustrate the components of multiple-choice items

In the QALab-2 collection, we classify items into five classes: slot-filling items, single-word answer items, combination items, true–false items, and normal items. Normal items refer to items that do not belong to any of the other four types. The characteristics and detecting methods of the other four item types are illustrated as follows:

Slot-Filling Items (SF)

Slot-filling items refer to questions that requires the respondents to fill a slot in a piece of text with a word such that the selected text can correctly describe a historical event. Because all items in the QALab-2 collection are written in the XML format, when there is a slot in the scenario to be filled, a special label is used to tag the blank. Therefore, the system recognizes a slot-filling item based on whether it contains the specific label.

Single-Word Answer Items (SW)

Single-word answer items refer to a multiple-choice item in which each option is only one word. We use the dependency parsing of the Stanford Parser to parse each option. If the parsing result of an option does not contain a nominal subject, direct object, object of a preposition, and conjunction, this option is interpreted as consisting

of a single semantic unit composition (including proper nouns, which are combined by several words). If each option of a question is composed of a single word, the question is marked as a single-word answer item.

True–False Items (TF)

The true–false item refers to a question whose stem usually provides two narratives, and the respondents must define whether either or both of the two narratives is correct. The options are usually the true–false permutations of the two narratives, respectively known as combinations of "true-true," "true-false," "false-true," or "false-false." Because this kind of stem usually contains the phrase "combination of 'correct' and 'incorrect'," if the module detects this text string, it will mark the question as a true–false item.

Combination Items (CB)

The characteristics of a combination item are that several words are provided for each option, and the respondents are required to select the option with the highest combined rate of similarity between each word in the options and the item. Similar to the true–false question, the word "combination" will appear in the stem. Therefore, if the module detects this word and if the item is not a true–false question, it will mark the question as a combination item.

3.2 Searching the Relevant Wikipedia Article

One of the key stages of our approach was to determine whether the contents of the option appeared in a Wikipedia article. Therefore, our method generated a vocabulary set for each option that could be input to the Google search engine (hereinafter referred to as *query set*). Our system would send the query set of each option to Google and locate the link that connects to a Wikipedia article among all links generated by the search results. This article contains most overlapping information between the selected option and Wikipedia as well as the reference to determine whether this option is correct.

However, using only the query sets, generated based on each option, might not accurately help find the information needed. For example, the options in true–false questions contain only the permutations of true and/or false, without any semantic information. Therefore, based on the item types, various methods were employed to generate query sets from the contents of scenario, stem, options, and emphasis. Table 1 shows the source and method of query sets generated for various item types.

Table 1: Source and Method Used to Generate Query Sets for Each Item Type

Item Type	**scenario**	**stem**	**option**	**emphasis**
SF	DP		All terms	
SW			All terms	
CB		DP	All terms	
TF	DP			
Normal			DP	DP

In Table 1, "all terms" means that all words in the source content are included in the query set. "DP" means that the content of components require dependency parsing by the Stanford Parser; after that, only four types of words, which are marked as nominal subject, direct object, object of a preposition, or conjunction will be included in the query set. The "all terms" method is employed because the corresponding components are presented as a word in the question, which means that it can be used directly as the search term. The "DP" method is applied when the corresponding component is presented as a phrase, sentence, or paragraph in the item. In this case, if all words are included directly as the search term, it will result in inaccurate search results, so only four types of words that play practical semantic roles are retained in the search term. The blank cells in the table indicate that the corresponding components are not used in the search term because their contents often do not contain information useful for searching.

In addition, true–false items provide two sentences in the scenario, asking respondents to define whether the statements are true or false. Therefore, for true–false items, the two sentences are treated as two options and the DP method is used to generate a query set for each of the two sentences. Moreover, the system will generate a word frequency table based on the training data. Words that are counted as high-frequency are seen as the "stop words" and are excluded from all query sets.

