

# Factual open cloze question generation for assessment of learner's knowledge

Bidyut Das<sup>1\*</sup>  and Mukta Majumder<sup>2</sup>

\*Correspondence:

bidyut2002in@gmail.com

<sup>1</sup>Department of Information  
Technology, Haldia Institute of  
Technology, Haldia, 721657 West  
Bengal, India

Full list of author information is  
available at the end of the article

## Abstract

Factual objective type questions are effectively used in active learning, information and communication technology based education and intelligent tutoring system for the assessment of learner's content knowledge. In this paper, we have presented an automatic factual open cloze question generation system which can generate fill-in-the-blank questions without alternatives. In order to generate the questions, the system first extracts a set of informative sentences from the given input corpus. The sentences are considered as informative based on part-of-speech tags and certain rules. After the identification of the informative sentences the questions are generated by omitting the answer-keys which are selected by identifying domain specific words in the sentences. The unbound option set of an open cloze question often confuses the examinees. However, open cloze questions require more productive knowledge from learners than cloze questions. Finally, we have also suggested answer hints for the examinees to reduce the number of possible answers that make assessment easier.

**Keywords:** Question generation, Open cloze question, Factual question, Fill-in-the-blank question, Learner knowledge

## Introduction

Question generation is the task of generating good quality assessment item automatically from a given input corpus to examine the content knowledge of the learner (Heilman & Smith, 2010b; Kunichika, Katayama, Hirashima, & Takeuchi, 2001). Automatic question generation system is getting popularity for generating objective test items (Goto, Kojiri, Watanabe, Iwata, & Yamada, 2009, 2010; Liu, Rus, & Liu, 2016). It is widely used in different levels of educational assessment (Brown, Frishkoff, & Eskenazi, 2005; Kunichika, Katayama, Hirashima, & Takeuchi, 2004; Lee & Seneff, 2007; Liu, Calvo, & Rus, 2012; Pino & Eskenazi, 2009). It can be also effective for building an automated examiner for evaluating the learner in intelligent tutoring systems (Conejo et al., 2004; Liu et al., 2016; Papasalouros, Kanaris, & Kotis, 2008; Pino, Heilman, & Eskenazi, 2008; Vinu & Sreenivasa Kumar, 2015). Questions are broadly categorized into two groups: Subjective Questions and Objective Questions. Objective Questions can be also classified into two types: Wh-Questions and Fill-in-the-blank Questions (Agarwal & Mannem, 2011; Erteschik-Shir, 1986). Questions begin with wh-word are called Wh-Questions like who, why, when, how, etc. Fill-in-the-blank Questions are again classified into two subcategories: Cloze Questions (CQs) and Open Cloze Questions (OCQs) (Agarwal, 2012).

A CQ contains a sentence that has one or more blanks and four options are provided to fill those blanks to complete the sentence. One of the four alternatives is correct and the others are wrong. The wrong alternatives are called distractors since they distract an examinee from choosing the correct answer. CQs and Wh-Questions with alternatives are also called Multiple Choice Questions (MCQs) or Multiple Choice Test Items (MCTIs) (Agarwal, 2012; Bhatia, Kirti, & Saha, 2013; Goto et al., 2009, 2010; Mitkov, Ha, & Karamanis, 2006; Nikolova, 2009, 2010). The OCQs are also similar to CQs without alternatives, which make them more difficult to solve. CQ generation consists of three main steps: (a) Informative sentence selection that can generate question (b) Identification of the answer-key that is the correct answer and (c) Generation of distractors which are the wrong answer set. OCQ generation has two steps (a) Sentence selection and (b) Answer-key identification, one step less than CQ generation, which is distractors generation.

Question can be generated to test the grammatical, vocabulary, content or subject related knowledge of a learner (Brown et al., 2005; Lee & Seneff, 2007; Lin, Sung, & Chen, 2007). We have concentrated our work to generate questions to test the content knowledge of the learner. This type of question has been termed as factual question (Heilman, 2011; Heilman & Smith, 2010a). Informative text is used to generate factual question which requires fact-based answer. The text could be an article from Wikipedia, any English article, or a chapter of a book. Factual question has one correct answer which can be verified by referring to the text. As for example a learner may be asked to go through a passage and then answer a set of factual questions based on the information he or she has gathered from the passage. The factual questions allow e-learning professionals to examine how much a learner be familiar with the passage, and what they need to know to fill the learning gap.

All sentences of a textual document are not suitable for factual question generation. The sentence which has sufficient and quality information can act as an informative sentence to generate the factual question. Therefore, the sentence selection has been performing a pioneer role in automatic factual question generation task. But unfortunately, the sentence selection has unable to achieve sufficient attention from the researchers and only restricted in a limited number of approaches.

