Mining Gap-fill Questions from Tutorial Dialogues

Nobal B. Niraula
Institute for Intelligent Systems
University of Memphis
nbnraula@memphis.edu

Vasile Rus
Institute for Intelligent Systems
University of Memphis
vrus@memphis.edu

Dan Stefanescu
Institute for Intelligent Systems
University of Memphis
dstfnscu@memphis.edu

Arthur C. Graesser
Institute for Intelligent Systems
University of Memphis
graesser@memphis.edu

ABSTRACT

Gap-fill questions are fill-in-the-blank questions which consist of a sentence with one or more gaps (blanks) and a number of choices for each gap. Such questions play crucial roles in creating test materials and tutorial dialogues. In this paper, we present a system that automatically generates such questions by exploiting previously recorded student-tutor interactions with an Intelligent Tutoring System. Our method is novel because it relies on mining questions' distractors, i.e. tempting incorrect answers, from tutorial dialogues unlike most of the existing approaches that rely on instructional contents. Experimental results show that the proposed system can generate high quality gap-fill questions.

Keywords

Question Generation, Tutoring System, Dialogue Systems

1. INTRODUCTION

Test construction is an expensive and time-consuming process for instructors and educational researchers. Computer-assisted test construction can dramatically reduce costs associated with such test construction activities [8]. As a result, particular attention has been paid by researchers to automatically generate several types of questions such as *gap-fill questions* that can be used in assessment instruments [4]. The more general problem of question generation has been systematically addressed via shared tasks [11].

In this paper, we present a novel method that mines *gap-fill* questions from tutorial dialogues. *Gap-fill* questions are *fill-in-the-blank* questions which consist of a sentence/paragraph with one or more gaps (blanks). Gap-fill questions can be of two types: with alternative options (*key and distractors*) and without choices. The former ones are called *cloze* questions and the latter ones are called *open-cloze* questions. In this paper, we use the term gap-fill questions to refer to the cloze questions. Consider the following *gap-fill* question:

Newton's ____ law is relevant after the mover doubles his force as we just established that there is a non-zero net force acting on the desk then.
(a) third (b) second (c) first (d) heating

One of the options in a *gap-fill* question is the correct answer to the question, called the *key*. The rest of the choices are the *distractors*, i.e. incorrect answers that are tempting less proficient students who often confuse them with the *key*. The question sentence containing gap(s) is also known as the *stem*. In the gap-fill question above, the question sentence contains a gap and there are four potential choices for the gap. The *key* is *second* and *first*, *third* and *heating* are three distractors. Two of distractors are very close to the key while another, *heating*, is quite remotely related.

The attractiveness of gap-fill questions is that they are well-suited for automatic marking because the correct answer is simply the original word corresponding to the gap in the original sentence. Furthermore, gap-fill questions are effective at diagnosing and assessing students' knowledge [5]. Many automatic gap-fill question generation techniques are reported in the literature [7, 12]. These techniques have been successfully used even in large scale evaluations (e.g. TOEFL[1] and TOEIC[2]) to measure learners' proficiency at various tasks, e.g. assessing second language learners' skills of the target language. Gap-fill questions are also important in Intelligent Tutoring Systems (ITSs)[2], a category of advanced educational systems that emphasizes interaction, active learning, and adaptation to the learner. Specifically, ITSs use such questions for assessing students' knowledge level and learning gains as part of their assessment. Furthermore, ITSs use such questions for scaffolding in their practice modules. We explain next the role of gap-fill questions for scaffolding purposes in dialogue-based ITSs.

In dialogue based or conversational ITSs, students typically solve problems (a.k.a. instructional tasks) with the help of the system. That is, during a tutoring session students are prompted to provide complete solutions to various problems. If some of the steps in their solutions are missing, the computer tutor will provide appropriate scaffolding through

[1]http://www.ets.org/toefl
[2]http://www.ets.org/toeic

the use of hints, some in the form of *open-cloze* questions, to help students articulate missing or vague parts. Table 1 shows a fragment of a real student-tutor interaction from the intelligent tutoring system DeepTutor [9]. The first student response (to a previous hint - not shown) is incorrect and therefore the system decides to provide a more informative hint in the form of a *open-cloze* hint/statement/question.

Table 1: A fragment of student-tutor interactions while solving a *task*.