3.3 Computing Option Validity

After locating the Wikipedia article for each option, the correctness of this option was assessed. Through observation of training data, it was discovered that on the search result article of the query set of the correct option, there was a sentence containing the key verbs and nouns from the option and the corresponding question. Based on this findings, this study utilized the part-of-speech (POS) tagging function of the Stanford Parser to tag the words in the option and the question. It then collected the nouns and verbs from these words, formed a word set, and named it the *comparison set*. Similar to the way we handle query sets, the study used various methods to generate comparison sets from the contents of scenario, stem, options, and emphasis based on different item types. Table 2 shows the source and method of comparison sets generated for various item types.

Table 2: Source and Method Used to Generate Comparison Sets for Each Item Type

Item Type	**scenario**	**stem**	**option**	**emphasis**
SF	N&V		All terms	
SW		N&V	All terms	
CB		N&V	All terms	
TF	N&V			
Normal			N&V	N&V

Most of the symbols in Table 2 have the same definition as those in Table 1. "N&V" presents that the nouns and verbs in the content of the corresponding components are included in the comparison sets, following the POS tagging process. Additionally, for true–false items, the sentence-comparison set was generated by retrieving verbs and nouns from the two narratives in the scenario. Similar to what was applied to query sets, stop words would be excluded from all comparison sets.

After collecting the comparison sets from each option, each sentence on the article could be examined, and the sentence that is the most relevant to the option could be located and named the *relevant sentence*. Given an option c, its comparison set is W. Supposing that all sentences in a Wikipedia article form a set P, the

relevance level *val* (*c*) of option *c* to the relevant sentences can be calculated through Formula (1). Because this value is used to determine whether the option *c* is the correct answer, we call it the validity of option *c*.

$$val(c) = \max_{s \in P}(fwc(s)) \Big/ \|W\| \quad (1)$$

where $\|W\|$ represents the total number of words in the set *W*, and the function of *fwc* can be calculated via Formula (2)

$$fwc(s) = \sum_{k \in W} occ(k, s) \quad (2)$$

where

$$occ(i, j) = \begin{cases} 1, & if\ word\ i\ appears\ on\ sentence\ \ j \\ 0, & otherwise \end{cases} \quad (3)$$

In a given article, after calculation using Formula (1), there may be several sentences with the same validity value as the option. In other words, these sentences are all relevant sentences, and we refer to them as the relevant set. In some of these relevant sentences, the comparison words are densely distributed. In such cases, the sentences must be analyzed further to identify which sentence in the key sentence set is more useful in determining the correctness of the option. We call the sentence with the highest density of comparison words within the relevant sentences set the *identity sentence*. Therefore, for a relevant set *V*, the following formula was employed to locate the identify sentence *S*:

$$S = arg \min_{s \in V}(dist(s)) \quad (4)$$

where

$$dist(s) = \frac{\max_{k \in U}(location(k, s)) - \min_{w \in U}(location(w, s))}{\|s\|} \quad (5)$$

where $\|s\|$ represents the total number of words in a sentence *s*, and *location (i, s)* indicates the location of the word *i* in the sentence *s*. *U* is a set that contains the words that are included in both the sentence *s* and the comparison set.

```
BestOption(x,y)
{
    // x, y are the two among the four options with greater validity
    // x.s is the identity sentence of option x; y.s is the identity
       sentence of option y
    // x.R represents the relevant sentence set of option x; y.R
       represents the relevant sentence set of option y
    // x.s.w is the number of comparison words for identity
       sentence x.s; y.s.w is the number of comparison words for
       identity sentence y.s
    If the difference in validity value between x and y is less than
       the threshold α
    then {
        x.nv = False; y.nv = False;
        while (x.nv is False) and (x.R is not empty) {
            retrieve one sentence h from x.R
            If sentence h contains the nouns and verbs included
               in the comparison set of x,
                then x.nv = True;
                else, remove sentence h from x.R
        }
        while (y.nv is False) and (y.R is not empty) {
            retrieve one sentence h from y.R
            If sentence h contains the nouns and verbs included
               in the comparison set of y,
                then y.nv = True;
                else, remove sentence h from y.R
        }
        if x.nv is True
            then if y.nv is False
                then { x is the best option; return; }
                else, if y.nv is True
                    then { y is the best option; return; }
            if dist(x.s) < dist(y.s)
                then { x is the best option; return; }
                else, if dist(x.s) > dist(y.s)
                    then { y is the best option; return; }
        }

        if the validity of x is larger than that of y
            then { x is the best option; return; }
        if the validity of x is smaller than that of y
            then { y is the best option; return; }

    // dealing with the situation when val(x) = val(y)
        if x.s.w > y.s.w
            then { x is the best option; return; }
            else if x.s.w < y.s.w
                then { y is the best option; return; }
        decide = False; best = x; vc = fwc(x.s);
        while (decide is Fasle) and (vc>0) {
            if fsc(x,vc) < fsc(y,vc)
                then { best = y; decide = True;}
                else, if fsc(x.vc) = fsc(y,vc)
                    then vc = vc-1;
                    else decide = True;
        }
        The option indicated by the best variable is the best option.
}
```