In this article, we have proposed an automatic question generation system for generating factual open cloze test item to assess the content knowledge of the learner. A new technique for informative sentence selection has been introduced here. The technique extracts the simple sentences from the input corpus and then selects some of those sentences as informative ones using few Part-of-Speech (POS) tagging based rules. Next, we have performed identification of answer-key to generate the question from the selected sentence. Answer-key selection is done by identifying domain specific words in the sentence. Though the open cloze question requires more productive knowledge from learner than cloze question but its unbound option set often confuses the examinees. Therefore finally we have suggested an answer hint based evaluation of the examinee to reduce the number of possible answers that makes assessment easier but still assess the active knowledge of a learner.

### Related works

Automatic question generation has come out as a promising area of research in the field of Natural Language Processing (NLP) and Educational Technology. In the last decade, we have seen that the researchers have paid a considerable amount of attention for objective type question generation semi-automatically or automatically. But most of the researches are confined to Multiple Choice Question (MCQ) or Cloze Question (CQ) generation. Only a limited number of approaches have been found which showed interest in Open Cloze Question (OCQ) generation. Here, we have listed some of the related works.

Manish Agarwal presented an automatic open cloze question generation (OCQG) system. This approach consisted of two steps. In the first step, relevant and informative sentences were selected and keywords were identified in the selected sentences in the second step. News reports on Cricket matches were taken by the system as input and produced factual OCQs as output (Agarwal, 2012). Pino and Eskenazi attempted to measure the level of hint in OCQs. They showed that the first few letters of a missing word in a fill-in-the-blank question gave information about that omitted word. Their goal was to adapt the difficulty level of a question to the student in an intelligent tutoring system for vocabulary learning (Pino & Eskenazi, 2009).

Narendra et al. directly applied a summarizer, *MEAD* for selecting informative sentences for automatic CQs generation (Narendra, Agarwal, & Shah, 2013). Correia et al. used supervised machine learning technique for selecting stem to generate cloze question. They utilized a set of features like *sentence length, word position, chunk, parts-of-speech, named entity, verb domain, known-unknown word, acronym etc.* to run Support Vector Machine classifier (Correia, Baptista, Eskenazi, & Mamede, 2012). Agarwal and Mannem described a system for generating gap-fill questions from a biology textbook. They used a number of features like *the position of the sentence in a document, is it the first sentence, contains token that occurs in the title, length, number of nouns and pronouns etc, whether it contains abbreviation or superlatives.* But they had not clearly reported how the features were combined, what should be the optimum value of these features or whether there was any relative weight among the features (Agarwal & Mannem, 2011). Pino et al. used a set of criteria like *well-defined context, probabilistic context-free grammar score, the number of tokens and the number of clauses.* They also manually calculated a sentence's score based on the occurrence of these criteria in a given sentence and identified the sentence as informative if the score was higher than a threshold (Pino et al., 2008). Hoshino and Nakagawa presented a semi-automatic system to assist teachers in order to produce cloze test items, based on online news articles. In their system, cloze test items were generated by removing one or more words from a passage and the learners were asked to fill the missing words. The system generated two types of distractors: grammar distractors and vocabulary distractors. User evaluation disclosed that 80% of the generated items were considered as suitable (Hoshino & Nakagawa, 2007). Silveira described a general framework for question generation. The input to the system was free text, which was parsed and annotated with metadata. Once annotated, an appropriate question model was selected, and then the question was formulated using natural language (Silveira, 2008). Brown et al. developed a system to generate vocabulary assessment questions automatically. In this task they used WordNet for finding the synonym, antonym, hyponym etc. in order to develop the questions and the distractors (Brown et al., 2005; Miller, 1995).

Coniam proposed one of the earlier methods of MCQ generation. He used word frequencies for an analyzed corpus in the various stages of the development. The author matched word frequency and parts-of-speech of each test item with word frequency and similar word class options to construct the test items (Coniam, 1997). Mitkov et al. presented a semi-automatic system for MCQ generation from a textbook on linguistics. They applied several NLP techniques like *term extraction, shallow parsing, computation of semantic distance and sentence transformation* for the task. They also used natural language corpora and ontology such as WordNet (Miller, 1995; Mitkov et al., 2006). Aldabe and Maritxalar and Aldabe et al. developed systems to generate MCQ in Basque language (Aldabe, de Lacalle, Maritxalar, Martinez, Uria, 2006; Aldabe & Maritxalar, 2010). Chen et al. proposed a technique for semi-automatic generation of grammar based test items by using NLP techniques. Their technique was based on manually designed patterns and it was used to find authentic sentences from the Web and transform into grammatical test items. Distractors were also taken from the Web with some modifications in manually designed patterns e.g. adding, deleting, replacing, reordering of words or changing part of speech. The experimental results of this approach showed that 77% of the generated MCQs were regarded as worthy. Their approach required a considerable amount of effort and knowledge to manually design patterns that were later used to generate grammatical test items (Chen, Liou, & Chang, 2006). Papasalouros et al. described an approach for automatic generation of MCQs from domain ontologies. For experimental purpose, they used five ontologies from different domains. Domain ontologies were represented in the Web Ontology Language (OWL) format thus conforming to Semantic Web technology standards (W3C 2004). Based on this approach, a prototype tool was developed which used OWL ontologies to provide multiple choice questionnaires as output (Papasalouros et al., 2008). Bhatia et al. presented a pattern based technique for selecting MCQ sentences from Wikipedia. The sentences were selected using a set of pattern extracted from the existing questions. They also proposed a novel technique for generating named entity distractors (Bhatia et al., 2013). Majumder and Saha used named entity recognition along with syntactic structure similarity for selecting informative sentences to generate MCQs. In another approach, Majumder and Saha used topic modeling and parse structure similarity to identify informative sentences. They selected the keyword based on domain specific word and named entity. Distractors were selected using gazetteer list based approach (Majumder & Saha, 2014, 2015).

