Generating Comprehension Questions Using Paraphrase

Ya-Min Tseng[1], Yi-Ting Huang[2], Meng Chang Chen[1], and Yeali S. Sun[2]

[1] Institute of Information Science, Academia Sinica, Taipei, Taiwan
{tym,mcc}@iis.sinica.edu.tw
[2] Department of Information Management, National Taiwan University, Taipei, Taiwan
{d97725008,sunny}@ntu.edu.tw

Abstract. As online English learning environment becomes more and more ubiquitous, English as a Foreign Language (EFL) learners have more choices to learning English. There is thus increasing demand for automatic assessment tools that help self-motivated learners evaluate their understanding and comprehension. Existing question generation systems, however, focus on the sentence-to-question surface transformation and the questions could be simply answered by word matching, even without good comprehension. We propose a novel approach to generating more challenging choices for reading comprehension questions by combining paraphrase generation with question generation. In the final evaluation, although there is a slight decrease in the overall quality, our results outperform the baseline system in challenging score and have a significantly smaller percentage of statements that remain intact from the sources sentences.

Keywords: question generation, automatic assessment, reading comprehension, e-learning, multiple choice questions, paraphrase generation, discourse relation.

1 Introduction

Online learning has become a popular choice for English learners. Reading online news and watching talks, for example, are ways to learning English. There are all sorts of learning material on the Internet but there are only a limited number of human quiz creators to provide assessments based on online resources. Automatic assessment tools could help evaluate whether the readers comprehend the text well. Aware of the demand, several Question Generation (QG) systems have focused on the generation of questions for reading comprehension. These work, however, tend to generate simplistic questions with doubtful ability to assess comprehension. The same wording as the source sentences are applied to the questions, like the question "*what is often voted as the best treat in Taiwan?*" and its source "*bubble tea is often voted as the best treats in Taiwan.*" Inevitably, such questions could be solved by searching for the same word spans in the article, even without good comprehension.

The over-simplicity problem might result from two common characteristics of existing QG systems. Firstly, the generating approaches have mostly focused on *wh*-questions or on question stems in the form of *cloze*. Answering these questions only requires a single piece of information, such as a location (*where*-question), a person

S.-M. Cheng and M.-Y. Day (Eds.): TAAI 2014, LNAI 8916, pp. 310–321, 2014.

(*who*-question) and time (*when*-question). On the other hand, due to the fact that in reading comprehension quizzes, the article is usually visible when the test takers attempt to answer the questions, it'd be hard for the automatically generated questions to reflect their comprehension rather than their test-taking skills. Most work concentrate on the surface transformation from declarative sentences to questions and barely discuss how different the resulting questions would look. While these questions are helpful in guiding the reading process and testing elementary English learners, the same might not be for more advanced ones. Self-motivated online learners tend to have higher English proficiency level, which enables them to learn independently without subscribing to any material and without human instructors.

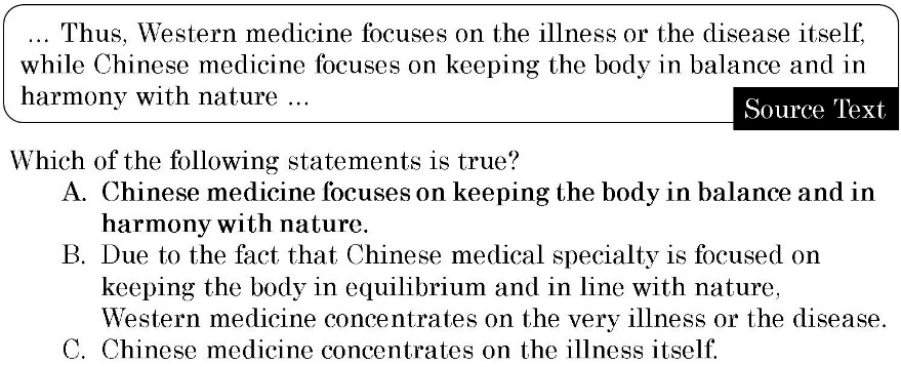

Fig. 1. Example question and choices

We approach the problem by developing generating approach for multiple-choice (non-) factual questions, as Fig. 1. The question form is selected because it's common in formal reading comprehension tests and it could be the container of different question types by casting each question into a statement with its answer. Fig. 1 (A) is transformed from the *what*-question that would be generated by many QG systems: "*what focuses on keeping the body in balance and in harmony with nature?*" along with its answer choice "*Chinese medicine*". We decode the task into generating true/false statements for these choices. By doing so, we could shift our focus from sentence-to-question transformation to increasing the difficulty of test choices. Our aim is to generate choices that test deeper knowledge and look different from the source sentences.

