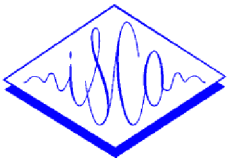

Filtering-based Automatic Cloze Test Generation

Kyusong Lee[1], *Soo-Ok Kweon*[2], *Hae-Ri Kim*[3], *Gary Geunbae Lee*[1]

[1]Department of Computer Science and Engineering,
[2]Division of Humanities and Social Sciences,
Pohang University of Science and Technology, Korea
[3]Department of English Education,
Seoul National University of Education, Korea

{[1]Kyusonglee, [2]soook, [1]gblee}@postech.ac.kr, [3]hrkim@snue.ac.kr

Abstract

We propose a method to generate high-quality cloze test questions using a computational approach. Previous methods for automatic cloze test generation have contained some problems; specifically, there can be multiple correct answers. We found that approximately 50% of the generated answers have such errors with previous methods, which requires human post-editing was necessary in previous research. We propose an N-gram filtering method that can detect the answer to a given question. We compare the errors of the generated questions before and after applying the filtering methods. We found that our filtering method can select quality distractors by reducing errors in the generated questions. Moreover, when we generate cloze tests using semantic similarity, non-native speakers are very hard to answer the questions.

Index Terms: Cloze Test Generation, Sentence Completion Task, Vocabulary Question Generation

1. Introduction

A cloze test is an activity in which students must fill in the blanks in a text with appropriate words. Although this type of test has been widely used in language learning to assess learner's proficiency in the target language, it requires significant labor to create because writing test questions by hand is a laborious task even for experienced teachers. To lessen the burden of test development, automatic question generation techniques [1];[2];[3];[4] have been sought. The goal of these techniques is to provide questions with constant quality and appropriate difficulty that, lead to an objective assessment. Mitkov and Ha proposed an NLP-based methodology for the construction of test items from instructive texts such as textbook chapters and encyclopedia entries [5]. The system for generation of multiple-choice test described in Mitkov and Ha [6] and in [7] was evaluated in a practical environment in which the user was offered the option to post-edit and in general to accept, or reject the test items generated by the system. The formal evaluation demonstrated that a significant fraction of the generated test items needed to be discarded. The primary motivation of our work is that a considerable number of generated questions are not sufficiently good or practical use in language learning. Human post-editing processing is still needed in automatic cloze test generation. Our purpose in this paper is to reduce the cost of the human post-editing step by filtering improper distractors.

In section 2, previous works are introduced. In section 3, we will introduce methods. And then, the experiment result is explained in section 4. Finally, we give a conclusion in section 5.

2. Precious Work

So far, very little work has been devoted to filering distractors. Several studies suggested implementing naïve filtering steps using a the web-based approach [8];[1]. In this approach, distracters are eliminated when sentences receive a non-zero number of hits in a web search because distracters must be incorrect. However, this approach has many problems. First, hits may come from non-native speakers' websites and contain invalid language usage. Second, even if sentence fragments cannot be located on the web, it does not necessarily imply that they are incorrect. Additionally, the number of web searches performed using Google and Bing each month is limited. Grammaticality and collocation to select distractors are used and semantically similar words are removed to avoid multiple answers [4]. However, semantic similarity is also a useful feature for generating cloze items and a mixed strategy demonstrates the best performance as described in [5]. To overcome these limitations, we suggest including a sophisticated filtering step in automatic question generation techniques to improve the acceptance rates of generated test items. If a distractor candidate is considered a possible answer to the question, it is eliminated from the candidate list. The filtering method is independent of candidate distractors selection. Thus, we can deploy more varied features and can apply a mixed strategy to generate cloze items using our filtering method. In this paper, we generated the distractors using semantic similarity for test data sets.