### **Proposed methodology**

For testing the content knowledge of the learner it is required to generate open cloze questions from a kind of sentences that carry proper information. Hence for generating open cloze questions, our first step is to identify the informative sentences. Next, we need to identify the answer-key that is the right answer for a given question. Therefore, our proposed methodology consists of two basic steps: sentence selection and answer-key identification.

Open cloze questions are difficult to answer than the other objective type questions like MCQs or CQs; moreover, question generated from complex or compound sentence has a more complicated answer rather than the simple sentence. Therefore, we have confined ourselves to generate the questions from only simple sentences. Sentence selection task is subdivided into two parts. The first part identifies simple sentences from the input

corpus. Then second part describes the selection of informative sentences from all simple sentences by which we can generate the suitable questions.

### Classification of sentences

Prior to describe the simple sentence identification; in this context, we need to mention the different types of sentences which are found in the input corpus. We have categorized them in following four classes.

*Simple sentence:* A sentence that has only one independent clause and no dependent clauses.

*Compound sentence:* A sentence that contains at least two independent clauses. The clauses are combined with coordinating conjunction.

*Complex sentence:* A sentence that has one or more dependent clauses (subordinate clauses). A dependent clause cannot stand alone. Therefore, a complex sentence must also have at least one independent clause. These clauses are combined by using subordinate conjunction (Klammer, Shultz, & Volpe, 2007).

*Compound-Complex sentence:* A sentence that has two or more independent clauses and one or more dependent clauses.

### Simple sentence identification

To identify simple sentences from the input text we have taken the help of openly available *Stanford Parser*<sup>1</sup> and *Stanford CoreNLP*<sup>2</sup> along with the help of *Stanford Typed Dependency Manual* (Marneffe & Manning, 2008). The Stanford Parser provides the dependency parsing of an input sentence and *Stanford Deterministic Co-reference Resolution System* which is a module of *CoreNLP Suit* helps us to solve the co-reference problem. We have proposed a mechanism that works on the dependency parsing to identify the simple sentences.

To identify the simple sentences we have analyzed the dependency structures of the input sentences. In a simple sentence only one *nsubj* or *nsubjpass* (subject) is there. If a sentence contains more than one *nsubj* or *nsubjpass* then it is considered as compound or complex. The *nsubj* and *nsubjpass* are categorized as subject according to *Stanford Typed Dependency Manual* (Marneffe & Manning, 2008). For the explanation, we have considered the following three example sentences.

*Simple sentence:* Amitabh Bachchan is married to actress Jaya Bhaduri.

*Compound sentence:* Jaya Bachchan joined the politics and became a Rajya Sabha member.

*Complex sentence:* Amitabh Bachchan, who was born 11th October 1942, is an Indian film actor.

A simple sentence is build up of one independent clause whereas a complex or compound sentence is built from at least two clauses. From the sample sentences, we have got the *Stanford Typed Dependency* notations as shown in Table 1.

The clauses in a sentence can be identified as the number of subjects (*nsubj*, or *nsubjpas*) occurred in the *Stanford Typed Dependency* notations of that sentence. From the dependencies in Table 1, the first sentence (*Simple sentence*) has one basic clause (*Bachchan married*), the second sentence (*Compound sentence*) has two basic clauses (*Bachchan joined* and *Bachchan became*) and the third sentence (*Complex sentence*) has two clauses