In this work, we present a new approach to generating more challenging choices for multiple choice questions. The novelty of this work lies in how we design choice generation and paraphrase generation towards the mutual goal and how to locate the best-quality choices among numerous variations, nice or erroneous. The Choice Generation System extracts and rewrites the sentences from the question generation aspect. We manually designed transformation rules, which use discourse relations as trigger, to bind up each generated statement with a specified testing purpose. The Paraphrase Generation System then moves on to enlarge the superficial difference by paraphrasing lexically, syntactically and referentially. We merged features from question generation and paraphrase generation to train the Acceptability Ranker, which

determines any choice candidate as either acceptable or unacceptable. In the final evaluation, we conduct an experiment with the baseline system and show the effect of our approach on quality and on difficulty.

The remainder of this paper is organized as follows. Section 2 introduces closely related QG work and explains how our work differs. The generation and ranking of choice candidates are illustrated in Section 3. We do not reveal much implementation detail in this paper due to the page limit, yet any interested reader is referred to [21]. Section 4 gives the setup and the results of the experiments that evaluate our output statements. Finally, in Section 5, we conclude this paper and list possible future work.

2 Related Work

Question Generation (QG) is the task of automatically generating questions from some form of input [20]. When it comes to language learning assessment, automated question generation research are more on grammar and vocabulary. Little work have claimed themselves as aiming at reading comprehension assessment. Mostow and Jang [16] introduced DQGen, a system that automatically generates multiple-choice cloze questions to assess children's reading comprehension. They proposed to diagnose three types of comprehension failures by different types of distractors–grammatical, nonsensical and plausible distractors. In our work, we avoid generating choices that are ungrammatical or do not make sense because, to higher-level learners, they would appear to be obviously wrong choices even without the need to take a look at the article. Heilman [6] proposed a syntactic-based approach to generate factual questions, or wh-questions, from which teachers could select and revise useful ones for future use. In these years, many work (such as [17]) take advantage of domain ontology to create assessments. The generated questions, however, are not based on any input text and are more suitable to test domain-specific knowledge, like the quizzes in science classes.

Generating choices are, partially, equivalent to generating distractors. There is no answer generation in the past because words/phrases in the source sentences of the questions are directly used as answers. Existing distractor generators, as noted by Moser, Gütl and Liu [15], mainly consider single-word choices, or they generate multi-word or phrasal distractors by applying simple algorithms. Mitkov and Ha [14] select multi-word concepts that are similar to the answer from WordNet [13] as distractors and if this fails, phrases with the same head as the answer are selected from a corpus as substitutes. Moser et al. [15] extract key-phrases that are semantically close to the answer as distractors, using LSA for their similarity calculation. Afzal and Mitkov [1] generate distractors for biomedical domain based on distributional similarity. The similarity score is calculated between the answer named entity, which are possibly multi-word, and each candidate from a set of biomedical named entities. The higher scoring ones are more desirable distractors. Different from these approaches, we focus on generating sentential choices. While a small part of our generating approaches is similar to the secondary approach in [14], our approach to generate both answers and distractors via recombination of discourse segments and relations is novel.

Several research have noted the problem caused by the same wording between the generated questions and their source counterparts. Afzal and Mitkov [1] brought up the concern that generating approaches which concentrate on sentence-to-question transformation, are likely to result in questions that could only evaluate test takers' superficial memorization. They solve this problem by generating questions based on semantic relations which are extracted using information extraction methodologies. Bernhard, De Viron, Moriceau and Tannier [3] approached the problem by using two of the many paraphrase skills. They specify the question words and nominalize the verbs. E.g., from "*Where has the locomotive been invented?*" to "*In which country has the locomotive been invented?*" and "*When was Elizabeth II crowned?*" to "*When was the coronation of Elizabeth II?*". On the other hand, Heilman and Smith [7] have developed sentence simplification for question generation based on syntactic rules. Although their work is intended to generate more concise questions, their simplification technique is also contributing to making surface difference. Our work is similar in intentions with most of these work, but paraphrase generation have never been systematically incorporated into these QG systems.