... ...
STUDENT: *Netwon's first law*
TUTOR: Let me give you a hint. The decomposition principle says that the analyses of forces and motion along two _____ directions, such as horizontal and vertical, can be done _____.
STUDENT: *perpendicular, separately*

In this paper, we propose a novel method to automatically generate *gap-fill* questions by exploiting recorded data from massive online education environments such as DeepTutor [10]. In such massive online courses (MOOCs) or massive online ITSs (MOITSs) instructional *tasks or problems* are solved by many students. Consequently, many student responses to hints in the form of questions, some of which are *open-cloze* questions, are collected and recorded in log files. Our approach here exploits this richness of information available in recorded tutorial dialogues from massive online training with ITSs. An advantage of mining these tutorial dialogues is the fact that we have access to actual students answers to *open-cloze* questions. That is, students' responses to these questions are words that they think best fill in the gaps in the *open-cloze* questions. Because not all responses are correct, we consider the incorrect responses as potential candidates for distractors. We rank these candidate distractors to find the best set of distractors. We will show later that this simple idea generates very good distractors for gap-fill questions.

2. RELATED WORKS

Before presenting the most related previous work, we describe the four main steps needed to generate gap-fill questions with choices from instructive texts or content-related documents. Understanding the four main steps will help better appreciate related work. The four main steps are : a) Selecting useful sentences from the text b) Identifying gaps (i.e. words to be deleted) in the selected sentences c) Generating distractor candidate list and d) Ranking the distractors in the list. The literature of gap-fill question generation contains methods that go through each of the steps or focus on particular steps.

Mitkov et al. [4] proposed a computer-aided procedure to generate multiple-choice questions from textbooks that goes through all the four steps. They find key terms by using regular expressions and thresholds. Hypernyms and coordinates of the terms are considered the distractors. The ranking of distractors is done using semantic similarity functions on the assumption that a distractor should be as semantically close to the key as possible. Agarwal and Mannem [1] also go through all four steps to automatically generates gap-fill questions from textbooks for reading comprehension tests.

Hoshino and Nakagawa[3] modeled the problem of generating multiple-choice questions as a learning problem. To decide whether a given word can be left blank in the declarative stem, they trained classifiers using a training data set. The distractors were random words from the same article excluding punctuation and the same word. Sumita *et al.* [13] generated gap-fill questions considering verbs as gaps in a sentence. Thesaurus was used to obtain distractors for the keys of the gaps. To rank distractors, they took each distractor, filled the gap using it, and searched the Web to get the hit counts of the sentence. Smith *et al.* [12] generated cloze questions in English language learning. They used distributional thesaurus to find distractors.

As one may note, most of these solutions require instructional texts such as textbook chapters and encyclopedia entries in addition to thesauri to generate gap-fill questions. Our method is unique because it is based on a generative approach, i.e. the potential distractors are generated by students themselves. Thus, our approach which works by mining questions and distractors from recorded dialogues complements the existing literature. Furthermore, this is the first approach, to the best of our knowledge, that relies on actual student answers to generate distractors.

3. THE METHODOLOGY

Since we do not start with instructional texts but with students' responses to *open-cloze* questions, we only need to generate distractors and rank them in order to generate gap-fill questions.

3.1 Generating Distractor Candidates

Finding plausible distractors that separate knowledgeable students from knowledge-poor students is one of the major challenges for cloze question generation. A good distractor is a concept that is semantically similar at some extent to the key but it is not a correct answer [4].

As already mentioned, we use student responses to *open-cloze* questions during tutorial dialogues as a source of distractors. When same open-cloze questions are answered by many students, there is a large pool of candidate distractors from which to select. We show in Table 2 an open-cloze question together with student responses and their *counts* or *votes*, i.e. the number of students that give the same response as the answer to the same question. Some open-cloze questions may not have enough student responses. In such cases, we follow some of the existing techniques for finding distractor candidates, e.g. we use WordNet as in [4]: extract the hypernyms and coordinated concepts (concepts with the same hypernym) of the key and consider them as the distractor candidates for the key.

3.2 Ranking Distractors

We used the following criteria to rank candidate distractors:
R1: *Use a semantic similarity score between the key and distractors.* This idea was used in the past by Mitkov *et al.* [4]. According to them, a good distractor is very related but not identical to the key. We used a Latent Semantic

Table 2: Students' responses and their frequencies (i.e. votes) to an open-cloze question.

While the wind is blowing, the shape of the sled's path will be _____.		
curved => 4	no => 1	idk => 1
straight => 3	linear => 1	a triangle => 1
diagonal => 2	uhm no => 1	west => 1

Analysis (LSA) based similarity measure [6] to compute the similarity between a key and its distractors.

R2: *Use votes.* We rank the candidates based on their votes/counts (the higher, the better). We break the tie using the semantic similarity score with the key.