**Table 1** Stanford typed dependency of the three sample sentences

Simple sentence	Compound sentence	Complex sentence
		compound(Bachchan-2, Amitabh-1)
	compound(Bachchan-2, Jaya-1)	nsubjpass(born-6, Bachchan-2)
	nsubj(joined-3, Bachchan-2)	nsubj(actor-15, Bachchan-2)
compound(Bachchan-2, Amitabh-1)	nsubj(became-7, Bachchan-2)	ref(Bachchan-2, who-4)
nsubjpass(married-4, Bachchan-2)	root(ROOT-0, joined-3)	auxpass(born-6, was-5)
auxpass(married-4, is-3)	det(politics-5, the-4)	acl:relcl(Bachchan-2, born-6)
root(ROOT-0, married-4)	dobj(joined-3, politics-5)	advmod(born-6, 11th-7)
case(Bhaduri-8, to-5)	cc(joined-3, and-6)	nmod:tmod(born-6, October-8)
compound(Bhaduri-8, actress-6)	conj:and(joined-3, became-7)	nummod(October-8, 1942-9)
compound(Bhaduri-8, Jaya-7)	det(member-11, a-8)	cop(actor-15, is-11)
nmod:to(married-4, Bhaduri-8)	compound(member-11, Rajya-9)	det(actor-15, an-12)
	compound(member-11, Sabha-10)	amod(actor-15, Indian-13)
	xcomp(became-7, member-11)	compound(actor-15, film-14)
		root(ROOT-0, actor-15)

(*Bachchan born* and *Bachchan actor*). Therefore, we can identify the simple sentence easily that has only one clause (one *nsubj*, or *nsubjpass*).

#### Informative sentence selection

Here, we have proposed a rule based method that retrieves the informative sentences. Analysis of the *Part-of-Speech tags (POS tags)*<sup>3</sup> in a sentence is the main backbone of our proposed rules (Santorini, 1990). To identify the informative sentences, first we have collected the simple sentences from the dependency parsing of the input corpus. Next, we have analyzed the simple sentences and considered those which are neither exceeding 20 words nor having the *RB/RBR/RBS tag (Adverb)* and with at least two disjoint *NNP/NNPS tags (Proper noun)*. The sentences which are having the aforementioned properties are further refined based on the following *POS tagging based rules*.

1. A sentence having *DT(Determiner)* followed by *NNP/NNPS(Proper noun)*.
2. A sentence having *DT(Determiner)* followed by *CD(Cardinal number)*.
3. A sentence having *DT(Determiner)* followed by *JJ/JJR/JJS(Adjective)*; then *JJ/JJR/JJS(Adjective)* is also followed by *NN/NNS(Noun)* or *NNP/NNPS(Proper noun)*.
4. A sentence having *DT(Determiner)* followed by *NN/NNS(Noun)*; then *NN/NNS(Noun)* is followed by *NNP/NNPS(Proper noun)* or *CD(Cardinal number)*.
5. A sentence having *DT(Determiner)* followed by *NN/NNS(Noun)* and *NN/NNS(Noun)* is followed by *JJ/JJR/JJS(Adjective)*; then *JJ/JJR/JJS(Adjective)* is also followed by *NN/NNS(Noun)* or *NNP/NNPS(Proper noun)*.
6. A sentence having multiple *DT(Determiner)*; then every *DT(Determiner)* must fulfill any one of the above rule.
7. A sentence having multiple *CD(Cardinal number)* and not having *DT(Determiner)*.

We have identified the sentences depending upon the aforementioned rules. According to *Penn Treebank Tagset* (Santorini, 1990), the *IN tag (Preposition or subordinating*

*conjunction*) is considered as a *Stop-Tag* or not important in our case and hence subsequently skipped while applying the above rules. Therefore, the term *DT(Determiner)* followed by *NNP/NNPS(Proper noun)* means one or more *NNP/NNPS tags (Proper noun)* come sequentially in a sentence after *DT(Determiner)* by ignoring *IN tag (Preposition or subordinating conjunction)*.

### Experimental results

To test the accuracy of the proposed system we have extracted the data from eight Wikipedia pages namely, Amitabh Bachchan, Ramayana, Mahabharat, Yoga, Internet, India, Sachin Tendulkar, Bengal Tiger. These pages contain a total of about 2275 sentences; out of them, 614 are simple sentences. Hence 614 sentences are given to the system which identifies 131 sentences as informative ones.

As there is no standard for computing the accuracy of such kind of system, we have taken the judgments of five human linguistic experts on the correctness of retrieved  $n$  sentences and considered the accuracy as the average of their judgments. They have considered 120, 118, 122, 124 and 121 sentences respectively as acceptable informative simple sentences. Hence the accuracy of our system is 92.367%. Table 2 summarizes the results of simple sentence identification from the input text. Table 3 describes the results of identifying informative simple sentences by which we can generate the open cloze questions.

From the evaluation scores given in Tables 2 and 3, it can be concluded that the proposed system is able to retrieve quality informative simple sentences from any input corpus.