The Penn Discourse Treebank (PDTB) [18] is a large scale corpus based on some early work of discourse structure and is annotated with related information of discourse semantics. A discourse relation captures two pieces of information and the logical relationship between them. Prasad and Joshi [19] evaluated the feasibility of using discourse relations in the content selection of *why*-questions. They showed that the source of 71% of the questions in an independent *why* question answering data set could be found in the same PDTB subset with a marked causal discourse relation. Agarwal, Shah and Mannem [2] followed the proved idea and used discourse cues (e.g., *because*, *as a result*) as an indicator of question type to generate *why*-questions and other question types based on temporal, contrast, concession and instantiation relations. These work suggest the usefulness of discourse relations in QG. While they use discourse relations in satisfying the form of certain question types, our work take advantage of discourse relations in the generation of comprehension questions and the development of distractors.

3 Approach

In this section, we introduce our approach to generate more challenging choices, or statements, for multiple-choice reading comprehension questions. To generate superficially different statements, our intuition is to rewrite with the four basic actions: to rephrase, to reorder, to simplify and to combine. Most paraphrase generation systems, in practice, are inclined to rephrase more often than to simplify or to combine because they do not paraphrase recursively. We improve this by incorporating the structural paraphrases into the design of choice generation rules.

The overall system consists of two sub-systems and a ranker, as shown in Fig. 2. The arrows represent the flows of the generating process and ideally, all these flows should work to satisfy different demand of test choices. In the experiment of this work, only the flow that visits the three components in the order of left to right, from

Choice Generation System, Paraphrase Generation System to Acceptability Ranker, is implemented.

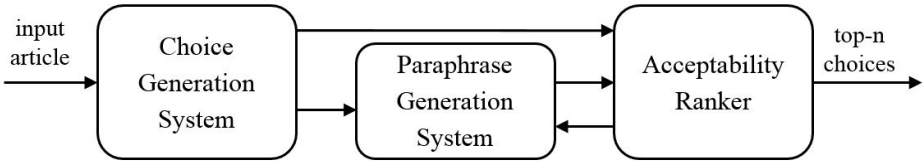

Fig. 2. System architecture

3.1 Choice Generation System

The Choice Generation System takes an article as input text and output a set of statements, each of which with a specific testing purpose. The testing purposes that are considered in this work are: understanding the cohesion of anaphora in the context, understanding the relationship (cause and effect, comparison, etc.) between details and identifying factual information that is explicitly stated in the passages. The overview of this system is given on the left of Fig. 3. In preprocessing, the information from the input article is extracted. The CoreNLP pipeline [11] splits the article into sentences and provides information on coreference chains, part-of-speech tags and syntactic trees. The PDTB-styled end-to-end discourse parser [10] recognizes intra- and inter-sentential discourse relations and the corresponding argument spans. Knowing the three basic elements (two arguments and the relation between them) allows the rules to rearrange them into new statements, with predetermined correctness. Since it's important not to produce vague statements, each pronoun, if not specified in the sentence, is replaced with the representative mention in the same coreference chain.

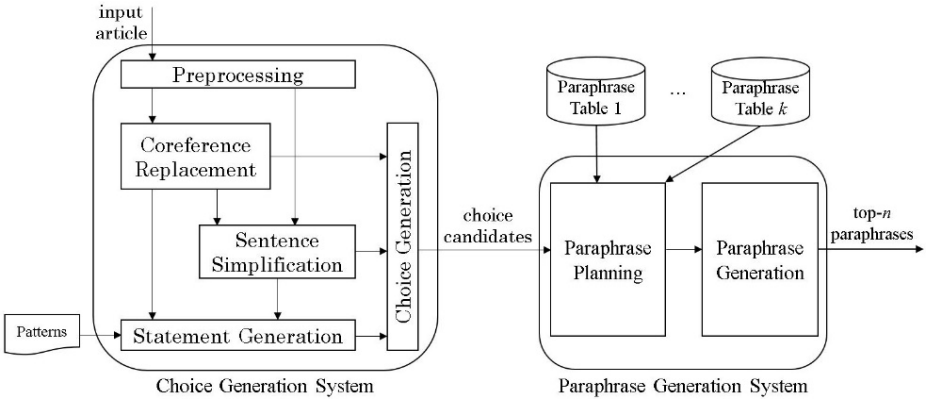

Fig. 3. Overview of the Choice Generation System and the Paraphrase Generation System

These clarified sentences are either sent to choice generation as choice candidates, which are intended to test the cohesion of anaphora, or enter the sentence simplification process. We utilize the sentence simplification work [7] that extracts simplified

statements from complex sentences using a set of hand-crafted Tregex patterns. The simplified statements satisfy the testing purpose of identifying explicitly written fact.