4. EXPERIMENTS AND RESULTS

We mined a collection of tutorial dialogues obtained from two of our experiments with the DeepTutor system ([9]). From the first experiment, we extracted tutorial dialogues for 297 students who solved 32 tasks (problems). Since a task was solved by zero or more students, we had 2,687 task sessions altogether (i.e. 9 tasks per student on average). Similarly, from the second experiment, we extracted 4,430 task sessions corresponding to 349 students and 37 tasks (i.e. 13 tasks per student on average). A total of 102 unique single-gap open-cloze questions were also mined. It is noted that some of the questions received a large number of responses while some others only a few. All of the single gap open-cloze questions received at least two responses, 82.85% of the questions received at least three responses, and 72.38% of questions received at least 4 responses.

4.1 Relation between a Response's Similarity and its Rank

We define the frequency rank (FR) of a student response i to a hint in the form of a open-cloze question q as : $FR(i) = \frac{100 * f_i}{\sum f_i}$ where f_i is the number of students who typed i as the answer to open-cloze question q (i.e. votes or counts of i). Then for each response, i.e. which could be either a correct response or candidate distractor, we computed its similarity with the corresponding key as well as its FR score. We discarded the student responses that were misspelled or contained emoticons. We used a small lexicons of emoticons for this purpose. Next, we computed correlation coefficients between the similarities and FR scores at different levels of response frequencies (see Table 3). The correlation coefficients for all responses (i.e. minimum frequency >=1) and for responses generated by at least two students (i.e. minimum frequency >=2) were 0.682 and 0.720 respectively. Similarly, the coefficients for responses with minimum frequencies of 3, 4, and 5 were 0.737, 0.733 and 0.754 respectively.

The positive correlation coefficients indicate that there is clearly a positive relation between the frequency of a response and its semantic similarity score. As we noticed, the correlation coefficients increased as we increased the minimum frequency. These results suggest that ranking student responses by their semantic similarity scores with the key

Table 3: Correlation between *Sim(key,responses)* & *FR(responses)* for responses with *freq >= Min Freq*

Min Freq	1	2	3	4	5
Correlation_LSA	0.682	0.725	0.737	0.733	0.754

can be approximated by their vote counts, i.e. how many students generated the answer. The higher the counts, the more similar the response is to the key. As the distractors for a key should be as semantically close to the key as possible, we can rank the responses by their votes and utilize them as potential distractors.

4.2 Evaluation of Distractor selection

We conducted two evaluations to determine the quality of the distractors generated by our automated method. In each evaluation, we asked two annotators to rate each distractor with one of the following quality ratings: *good*, *ok*, and *bad*. The *good* distractors are ideal distractors, the *ok* distractors can be considered as potential distractors but are not as appropriate as the good distractors. The *bad* distractors do not make sense as a distractor or have the exact meaning with the key.

In the first evaluation, we considered questions that had at least three different student responses and had at least two votes per response. There were 23 questions that satisfied this condition. We ranked the distractor candidates by using *R2* as presented in Section 3.2 and chose the top 3 candidates as distractors. We rejected the candidates if they were synonyms of the key. We considered a key and distractor synonyms when their semantic similarity score was above or equal to 0.9. We also removed duplicate distractors in the final list. To reduce the annotation bias, we introduced a random word from a Wikipedia article as the fourth distractor. The order of the four distractors were randomized.

Next, we asked the annotators to annotate the instances, each consisting of a question sentence, its key and the distractors. A typical annotated instance is showed in the Table 4. The inter-rater agreement using the unweighted version of the Cohen's kappa statistic was 0.64 when we considered *good*, *ok* and *bad* groups separately. It increased to 0.86 when we merged *good* and *ok* groups into a single group. The detailed annotation results are presented in Table 5. The proportion of the good questions is the highest for both annotators. Since we introduced one bad distractor per question and we had 23 questions, we expected at least 23 bad distractors per annotator. Discounting this number in the table, one can notice that we can achieve very good distractors using the voting scheme.

