

A System for Generating Multiple Choice Questions: With a Novel Approach for Sentence Selection

Mukta Majumder

Computer Centre, Vidyasagar University
Midnapore, West Bengal, India-721102
mukta_jgec_it_4@yahoo.co.in

Sujan Kumar Saha

Computer Science and Engg. Department
BIT, Mesra, Jharkhand, India-835215
sujan.kr.saha@gmail.com

Abstract

Multiple Choice Question (MCQ) plays a major role in educational assessment as well as in active learning. In this paper we present a system that generates MCQs automatically using a sports domain text as input. All the sentences in a text are not capable of generating MCQs; the first step of the system is to select the informative sentences. We propose a novel technique to select informative sentences by using topic modeling and parse structure similarity. The parse structure similarity is computed between the parse structure of an input sentence and a set of reference parse structures. In order to compile the reference set we use a number of existing MCQs collected from the web. Keyword selection is done with the help of occurrence of domain specific word and named entity word in the sentence. Distractors are generated using a set of rules and name dictionary. Experimental results demonstrate that the proposed technique is quite accurate.

1 Introduction

MCQ generation is the task of generating questions from various text inputs, having prospective learning content. MCQ is a popular assessment tool used widely in various levels of educational assessment. Apart from assessment MCQ also acts as an effective instrument in active learning. It is studied that, in active learning classroom framework conceptual understanding of the students can be boosted by posing MCQs on the concepts just taught (Mazur, 1997; Nicol

2007). Thus the MCQ is becoming an important aspect for next generation learning, training and assessment environments.

Generation of Multiple Choice Question manually is a time-consuming and tedious task which also requires domain expertise. Therefore an automatic MCQ generation system can leverage the active learning and assessment process. Consequently automatic MCQ generation became a popular research topic and a number of systems have been developed (Coniam 1997; Mitkov, Ha, & Karamanis, 2006; Karamanis, Ha, & Mitkov, 2006; Pino, Heilman, & Eskenazi, 2008; Agarwal & Mannem, 2011). Generation of MCQ automatically consists of three major steps; (i) selection of sentences from which question can be generated, (ii) identification of the keyword which is the correct answer and (iii) generation of distractors that are the wrong answers (Delphine Bernhard, 2010).

All the sentences of a textual document cannot be the candidates for being question sentences or stems. The sentence that contains sufficient and quality information can act as MCQ stem; moreover keyword and corresponding distractors should be available. Hence the target is to select only the informative sentences from which factual MCQs can be generated for testing the content knowledge of the learner. Therefore, selection of sentence has been playing a pioneer role in automatic MCQ generation task. But unfortunately in the literature we have found that the sentence selection task has become unable to achieve adequate attention from the researchers. As a result, the sentence selection task is confined in a limited number of approaches by using only a set of rules or checking the occurrence of a set of pre-defined features and pattern. Success of such approaches suffers from

the quality of the rules or features and thus become extremely domain reliant.

In this paper we propose an efficient technique for informative sentence selection and generation of MCQs from the selected sentences. Here we select the informative sentences based on certain words that are important to define the domain or topic and parse structure similarity. The proposed system is robust and expected to work in a wide range of domains. As input to the system we consider the Wikipedia and news article which are trusted sources of information. To generate a MCQ from a sentence, first we perform a set of pre-processing tasks like, converting complex and compound sentences into simple sentences and co-reference resolution. Then we use topic modeling as another pre-processing step that finds the subject words or topics of the domain and check whether the sentence contains any of these topics. This will reduce our overhead in subsequent steps. We have found that two sentences contain similar parse structures, are generally of similar type and carry same type of facts. Therefore, parse structure of a sentence may play an important role in sentence selection. We collect a set of MCQs available in the Internet in the domain of interest and form sentences from them. Here we like to mention that we have chosen sports domain specially cricket as a case study because of wide availability of existing MCQs in this domain. We obtain parse structures of these sentences and the common structures are saved as a reference set. Next we compare the parse tree of an input sentence with the reference set structures. If the sentence has structural similarity with any of the reference set structures then it is considered as an informative sentence for MCQ stem generation.

Next we perform other subtasks namely, keyword selection and distractor generation. Keyword selection is done by a rule based approach based on cricket domain specific words and named entities (NE) in the sentence. Generation of distractors is done using a gazetteer list based approach. The following sections present the details of the system.

