Multimedia Tools and Applications (2021) 80:31907–31925
https://doi.org/10.1007/s11042-021-11222-2

Multiple-choice question generation with auto-generated distractors for computer-assisted educational assessment

Bidyut Das[1] · **Mukta Majumder**[2] · **Santanu Phadikar**[3] · **Arif Ahmed Sekh**[4]

Received: 1 May 2020 / Revised: 26 January 2021 / Accepted: 7 July 2021 /
Published online: 21 July 2021

Abstract
Multiple-choice questions (MCQs) are used as instrumental tool for assessment, not only in various competitive examinations but also in contemporary information and communications Technology (ICT)-based education, active learning, etc. Therefore, automatic generation of multiple-choice test items from text-based learning material is a truly demanding task in computer aided-assessment. A lot of systems were developed in the past two decades for this purpose, but the system generated questions have failed to satisfy the needs of computer-based automated assessment. As a consequence, this is still an open area of research in education technology and natural language processing. This article presents an automated system for generating multiple-choice test items with distractors. The system first selects informative sentences using the topic-words or keywords (one or more words). The best keyword from a selected sentence is chosen as an answer key. Next, the system eliminates the answer key from this sentence and transforms it into a question-sentence (stem). The wrong options or distractors are generated automatically using a feature-based clustering approach, without using any external information or knowledge-base. The result highlights the efficiency of the proposed system for generating MCQs with distractors.

Keywords Computer-assisted learning; Multiple-choice question · Distractor generation · Unsupervised clustering · Computer-aided assessment

1 Introduction

Question plays a significant role in the teaching-learning process [28]. Preparing the questions and assessing their answers manually are time-consuming and laborious task [12]. Therefore, automatic question generation and grading the answer automatically catch the attention of educationalists and researchers [17]. Questions are of two types: objective and subjective [14]. In the case of objective questions, the examinees are asked to select

✉ Bidyut Das
bidyut2002in@gmail.com

Extended author information available on the last page of the article.

the correct answer from a set of options or fill the blanks with words to answer a question. Multiple-choice, true-false, and fill-in-the-blank are the popularly used assessment tools [16].

Multiple-choice question (MCQ) has many advantages, including quick evaluation, uniform scoring, and less testing time [12]. Therefore, many competitive examinations use MCQ papers for assessing the candidate's merit. MCQ is also effective in active learning environment [49] and outcome-based education (OBE) [43] system.

MCQ has three main components [46]. These are stem, answer key, and distractors. A stem forms the body of a MCQ, which is an interrogative sentence (for wh-question) or a sentence with gap (for fill-in-the-blank question). An answer key is the correct option of an MCQ, and the distractors are the wrong options that confuse the examinee to select the correct answer.

All sentences of a text are not suited to generate MCQ stems [36]. A sentence with appropriate information can lead to a stem. Therefore, identifying informative sentences from a text plays an important role in MCQ generation. Different techniques are employed in the literature for selecting informative sentences such as sentence length [21], appearance of a particular word [50], parts-of-speech pattern [14], summarization [9] and parse structure [36].

Similarly, all words of an informative sentence are not chosen for the answer key. Therefore, the answer key selection is a task that determines which word or phrase will be replaced/removed from the sentence to generate stem [36]. Term frequency (TF) is the starting and probably the efficient approach to determine the key in a sentence [13]. Sometimes TF-IDF is applied as an alternative to term frequency [25]. The other techniques such as part-of-speech matching [42], parse structure [23], pattern matching [23], and semantic information [1] are used in the literature for selecting key for MCQ.

After selecting the key from an informative sentence, the next task is transforming it into a question form (stem). Several approaches are used such as appropriate wh-word [35], dependency structure [1], discourse connectives [3], and semantic information [40] in the literature to generate the stem for MCQ.

The distractors are also important in MCQ generation [22]. The quality of distractors improves the quality of an MCQ. The examinees choose the correct answers easily when the distractors are not able to confuse them. As a result, the quality of the MCQ degrades. Parts-of-speech information [2], frequency count [13], WordNet [27], domain ontology [29], distributional hypothesis [1], and semantic analysis [4, 41] are used in the literature to generate distractors for the MCQ.

After a lot of efforts by the researchers, generating MCQs with suitable distractors is still a challenging task and also not effective in real educational applications [46]. We have noted that the simple sentences are more useful to generate MCQs than the complex and compound sentences. In this paper, we have used a pipeline for simple sentence generation [15]. Next, the simple sentences are ranked based on topic-words to select informative sentences for creating MCQ stem. The topic-words are identified using the rapid automatic keyword extraction (RAKE) [48]. The distractors generation technique is proposed here using feature-based unsupervised clustering. Finally, the string similarity and semantic similarity are explored within clusters for selecting final distractors, which are closest to the answer key. The salient features of the article, which contributes to the literature in multiple-ways are as follows:

- We have proposed a complete framework for MCQ generation that includes stem generation, answer-key identification, and distractor generation from a text-based learning material for educational assessment.
- We have proposed a semantic feature-based clustering approach for distractor generation that improves state-of-the-art accuracy.
- The system can able to generate multiword distractors, which makes it more attractive.

2 Related work

This section presents the related existing methods found in the literature. Table 1 shows methods and limitations used for MCQ generation. We also discuss the challenges and bridge gaps by our proposed method.

NLP based methods: Agarwal and Mannem [2] proposed a system, which generates gap-filling questions from a textbook. They used syntactic and lexical features of the document for generating questions without relying on any external resource. Narendra et al. [44] employed a summarizer (MEAD) to select informative sentences for cloze question generation. They proposed an approach to select distractors using a knowledge-base for a specific domain. Bhatia et al. [10] described a pattern-based approach to select sentences for generating MCQ. They used a set of patterns of the existing questions for selecting sentences from Wikipedia. They also proposed an approach for generating named-entity distractors. Afzal and Mitkov [1] proposed a dependency-based unsupervised approach for extracting semantic-relations to generate MCQs automatically. They generated questions using these semantic relations and finally, generated distractors using a distributional similarity measure. Majumder and Saha [35] presented a parse-tree matching approach for selecting informative sentences. They mainly focused on selecting suitable-sentences for generating MCQs and considered distractors generation as their future work. In another work, Majumder and Saha [36] also applied topic modeling and parse structure similarity for selecting informative sentences. The distractors were generated using a name-dictionary and a set of rules. Alsubait et al. [5] proposed an ontology-based MCQ generation system and evaluated the approach by domain experts. Pugh et al. [47] developed a framework for generating high quality MCQs employing cognitive models. This approach created quality test-items that assess clinical decision-making. Santhanavijayan et al. [49] proposed an automatic system for generating MCQs on any user-defined domain. Their system transformed summary-sentences into the stem for generating MCQs. They used similarity-metrics such as hypernyms and hyponyms to generate distractors. Patra and Saha [46] presented a method to generate named entity distractors for generating MCQs.