### Answer-key identification

Answer-key identification is the task where we select a word or a group of words ( $n$ -gram) which has the potential to become the correct answer of the OCQ. An OCQ has one correct answer-key without alternatives. Therefore we need to identify the answer-key from the selected sentence for question formation. Every informative sentence consists of Unigram and/or Bigram and/or Ngram answer-keys. We have seen that the Ngram key gives more information about the sentence as well as the topic from where we have taken the sentence than the Unigram key. Therefore first we have tried to identify the Ngram answer-key from the sentence; if there is no Ngram key available then we have considered the Unigram answer-key to generate the question. For an open cloze question, it is difficult to guess the answer. Therefore we have considered the number of words in Ngram upto three for our experimental work.

**Table 2** Result of identifying simple sentences

Wiki pages	Total number of sentences	Number of simple sentences
Amitabh Bachchan	224	70
Ramayana & Mahabharat	590	146
Yoga	390	84
Internet	390	61
India	389	65
Sachin Tendulkar	436	109
Bengal Tiger	356	79

**Table 3** Accuracy of informative sentence generation

Number of simple sentences	Number of sentences having at least two NNP/NNPS	Informative sentences	Correct informative sentences (evaluators judgment)	Accuracy (%)
614	253	131	Evaluator 1: 120	92.367
			Evaluator 2: 118	
			Evaluator 3: 122	
			Evaluator 4: 124	
			Evaluator 5: 121	

The key identification task is subdivided into two phases. In the first phase, we have preprocessed the source text from where we have identified the sentences. The text is preprocessed in such a way that the frequency of *NNP/NNPS(Proper noun)* and the co-occurrence of *NNP/NNPS(Proper noun)* are easily counted. We have used Dice-coefficient (Dice, 1945) association technique for identifying a set of Ngram keys  $G_1$  from the co-occurrence frequency of *NNP/NNPS(Proper noun)*. Set  $G_2$  contains the unigram frequency of *NNP/NNPS(Proper noun)*. The Dice Coefficient is described for bigrams as

$$Dice = 2 * x_{11} / (x_{1p} + x_{p1}) \tag{1}$$

Where  $x_{11}$  is the joint frequency and  $x_{1p}$  and  $x_{p1}$  are the marginal totals of the bigram. This measure also easily expands to Ngrams of any size, for example, the Dice Coefficient for Trigrams can be defined as

$$Dice = 2 * x_{111} / (x_{1pp} + x_{p1p} + x_{pp1}) \tag{2}$$

Where  $x_{111}$  is the joint frequency of Trigram,  $x_{1pp}$  is the number of times where *token<sub>1</sub>* appears in the first position,  $x_{p1p}$  is the number of times *token<sub>2</sub>* appears in the second position and  $x_{pp1}$  is the number of times *token<sub>3</sub>* appears in the third position.

In the next phase, for each sentence  $S_i$ , we have extracted a match from  $G_1$  that has a maximum number of words and highest Dice Coefficient score. As we have mentioned earlier, the maximum number of words in a answer-key is three (Trigram). If no match is found then we have tried with the Unigram key whose frequency is highest from  $G_2$ . For the explanation of our proposed technique we have considered the following two sentences: “*Amitabh Bachchan is married to actress Jaya Bhaduri*” and “*Dasharatha was the king of Ayodhya*”. The first sentence finds matches with the two bigram keys; *Amitabh Bachchan* and *Jaya Bhaduri*. The association score of *Jaya Bhaduri* is greater than *Amitabh Bachchan* because we have noticed that *Bachchan* comes individually many times in the source text. Therefore, we have omitted *Jaya Bhaduri* in the sentence to generate the question and identified *Jaya Bhaduri* as the answer-key.

*Question: Amitabh Bachchan is married to actress \_\_\_\_\_.*

*Answer: Jaya Bhaduri*

For the second sentence, there is no match for Ngrams in  $G_1$ . So we have considered the Unigram keys, and extracted match from the Set  $G_2$ . *Dasharatha, king* and *Ayodhya* are found as matches, because *POS tagger* identifies these words as *NNP/NNPS(Proper noun)*. The frequency of *king* is greater than the other two words; therefore we have replaced the



word to generate question and *king* is considered as the answer-key. The Table 4 shows the promising accuracy of answer-key identification.

*Question: Dasharatha was the \_\_\_\_\_ of Ayodhya.*

*Answer: king*

**Hints in OCQ to make assessment easier**

To solve an open cloze question is difficult, compare to other objective type questions for a learner. For open cloze questions, providing hints is a way to reduce the number of possible answers that makes assessment easier, although it still requires active knowledge from the learners. Hence we have proposed here a way of generating hints of open cloze questions for evaluating the learner’s knowledge. The hints include one or more of the following, depending on the content knowledge of the learners: “*Number of words in the answer-key*”, “*First two letters of the unigram answer-key*”, “*Second word of the bigram answer key*”, “*Last word of the trigram answer-key*”, “*Middle and last word of the trigram answer-key*”, “*First two letters of the first missing word of bigram or trigram*”, etc.