2 Previous Work

Generating Multiple Choice Question automatically is a relatively new and important research area and potentially useful in Education Technology. Here we first discuss a few systems for MCQ generation.

Coniam (1997) presented one of the earlier attempts of MCQ generation. They used word frequencies for an analyzed corpus in the various phases of the development. They matched parts-of-speech and word frequency of each test item with similar word class and word frequency options to construct the test items. Mitkov and Ha (2003) and Mitkov et al. (2006) used NLP techniques like shallow parsing, term extraction, sentence transformation and computation of semantic distance in their works for generating MCQ semi automatically from an electronic text. They did term extraction from the text using frequency count, generated stems using a set of linguistic rules, and selected distractors by finding semantically close concepts using WordNet. Brown (2005) developed a system for automatic generation of vocabulary assessment questions. They used WordNet for finding definition, synonym, antonym, hypernym and hyponym in order to generate the questions as well as the distractors. Aldabe et al. (2006) and Aldabe and Maritxalar (2010) developed systems to generate MCQ in Basque language. They have divided the task into six phases: selection of text (based on learners and length of texts), marking blanks (manually), generation of distractors, selection of distractors, evaluation with learners and item analysis. Papasalouros et al. (2008) proposed an ontology based approach for development of an automatic MCQ system. Agarwal et al. (2011) presented a system that automatically generates questions from natural language text using discourse connectives.

As in this paper we focus on sentence selection, next we like to discuss the sentence selection strategies used in various works. In order to MCQ stem generation different types of rules have been defined manually or semi-automatically for selecting informative sentences from a corpus; these are discussed as follows. Mitkov et al. (2006) selected sentences if they contain at least one term, is finite and is of SVO or SV structure. Karamanis et al. (2006) implemented a module to select clause, having some specific terms and filtering out sentences which having inappropriate terms for multiple choice test item generation (MCTIG). For sentence selection Pino et al. (2008) used a set of criteria like, number of clause, well-defined context, probabilistic context-free grammar score and number of tokens. They also manually computed a sentence score based on occurrence of these criteria in a given sentence and select the sentence as informative if the score is higher than a threshold. For sentence selection Agarwal and Mannem (2011) used a number of features like:

is it first sentence, contains token that occurs in the title, position of the sentence in the document, whether it contains abbreviation or superlatives, length, number of nouns and pronouns etc. But they have not clearly reported what should be optimum value of these features or how the features are combined or whether there is any relative weight among the features. Kurtsov (2013) applied some predefined rules that allow selecting sentences of a particular type. For example, the system recognizes sentences containing definitions, which can be used to generate a certain category of test exercise. For ‘Automatic Cloze-Questions Generation’ Narendra et al. (2013) in their paper directly used a summarizer, MEAD for selection of important sentences. Bhatia et al. (2013) used pattern based technique for identifying MCQ sentences from Wikipedia. Apart from these rule and pattern based approaches we also found an attempt on using supervised machine learning technique for stem selection by Correia et al. (2012). They used a set of features like parts-of-speech, chunk, named entity, sentence length, word position, acronym, verb domain, known-unknown word etc. to run Support Vector Machine (SVM) classifier. Another approach was presented by Majumder and Saha (2015), which used named entity recognition, based rule mining along with syntactic structure similarity for sentence selection.

3 Pre-processing on Input Text

MCQ is generally made from a simple sentence but we have found that many of the Wikipedia and news article sentences are long, complex and compound in nature. Moreover, a number of these sentences are having coreference issues. Our system first aims to identify informative sentences from Wikipedia and news articles for stem generation. The proposed technique is based on parse structure similarity; hence the structure of the sentences plays a major role in the task. In order to obtain better structural similarity we first apply a few pre-processing steps that are discussed below.

3.1 Co-reference Resolution and Simple Sentence Generation

First preprocessing step we employ is transforming complex and compound sentences into simple form. Moreover, to resolve the coreference issues we perform coreference resolution. Coreference has been defined as, referring of the same object (e.g., person) by two or more expressions

in a corpus. For generating question the referent must be identified from such sentences. We consider the following sentence as an example.

The 2012 ICC World Twenty20 was the fourth ICC World Twenty20 competition that took place in Sri Lanka from 18 September to 7 October 2012 which was won by the West Indies.