ML based methods: Goto et al. [24] developed a system for multiple-choice cloze-question generation from text and it's evaluation. The system extracted informative sentences for generating questions based on preference learning. It estimated blank parts using a sequence labeling model, conditional random field (CRF). It was unable to generate distractors for the blank part, which required more than two words. Du et al. [19] proposed the attention mechanism framework and later use an encoder-decoder [20] for generating questions from a given paragraph. The method did not address the problems of distractor generation. Yuan et al. [56] proposed a text-to-text learning method for question generation. Subramanian et al. [51] suggested a key-phrase detection framework for question generation, where the key-phrase was detected using a neural network. Liu et al.

Table 1 Comparative analysis of MCQ generation methods in the previous literature

Year	Ref.	Method	Limitation
2011	[2]	Sentence selection method used some features such as common tokens, abbreviation, sentence position, sentence length, presence of nouns, etc. Key generation approach used frequency and some rules. Distractor generation method used syntactic and lexical features without any external resource or ontology	The method did not use semantic features and generated all single word distractors.
2014	[1]	Generate question using dependency based semantic relations. Generate distractors using a distributional similarity measure.	The study only focused on the biomedical domain.
2015	[36]	Sentence selection method used topic-word and parse-structure similarity. A rule-based approach and named entities are used for keyword identification. Distractor generation is performed using a gazetteer list-based approach.	The gazetteer list-based distractor generation depends on the content of the corpus.
2016	[47]	High-quality MCQs are generated using cognitive models.	Require more studies to compare the cost and psychometric properties of MCQs that developed from cognitive models.
2017	[49]	Stem generation method used summarizer and Boom's taxonomy. Key generation approach used proper-noun and adjective phrases, and distractor generation method used similarity metrics such as hypernyms and hyponyms.	The sentence selection method did not include the technique of co-reference resolution. The distractor generation approach did not use ngrams.
2017	[19]	A hierarchical neural model is used for identifying question worthy sentences	This model did not include the distractor generation approach
2018	[53]	Semantic similarity and collocation information is used to rank the distractors.	The method only focuses on distractor generation. Sentence selection, stem generation, and answer key identification are not included with this method.
2019	[46]	Generate distractors using named-entity. The system takes the question sentence and the correct answer as input and generates three distractors	The system did not focus on the generation of question sentences and their answer keys.
2020	[39]	Generate distractors using hierarchical Multi-Decoder Network (HMD-Net). It consists of one encoder and three decoders. Each decoder generates a single distractor	The method did not focus on the other phases of MCQ generation, such as selecting sentences for stems and identifying answer keys.

[32] proposed a regression model using orthographic, phonological, and semantic features. It automatically generated Chinese MCQs using a mixed similarity strategy. They employed a machine learning approach for generating Chinese MCQ distractors. Sun et al. [52] utilized a sequence-to-sequence model considering the answer as a cue for the question. Kim et al. [26] also suggested an answer separation module for generating questions.

Challenges and bridge gaps: The primary challenges of MCQ generation from a text are: selecting suitable-sentences for questions, answer-phrase identification, and relevant distractors selection. Existing NLP-based methods mainly addressed this problem of question generation but suffers from many poorly performed sub-tasks such as sentence selection and simplification. We have noted that suitable distractor generation for MCQs needs much attention.

Recently, the sharp advancement in computational hardware and machine learning algorithms open up new possibilities in NLP. The main drawback of it is that it demands a large volume of training corpus, which is difficult in many cases. The selection of question sentences, answer keys, distractors are also dependent on the content of the corpus and require learning beyond sequence-to-sequence. Therefore, most of the researches in the last decade focused on NLP-based methods to solve the problem. This study proposed an automatic system for generating MCQs from text-based learning materials. It also focused on distractor generation for MCQs from the same learning materials using a novel distance metric approach. This research will help teachers or organizations to generate MCQs automatically from the learning content to assess learners automatically.

3 Proposed method

MCQ stems are generated from simple sentences using topic-words. The topic-words or keywords define the domain or topic of the corpus. In this paper, first, we have used a technique that focuses on identifying the existing simple sentences from the text corpus and generating simple sentences from complex and compound sentences. Next, the useful keywords are fetched from this corpus. The simple sentences are ranked based on the keywords for identifying informative sentences. The system also used a preprocessing step for

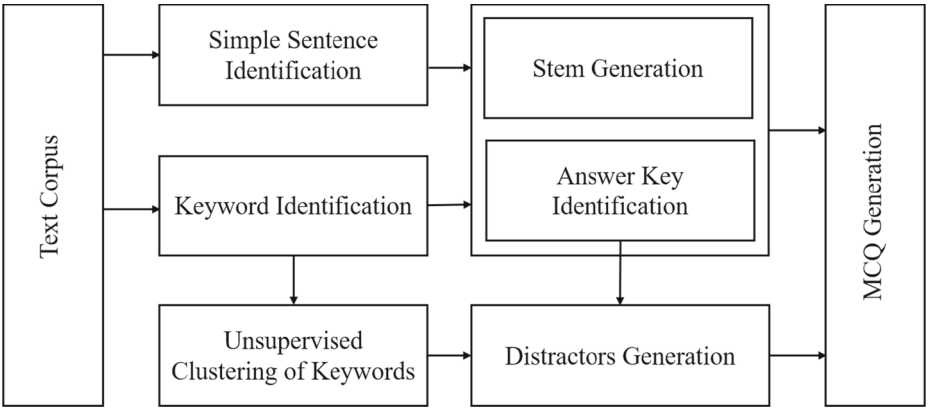

Fig. 1 The overall view of the proposed MCQ generation system

resolving co-references [37]. The best keyword of an informative sentence is selected as an answer key. Finally, a new feature-based clustering approach is proposed for distractor generation. Figure 1 shows the overall view of the proposed MCQ generation system. In the next subsections, we have elaborated the steps of the system. The complete system is shown in Algorithm 1.

Algorithm 1 MCQ Generation system.