The statement generation matches the source, clarified and simplified sentences with manually-defined rules. The discourse-based rules either recompose pairs of arguments with wrong discourse relations to form false statements or reorder the argument pairs and reunite with other discourse connectives in the same relations to create true statements. If the logical relation stays true, the generated statement is true and vice versa. In the experiment, the rules are applied only if the discourse relations that are involved are explicit because there is still room for improvement in the recognition of implicit discourse relations and because QG is more precision-favored. The relations we include in the transformation rules are: conjunction, cause, contrast, concession, condition and comparison. These allow us to provide more variety to the second testing purposes and to generate choices that test more than one piece of information. The SST (SuperSense Tag) -based rules transform a sentence by replacing a noun/verb phrase by another noun/verb phrase with the same SST for their head words. The generated statements should be plausible but false and should act as choices that assess learners' ability to identify explicitly written fact. A few sample rules are listed in Table 1. Rule #1 and #2 are discourse-based while #3 is SST-based. Fig. 1 (B) is the result of instantiating Rule #1 and Fig. 1 (C) is the statement generated by applying Rule #3. The head of the noun phrases *Chinese medicine* and *Western medicine* are categorized to the same SST (*B-noun.cognition*). The two choices have both been paraphrased whereas Fig.1 (A) has not, which has only undergone simplification and leaves the wording largely the same. The full set of rules can be found in [21].

Table 1. Sample rules

#	Rule	T/F
1	[Arg1] CONTRAST [Arg2] → [Arg2] CAUSE [Arg1]	False
2	[Arg1] CONJUNCTION [Arg2] CONTRAST [Arg3] → [Arg1] CONCESSION [Arg3]	True
3	Sentence={...NP_1…} & SST_{NP1}= SST_{NP2} → Sentence={...NP_2…}	False

3.2 Paraphrase Generation System

This system generates a ranked list of sentential paraphrases given an input sentence and a source article. It enables the overall system to produce lexically different statements and to avoid direct usage of text from the input article that would be easily answerable by word match. The architecture is given on the right of Fig. 3.

Paraphrases are 'sentences or phrases that convey approximately the same meaning using different words' [4]. Abiding by the definition, the correctness should remain unchanged for any true or false statement after paraphrasing. Research on paraphrasing is mainly divided into two lines, paraphrase extraction and paraphrase generation.

Paraphrase extraction focuses on approaches that automatically acquire paraphrases from corpora and paraphrase generation produces paraphrase for any input sentence.

Table 2. Paraphrase resources and likelihood

Alias	Resource	Paraphrase likelihood
PT-1	PPDB lexical paraphrase	$p(t\|s) \approx \sum_e p(t\|e)p(e\|s)$
PT-2	PPDB phrasal paraphrase	
PT-3	PPDB syntactic paraphrase	
PT-4	WordNet synonyms/entailments	e^1
PT-5	Inference rules for predicates	$score_{WT}(LHS \rightarrow RHS, w_x, w_y) = \sqrt{sim(v_l^x, v_r^x, w_x) \cdot sim(v_l^y, v_r^y, w_y)}$
PT-6	Nominal Coreference	Representative mentions: e^1 Other mentions: $e^{0.8}$
PT-7	Self[1]	e^{-1}

Among the many paraphrase generation framework, we favor the idea proposed in [23] to combine multiple paraphrase resources, which allows us to flexibly introduce application-specific resources to the framework. We incorporate pairs of mentions extracted from the same coreference chain as paraphrases, which hasn't been exploited in existing paraphrase generation systems because they do not consider the article information. Besides coreference, resources like the ParaPhrase DataBase (PPDB) [5], WordNet and context-sensitive inference rules for predicates [12] are also included. These resources provide a diversity of paraphrases, from lexical, phrasal, syntactic to referential. For any input sentence, the paraphrase planning phase in Fig. 3 cuts the sentence into segments and transforms them into the search patterns of each resource. It outputs all possible paraphrases for all segments in the input sentence. In the next phase, to form a paraphrased sentence from all possible substitutes, we use a log-linear model [22] to score the combination:

$$\mathrm{p(t|s)} = \sum_{k=1}^{K} \lambda_k \sum_{k_i} \ln \varphi_{PT_k}(\bar{s}_{k_i}, \bar{t}_{k_i}) + \lambda_{lm} \sum_{j=1}^{J} \ln p(t_j | t_{j-2} t_{j-1}) \quad (1)$$