3. Method

3.1. Overview

We propose performing automatic cloze test generation using the following steps (Figure 1). 1) Input the sentence with a blank position 2) We select the distractor candidates using a target based on previous research such as semantic and phonetic similarity [5], synonym or related word using thesaurus extraction [9], WordNet [10], collocation and grammaticality [4] methods, or mixed strategy etc. 3) To make all distractors have the same Part-of-

speech (POS) form, we save the part-of-speech of the target word (answer word); then, we change all words to the same form as the answer word using the English Synthesizer[1]. 4) N-grams can be used to remove potential multiple answers

3.2. Filtering Distractor

N-gram scores are used for filtering the words among distractor candidates. We build a probabilistic language model using the Google Web1T N-gram Count corpus[2], which is built by a huge amount of data. One of baselines is established with Laplace smoothing the N-gram model [11] which avoids zero probability in equation (1).

$$\widehat{\mathrm{P}}_{\mathrm{Laplace}}(\mathrm{w_i}|\mathrm{w_{i-3}w_{i-2}w_{i-1}}) = \frac{\mathrm{C(w_{i-3}w_{i-2}w_{i-1}w_i)}+1}{\mathrm{C(w_{i-3}w_{i-2}w_{i-1})}+|\mathrm{V}|} \quad (1)$$

Zero probability problems are not evitable when considering 5-ngram probability. Backoff N-gram models were introduced by Katz (1987). If the N-gram that we need has zero counts, we approximate it by backing off to the (N-1)-gram, as shown in equation (2).

$$P_{katz}(w_n|\mathrm{w}_{n-N+1}^{n-1}) = \begin{cases} P^*(w_n|\mathrm{w}_{n-N+1}^{n-1}) & if\ C(\mathrm{w}_{n-N+1}^{n}) > 0 \\ \alpha(\mathrm{w}_{n-N+1}^{n-1})P_{katz}(w_n|\mathrm{w}_{n-N+2}^{n-1}) & otherwise \end{cases} \quad (2)$$

However, we recognize that the Google N-gram corpus cannot be used to build previous smoothing methods because of the frequency cut-offs, which implies that only N-grams appearing more than 40 times were kept and appear in the N-gram tables. Some methods need low-order word counts. However, all N-grams with counts lower than 40 were discarded, we cannot use most of previous smoothing methods Calculating normalized factor α in the Katz backoff is also difficult. Our purpose to use N-gram is to find the most proper word among distractor candidates in the question sentence as equation (3).

$$\mathrm{E} = \underset{x \in \{C_0, C_1, .., C_n\}}{\operatorname{argmax}} P(x|Stem) \quad (3)$$

C denotes that distractor candidates.(e.g., C = {addressing, coming, recognizing, greeting, saluting, presenting, receiving, contenting, bidding, accosting}). Stem denotes the words in the question sentence (e.g., Stem = "The manager suggested [ ] the resort guests with a culture themed show after they had finished settling into their private rooms.") (Figure 1). E denotes the potential answers from N-gram filtering model.

Previous N-gram model has low performance (the accuracy is 52%) for the sentence completion challenge (see the results in Table 4). Only half of questions in the close test could be correctly answered by the N-gram approach which implies that it is a challenging task. However, to filter the potential answer words in the distractor candidates, the accuracy of finding the correct answer by N-gram must be much higher than current state-of-art performance. Thus, we improved the performance of N-gram model on sentence completion task by considering more effective features when calculating the N-gram probability. The preceding words and following words are already given in a question sentence, so we can consider the both directions and various

1. Input (Text with Black position)
The manager suggested **[welcoming]** the resort guests with a culture themed show after they had finished settling into their private rooms .

2. Candidates distractors (Target : welcoming)
address, come, recognize, greet, salute, present, receive, content, bid, accost

3. English Synthesizer (Target: welcome, POS: VBG)
addressing, coming, recognizing, greeting, saluting, presenting, receiving, contenting, bidding, accosting

4. Removing multiple answer by N-gram
addressing, coming, recognizing, ~~greeting~~, saluting, ~~presenting~~, ~~receiving~~, ~~contenting~~, bidding, accosting

5. Final Selection
addressing, coming, recognizing, saluting, bidding, accosting

6. Output (cloze Test)
The manager suggested _____ the resort guests with a culture themed show after they had finished settling into their private rooms .
a) addressing b) coming c) accosting d) **welcoming**

Figure 1. Overall Process of Generating Cloze test

ranges. Our proposed for using N-gram model can consider both forward and backward N-gram probability and has much less data sparseness problem for the filtering model. From 5-gram to 2-gram count information are used to develop the proposed N-gram model.