Input: Text (T)
Output: Question sentence (Q), Answer (A), Distractors (D)

1: Identify all sentences $\{sentence_1, sentence_2,..., sentence_n\}$ from the text T by text pre-processing using natural language toolkit (NLTK).
2: Separate simple sentences (SS) from other sentences that have only a single clause.
3: Other complex and compound sentences are split to generate simple sentences.
4: Accumulate all simple sentences into (L_s) from the text (T). $L_s = \{ss_1, ss_2, ...; ss_n\}$
5: A list L_K of keywords is extracted from the text using the RAKE method.
6: The simple sentences are ranked based on their weights of keywords (3)
7: Top-ranked simple sentences are selected as informative-sentences (IS) for stem generation.
8: Then identify the best keyword of an informative sentence as answer-key (A) and translate the sentence into a question sentence or stem (Q).
9: Call Algorithm 2 for generating distractors (D)
10: Return Q, A, D

3.1 Simple sentence identification

A simple sentence is built of one independent clause; on the other hand, a compound or complex sentence is consisted of minimum two clauses [6]. First, we have separated all existing simple sentences from other sentences using the identification of one independent clause in the sentence, using the technique as described in [14]. Das et al. [15] analyzed the dependency structure [38] of input sentences and proposed a technique to generate simple sentences from complex and compound sentence. A compound sentence consists of two or more independent clauses. Their approach generated two or more simple sentences by splitting a compound sentence. A complex sentence has at least one independent clause and one or more dependent clauses. Their approach also generated one or more simple sentences from a complex sentence by extracting the independent clauses with ignoring dependent clauses. This technique is inherited here for generating simple sentences from complex and compound sentences.

3.2 Keywords identification

A keyword is a word or a set of words that provides the content clue of a document. The term frequency (TF) is a popular approach to determine the keywords in a document [33]. Sometimes, the term frequency and inverse document frequency (TF-IDF) is applied alternately to identify the keywords from an individual document [18]. But the TF and TF-IDF are not useful for finding multiword keywords. Several statistical association measures such as *Pointwise mutual information* (PMI), *Dice-coefficient* [14], *Jaccard similarity* are used most often to determine the multiword keywords in a document. A well-known approach *TextRank* used *Jaccard similarity* for extracting keywords [31]. Another popularly known

technique is RAKE (*Rapid Automatic Keyword Extraction*) [55]. It is an unsupervised statistical method used for extracting keywords, which is independent of the corpus domain and language. It can generate more complicated keywords that might have more meaning than individual words. The RAKE is computationally more effective than *TextRank* while obtaining comparable higher precision and recall scores. We have used the RAKE method to identify the keywords from our corpus. We have customized the RAKE method and considered the keywords tagged with 'NNP' or 'NNPS' (proper nouns: required POS = ['NNP','NNPS']) and 'CD' (numbers: required POS = ['CD']) to generate more suitable distractors. The RAKE score of a word is calculated in equation (1), where $deg(w)$ is the degree, and $freq(w)$ is the frequency of a word w in the corpus.

$$\rho(w) = \frac{deg(w)}{freq(w)} \tag{1}$$

This problem is represented by an undirected graph considering the words as nodes. The degree of a word $deg(w)$ is defined by the degree of a node or vertex ($deg(v)$) in the graph. Two nodes are connected via an undirected edge when they are linked with the same candidate keyword. The higher-degree of a node means that it has more connections in the graph. It means that the word occurs more often and appear in the longer candidate keywords. Therefore, the degree of a word presents, how frequently it co-occurs with other words in the candidate keywords. To find the multiword keyword, the RAKE looks for pairs of words that are adjacent to one another in the same order and at least twice in the same document. Next, a new candidate keyword is formed as a combination of those words. The RAKE score $\rho(k)$ of the keyword (k) is computed by summing the score of adjacent member words $\rho(w_i)$, which is shown in equation (2), where w_i is the i_{th} adjacent member-word of the keyword $(1 \geq i \leq n)$, and n is the number of individual words present in the keyword.

$$\rho(k) = \sum_{i=1}^{n} \rho(w_i) \tag{2}$$

3.3 Stem generation and answer key identification

A sentence consists of some meaningful keywords (one or more words) and stopwords. Since the stopwords do not have any weight-information in the sentence, we exclude them for calculating the sentence weight for informative sentence selection. Therefore, the sentence weight $w(s)$ is calculated by combining the weights of individual keywords that belong to the sentence. We assign a higher weight to a keyword, which has more words. The weights of i_{th} keyword $w(k_i)$ in a sentence is defined by the number of individual words present in the keyword. Finally, the weight of the sentence $s(w)$ is calculated using equation (3), where p is the number of keywords present in the sentence (s). Top-ranked sentences are selected as informative sentences to generate MCQ stems.

$$w(s) = \sum_{i=1}^{p} w(k_i) \tag{3}$$

Among several candidates, the best keyword is identified as an answer key depending on the word length and the RAKE score from an informative sentence. We have noticed that the multiword keyword has more significant meaning than the single word keyword to act as the answer key. After the answer key is identified, the Stanford Named Entity Recognizer

(NER) is used to identify the category of answer key.[1] The stem is formed by replacing the answer key with a suitable *'wh-word'*. Parse-tree structure of the identified informative sentence is interviewed to place the *'wh-word'* at an appropriate position or how long the sentence is taken before truncating it [36]. Figure 2 shows the parse tree structure of an informative sentence using Stanford Tregex [30]. For example, 'who', 'where' and 'when' are appropriately used for 'person', 'location', and 'time/date' respectively. The system also generates fill-in-the-blank MCQ stems from the informative sentences by simply omitting the answer-key with a blank when the answer key cannot be categorized by the NER.

3.4 Distractors generation

Several researchers have proposed different approaches for generating distractors, but they could not achieve adequate success [46]. Generating distractors for multiword answer key is more complex than the unigram key [12]. Here, we have proposed a method to identify multiword distractors using the K-means clustering algorithm. The number of clusters for candidate distractors is identified automatically using the elbow method [11]. The elbow method is applied to determine the nearly optimal number of clusters K in K-means. Next, we have selected the final three distractors from a cluster which contains answer key. The overall distractors generation technique is presented in Algorithm 2.

Algorithm 2 Distractors generation.