This sentence is complex in nature and it has coreference problem. In this sentence ‘that’ and ‘which’ are referring to ‘2012 ICC World Twenty20’. A simple sentence is built up from one independent clause where a compound or complex sentence is consisted of at least two clauses. So the task is to split complex or compound sentence into clauses that can form simple sentences.

To convert the sentence into simple form we use the openly available ‘Stanford CoreNLP Suite’¹. The tool is not directly converting the complex and compound sentences into simple ones. It provides the parse result of the example sentence in Stanford typed dependency (SD) notations (Marneffe et al., 2008). We analyze the dependency structure provided by the tool in order to convert it. We use ‘Stanford Deterministic Coreference Resolution System’, which is basically a module of the ‘Stanford CoreNLP Suite’, for coreference resolution. Finally we get the following simple sentences from the aforementioned example sentence.

Simple1: The 2012 ICC World Twenty20 was the fourth ICC World Twenty20 competition.

Simple2: The 2012 ICC World Twenty20 took place in Sri Lanka from 18 September to 7 October 2012.

Simple3: The 2012 ICC World Twenty20 was won by the West Indies.

3.2 Subject or Topic Word Identification and Potential Candidate Sentence Selection

The sentence selection strategy for MCQ stem generation is based on parse tree similarity. We need to compare an input sentence with reference set of structures for selecting it as the basis of a MCQ. But the size of such input text is huge. Therefore comparing these vast numbers of sentences with reference structures will be a gigantic task. To reduce this overhead we have taken the help of topic modeling which can identify the topic words of the domain and if the test sentence is not containing a topic then reject it. We also found that the sentence with the topic word

¹ <http://nlp.stanford.edu/software/corenlp.shtml>

is more informative than the sentences which are not containing any domain or topic specific words. This approach will identify a set of potential candidate sentences and simplifies the task of parse tree comparison.

We use the openly available Topic Modeling Tool (TMT)² to identify the topic words as well as the distribution of these words in the sentences. We run the topic modeling tool on the Wikipedia pages and news articles that we considered as input for sentence selection, and get the topic words. Some of the identified topic words are, 'World Cup', 'World Twenty20', 'Champions Trophy', 'Knock Out Tournament', 'Indian Premier League or IPL' etc. Now we check whether an input sentence is containing any of these topic words or not.

4 Sentence Selection for MCQ Stem Generation

The syntactic structure can play a key role in sentence selection for MCQ. The parse tree of a particular question sentence is able to retrieve many informative sentences have similar structure. For example, the aforementioned Wikipedia sentence 'Simple3' (in Section 3.1) is defining the fact that a team has won a series/tournament. The parse structure of the sentence is similar with many sentences carrying 'team wins series' fact. The sentences like '1983 ICC World Cup was won by India.', '2006 ICC Champions Trophy was won by Australia.' have similar parse trees and these can be retrieved if the parse structure shown in Figure 1 is considered as a reference structure. From this observation we aim to collect a set of such syntactic structures that can act as the reference for retrieving new sentences from the web.

4.1 Reference Sentence Formation

For the parse tree matching we require a reference set of parse structures with which the input sentences will be compared. We compile the reference set from existing MCQs. We found that in the sports domain a large number of MCQs are available in the Internet. We collect about 400 MCQs for the reference set creation.

As we have discussed earlier, a MCQ is mainly composed of a stem and a few options. Generally the stems are interrogative in nature. Our system is supposed to identify informative sentences from Wikipedia and news articles. Most

of the sentences in Wikipedia pages and news articles are assertive. In order to get the structural similarity the reference sentences and the input sentences should be in same form. Therefore we convert the collected stems into assertive form. For this conversion we replace the 'wh' phrase or the blank space of the stem by the first alternative of the option set. For example:

MCQ: Which country won the first World Cup Cricket tournament held in England in 1975?

a) England b) India c) Australia d) Pakistan e) West Indies

Reference Sentence: England won the first World Cup Cricket tournament held in England in 1975.

Here we like to mention that in this phase our target is to compile a reference set containing a number of grammatically correct sentences, not to extract the fact from the existing MCQ. Even if the first option is not the correct answer of the given question, our target of reference set creation is satisfied. The set of sentences generated using the approach is referred as 'reference sentence'.