In Equation 1, *s* represents the source sentence and *t* is the target sentence. *K* is the total number of paraphrase tables and *J* is the unit of the *J*-gram language model. $\varphi_{PT_k}(\bar{s}_{k_i}, \bar{t}_{k_i})$ is the sum of the paraphrase likelihood scores of the substitutes for the *i*-th segment that are found in PT-*k*. The likelihood scores for each PT is defined in Table 2. The second part of the addition is the *J*-gram (*J* = 3) language model score of *t* and is retrieved via Microsoft web n-gram services[2]. λ_k and λ_{lm} are the parameters that represent the weights of the sub-scores. The calculation is reduced to the Viterbi algorithm and the top-scoring target sentences can be easily found.

[1] The self-table is created dynamically for each word in the input sentence. This allows words in the sentence to remain unchanged when there is no better substitute.

[2] `http://weblm.research.microsoft.com/`

3.3 Acceptability Ranker

Processed by the Choice Generation System and the Paraphrase Generation System, most source sentences are transformed into various statements with different testing purposes and with different appearances. Obviously, we don't need all these for the final application. A two-way classifier is trained to answer the question, "*can this statement be accepted as a choice?*" The probability scores provided by the classifier should help rank the choice candidates according to its acceptability in an assessment.

The features that the ranker is based on can be grouped into five types by function. We combine features commonly used in QG as well as those that are frequently concerned in paraphrase scoring. Surface features describe the appearance of the choice candidate from the view of grammaticality and length. Vagueness features include features that would tell the vagueness of the sentence. Grammar features [8] are part of the vagueness features because the information of part-of-speech tags and the grammatical structures may suggest how descriptive the sentence is. Transformation rule features capture the inherent accuracy of each transformation rule. Replacement features measure the quality of the replacement by considering the content and the context of the replacing phrase and the replaced phrase. QG challenging features suggest how challenging the choice candidate might be by features that summarize the category and the extent of paraphrasing. There are 90 features in total.

4 Experiment and Results

4.1 Parameter Estimations

The parameters in Equation 1 is estimated according to the settings in [23] and the optimization function in [22] with minor adjustment. The Acceptability Ranker is trained on the data that are partly rated by two human experts. The other part is rated by the workers on Amazon Mechanical Turk[3] (MTurk) service. The human experts worked individually and the ratings of any Turker should correlate with the others to at least a moderate degree on a batch basis. The raters were asked to rate the acceptability on a Likert scale rating, where the definition follows [9]. From 1 to 5, the acceptability score represents bad, unacceptable, borderline, acceptable and good, respectively. We binarize the rating to have scores that exceed 3.5 as acceptable and unacceptable otherwise. We also asked the raters to mark the choices as true or false, given the article.

In total, 10 articles, with 1065 related statements that are generated by our work, are annotated. 200 statements are randomly selected as the held-out test set while the rest are on the training set for logistic regression. The Acceptability Ranker that we trained in this work reflects an accuracy of 0.73 on the test set, as shown in Table 3. Since there is concern that the working quality of Turkers might not be as good as human experts, we also trained the Acceptability Ranker using only the data annotated by the human experts on the training set and the ones by the Turkers, respectively. The former subset hits a higher

[3] https://www.mturk.com/

accuracy of 0.7596 while the sub data set by Turkers reaches a significantly lower accuracy of 0.6875, suggesting that the work done by Turkers might be less consistent.

Table 3. Accuracy of the Acceptability Ranker

	HE+MTurk	*HE*	*MTurk*
Accuracy	0.73	0.7596	0.6875

HE: the data tagged by human experts
MTurk: the data tagged by Turkers

4.2 Experimental Settings

To show the overall performance, we evaluate the top-ranked statements from the view of question generation. The baseline system is proposed by Heilman and Smith [7], which is also intended to facilitate QG and outputs statements. Since the baseline is included in our system as the simplification component, the effect of adding other components could be shown.