Input: Answer (A) and List of all keywords $L_K = \{k_1, k_2, ..., k_n\}$
Output: Distractors (D)

A set of clusters $C = \{c_1, c_2, ..., c_n\}$ of L_K using features $f(k)$ (4)
K_{c_i} is the set of keywords in i^{th} cluster $K_{c_i} = \{k_1, k_2, ..., k_m\}$
for all $c_i \in C$ **do**
    for all $k_j \in K_{c_i}$ **do**
        if $k_j = A$ **then**
            Calculate similarity $\forall k_j$ with A when $k_j \neq A$
        end if
    end for
    Similarity score of keywords is ranked in descending order
    D $\leftarrow$ Chose top three keywords from c_i
end for

The bag of words (BOW) [58] model is the simplest approach for clustering words. The word2vec [34] is a well-known method in feature learning and language modeling techniques in natural language processing (NLP). Therefore, we have used word2vec instead of the bag of words (BOW) model for clustering the keywords. Figure 3 shows the sample clustering result using word2vec features. The result shows that word2vec features are not adequate for generating distractors. For example, we have found *'Raja Ram Mohan Roy'* and *'1931'* are grouped into the same cluster.

Feature selection is one of the biggest challenges in distractors generation. We have combined different features to generate our proposed feature set. The feature set is used in the experiment for evaluating the technique of distractors generation using unsupervised K-means clustering. The more refined feature set can generate more accurate clusters for

[1] http://nlp.stanford.edu:8080/ner/

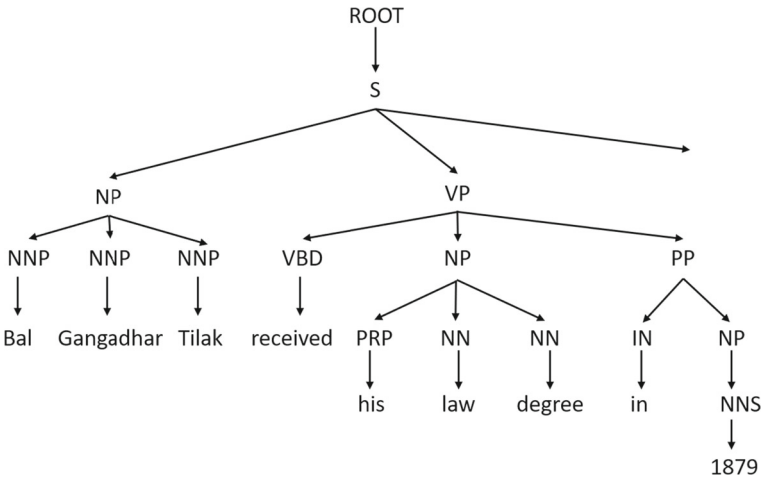

Fig. 2 The parse tree structure of a sentence using Stanford Tregex [30]

candidate distractors. In the first stage of the experiment, RAKE score ($\rho(k)$), unigram keyword (k_u), bigram keyword (k_b), trigram keyword (k_t), and quadgram keyword (k_q) are taken as the features. Then we have added part-of-speech (e.g., noun (nn), proper-noun (nnp), number (cd) etc.) and named-entity (e.g., person (per), organization (org), location (loc) and date ($date$)) features for the experiment. Finally, we have taken these twelve features of keywords to group them. The feature set of a keyword $f(k)$ is represented in the equation (4). For example, the feature set of the keyword *'Raja Ram Mohan Roy'* is presented by {9, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0}. Figure 4 shows the clustering results using our proposed feature set.

$$f(k) = \{\rho(k), k_u, k_b, k_t, k_q, nn, nnp, cd, per, org, loc, date\} \tag{4}$$

It is difficult to determine the appropriate number of clusters for the candidate distractors. It depends on the corpus. Here, we have used the state-of-the-art elbow method [11] that automatically identifies the number of clusters for candidate distractors depending on the

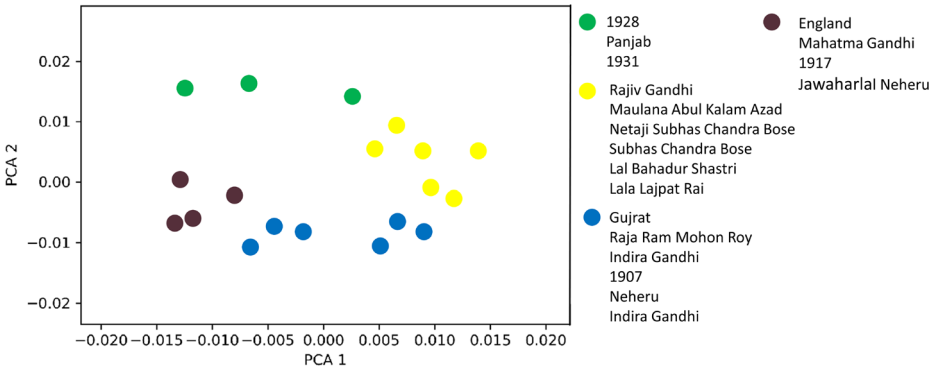

Fig. 3 Toy example of K-means clustering (K=4) on keywords (PCA-reduced data using word2vec features)

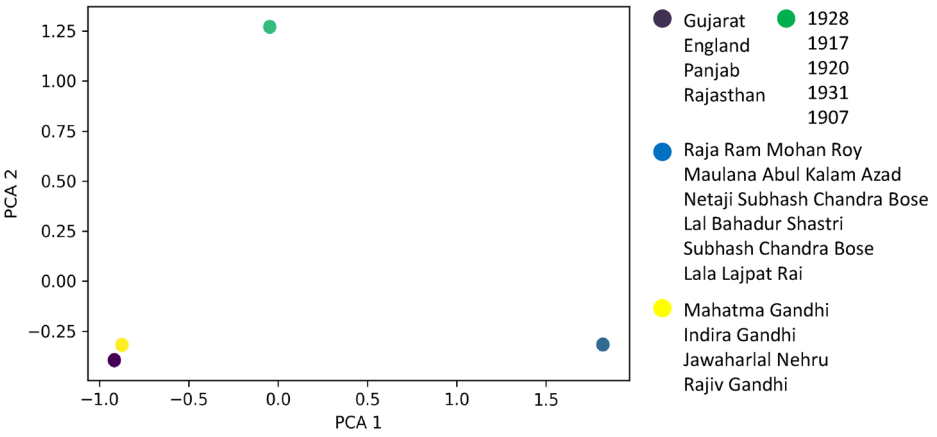

Fig. 4 Toy example of K-means clustering (K=4) on keywords (PCA-reduced data using our proposed feature set)

features of keyword in the corpus. Figure 5 shows the typical elbow-based cutoff to determine the number of clusters. The distractors are selected from a cluster when the answer key is also present in the same cluster.