It is also difficult to decide the number of hints which are to be provided to the learners. To attempt the OCQs easily we have provided the first hint that indicates the number of words of the answer-key for all examinees. For unigram answer-key only two hints are given to guess the correct answer; the second hint shows the first two letters of the answer-key. For bigram answer-key three hints are there; the second hint shows the last word of the bigram key and the third hint is similar with the second hint of unigram key. For the trigram answer-key we have provided four hints; the second hint is similar to the second hint of bigram, the additional third hint shows the middle word of the trigram answer-key and the fourth hint is similar with the second hint of unigram key.

For explanation, we have considered three sample questions with three different types of answer-keys:

*Question: Father Kamil Bulke author of Ramakatha has identified over 300 variants of \_\_\_\_\_.*

*First Hint: One (Number of words of the Answer-Key)*

*Second Hint: Ra \_\_\_\_\_ (First two letters of the Answer-Key)*

*Unigram Answer-Key: Ramayana*

*Question: The Ramayana written by \_\_\_\_\_ is one of the most popular verses in nepal.*

*First Hint: Two (Number of words of the Answer-Key)*

*Second Hint: \_\_\_\_\_ Acharya (Last word of the Answer-Key)*

*Third Hint: Bh \_\_\_\_\_ Acharya (First two letters of the Answer-Key)*

*Bigram Answer-Key: Bhanubhakta Acharya*

*Question: In \_\_\_\_\_ there is description of two types of Ramayana.*

*First Hint: Three (Number of words of the Answer-Key)*

*Second Hint: \_\_\_\_\_ Sahib (Last word of the Answer-Key)*

*Third Hint: \_\_\_\_\_ Granth \_\_\_\_\_ (Middle word of the Answer-Key)*

**Table 4** Accuracy of answer-key identification

No. of sentences	Sentences with trigram answer-key	Sentences with bigram answer-key	Sentence with unigram answer-key	No. of question with correct answer-key	Accuracy (%)
122	14	48	60	117	95.902

*Fourth Hint: Gu\_\_\_\_\_ Granth Sahib (First two letters of the Answer-Key)*  
*Trigram Answer-Key: Guru Granth Sahib*

We calculate the evaluation score of a learner depending on the right answer and the number of hints he or she has used to solve the question. In the above section, we have mentioned that the question with unigram answer-key has 2 hints, bigram answer-key has 3 hints and trigram key has 4 hints. Therefore the question with unigram answer-key carries 2 credits, bigram key 3 credits and the trigram key 4 credits. The evaluation score  $S$  for the right answer of  $n$  questions is calculated by the following formula.

$$S = \sum_{i=1}^n Q_i(C_r) + Q_i(C_r - H_u) \tag{3}$$

Here,  $C_r$  is the credit of question ( $1 \leq C_r \leq 4$ ) and  $H_u$  is the number of hints used to guess the correct answer ( $1 \leq H_u \leq 4$ ).

The following Table 5 shows the evaluation scores for 5 learners. A set of 30 questions is given to each learner. The system is designed in such a way that the number of hints and the credit of a question are unknown to the learner at prior to the test. There is a button for hints. The same button can be pressed repeatedly when the learner tries to guess the answer and the hints will be shown one by one respectively. The number of hints used by the learner is easily counted by the number of clicks on the button. The button will be disabled when all the available hints have been shown.

**Conclusion**

In this article, we have described a novel technique for identifying informative sentences to generate factual open cloze fill-in-the-blank assessment items. The proposed technique retrieves factual sentences based on POS tags and certain rules. To form the fill-in-the-blank test items we have omitted the answer-keys which are selected by identifying domain specific words in the sentences. To test the content depth of the learners, the proposed system generates open cloze questions without giving possible answer set. Though this unbound option set of an open cloze question demands more intensive knowledge from the learners, it often seems to be complicated to solve. To make assessment easier and to reduce the number of possible answers for open cloze question we have also proposed an answer hint based approach for evaluation purpose. The experimental results reveal that the proposed system can be used to judge the prolific knowledge of the learners and it can enhance the assessment procedure of modern generation education technology.

The system selects only 131 candidate sentences out of 2275 input sentences for OCQs formation. Hence, to generate sufficient number of OCQs by our proposed system, a huge size of input corpus is a prerequisite. The proposed approach sometimes discards

**Table 5** Learner evaluation score and ranking (Total credit=180)

	Number of correct answer given by learner			Score	Ranking
	10 Questions (Credit 4)	10 Questions (Credit 3)	10 Questions (Credit 2)		
Learner 1	3 (Used hints 8)	6 (Used hints 13)	7 (Used hints 7)	60	4th
Learner 2	4 (Used hints 10)	3 (Used hints 4)	8 (Used hints 10)	58	5th
Learner 3	5 (Used hints 12)	5 (Used hints 7)	9 (Used hints 5)	82	1st
Learner 4	2 (Used hints 5)	7 (Used hints 10)	8 (Used hints 8)	67	3rd
Learner 5	5 (Used hints 2)	4 (Used hints 5)	9 (Used hints 12)	81	2nd

informative ones in sentence selection phase while filtering out the less informative sentences. We have intensely studied the discarded sentences and observed that better preprocessing steps like machine learning or pattern based approaches may be followed to increase the accuracy of sentence selection phase. We have also considered the distractors creation for CQs or MCTIs generation as future work.