4.2 Parse Tree Comparison

We generate the parse tree of the reference set sentences using the openly available Stanford Parser 3. In the sports domain the questions (MCQs) deal with the facts embedded in the sentences. Therefore, the tense information of the sentences is not so important for question formation but tense information leads to alter the parse structure. For example, 'In the 2012 season Sourav Ganguly has been appointed as the Captain for Pune Warriors India.' and 'In the 2013 season Graeme Smith was announced as the captain for Surrey County Cricket Club.' the two sentences are describing similar type of fact but parse structure is different due to the difference in verb form. This type of phenomena occurs in 'noun' subclasses also: singular noun vs plural noun, common noun vs proper noun etc. For the sake of parse tree matching we have used a coarse-grain tagset where a set of subcategories of a particular word class is mapped into one broader category. From the original Penn Treebank Tagset (Santorini, 1990) used in Stanford Parser we derive the new tagset and modify the sentences accordingly. For this purpose first we create parse trees and replace the tags or words according to the new tagset in the parse structures.

² <http://code.google.com/p/topic-modeling-tool/>

³ <http://nlp.stanford.edu/software/lex-parser.shtml>

For example, we map VBZ-VBN (has been), VBD (was) and VBG (chasing) into ‘VB’; similarly ‘NN’, ‘NNS’, ‘NNP’ and ‘NNPS’ are mapped into ‘NN’ etc.

Once we get the parse trees of the reference sentences and test sentences, we need to find the similarity among them. In order to find the similarity in these parse trees we have proposed the Parse Tree Matching (PTM) Algorithm.

The algorithm is basically trying to find whether the sentences have similar structure. The parse tree matching algorithm considers only the non-leaf nodes during the matching process. All other words that occur as leaf of the tree are not playing any role in the parse tree matching.

Algorithm 1: Parse Tree Matching (PTM) Algorithm

input : Parse Tree T1, Parse Tree T2

output : 1 if T1 is similar with T2, 0 otherwise

1. T1 and T2 are using the coarse-grain tagset.
 2. Set Cnode1 as root of T1 and Cnode2 as root of T2;
 3. if (label (Cnode1) = label (Cnode2) and number of children (Cnode1) = number of children (Cnode2)) then
 4. n=number of children of Cnode1;
 5. for (i= 1 to n) do
 6. if both Cnode1_child_i and Cnode2_child_i are non-leaf then
 7. if label(Cnode1_child_i) !=label(Cnode2_child_i)
 8. then return 0 and exit;
 9. end
 10. if Only one of Cnode1_child_i and Cnode2_child_i is leaf then
 11. return 0 and exit;
 12. end
 13. end
 14. Increase level by 1, update Cnode1 and Cnode2, and Go to Step 4;
 15. return 1;
 16. else
 17. return 0 and exit;
 18. end
-

We have found that some of the reference sentences are having similar parse structures. Therefore first we run the PTM Algorithm among these parse trees generated from the reference set of sentences to find the unique set of structures. During this phase argument ‘T1’ of the algorithm is a parse tree of the reference set sentence and the argument ‘T2’ is the parse tree of another reference set sentence. We run this algorithm for several iterations: by keeping ‘T1’ fixed and varying ‘T2’ for all the parse trees.

The sentences for which the matches are found are basically of similar type and we keep only one of these in the reference set and discard the others. By applying the procedure finally we generate the reduced set of parse structures.

Once the reference structures are finalized, we used them for finding new Wikipedia and news article sentences which have similar structure. For this purpose we run the proposed PTM Algorithm repeatedly in the same way as mentioned above. Here we set the argument ‘T1’ as the parse structure of a test sentence and argument ‘T2’ as a reference structure. We fix ‘T1’ and vary the ‘T2’ among the reference set structures until a match is found or we come to the end of the reference set. If a match is found then the sentence (whose structure is ‘T1’) is selected.

1998 ICC Knock Out Trophy was won by South Africa.
 (ROOT
 (S
 (NP (CD 1998) (NN ICC) (NN Knock) (NN Out) (NN Trophy))
 (VP (VB was)
 (VP (VB won)
 (PP (IN by)
 (NP (NN South) (NNP Africa))))))
 (. .)))

Figure 1. Reference Structure One

Figure 1 is a reference structure and Figure 2 and Figure 3 are showing two input structures. When the PTM Algorithm is executed a match is found in between Figure 1 and Figure 2. The other input structure (Figure 3) does not have similarity with any of the reference trees.