The efficiency of the clustering is measured using the Rand Index (RI) [54]. The RI computes the similarity between two clustering results, considering all pairs of samples and counting the pairs that are assigned in the same or different clusters in the predicted and true-clusters. The true-clusters of keywords are generated manually based on the relevance of candidate distractors. The RI score is then transformed into *'adjusted for chance'* ARI score using the equation (5).

$$ARI = (RI - Expected\_RI)/(max(RI) - Expected\_RI) \tag{5}$$

The ARI is thus assured to have a score close to 0.0 for random labeling, independently of the number of clusters and samples, and 1.0 when the clusters are identical (upto a permutation). The ARI score of cluster similarity with true cluster is shown in Fig. 6. We have noticed that the ARI score is maximum when the number of clusters (K) is 21.

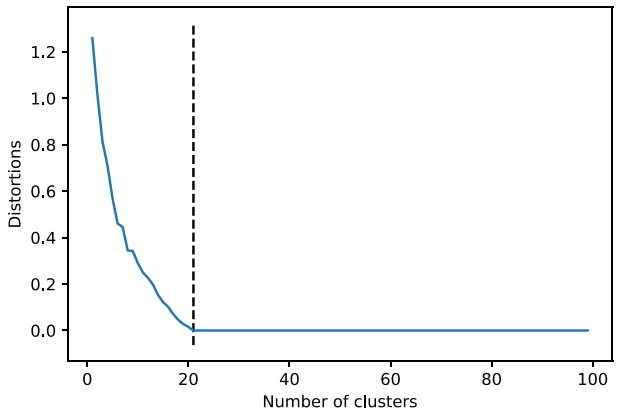

Fig. 5 The elbow method to find the optimal number of clusters (K=21) for candidate distractors

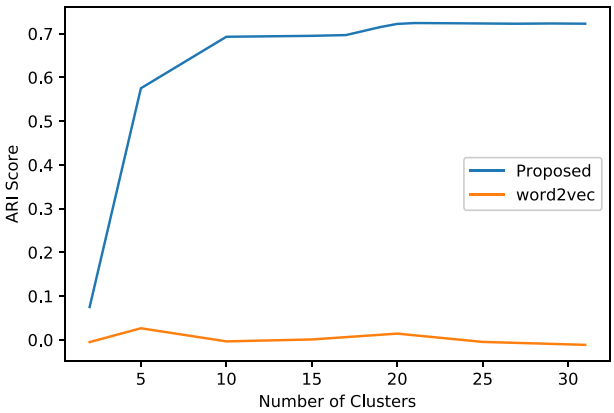

Fig. 6 The ARI Score between predicted and true cluster

4 Results

This section evaluates the result of the proposed MCQ generation system. The system has different modules. Therefore, we have taken three experiments to test the system quality in different ways. This section has four subsections: performance evaluation metrics, used dataset, the experiments, and the discussion of results.

4.1 Performance evaluation metrics

The effectiveness is mainly measured using the following set of metrics [45]. The precision and recall metrics are defined as follows, where TP is the True positive rate, FP is the False Positive rate, FN is the False Negative rate, and TN is the True Negative rate (Fig. 7)

$$Precision\ (PE) = \frac{TP}{TP + FP} \tag{6}$$

$$Recall\ (RE) = \frac{TP}{TP + FN} \tag{7}$$

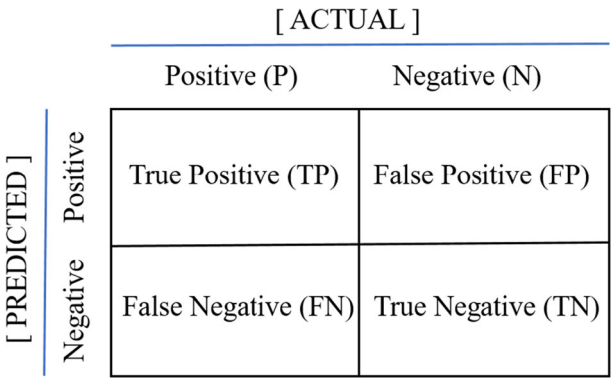

Fig. 7 The confusion matrix

Another popular metric is the F1 score. It is the harmonic mean of precision and recall. It could be applicable when a balance between the precision and recall is needed, and the class distribution is uneven (i.e., high TN +FP). F1 score is defined as follows:

$$F1\ Score(FS) = 2 * \frac{PE * RE}{PE + RE} \tag{8}$$

The accuracy of the proposed system is evaluated using the following equation (9).

$$Accuracy\ (ACC) = \frac{(TP + TN)}{TP + FP + FN + TN} \tag{9}$$

4.2 Dataset

Several in-house datasets are used in the literature to measure the correctness of MCQ generation systems, and most of the MCQ generation systems are evaluated by human evaluators [46]. There is no openly available gold-standard data to evaluate the proposed system [12]. Therefore, we have created a test dataset to check the performance of the system using human evaluators. We employed five evaluators to check the correctness of the system generated results. We have tested the system using web documents. The test corpus was created by extracting the web pages of fourteen Indian leaders and eleven Indian social reformer's [2]. The test corpus has 25 documents that consist of 1893 sentences.

4.3 Experiments

We have taken three different experiments to assess the system quality. Experiment 1 evaluates the accuracy of informative sentences. Experiment 2 evaluates the accuracy of system generated stem with answer key, and Experiment 3 evaluates the accuracy of distractors generation.

Experiment 1 *In the first experiment, we have evaluated the selected informative sentences for generating stem. The sentence selection task mainly depends on the keywords and simple sentences. The visualization of the extracted top-ranked keywords is illustrated in* Fig. 8. *The simple sentences are ranked based on keywords to select informative sentences for creating suitable stems. After selecting the informative sentences, five experts are asked to mark relevant/irrelevant sentences, and ground truths are generated based on the average of their voting. The average accuracy of informative sentence selection is shown in* Fig. 9.