## Endnotes

<sup>1</sup> <http://nlp.stanford.edu:8080/parser/index.jsp>

<sup>2</sup> <http://nlp.stanford.edu:8080/corenlp/process>

<sup>3</sup> <https://cs.nyu.edu/grishman/jet/guide/PennPOS.html>

## Authors' contributions

Both authors read and approved the final manuscript.

## Competing interests

The authors declare that they have no competing interests.

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Author details

<sup>1</sup>Department of Information Technology, Haldia Institute of Technology, Haldia, 721657 West Bengal, India. <sup>2</sup>Department of Computer Centre, Vidyasagar University Midnapore, 721102 West bengal, India.

Received: 23 February 2017 Accepted: 16 May 2017

Published online: 09 August 2017

## References

- Aldabe, I., de Lacalle, M. L., Maritxalar, M., Martinez, E., & Uria, L. (2006). Arikurri: An automatic question generator based on corpora and nlp techniques. In *Proceedings of the 8th International Conference on Intelligent Tutoring Systems* (pp. 584–594). Heidelberg: Springer-Verlag Berlin.
- Aldabe, I., & Maritxalar, M. (2010). Automatic distractor generation for domain specific texts. In *Proceedings of the 7th International Conference on Advances in Natural Language Processing* (pp. 27–38). Heidelberg: Springer-Verlag Berlin.
- Agarwal, M. (2012). Cloze and open cloze question generation systems and their evaluation guidelines. Master's thesis, International Institute of Information Technology, Hyderabad.
- Agarwal, M., & Mannem, P. (2011). Automatic gapfill question generation from text books. In *Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 56–64). Stroudsburg: Association for Computational Linguistics.
- Bhatia, A. S., Kirti, M., & Saha, S. K. (2013). Automatic generation of multiple choice questions using wikipedia. In *Proceedings of the Pattern Recognition and Machine Intelligence* (pp. 733–738). Heidelberg: Springer-Verlag Berlin.
- Brown, J. C., Frishkoff, G. A., & Eskenazi, M. (2005). Automatic question generation for vocabulary assessment. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing* (pp. 819–826). Canada: Association for Computational Linguistics.
- Chen, C. Y., Liou, H. C., & Chang, J. S. (2006). Fast: An automatic generation system for grammar tests. In *Proceedings of the COLING/ACL on Interactive Presentation Sessions* (pp. 1–4). Stroudsburg: Association for Computational Linguistics.
- Conejo, R., Guzmán, E., Millán, E., Trella, M., Pérez-De-La-Cruz, J. L., & Ríos, A. (2004). Siette: A web-based tool for adaptive testing. *International Journal of Artificial Intelligence in Education*, 14(1), 29–61.
- Coniam, D. (1997). A preliminary inquiry into using corpus word frequency data in the automatic generation of english language cloze tests. *Calico Journal*, 14(2-4), 15–33.
- Correia, R., Baptista, J., Eskenazi, M., & Mamede, N. (2012). Automatic generation of cloze question stems. In *Computational Processing of the Portuguese Language* (pp. 168–178). Heidelberg: Springer-Verlag Berlin.
- Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, 26(3), 297–302.
- Erteschik-Shir, N. (1986). Wh-questions and focus. *Linguistics and Philosophy*, 9(2), 117–149.
- Goto, T., Kojiri, T., Watanabe, T., Iwata, T., & Yamada, T. (2009). An automatic generation of multiple-choice cloze questions based on statistical learning. In *Proceedings of the 17th International Conference on Computers in Education* (pp. 415–422). Hong Kong: Asia-Pacific Society for Computers in Education.
- Goto, T., Kojiri, T., Watanabe, T., Iwata, T., & Yamada, T. (2010). Automatic generation system of multiple-choice cloze questions and its evaluation. *Knowledge Management & E-Learning: An International Journal*, 2(3), 210–224.
- Heilman, M. (2011). Automatic factual question generation from text. PhD thesis, Carnegie Mellon University.
- Heilman, M., & Smith, N. A. (2010a). Extracting simplified statements for factual question generation. In *Proceedings of QG2010: The Third Workshop on Question Generation* (pp. 11–20).
- Heilman, M., & Smith, N. A. (2010b). Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 609–617). Stroudsburg.
- Hoshino, A., & Nakagawa, H. (2007). Assisting cloze test making with a web application. In *Proceedings Of Society for Information Technology and Teacher Education International Conference* (pp. 2807–2814). San Antonio.
- Klammer, T. P., Shultz, R. M., & Volpe, A. D. (2007). *Analyzing English Grammar*. India: Pearson Education.