A backoff proposed N-gram model is used for the N-gram filtering model as in equation (4). If the N-gram probabilities we need have all zero counts for every C, we approximate them by backing off to the (N-1)-grams.

$$\mathrm{E} = \underset{x \in \{C_0, C_1, .., C_n\}}{\operatorname{argmax}} P_{\mathrm{N}}(x|Stem) \quad (4)$$

where

$$P_{\mathrm{N}}(x|Stem) = \begin{cases} P_{\mathrm{N}}^*(x|Stem) & if\ not \sum_{i=0}^{n} P_{\mathrm{N}}(C_i|Stem) = 0 \\ P_{\mathrm{N-1}}(x|Stem) & otherwise \end{cases}$$

The model uses a variety of contexts and different sizes and positions to replace the distractor candidates' words in C, where C = { addressing, coming, recognizing, greeting, saluting, presenting, receiving, contenting, bidding, accosting } in Figure 1. We can retrieve a count for each context pattern of length-N with a filler word replacing c_i, which constitutes a single N-gram. We then retrieve a count using the Google 5-gram for sequences

[1] http://www.languagetool.org

[2] Available from the LDC as LDC2006T13.

Table 1: Proposed back off N-gram (Example: The manager suggested $[C_i]$ the resort guests with a culture themed show after they had finished settling into their private rooms.)

5-GRAM	
P_5	P(Ci | the resort guests with) + P(Ci | <s> The manager suggested) + P(suggested Ci |<s> The manager)+ P(Ci the resort | guests with)+ P(manager suggested Ci | <s> The)+ P(Ci the resort guests | with)+ P(The manager suggested Ci | <s>)+ P(suggested Ci | the resort guests)+ P(Ci the | The manager suggested)+ P(suggested Ci the | resort guests)+ P(suggested Ci the | The manager)+ P(suggested Ci the resort | guests)+ P(manager suggested Ci the | The)+ P(manager suggested Ci | the resort)+ P(Ci the resort | manager suggested)+ P(manager suggested Ci the | resort)+ P(suggested Ci the resort | manager)+ P(The manager suggested Ci | the)+ P(Ci the resort guests | suggested)
4-GRAM	
P_4	P(Ci | the resort guests)+ P(Ci | The manager suggested)+ P(Ci the | resort guests)+ P(suggested Ci | The manager)+ P(Ci the resort | guests)+ P(manager suggested Ci | The)+ P(suggested Ci | the resort)+ P(Ci the | manager suggested)+ P(suggested Ci the | resort)+ P(suggested Ci the | manager)+ P(manager suggested Ci | the)+ P(Ci the resort | suggested)
3-GRAM	
P_3	P(Ci | the resort)+ P(Ci | manager suggested)+ P(Ci the | resort)+ P(suggested Ci | manager)+ P(suggested Ci | the)+ P(Ci the | suggested)
2-GRAM	
P_2	P(Ci |the)+ P(Ci | suggested)

including N-grams of length 2 to 5. For each target word c_i, five separate 5-gram context patterns that span its range are found. We describe the notation of P_N in more detail in Table 1).

4. Experiment and Result

4.1. Data

For our proposed filtering strategies, we used Google N-gram corpus which contain 1 trillion words of running text and the counts for all 1 billion five-word sequences that appear at least 40 times. There are 13 million unique words after words that appear less than 200 times are discarded. We used WordNet-based semantic similarity to generate candidate distracters in this paper. For computing the WordNet-based semantic similarity, we employed a popular word similarity measure using the Python NLTK package [12]: Jiang and Conrath's (JCN) measure [13] to generate the distractor candidates. A total 100 questions are generated using WordNet-based semantic similarity. Each question has 10 distractor candidates. Four experts in English Education annotated every generated distractors in the JCN measure regarding whether distractors could be potential answers. Two annotators' native language is English, the others are non-native speakers. Kappa value between two native was 0.7. However, the agreement between non-native speakers and native speakers was 0.409, 0.401, 0.45, and 0.455. Between two non-native speakers, Kappa value is 0.764. Even though they are all experts in English Education, it is a challenging tasks for non-native speakers; the kappa value is about 0.4 (Table 2, Table3). It indicates that questions by semantic similarity could be good test items for identifying native speakers. Moreover, it would be good test items for non-native speakers to teach the real usage of words among the similar meanings. To investigate the performance of the filtering method that uses N-gram model, we use 500 semantic

Table 2. Multiple answer annotations, NonNS1 denotes Non-native speaker 1, NS denotes Native speaker. O means proper distractor, X means potential answer.