Experiment 2 *In the next experiment, we have evaluated the relevant stems of the MCQs. The stem generation depends on the accuracy of the informative sentences. Four experts are asked to mark relevant/irrelevant stems, and ground truths are generated based on their voting.* Figure 10 *shows the stem generation result from top-ranked 50% informative sentences.*

Experiment 3 *After selecting the candidate set of distractors, the string similarity is checked using Levenshtein Distance* [57]*, and semantic similarity is checked using Latent Semantic Analysis (LSA)* [7] *to generate a set of distractors which are close enough to answer key. Then, aggregate the scores for ranking the candidate distractors in a category. The top three are chosen as a final set of distractors for the answer key. For the*

[2] http://www.culturalindia.net

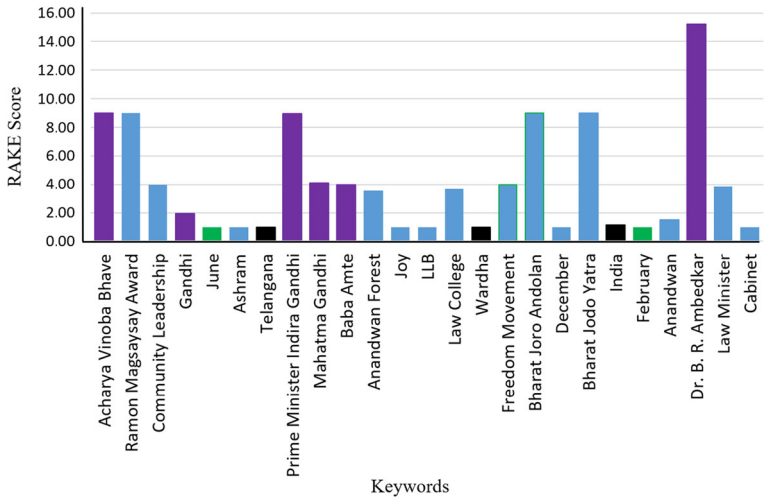

Fig. 8 Keywords identification (Required POS ['NNP', 'NNPS']) using RAKE score. Purple colours denote the name of the persons, Blacks are the name of locations, greens denote dates, and blues belong to the miscellaneous category identified by the NER with 4 classes

evaluation purpose, the correctness of distractors is measured by the average scoring of distractors. The distractor's score of a question q_t *is denoted by* $\delta(q_t)$*. The* $\delta(q_t) = 1$ *for one,* $\delta(q_t) = 2$ *for two and* $\delta(q_t) = 3$ *for three correct distractors of a question qt. If the number of questions is z, then the total distractors are* $3z$*. The accuracy of distractors* (α) *is measured in the equation* (10)*.* Table 2 *presents the accuracy of the proposed cluster-based distractor generation method with the different state-of-art methods using our dataset. Four system-generated sample MCQs are shown in* Table 3*.*

$$\alpha = \frac{\sum_{t=1}^{z} \delta(q_t)}{3z} \times 100 \tag{10}$$

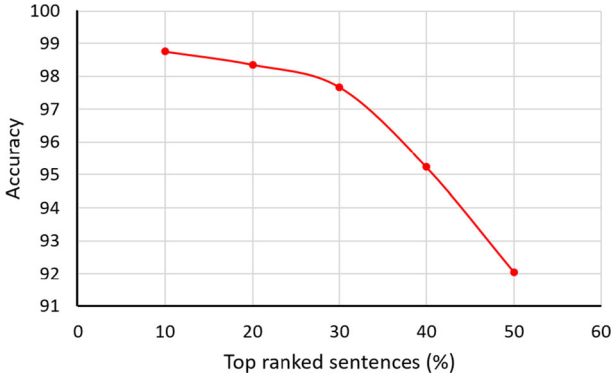

Fig. 9 The accuracy of the selection of top ranked informative sentences. The accuracy varied from 99% to 92% when we choose informative sentences from 10% to 50%

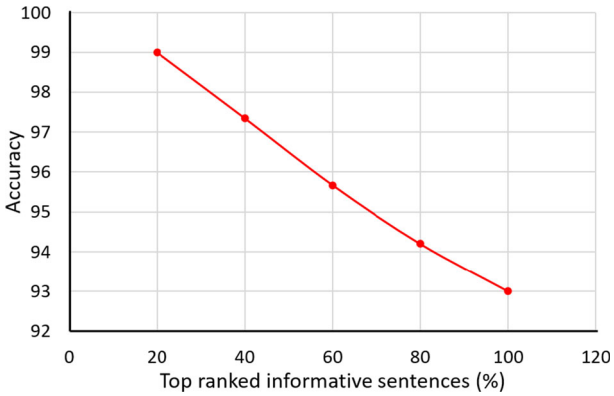

Fig. 10 The accuracy of stem generation with respect to top 50% informative sentences

4.4 Discussion of results

The system is tested in various ways in the Experiments in Section 4.3. The accuracy of top-ranked keywords and sentences are adequate. It is mentioned that here we only considered the precision. For question generation, precision is more important than recall because the exactness of generating questions is more important than completeness [1]. Figure 9 presents a curve that indicates an upper-ranked sentence has more potential to be selected as informative. The reason is an upper-ranked sentence has more meaningful keywords, which make the sentence more informative. Figure 10 similarly shows the linear curve. Top-ranked informative sentences can generate more suitable stems for MCQ generation. We considered the top half of the informative sentences for stem generation to increase the precision. Due to the lack of dataset, we used a clustering approach based on keyword features for generating distractors.

4.5 Computational efficiency of the system

The proposed system is a two-step process: question generation (stem and answer key), and distractors generation. The question generation step depends on the size of sentences in the dataset. This step has a complexity of $O(n)$, where n is the number of sentences.

Table 2 Accuracy of distractor generation

Method	Accuracy (%)
Frequency count [13]	57.72
Parts-of-speech information [2]	52.45
WordNet [27]	56.23
Pattern matching [24]	64.39
Distributional hypothesis [1]	61.45
Semantic analysis [8]	62.12
Statistic+Semantic [46]	70.24
Proposed feature-based clustering	71.86

Table 3 The sample MCQs that generated automatically by the system. The asterisk (*) indicates the correct answer and the other three options are the distractors

Sentence:	Ishwar Chandra Vidyasagar helped revered Bengali poet Michael Madhusudan Dutta to relocate from France to England and study for the bar.
Question:	Who helped revered Bengali poet Michael Madhusudan Dutta to relocate from France to England and study for the bar?
Answers:	1. Swami Dayanand Saraswati 2. Ishwar Chandra Vidyasagar* 3. Sri Ramakrishna Paramhansa 4. Acharya Vinoba Bhave
Sentence:	In 1841 Jyotiba Phule got admission in the Scottish Missions High School Poona and completed his education in 1847.
Question:	In 1841 who got admission in the Scottish Missions High School Poona and completed his education in 1847?
Answers:	1. Mother Teresa 2. Chhatrapati Shahu 3. Jyotiba Phule* 4. Swami Vivekananda
Sentence:	Swami Dayanand Saraswati was born on February 12 1824 in Tankara Gujarat as Mool Shankar to Karshanji Lalji Tiwari and Yashodabai.
Question:	Where Swami Dayanand Saraswati was born on February 12 1824?
Answers:	1. England 2. Gujarat* 3. Panjab 4. Rajasthan
Sentence:	Chhatrapati Shahu was married to Lakshmibai Khanvilkar daughter of a nobleman from Baroda in 1891.
Question:	When Chhatrapati Shahu was married to Lakshmibai Khanvilkar daughter of a nobleman from Baroda?
Answers:	1. 1956 2. 1891* 3. 1890 4. 1892

The distractors generation method is executed by K-mean clustering and depends on the number of keywords that are used for clustering. We have quantified the execution time of the proposed method on the dataset. We have used Intel i7 processor (3.2 GHz speed) with 8 GB of RAM for the experiments. Figure 11 shows the execution time of the system with varying numbers of sentences. It is observed that the system is almost linear to the number of sentences in the dataset.