Table 4: Sample Annotation

Question	The force of gravity exerted by the Earth on the cat is ___ all the time.			
Key	constant			
Distractors	relative	horizontal	zero	smile
Annotation	good	good	good	bad

Table 5: Annotation results for 23 questions with 4 distractors each

	good	*ok*	*bad*	*expected bad*
Annotator 1	46	11	35	23
Annotator 2	41	16	35	23

In a second evaluation, we addressed the case when we could not get sufficient distractors for a key due to too few responses available in our tutorial dialogue dataset. We had 100 such questions in our corpora. In such cases, we generated distractor candidates for a question from three different sources: student responses corresponding to the question, different questions with the same key, and WordNet. For each candidate, we checked whether its parts-of-speech matched with that of the key. If matched, we marked the candidate as a potential distractor for the key. Once we had three potential distractors, we stopped. The fourth distractor was a random word from Wikipedia. Annotation results showed that WordNet-based approach could generate distractors out of context. For example, it generated *one*, *two*, and *three* as the three distractors for the key *zero* for the question: *The net force is* ___. The three candidates may look good but for the given question, they are bad distractors. Since the students' responses are mostly contextual,they are preferred over the WordNet-based distractors.

4.3 Error Analysis

The most challenging issue was finding similarities between student answers and the key. Although word pairs such as (*is*, *equals*), (*vertical*, *y-direction*), (*identical*, *constant*) include words with same meaning in the context of Newtonian Physics, LSA failed to find that due to lack of domain knowledge. Use of numbers (e.g. 1st for first, 9.8m/s for constant acceleration) and misspellings of the words (e.g. *seperately* for *separately*, *thirrd* for *third*, *on* for *no*) in students responses were other factors limiting the performance of the proposed approach.

5. CONCLUSION AND FUTURE WORK

We presented in this paper a unique method to generate gap-fill questions. We also proposed different ranking functions to prioritize the list of potential distractor candidates. Since we exploit the students responses corresponding to a problem, our approach would be particularly useful for scalable ITSs and MOOCs and where thousands of students solve the same problem. In future, we exploit the open-cloze questions with multiple gaps to generate more gap-fill questions.

6. ACKNOWLEDGMENTS

This research was supported in part by Institute for Education Sciences under awards R305A100875. Any opinions, findings, and conclusions or recommendations expressed in this material are solely the authors'.

7. REFERENCES

[1] M. Agarwal and P. Mannem. Automatic gap-fill question generation from text books. In *Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 56–64. Association for Computational Linguistics, 2011.

[2] A. C. Graesser, S. D'Mello, X. Hu, Z. Cai, A. Olney, and B. Morgan. Autotutor. In P. M. McCarthy and C. Boonthum-Denecke, editors, *Applied Natural Language Processing: Identification, Investigation and Resolution*, pages 169–187. PA: IGI Global, 2012.

[3] A. Hoshino and H. Nakagawa. 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*, pages 17–20. Association for Computational Linguistics, 2005.

[4] R. Mitkov, L. A. Ha, and N. Karamanis. A computer-aided environment for generating multiple-choice test items. *Natural Language Engineering*, 12(2):177–194, 2006.

[5] R. Mitkov, L. A. Ha, A. Varga, and L. Rello. Semantic similarity of distractors in multiple-choice tests: extrinsic evaluation. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 49–56. Association for Computational Linguistics, 2009.

[6] N. Niraula, R. Banjade, D. Ştefănescu, and V. Rus. Experiments with semantic similarity measures based on lda and lsa. In *Statistical Language and Speech Processing*, pages 188–199. Springer, 2013.

[7] J. Pino, M. Heilman, and M. Eskenazi. 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*, pages 22–32, 2008.

[8] M. Pollock, C. Whittington, and G. Doughty. Evaluating the costs and benefits of changing to caa. In *Proceedings of the 4th CAA Conference*, 2000.

[9] V. Rus, S. D'Mello, X. Hu, and A. C. Graesser. Recent advances in conversational intelligent tutoring systems. *AI Magazine*, 34(3), 2013.

[10] V. Rus, D. Stefanescu, N. Niraula, and A. C. Graesser. Deeptutor: Towards macro- and micro-adaptive conversational intelligent tutoring at scale. In *Work in Progress Learning At Scale*, 2014.

[11] V. Rus, B. Wyse, P. Piwek, M. Lintean, S. Stoyanchev, and C. Moldovan. The first question generation shared task evaluation challenge. In *Proceedings of the 6th International Natural Language Generation Conference*, pages 251–257. Association for Computational Linguistics, 2010.

[12] A. K. S. Smith, P. Avinesh, and A. Kilgarriff. Gap-fill tests for language learners: Corpus-driven item generation. In *Proceedings of ICON-2010: 8th International Conference on Natural Language Processing*, 2010.

[13] E. Sumita, F. Sugaya, and S. Yamamoto. Measuring non-native speakers' proficiency of english by using a test with automatically-generated fill-in-the-blank questions. In *Proceedings of the second workshop on Building Educational Applications Using NLP*, pages 61–68. ACL, 2005.