Question) Remember to [ ] your complete company information when filling out the tax form .

Answer) include

Distractor Candidates)

	NonNS1	NonNS2	NS1	NS2
bear	O	O	O	O
involve	X	X	O	O
carry	O	O	O	O
embroil	O	O	O	O
admit	O	O	O	O
add	O	O	X	X
hold	O	O	O	O
tangle	O	O	O	O
drag	O	O	O	O
contain	X	X	O	O

Table 3: Kappa value between annotators

	NonNS1	NonNS2	NS1	NS2
NonNS1	1			
NonNS2	0.764	1		
NS1	0.409	0.401	1	
NS2	0.455	0.455	**0.70**	1

questions from TOEIC data[1] and the MSR sentence completion challenge data[2] [14].

4.2. Filtering By N-Gram

We only explore content words (verbs and nouns) in this paper. To evaluate our filtering method, we must select the N-best candidate distractors. For the experiment, we selected the N-best semantically similar words to the target vocabulary scheduled by JCN WordNet based semantic similarity measure in Figure 1, such as *addressing, coming, recognizing, greeting, saluting, presenting, receiving, contenting, bidding,* and *accosting*. The goal of the filtering method is to remove *Greeting, presenting, receiving,* and *contenting* which labeled as multiple answers using N-gram model. Among the 100 test items, each item has 10 distractor candidates, which yields a total 1000 distractors candidates labeled as multiple answers or proper distractors. If a distractor could be an answer in the given text, we count the question as an instance of "multiple answers". The performances are quantified using precision, recall, and F-score. To explore the filtering performance for multiple answers, we deploy the proposed N-gram. Because the number of web searches performed using Google and Bing in a month is limited, it is improper to use those corpora as the baseline system. Thus, the baseline is randomly selected from distractor candidate. We consider two perspectives of the results: one is the proper distractors perspective and the other is filtering perspective.

Proper distractor perspective:

$$\text{Precision} = \frac{\#\text{ of proper distractor in final}}{\#\text{ of distractors in final}}$$

$$\text{Recall} = \frac{\#\text{ of proper distracotrs in final}}{\#\text{ of proper distractors in candidates}}$$

Filtering perspective:

$$\text{Precision} = \frac{\#\text{ of filtered improper distractors in final}}{\#\text{ of total filtered distractors}}$$

$$\text{Recall} = \frac{\#\text{ of filtered improper distractors in final}}{\#\text{ of total improper distractors in candidates}}$$

From the proper distractor perspective, eliminating the proper distractors is not a critical problem when final distractors are all proper distractors which indicate precision is important from this perspective. The reason of the low recall rate is that many proper distractors are also removed until the final number of distractors K is remains (K=the number of Distractors). Therefore, low recall is not a critical problem. However, from the filtering perspective, recall is much more important than precision. If recall is 100%, the filtering method can eliminate every improper distractor. We found that the recall rate is much higher than the baseline which implies that our method filters improper distractors properly (Table 3). We compared the performance after applying filtering model in as shown (Table 4). The 46.8% of generated questions have distractors that are improper for practical use without human editing. After applying our proposed filtering strategies, significant parts of questions are removed the improper distractors.

Table 3. Performance of Proper Distractor Selection

		Precision	Recall	F-Score
Proper Distractor Perspective	PROPOSED	**90.9**	44.64	59.90
	BASELINE	83.51	40.99	54.99
Filtering Perspective	PROPOSED	24.82	**80.45**	37.94
	BASELINE	10.99	35.63	16.80

Table 4. The portion of errors on generated Cloze Test

	Suitable Questions
Baseline	53.2
After Filter	70.2

Table 5. Performance of methods on the TOEIC test

	TOEIC data
Chance	25%
Baseline Laplace Smoothing	61
*Proposed Method	**74.0***

Table 6. MSR sentence completion (SC) performance with the proposed N-gram method

	MSR SC
Chance	20 %
(1) GT N-gram LM	39
(2) LSA- Total Similarity	49
Combination (1) + (2)	52
*Proposed Method	**87.4***

We found a 17% improvement gain after applying our filtering methods.