5 Conclusion

To meet the increasing demand of MCQs in competitive examinations and educational assessment, especially in e-learning and active learning framework, automatic generation of multiple choice test items from text-based course material has become a popular research area among educationalists and natural language processing researchers. In this paper, we have proposed an approach to generate MCQ with auto generated distractors. First, we have extracted the topic-words from the corpus. Then, we have identified the simple sentences and ranked them based on the topic-words. The question sentence (stem) is generated by replacing the answer key with an appropriate wh-word or a blank (gap). To generate the

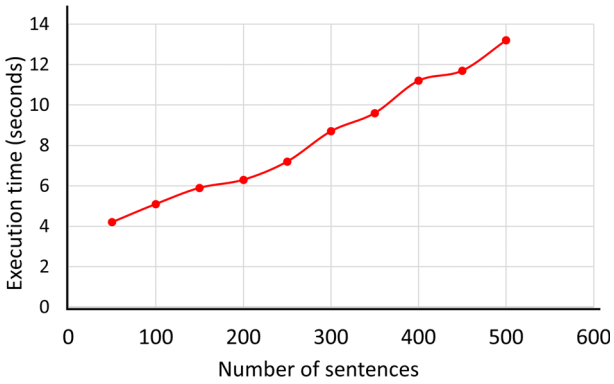

Fig. 11 The execution time varying number of sentences

option set of the MCQ, dictractors are selected such a way that they are closely related to answer key. Feature based clustering technique is employed to select the candidate set of distractors. The distractors category is identified automatically using the elbow method in K-means clustering. Levenshtein Distance and Latent Semantic Analysis are explored to combine string similarity and semantic similarity for selecting the final set of distracters. The findings suggest that the proposed approach can produce good quality MCQs and be useful at various levels of assessment.

Funding This study is not funded from anywhere.

Declarations

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Conflict of Interests The authors declare that there is no conflict of interest regarding the publication of this paper.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

1. Afzal N, Mitkov R (2014) Automatic generation of multiple choice questions using dependency-based semantic relations. Soft Comput 18(7):1269–1281
2. Agarwal M, Mannem P (2011) Automatic gap-fill question generation from text books. In: Proceedings of the 6th workshop on innovative use of NLP for building educational applications, pp 56–64. Association for computational linguistics
3. Agarwal M, Shah R, Mannem P (2011) 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
4. 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. Springer, Berlin, pp 27–38
5. Alsubait T, Parsia B, Sattler U (2016) Ontology-based multiple choice question generation. KI-Künstliche Intelligenz 30(2):183–188

6. Andersen S (2014) Sentence types and functions. California: San José State University Writing Center
7. Aouicha MB, Taieb MAH, Hamadou AB (2018) Sisr: system for integrating semantic relatedness and similarity measures. Soft Comput 22(6):1855–1879
8. Araki J, Rajagopal D, Sankaranarayanan S, Holm S, Yamakawa Y, Mitamura T (2016) Generating questions and multiple-choice answers using semantic analysis of texts. In: Proceedings of COLING 2016, the 26th International conference on computational linguistics: technical papers, pp 1125–1136
9. Becker L, Basu S, Vanderwende L (2012) Mind the gap: learning to choose gaps for question generation. In: Proceedings of ACL on human language technologies, pp 742–751. Association for Computational Linguistics
10. Bhatia AS, Kirti M, Saha SK (2013) Automatic generation of multiple choice questions using wikipedia. In: Proceedings of the pattern recognition and machine intelligence. Springer, Berlin, pp 733–738
11. Bholowalia P, Kumar A (2014) Ebk-means: A clustering technique based on elbow method and k-means in wsn. Int J Comput Appl 105(9)
12. Ch DR, Saha SK (2018) Automatic multiple choice question generation from text: A survey. IEEE Transactions on Learning Technologies
13. 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
14. Das B, Majumder M (2017) Factual open cloze question generation for assessment of learner's knowledge. Int J Educ Technol High Educ 14:1–12
15. Das B, Majumder M, Phadikar S (2018) A novel system for generating simple sentences from complex and compound sentences. Int J Modern Educ Comput Sci 10(1):57
16. Das B, Majumder M, Phadikar S, Sekh AA (2019) Automatic generation of fill-in-the-blank question with corpus-based distractors for e-assessment to enhance learning. Comput Appl Eng Educ 27(6):1485–1495
17. Divate M, Salgaonkar A (2017) Automatic question generation approaches and evaluation techniques. Current Science (00113891) 113(9)
18. Dostal M, Ježek K (2011) Automatic keyphrase extraction based on nlp and statistical method. Poster presentation of SVK, pp 140–145
19. Du X, Cardie C (2017) Identifying where to focus in reading comprehension for neural question generation. In: Proceedings of the 2017 conference on empirical methods in natural language processing, pp 2067–2073
20. Du X, Shao J, Cardie C (2017) Learning to ask: Neural question generation for reading comprehension. In: Proceedings of the 55th annual meeting of the association for computational linguistics (Volume 1: Long Papers), pp 1342–1352
21. Effenberger T (2015) Automatic question generation and adaptive practice. PhD thesis, Masarykova univerzita, Fakulta informatiky
22. Gao L, Gimpel K, Jensson A (2020) Distractor analysis and selection for multiple-choice cloze questions for second-language learners. In: Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications, pp 102–114. Association for Computational Linguistics
23. Gates DM (2011) How to generate cloze questions from definitions: A syntactic approach. In: 2011 AAAI Fall symposium series, pp 19–22
24. Goto T, Kojiri T, Watanabe T, Iwata T, Yamada T (2010) Automatic generation system of multiple-choice cloze questions and its evaluation. Int J Knowl Manag E-Learning 2(3):210–224
25. Karamanis N, An HL, Mitkov R (2006) Generating multiple-choice test items from medical text: A pilot study. In: Proceedings of the fourth international natural language generation conference, pp 111–113. Association for Computational Linguistics
26. Kim Y, Lee H, Shin J, Jung K (2019) Improving neural question generation using answer separation. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, pp 6602–6609
27. Knoop S, Wilske S (2013) Wordgap-automatic generation of gap-filling vocabulary exercises for mobile learning. In: Proceedings of the second workshop on NLP for computer-assisted language learning at NODALIDA; Oslo; Norway, number 086 in 17. Linköping University Electronic Press, pp 39–47
28. Kurdi G, Leo J, Parsia B, Sattler U, Al-Emari S (2020) A systematic review of automatic question generation for educational purposes. Int J Artif Intell Ed 30(1):121–204
29. Leo J, Kurdi G, Matentzoglu N, Parsia B, Sattler U, Forge S, Donato G, Dowling W (2019) Ontology-based generation of medical, multi-term mcqs. Int J Artif Intell Education, pp 1–44
30. Levy R, Andrew G (2006) Tregex and tsurgeon: tools for querying and manipulating tree data structures. In: LREC. Citeseer, pp 2231–2234
31. Li J, Huang G, Fan C, Sun Z, Zhu H (2019) Key word extraction for short text via word2vec, doc2vec, and textrank. Turkish J Electrical Eng Comput Sci 27(3):1794–1805