Figure 2: The Algorithm for Determining the Best Option

Based on the above description, although the query sets and comparison sets are generated with similar methods, they have different functionalities, so it is necessary to generate two different sets. The purpose of a query set is to identify the Wikipedia article associated with the option correctly, so the search terms must be those that can make the Google search yield the correct article. Generally, verbs are unsuitable as search words. Conversely, the purpose of a comparison set is to determine the existence of a sentence on an article that can express the meaning of the option, in which case verbs become one of the influencing factors in the validity calculation of the option.

3.4 Defining the Best Option

According to the training data, except for the true–false items, the validity of the correct option for other item types usually ranked within the top two among the four options. Thus, after calculating the validity of each option, we selected the top two options with the highest validity and applied the algorithm illustrated in Figure 2 to determine the best option. The relevant sentence and function *dist* used in the algorithm can be calculated using Formulae (3) and (4), respectively. *fsc(a,i)* represents the total number of sentences in set *S*, and $S = \{s | s \in G \text{ and } fwc(s) = i\}$, where *G* is a set formed by all sentences in the article that was located for option a.

The basic flow of this algorithm is to first analyze the situation in which the validity difference between the two options is below the threshold. If there is no obvious evidence proving that the option with the lowest validity is the best option, then the algorithm directly defines that the option with the larger validity is the best option. The situation in which both validity values are the same is processed in the last part of the algorithm. This algorithm targets only items that require "the selection of the correct option." In case an item requires "the selection of the wrong option," the option with the smallest validity is then selected directly as the best option.

In addition, because the system considers the two narrative sentences of true–false items as two options when calculating validity and then assesses whether the two narratives are correct or not, respectively, it is possible that both narratives are defined as correct or both as incorrect in the results. Because the aforementioned algorithm can determine the narrative of only one option being the correct answer, this algorithm cannot be applied to true–false items. Concerning the true–false items, we adopted a simple rule to analyze the options: if the validity of a narrative is greater than a threshold β, then the narrative is correct; otherwise, the narrative is incorrect.

4. EXPERIMENTAL RESULTS

The collections used in the experiment were taken from the QALab-2 task of the NTCIR-12 Conference, which originated from World History B of the National Center Test for University Admissions in Japan. The training data included the items from 1997, 1999, 2001, 2003, 2005, 2007, and 2009, accounting for 271 items. The experiment data consisted of 36 items, originating from the exam in 2011. All data are in the XML format.

Table 3 shows the total number of correct answers, the accuracy rate, and the number of correct answers and accuracy rate per item type when applying the method proposed by this paper to the experiment data. Because single-word answer questions were included in the training data but not in the experiment data, the accuracy rate of this item type could not be evaluated. Moreover, there was only one combination item in the experiment data, and it was not answered correctly, so the accuracy rate of this item class was 0. There was a limited number of slot-filling items. Owing to the limited number of these two item types, Table 3 cannot be used to directly assess the accuracy of the system when answering questions within these two item classifications.