The 2002 ICC Champions Trophy was held in Sri Lanka.
 (ROOT
 (S
 (NP (DT The) (CD 2002) (NN ICC) (NN Champions) (NN Trophy))
 (VP (VB was)
 (VP (VB held)
 (PP (IN in)
 (NP (NN Sri) (NN Lanka))))))
 (. .)))

Figure 2. Input Structure which matches with reference set

After this phase we have successfully selected a set of sentences which is used to form MCQ stems. Keyword extraction and distractors generation are also done from these selected sentences. Question generation, keyword extraction and distraction are discussed as follows.

5 Keyword Identification, Question Formation and Distractors Generation

A MCQ consists of a stem along with the option set which contains a keyword and distractors.

Therefore we need to identify the keyword and form the distractors to generate a multiple choice question.

```

The Kolkata Knight Rider is the champions, having won the IPL 2014.
(ROOT
(S
(NP (DT The) (NN Kolkata) (NN Knight) (NN Rider))
(VP (VB is)
(NP (DT the) (NN champions))
(, .)
(S
(VP (VB having)
(VP (VB won)
(NP (DT the) (NNP IPL) (CD 2014))))))
( . )))

```

Figure 3. Input Structure which does not matches with reference structure

5.1 Keyword Identification

Keyword identification is the next phase where we select the word (or n-gram) that has the potential to become the right answer of the MCQ. We have found that some particular patterns are followed by these potential sentences which are having some specific named entities (NEs). For the identification of these keys we have taken the help of the named entity recognition (NER) system developed by Majumder and Saha (2014). And the domain specific words like, tournament, series, trophy, captain, wicket, bowler, batsman, wicket-keeper, umpire, pitch, opening ceremony, etc are very important to identify these patterns in the sentences. Therefore we have also compiled a list of such domain specific words. For example, “opening ceremony was held in” pattern retrieves sentences containing the name of the location (city name or ground name) where the opening ceremony of a tournament was held. Therefore the key for this pattern is the location name in the retrieved sentence. Similarly, “the man of the tournament” pattern extracts sentences having the name of the player who got the man of the tournament in a particular tournament. Here the key for the pattern is the person name. The pattern “team won the tournament/series” is retrieving the team or country name that won the series or tournament; therefore the corresponding key is the country or team or franchise name. The sentences are tagged using the NER system and the corresponding entity is selected as the key.

5.2 Question Formation

After the keyword is identified we can form the question by replacing it with proper ‘wh-word’. We have also consulted the parse tree structure of the sentence to bring the ‘wh-word’ at the appropriate position in the stem of the MCQ. For different type of keyword appropriate ‘wh-word’ is selected. For example if the category is location then the ‘wh-word’ is where; similarly, for person: who, for date: when, for number: how many etc.

5.3 Distractors Generation

Distractors are closely related to keyword. These are the distraction for the right answer in a MCQ. In this cricket domain majority of the distractors are named entity. Here first we identify the class of the key and search for a few close members using a gazetteer list based approach.

We compile a few gazetteer lists using the web. In this cricket domain the major categories of key (or, distractors) are: person name (cricketer, bowler, batsman, wicketkeeper, captain, board president, team owner etc.), organization name (country name, franchise name, cricket boards like ICC etc.), event name (cup, tournament, trophy, championship etc.), location name (cricket ground, city etc.). For each of the name categories we extract lists of names from relevant websites. For example, for cricketers we search the Wikipedia, Yahoo! Cricket and Espncricinfo player’s lists. Then we search the key in these lists to determine the class of the key.

For each name category we select a set of attributes. The Wikipedia pages normally contain an information template on the title (at the top-right portion of the page) that contains a set of properties defining the class. Additionally, majority of the cricket related pages contain a table for summarizing the topic. Those fields of the tables are extracted to become member of the attribute set. For example, if we consider the category batsman, the attribute set may include date-of-birth, span, team name, batting style, last match, total run, batting average, strike rate, number of century, number of half-century, highest score etc. The detailed strategy is discussed as follows.