32. Liu M, Rus V, Liu L (2018) Automatic chinese multiple choice question generation using mixed similarity strategy. IEEE Trans Learn Technol 11(2):193–202
33. Lott B (2012) Survey of keyword extraction techniques. UNM Education, 50
34. Ma L, Zhang Y (2015) Using word2vec to process big text data. In: 2015 IEEE International Conference on Big Data (Big Data). IEEE, pp 2895–2897
35. Majumder M, Saha SK (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
36. Majumder M, Saha SK (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
37. Manning CD, Surdeanu M, Bauer J, Finkel JR, Bethard S, McClosky D (2014) The stanford corenlp natural language processing toolkit. In: Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations, pp 55–60
38. Marneffe Marie-CatherineD., Manning CD (2008) Stanford typed dependencies manual. Technical report, Technical report, Stanford University
39. Maurya KK, Desarkar MS (2020) Learning to distract: a hierarchical multi-decoder network for automated generation of long distractors for multiple-choice questions for reading comprehension, pp 1115–1124. Association for Computing Machinery, New York, NY, USA
40. Mazidi K, Nielsen RD (2014), vol 2, pp 321–326
41. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space.arXiv:1301.3781
42. Mitkov R, An HL, Karamanis N (2006) A computer-aided environment for generating multiple-choice test items. Nat Lang Eng 12(2):177–194
43. Naqvi SR, Akram T, Haider SA, Khan W, Kamran M, Muhammad N, Qadri NN (2019) Learning outcomes and assessment methodology: case study of an undergraduate engineering project. Int J Electri Eng Educ 56(2):140–162
44. Narendra A, Agarwal M, Shah R (2013) Automatic cloze-questions generation. In: Proceedings of recent advances in natural language processing, pp 511–515. Hissar, Bulgaria
45. Olszewska JI (2019) Designing transparent and autonomous intelligent vision systems. In: ICAART (2), pp 850–856
46. Patra R, Saha SK (2019) A hybrid approach for automatic generation of named entity distractors for multiple choice questions. Educ Inf Technol 24(2):973–993
47. Pugh D, Champlain AD, Gierl M, Lai H, Touchie C (2016) Using cognitive models to develop quality multiple-choice questions. Medical Teacher 38(8):838–843
48. Rose S, Engel D, Cramer N, Cowley W (2010) Automatic keyword extraction from individual documents. Text mining: applications and theory, pp 1–20
49. Santhanavijayan A, Balasundaram SR, Narayanan SH, Kumar SV, Prasad VV (2017) Automatic generation of multiple choice questions for e-assessment. International Journal of Signal and Imaging Systems Engineering 10(1-2):54–62
50. Smith S, Avinesh PVS, Kilgarriff A (2010) Gap-fill tests for language learners: Corpus-driven item generation. In: Proceedings of ICON: 8th international conference on natural language processing, pp 1–6
51. Subramanian S, Wang T, Yuan X, Zhang S, Trischler A, Bengio Y (2018) Neural models for key phrase extraction and question generation. In: Proceedings of the workshop on machine reading for question answering, pp 78–88. Association for Computational Linguistics
52. Sun X, Liu J, Lyu Y, He W, Ma Y, Wang S (2018) Answer-focused and position-aware neural question generation. In: Proceedings of the 2018 conference on empirical methods in natural language processing, pp 3930–3939
53. Susanti Y, Tokunaga T, Nishikawa H, Obari H (2018) Automatic distractor generation for multiple-choice english vocabulary questions. Res Pract Technol Enhanc Learn 13(1):15
54. Warrens MJ, van der Hoef H (2020) Understanding the rand index. In: Advanced studies in classification and data science. Springer, pp 301–313
55. Wongso R, Hanafiah N, Hartanto J, Alexander K, Sutanto C, Kesuma F (2018) Complaint analysis in indonesian language using wpke and rake algorithm. Int J Electr Comput Eng 8(6):5311
56. Yuan X, Wang T, Gulcehre C, Sordoni A, Bachman P, Zhang S, Subramanian S, Trischler A (2017) Machine comprehension by text-to-text neural question generation. In: Proceedings of the 2nd Workshop on Representation Learning for NLP, pp 15–25. Association for Computational Linguistics

57. Zhang S, Hu Y, Bian G (2017) Research on string similarity algorithm based on levenshtein distance. In: 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). IEEE, pp 2247–2251
58. Zhang Y, Jin R, Zhou Zhi-Hua (2010) Understanding bag-of-words model: a statistical framework. Int J Mach Learn Cybern 1(1-4):43–52

Affiliations

Bidyut Das[1] · Mukta Majumder[2] · Santanu Phadikar[3] · Arif Ahmed Sekh[4]

Mukta Majumder
mukta_jgec_it_4@yahoo.co.in

Santanu Phadikar
sphadikar@yahoo.com

Arif Ahmed Sekh
skarifahmed@gmail.com

1 Haldia Institute of Technology, Haldia, India

2 University of North Bengal, Siliguri, India

3 Maulana Abul Kalam Azad University of Technology, West Bengal, India

4 XIM University, Bhubaneswar, India