An essential step in the filtering process is to eliminate potential answers, so we believe testing our method's ability to find the correct answers in sample questions is a useful assessment. Therefore, for additional evaluation, we explore how effective the filtering method is in selecting potential answers, we apply our filtering method on TOEIC questions. In all, 74% of the questions are correctly answered (the fraction that would be answer by chance is 25%, and baseline is 61%) (Table 5). The TOEIC questions that we use consist of all semantically related questions. Moreover, we explore the accuracy for the sentence completion challenge [14] with the proposed N-gram model using the Google corpus. The challenge was designed as a benchmark for semantic models and consists of SAT-style sentence completion problems. Given 1,040 sentences, each of which is missing a word, the task is to select the correct word out of the candidates provided for each sentence. The best result 52% was produced by a combination of latent semantic allocation total similarity and N-gram models (Zweig & Burges, 2011). Our method achieves better performance than these previous the results from [15], as indicated in Table 6. We used the evaluation tool that MS provided. Note that the result

[1] http://www.toeflgoanywhere.org/, data cannot open because of license problem

[2] http://research.microsoft.com/en-us/projects/scc/, The evaluation tool is also available in the website.

of "*proposed N-gram" in Table 6 are only our experimental result, others results such as (1), (2), and the combination (1)+(2) are from the [15] paper. The accuracy of our proposed method is 87.4 %, which is significantly improved. The annotated distractors and experimental results are made available to the public[1].

5. Conclusions

We found that machine generated test items have many errors, such as multiple answers. To solve these problems, we proposed filtering method to remove improper words from candidate distractors. We found that proposed methods significantly reduce the generated test item error rate. Moreover, our method also performs well on sentence completion challenge. We also found that the annotation agreement between native speakers are much higher than between native speaker and non-native speakers. It indicates that questions made by semantic similarity are challenging test items for non-native speakers. We have plan to explore a real student test for evaluations on generated cloze test set.

6. Acknowledgements

Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012-0008835). "This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency)" (NIPA-2012-C1090-1231-0009)

7. References

[1] T. Goto, T. Kojiri, T. Watanabe, T. Iwata, and T. Yamada, "Automatic Generation System of Multiple-Choice Cloze Questions and its Evaluation," *Knowledge Management & E-Learning: An International Journal (KM&EL),* vol. 2, pp. 210-224, 2010.

[2] C. Y. Chen, H. C. Liou, and J. S. Chang, "FAST: an automatic generation system for grammar tests," in *COLING-ACL '06 Proceedings of the COLING/ACL on Interactive presentation sessions*, 2006, pp. 1-4.

[3] J. Lee and S. Seneff, "Automatic generation of cloze items for prepositions," 2007.

[4] 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*, 2008, pp. 22-32.

[5] R. Mitkov, L. A. Ha, A. Varga, and L. Rello, "Semantic similarity of distractors in multiple-choice tests: extrinsic evaluation," 2009, pp. 49-56.

[6] R. Mitkov and L. A. Ha, "Computer-aided generation of multiple-choice tests," 2003, pp. 17-22.

[7] R. Mitkov, L. A. Ha, and N. Karamanis, "A computer-aided environment for generating multiple-choice test items," *Natural Language Engineering,* vol. 12, pp. 177-194, 2006.

[8] 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*, 2005, pp. 61-68.

[9] M. Heilman and M. Eskenazi, "Application of automatic thesaurus extraction for computer generation of vocabulary questions," in *Proceedings of the SLaTE Workshop on Speech and Language Technology in Education*, 2007, pp. 65-68.

[10] J. C. Brown, G. A. Frishkoff, and M. Eskenazi, "Automatic question generation for vocabulary assessment," in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 2005, pp. 819-826.

[11] G. J. Lidstone, "Note on the general case of the Bayes-Laplace formula for inductive or a posteriori probabilities," *Transactions of the Faculty of Actuaries,* vol. 8, p. 13, 1920.

[12] E. Loper and S. Bird, "NLTK: The natural language toolkit," 2002, pp. 63-70.

[13] J. J. Jiang and D. W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," *Arxiv preprint cmp-lg/9709008,* 1997.

[14] G. Zweig and C. J. C. Burges, "The Microsoft Research Sentence Completion Challenge," 2011.

[15] G. Zweig, J. C. Platt, C. Meek, C. J. C. Burges, A. Yessenalina, and Q. Liu, "Computational Approaches to Sentence Completion," in *the Association for Computational Linguistics*, Jeju, Korea, 2012.

[1] https://sites.google.com/site/dataforslate/