Two articles, one from BBC news (22 sentences) and the other from GSAT English 2009 (15 sentences), are randomly selected. They represent different writing styles, one as news report and the other in a more formal way. They are processed by both the baseline and our system into factual statements. Two human experts, graduate students who are non-native English speakers but with high English proficiency, are asked to fulfill half of the rating work. A moderate degree of Pearson correlation coefficient is achieved. The evaluation metrics include *grammaticality* (1–5), *makesense* (1–3), *challenging score* (1–3) and *overall quality* (1–5).

For each article, the baseline generated around 20-35 simplified statements while our system generated over 700 variations. All the statements from the baseline are evaluated. Since these statements cover all source sentences in the input, to make a fair comparison, the top-5 choice candidates for each source sentence are generated by our system for evaluation.

4.3 Experimental Results

If all transformations go well without errors, the transformation rules should determine whether the choice is true or false. A contingency table that summarizes the intended correctness and the actual correctness is shown in Table 4. The statistics are summed up based on the training and the testing data for the Acceptability Ranker. In consideration of the quality of work on MTurk, as Table 3 suggests, we only take the human-annotated data for evaluation in order to obtain more reliable results. Excluding the choice candidates that are unacceptable, 83% of the correctness labels remain identical as planned. For statements that are made to be true, 94% of them are successful. On the contrary, for statements that are designed to be distractors, a lower ratio of 75% is attained. True statements are more likely to maintain their correctness

while false ones, or distractors, may be true when the transformation is based on weak discourse relations or on phrases with similar meaning.

Table 5 shows the evaluation results of the baseline and our system. The baseline system attains better overall quality. This matches what we predicted because our system integrates multiple components, each of which used to be an independent system and has distinctive errors, such as the simplification system, the paraphrase generation system and the question generation system. The errors that these systems bring in would definitely harm the overall quality as well as the grammaticality and the score of make-sense. Still, it's delightful to see that the decrease in these scores is slight and to have made the average difficulty of these choices higher. The challenging score is increased but not as much as we expected. This might be because the discourse-based rules are much less productive than the SST-based ones. The top-5 choices that we evaluated are overwhelmingly occupied the SST-based choices, which are on average not as difficult as those that involve discourse relations.

Table 4. Number of intended and actual TRUE/FALSE

	Actual TRUE	Actual FALSE	Total
Intended TRUE	257 (41%)	16 (3%)	273
Intended FALSE	90 (14%)	264 (42%)	354
Total	347	280	627

Table 5. Extrinsic evaluation results

	Grammaticality (1–5)	Make-sense (1–3)	Challenging score (1–3)	Overall quality (1–5)	Unchanged sentences
Baseline	4.86	2.5	1.2	3.76	38.10%
Our system	4.22	2.39	1.51	3.53	8.57%

The statistics also suggest that our system is generating statements with more variation. The percentage of unchanged sentences is 38.1% for the baseline system while only 8.57% of the sentences in our system output are identical to the source counterparts. Keeping a source sentence intact is sure to produce a grammatically perfect statement, which might be an easy test choice. On the contrary, making most of the source sentences changed should have largely affected the quality and the grammaticality but our Acceptability Ranker has successfully performed to maintain the good quality of the top-ranked choices.

5 Conclusion

In this paper, we presented a novel approach to generate statements for multiple-choice reading comprehension questions. By exploiting discourse relations, our system creates

artificial statements that could test the knowledge of multiple spans of information. We introduced the concept of paraphrase when designing the rules, allowing them to perform paraphrasing actions. The Paraphrase Generation System includes paraphrase resources that are suitable to our system. Particularly, we added QG-specific resource, nominal coreference, to capture the article-wide coreferential relations. Finally, a two-way classifier, the Acceptability Ranker, was trained from an annotated data set generated by our system. We integrated useful features from both rankers for question generation and paraphrase generation. The experimental results suggest that our system are more capable of generating challenging test choices that would not be simply solved by matching exact word span and would be more likely to distinguish those who do not comprehend the reading article well from those who do.

In the future, we plan to investigate the possibility of using implicit discourse relations and incorporate entailment-based rules into our system. We believe that implicit discourse relations would test a higher level of comprehension than explicit ones because the former do not give obvious clues, like connectives. The idea of rewriting a statement while pertaining/changing its correctness conforms to rewriting a statement into another with/without an entailment relationship between them. Entailment is expected to increase the variety of the generated statements. Ultimately, we hope to develop directly applicable question generation system that benefits e-learning environment in the near future.