Table 3: Accuracy Rate of the Experiment

	# of Items	# of Correct Answers	Accurate Rate
Normal	26	17	0.65
SF	3	1	0.33
SW	0	0	N/A
CB	1	0	0.00
TF	6	3	0.50
Total	36	21	0.58

5. DISCUSSION and CONCLUSION

It is believed that there are three major reasons why this system can achieve good results by using a relatively simple method. First, the items have clear and regular descriptive formats and knowledge-presenting methods, without overly complex grammatical structures; therefore, there is a high accurate rate when retrieving search words and comparison words. Second, a considerable number of the items were aimed to test whether respondents understood the contents and authenticity of real historical events; hence, it is easy to evaluate the correctness of options by comparing them with Wikipedia articles, which also have clearly defined content. Third, this system uses Google search to identify relevant Wikipedia articles, and the validity of the search results is quite reliable.

However, there are two essential limitations of this approach. First, this system must use search engines to locate the most relevant Wikipedia articles to the options. Because Google provides search results at a designated time, the search results tend to vary if same search action is to be carried out after a few days. Although in our experiment, we found that the influence of this change on the accuracy of the results was small, it created difficulties when analyzing the efficacy of the system. When the system fails to answer the items using a Wikipedia article, it is difficult to identify whether it is because Google misses the correct Wikipedia article or because such a correct article does not exist at all. Second, for items that require not only a simple judgment of right and wrong but also further inference, this method is unlikely to be of any use. For that reason, this system must employ further textual entailment inference techniques when processing items that require inference before answering. These limitations can provide the direction of future techniques that await further development. Additionally, this method utilized a rule-based algorithm for the selection of the best option. According to the experience of previous studies, the use of machine-learning models to replace this algorithm might achieve better performance.

6. ACKNOWLEDGMENTS

This work is supported in part by the Ministry of Science and Technology, Taiwan, R.O.C. under the Grants MOST 103-2511-S-151-001 and MOST 104-2511-S-151-001-MY3.

7. REFERENCES

[1] Gronlund, N. E. 1998. *Assessment of student achievement*. Allyn and Bacon, Needham Heights, MA.

[2] Chen, Y. Y., Liu, C. L., Chang, T. H. & Lee, C. H. 2010. An Unsupervised Automated Essay Scoring System. *IEEE Intelligent System*, 25(5), 61-67.

[3] Chang, T. H., Liu, C. L., Su, S. Y. & Sung, Y. T. 2014. Integrating Various Features to Grade Students' Writings Based on Improved Multivariate Bernoulli Model. *Information: An International Interdisciplinary Journal*, 17(1), 45-52.

[4] Sung, Y. T., Chang, T. H., Lin, W. C., Hsieh, K. S. & Chang, K. E. 2015. CRIE: An Automated Analyzer for Chinese Texts. *Behavior Research Methods*, in press. DOI: 10.3758/s13428-015-0649-1

[5] Miyashita, H., Ishii, A., Kobayashi, M. & Hoshino, C. 2014. NUL System at QALab tasks. In *Proceedings of the 11th NTCIR Conference*, 556-558, 2014, Tokyo, Japan.

[6] Kano, Y. 2014. Solving History Exam by Keyword Distribution: KJP System at NTCIR-11 QALab Task. In *Proceedings of the 11th NTCIR Conference*, 530-531, Tokyo, Japan.

[7] Kimura, Y., Ashihara, F., Jordan, A., Takamaru, K., Uchida Y., Ototake, H., Shibuki, H., Ptaszynski, M., Rzepka, R., Masui, F. & Araki, K. 2014. Using Time Periods Comparison for Eliminating Chronological Discrepancies between Question and Answer Candidates at QALab NTCIR11 Task. In *Proceedings of the 11th NTCIR Conference*, 550-555, Tokyo, Japan.

[8] Makino, T., Okura, S., Okajima, S., Song, S. & Suzuki, H. 2014. FLL: Answering World History Exams by Utilizing Search Results and Virtual Examples. In *Proceedings of the 11th NTCIR Conference*, 537-541, Tokyo, Japan.

[9] Sakamoto, K., Matsui, H., Matsunaga, E., Jin, T., Shibuki, H., Mori, T., Ishioroshi, M. & Kando, M. 2014. Forst: Question Answering System Using Basic Element at NTCIR-11 QA-Lab Task. In *Proceedings of the 11th NTCIR Conference*, 532-536, Tokyo, Japan.