Next we search for a list of related tokens of the same category in the Wikipedia. For a cricketer key we run a search query “list of <national side> cricketers”; if the ‘is-captain’ attribute value is true, then the query is “List of <national side> national cricket captains”. From the search result in Wikipedia pages we extract a set of sim-

ilar entities. *Similar entity* is defined as the entities that have certain attribute value same as the *key*. We have predefined a set of attributes as 'important' for each class. For the *cricketer* class we consider the attributes *country*, *span* (overlapping), *batting average* (difference less than ten) or *bowling average* (difference less than five). Similarly, for the *ground* class we use only the *country* attribute; for the *team* class we consider the *country* and common *trophy/tournament* attributes as important. The entities which have match in important attributes are considered as candidate distractor. And from these candidate distractors we randomly pick three to four entities as the list of distractors.

6 Result and Discussion

We have already mentioned that the system is tested on cricket related Wikipedia pages and news article. In order to evaluate the performance of the sentence selection module we consider the quality of the retrieved sentence - whether this is really able to act as a MCQ stem.

There is no benchmark or gold- standard data in the task. In order to evaluate the performance of the system we have taken a few Wikipedia pages and news articles as input on which we run the system. The question formation capability of the retrieved sentences is examined by a set of human evaluators. The evaluators count the number of sentences that are potential to become the basis of a MCQ ('correct retrieval'). The average of the percentage of correct retrieval is considered as the accuracy of the system.

For computing the accuracy of the system we consider six Wikipedia pages. These are the pages on 2003, 2007, 2011 ICC Cricket World Cup, ICC Champions Trophy, IPL 2014 and T20 World Cup 2014 and four sports news articles from The Times of India, a popular English daily of India related to the T20 World Cup 2014, namely, 'Sri Lankans Lord Over India', 'Yuvi cuts a sorry figure in final', 'Virat, the lone man standing for India' and 'Mahela, Sangakkara bow out on a high'. Only the text portions of these pages are taken as input that contains a total of ~795 sentences. From these input text ~508 sentences were selected after the topic word based filtering. Then we apply the parse tree matching algorithm which finally considers 112 sentences. These sentences are examined by five human evaluators. They consider 105, 104, 103, 106 and 104 sentences respectively as correct retrieval. Therefore the accuracy of the system is

93.21%. Table 1 summarizes the accuracy of the system.

Input Sentence	Sentence After TMT	Sentence After PTMA	Evaluators Judgment	% Accuracy
~795	508	112	Evaluator1: 105	93.21%
			Evaluator2: 104	
			Evaluator3: 103	
			Evaluator4: 106	
			Evaluator5: 104	

Table 1: Performance of the developed System

From the evaluation score given by the human evaluators it is clear that the proposed system is capable of retrieving quality sentences from an input document. In addition to the correct retrievals, the system also selects a few sentences that are not considered as 'good' by the evaluators. We have analyzed these sentences. As for example we have listed the following sentences:

Netherlands and Canada were both appearing in the Cricket World Cup for the second time.

Ireland had been the best-performing associate member since the previous World Cup.

These sentences are containing the topic words and matching with the reference set structures. But these are missing out of some important information for which the fact is incomplete. The time or year related information is missing in both the sentences. A modified topic modeling system may be used to consider a tournament name with year is a topic but only the tournament name without year is not.

While comparing with the existing technique (Majumder and Saha, 2015), we found that the proposed technique identifies more number of sentences after pre- processing and post-processing steps. Omission of domain specific word and NER based rule mining restriction not only make the proposed system domain independent but also it outperforms the existing system in terms of selecting number of sentences.

Next we measure the performance of the overall MCQ system. After sentence selection, key selection and distractor generation are the major modules. We evaluate the performance of these modules using: key selection accuracy (whether the key is selected properly), distractor quality (whether the distractors are good). Again we employ the human evaluators to assess the system. The average evaluation accuracy of key selection is 83.03% (93 out of 112) and in distractor quality the accuracy is 91.07% (102 out of 112).

A few examples of the generated MCQs are given below:

1. Which country won the 2014 ICC World Twenty20?
 - a) Australia b) India c) West Indies d) Sri Lanka
2. Who was the man of the series of the 2011 ICC Cricket World Cup?
 - a) Sachin Tendulkar b) Tillakaratne Dilshan c) Yuvraj Singh d) Kumar Sangakkara

7 Conclusion

In this paper we have presented a novel technique for selecting informative sentences for multiple choice questions generation from an input corpus. The proposed technique selects informative sentences based on topic word and parse structure similarity. The system also uses a set of pre-processing steps like simplification of sentences, co-reference resolution etc. The selected sentences are used in the key selection and distractor generation modules to make a complete automatic MCQ system. We test the system in sports domain and use Wikipedia pages and news articles as input corpus. But we feel the system is generic and expected to work well in other domains also.