- Kunichika, H., Katayama, T., Hirashima, T., & Takeuchi, A. (2001). Automated question generation methods for intelligent english learning systems and its evaluation. In *Proceedings of International Conference on Computers in Education* (pp. 1117–1124).
- Kunichika, H., Katayama, T., Hirashima, T., & Takeuchi, A. (2004). Automated question generation methods for intelligent english learning systems and its evaluation. In *Proceedings of International Conference on Computers in Education*.
- Lee, J., & Seneff, S. (2007). Automatic generation of cloze items for prepositions. In *Proceedings of Interspeech 2007* (pp. 2173–2176). Antwerp. International Speech Communication Association (ISCA).
- Lin, Y. C., Sung, L. C., & Chen, M. C. (2007). An automatic multiple-choice question generation scheme for english adjective understanding. In *Workshop on Modeling, Management and Generation of Problems/Questions in eLearning, the 15th International Conference on Computers in Education (ICCE 2007)*. <http://www.apsce.net:8080/icceasp/icce2007/> (pp. 137–142).
- Liu, M., Calvo, R. A., & Rus, V. (2012). G-asks: An intelligent automatic question generation system for academic writing support. *D&D*, 3(2), 101–124.
- Liu, M., Rus, V., & Liu, L. (2016). Automatic chinese factual question generation. *IEEE Transactions on Learning Technologies*, 1(1), 1–12. doi:10.1109/TLT.2016.2565477, <http://ieeexplore.ieee.org/document/7467571/>.
- Majumder, M., & Saha, S. K. (2014). Automatic selection of informative sentences: The sentences that can generate multiple choice questions. *Knowledge Management and E-Learning: An International Journal*, 6(4), 377–391.
- Majumder, M., & Saha, S. K. (2015). A system for generating multiple choice questions: With a novel approach for sentence selection. In *Proceedings of the 2nd Workshop on Natural Language Processing Techniques for Educational Applications* (pp. 64–72). Beijing. Association for Computational Linguistics and Asian Federation of Natural Language Processing.
- Marneffe, M. CD., & Manning, C. D. (2008). Stanford typed dependencies manual. In *Technical Report*. [https://nlp.stanford.edu/software/dependencies\\_manual.pdf](https://nlp.stanford.edu/software/dependencies_manual.pdf) (pp. 338–345). Stanford University.
- Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11), 39–41.
- Mitkov, R., Ha, L. A., & Karamanis, N. (2006). A computer-aided environment for generating multiple choice test items. *Natural Language Engineering*, 12(2), 177–194.
- Narendra, A., Agarwal, M., & Shah, R. (2013). Automatic cloze-questions generation. In *Proceedings of Recent Advances in Natural Language Processing*. <https://aclweb.org/anthology/R/R13/R13-1067.pdf> (pp. 511–515). Hissar.
- Nikolova, I. (2009). New issues and solutions in computer-aided design of mcti and distractors selection for bulgarian. In *Proceedings of the Workshop on Multilingual Resources, Technologies and Evaluation for Central and Eastern European Languages* (pp. 40–46). Association for Computational Linguistics.
- Nikolova, I. (2010). Language technologies for instructional resources in bulgarian. In *Interfaces: Explorations in Logic, Language and Computation* (pp. 114–123). Springer.
- Papasalouros, A., Kanaris, K., & Kotis, K. (2008). Automatic generation of multiple choice questions from domain ontologies. In *Proceedings of the e-Learning* (pp. 427–434).
- Pino, J., & Eskenazi, M. (2009). Measuring hint level in open cloze questions. In *Proceedings of the 22nd International Florida Artificial Intelligence Research Society Conference (FLAIRS-22)*. [http://mi.eng.cam.ac.uk/~jmp84/papers/pino\\_eskenazi\\_hintlevel\\_FLAIRS\\_2009.pdf](http://mi.eng.cam.ac.uk/~jmp84/papers/pino_eskenazi_hintlevel_FLAIRS_2009.pdf) (pp. 460–465). Florida.
- Pino, J., Heilman, M., & Eskenazi, M. (2008). A selection strategy to improve cloze question quality. In *Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains, 9th International Conference on Intelligent Tutoring Systems* (pp. 22–34). Montreal.
- Santorini, B. (1990). Part-of-speech tagging guidelines for the penn treebank project, 3rd edn.
- Silveira, N. (2008). Towards a framework for question generation. In *Proceedings of the Workshop on the Question Generation Shared Task and Evaluation Challenge*. Arlington.
- Vinu, E. V., & Sreenivasa Kumar, P. (2015). *Automated generation of assessment tests from domain ontologies*. IOS Press.

Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)

---