Acknowledgement. This research was partially supported by National Science Council of Taiwan under grant NSC100-2221-E-001-015-MY3.

References

1. Afzal, N., Mitkov, R.: Automatic generation of multiple choice questions using dependency-based semantic relations. Soft Computing 18, 1269–1281 (2014)
2. Agarwal, M., Shah, R., Mannem, P.: Automatic question generation using discourse cues. In: Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications, pp. 1–9. Association for Computational Linguistics (2011)
3. Bernhard, D., De Viron, L., Moriceau, V., Tannier, X.: Question Generation for French: Collating Parsers and Paraphrasing Questions. Dialogue and Discourse 3(2), 43–74 (2012)
4. Bhagat, R., Hovy, E.: What is a paraphrase? Computational Linguistics 39(3), 463–472 (2013)
5. Ganitkevitch, J., Callison-Burch, C., Van Durme, B.: Ppdb: The paraphrase database. In: Proceedings of NAACL-HLT, pp. 758–764 (2013)
6. Heilman, M.: Automatic Factual Question Generation from Text. Ph.D. Dissertation, Carnegie Mellon University. CMU-LTI-11-004 (2011)
7. Heilman, M., Smith, N.A.: Extracting Simplified Statements for Factual Question Generation. In: Proceedings of QG 2010: The Third Workshop on Question Generation, pp. 11–20 (2010)
8. Heilman, M., Smith, N.A.: Good question! statistical ranking for question generation. In: NAACL-HLT, pp. 609–617 (2010)

9. Heilman, M., Smith, N.A.: Rating computer-generated questions with Mechanical Turk. In: Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, pp. 35–40 (2010)
10. Lin, Z., Ng, H.T., Kan, M.: A PDTB-styled end-to-end discourse parser. Comput. Res. Repository (2011)
11. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP Natural Language Processing Toolkit. In: 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55–60 (2014)
12. Melamud, O., Berant, J., Dagan, I., Goldberger, J., Szpektor, I.: A Two Level Model for Context Sensitive Inference Rules. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pp. 1331–1340 (2014)
13. Miller, A.G.: WordNet: A lexical database for English. Communications of the ACM 38(11), 39–41 (1995)
14. Mitkov, R., An, L.: Computer-aided generation of multiple-choice tests. In: Proceedings of the 1st Workshop on Building Educational Applications Using NLP, HLT-NAACL, pp. 17–22 (2003)
15. Moser, J., Gütl, C., Liu, W.: Refined Distractor Generation with LSA and Stylometry for Automated Multiple Choice Question Generation. In: Thielscher, M., Zhang, D. (eds.) AI 2012. LNCS, vol. 7691, pp. 95–106. Springer, Heidelberg (2012)
16. Mostow, J., Jang, H.: Generating diagnostic multiple choice comprehension cloze questions. In: Proceedings of the Seventh Workshop on Building Educational Applications Using NLP, pp. 136–146. Association for Computational Linguistics (2012)
17. Papasalouros, A., Kanaris, K., Kotis, K.: Automatic Generation of Multiple Choice Questions from Domain Ontologies. In: IADIS International Conference e-Learning, pp. 427–434. IADIS Press (2008)
18. Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A., Webber, B.: The Penn Discourse Treebank 2.0. In: Proceedings of the 6th LREC (2008)
19. Prasad, R., Joshi, A.: A Discourse-based Approach to Generating why-Questions from text. In: Proceedings of the Workshop on the Question Generation Shared Task and Evaluation Challenge (2008)
20. Rus, V., Graesser, A.C. (eds): Workshop Report: The Question Generation Task and Evaluation Challenge, Institute for Intelligent Systems, Memphis, TN (2009) ISBN: 978-0-615-27428-7
21. Tseng, Y.M.: Generating Reading Comprehension Questions using Paraphrase. Master's thesis (to be published), National Taiwan University, Taipei, Taiwan (2014)
22. Zhao, S.Q., Lan, X., Liu, T., Li, S.: Application-driven statistical paraphrase generation. In: Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pp. 834–842 (2009)
23. Zhao, S.Q., Cheng, N., Zhou, M., Liu, T., Li, S.: Combining multiple resources to improve SMT-based paraphrasing model. In: Proceedings of ACL 2008: HLT, pp. 1021–1029 (2008)