We have deeply studied the false identifications and observed that the accuracy of the system can be further improved by incorporating better pre-processing and post processing steps. A deeper co-reference resolution system can be used to remove a number of semi-informative sentences. Better identification of domain specific phrases or topics can also be helpful to handle a number of false detections. These observations may lead us to continue work in future.

Reference

- Agarwal, M., and 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.
- Agarwal, M., Shah, R., and 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.
- Aldabe, I., Lopez de Lacalle, M., Maritxalar, M., Martinez, E., Uria, L., 2006. ArikIturri: An Automatic Question Generator Based on Corpora and NLP Techniques. In ITS. LNCS 4053, pp. 584-594.
- Aldabe, I., Maritxalar, M., 2010. Automatic Distractor Generation for Domain Specific Texts. Proceedings of IceTAL, LNAI 6233. pp. 27-38.
- Bernhard, D., 2010. Educational Applications of Natural Language Processing. In NATAL. pp. 1-123.
- Bhatia, A. S., Kirti, M., and Saha, S. K., 2013. Automatic Generation of Multiple Choice Questions Using Wikipedia. In Proceedings of Pattern Recognition and Machine Intelligence, Springer Berlin Heidelberg, pp. 733-738.
- Brown, J. C., Frishkoff, G. A., and 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, Association for Computational Linguistics, pp. 819-826.
- 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), pp. 15-33.
- Correia, R., Baptista, J., Eskenazi, M., and Mamede, N., 2012. Automatic generation of cloze question stems. In Computational Processing of the Portuguese Language, Springer Berlin Heidelberg, pp. 168-178.
- De Marneffe, M. C., and Manning, C. D., 2008. The Stanford typed dependencies representation. In Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation, Association for Computational Linguistics, pp. 1-8.
- Karamanis, N., Ha, L. A., and Mitkov, R., 2006. Generating multiple-choice test items from medical text: A pilot study. In Proceedings of the Fourth International Natural Language Generation Conference, Association for Computational Linguistics, pp. 111-113.
- Kurtasov, A., 2013. A System for Generating Cloze Test Items from Texts in Russian. In Proceedings of the Student Research Workshop associated with RANLP 2013, pp. 107-112.
- Majumder, M., Saha, S. K., 2014. Development of NER System for Wikipedia without using Wikipedia text as training data: Sports (Cricket) a case study. In Proceedings in EIIC-The 3rd Electronic International Interdisciplinary Conference (No. 1).
- Majumder, M., Saha, S. K., 2015. Automatic selection of informative sentences: The sentences that can generate multiple choice questions. Knowledge Management & E-Learning: An International Journal (KM&EL), 6(4), 377-391.
- Mazur, E., 1997. Peer instruction. Upper Saddle River, NJ: Prentice Hall. pp. 9-18.

- Mitkov, R., Ha, L.A., 2003. Computer-aided generation of multiple-choice tests. Proceedings of the HLT/NAACL Workshop on Building educational applications using Natural Language Processing, pp. 17–22.
- Mitkov, R., Ha, L. A., and Karamanis, N., 2006. A computer-aided environment for generating multiple-choice test items. *Natural Language Engineering*, Vol. 12(2), pp. 177-194.
- Narendra, A., Agarwal, M. and Shah, R., 2013. Automatic Cloze-Questions Generation. In Proceedings of Recent Advances in Natural Language Processing, Hissar, Bulgaria, pp. 511–515.
- Nicol, D., 2007. E-assessment by design: using multiple-choice tests to good effect. *Journal of Further and Higher Education*, Vol. 31(1), pp. 53-64.
- Papasalouros, A., Kanaris, K., and Kotis, K., 2008. Automatic Generation Of Multiple Choice Questions From Domain Ontologies. In *e-Learning*, pp. 427-434.
- Pino, J., Heilman, M., and 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, Montreal, Canada, pp. 22-32.
- Santorini, B., 1990. Part-of-Speech Tagging Guideline for Penn Treebank Project (3rd Revision, 2nd Printing).