Automatic selection of informative sentences: The sentences that can generate multiple choice questions

Mukta Majumder
Sujan Kumar Saha
Birla Institute of Technology, Mesra, Ranchi, India

Recommended citation:
Automatic selection of informative sentences: The sentences that can generate multiple choice questions

Mukta Majumder*
Department of Computer Science and Engineering
Birla Institute of Technology, Mesra, Ranchi 835215, India
E-mail: mukta_igeec_it_4@yahoo.co.in

Sujan Kumar Saha
Department of Computer Science and Engineering
Birla Institute of Technology, Mesra, Ranchi 835215, India
E-mail: sujan.kr.saha@gmail.com

*Corresponding author

Abstract: Traditional education cannot meet the expectation and requirement of a Smart City; it require more advance forms like active learning, ICT education etc. Multiple choice questions (MCQs) play an important role in educational assessment and active learning which has a key role in Smart City education. MCQs are effective to assess the understanding of well-defined concepts. A fraction of all the sentences of a text contain well-defined concepts or information that can be asked as a MCQ. These informative sentences are required to be identified first for preparing multiple choice questions manually or automatically. In this paper we propose a technique for automatic identification of such informative sentences that can act as the basis of MCQ. The technique is based on parse structure similarity. A reference set of parse structures is compiled with the help of existing MCQs. The parse structure of a new sentence is compared with the reference structures and if similarity is found then the sentence is considered as a potential candidate. Next a rule-based post-processing module works on these potential candidates to select the final set of informative sentences. The proposed approach is tested in sports domain, where many MCQs are easily available for preparing the reference set of structures. The quality of the system selected sentences is evaluated manually. The experimental result shows that the proposed technique is quite promising.

Keywords: Educational assessment; Multiple choice questions; Question generation; Sentence selection; Parse tree matching; Named entity recognition

Biographical notes: Mukta Majumder is a Ph.D scholar in Computer Science and Engineering Department, Birla Institute of Technology, Mesra, Ranchi, India. He has completed his post graduation from National Institute of Technical Teachers Training and Research’s, Kolkata, India and graduation from Jalpaiguri Government Engineering College, Jalpaiguri, India. His main research interests include Text Processing, Machine Learning, Micro-fluidic System, and Biochip etc.

Dr. Sujan Kumar Saha is working as Assistant Professor in the department of Computer Science and Engineering, Birla Institute of Technology Mesra, Ranchi, India. His main research interests include Natural Language Processing,
1. Introduction

The concept of Smart Cities comes from the urbanization and its consequences in today’s modern cities. Almost more than fifty percent of the world population lives in the urban areas (Dirks, Gurdgiev, & Keeling, 2010; Dirks & Keeling, 2009; Dirks, Keeling, & Dencik, 2009). And as mentioned by Chourabi et al. (2012) people and communities are two important aspects of a smart city development. To design and develop a smart city, technological and educational development of its population is highly important to sustain the smart city initiatives. Naturally the literacy of its human resource becomes a significant issue. Thus to build a smart city, education and learning technology plays a vital role. Giovannella et al. (2013) showed the importance of education in the Smart City initiatives. Not only the conventional education is enough; to match smart city requirement we need next generational education technology like Active Learning where the learner also lively participate in the process.

Multiple choice question (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. This implies the significance of MCQs in Modern Age and Smart Cities education.

Manual creation of questions is time-consuming and requires domain expertise. Therefore the questions are required to be prepared by the instructors and this laborious task often brings boredom to the instructor and as a result the benefits of active learning are suppressed. Therefore an automatic system for MCQ generation 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, & Eskkenazi, 2008; Agarwal & Mannem, 2011). An automatic MCQ generation system consists of three major components (i) selection of sentence from which question sentence or stem can be formed, (ii) identification of the keyword that will be the correct alternative and (iii) generation of distractors which will be the wrong answers set (Bernhard, 2010).

In general, MCQ is effective to assess well-defined knowledge or concepts. Such concepts are embedded in the relevant study materials i.e., the text. Moreover, majority of the MCQ concepts are confined to a few sentences of the text. Additionally, a conceptual question cannot be formed from every sentences of the text; a portion of the sentences carry the concepts that can be asked to the examinee. Therefore selection of the sentence from which a question can be made plays a vital role in the automatic MCQ generation task. But unfortunately in the literature we observe that the sentence selection phase failed to achieve sufficient attention of the researchers. As a result, the sentence selection task is confined in a limited number of approaches. Most of the available systems select sentences by using a set of rules or checking the occurrence of a set of pre-defined features. Effectiveness of such approaches suffers from the quality of the rules or features and these are highly domain dependent.
As an alternative, in this paper we propose a novel parse-tree matching based approach for potential MCQ sentence selection. The approach is based on computation of parse tree similarity of a target sentence with a set of reference sentences. Therefore, for the task we need a set of sentences that acts as a reference set. In order to create the reference set we collect a number of existing MCQs from which the reference sentences are extracted. Most of the collected stems are interrogative in nature. These are converted into assertive sentences by a set of simple steps, that basically targets to replace the ‘wh-words’ or the ‘blank space’ by the first alternative. We are primarily interested in the structure of the sentence, not the fact embedded in the sentence; therefore we do not judge whether the first alternative is correct or not. Then we generate the parse structures of these sentences and find the most common parse structures that act as the reference structure set. Now a set of pre-processing tasks, like, converting complex and compound sentences into simple sentences and co-reference resolution, are performed on the input sentences to make them simple. Then the parse structure of a simplified input sentence is matched with the reference set of structures. If there is a match then we conclude that the sentence is a potential candidate to generate MCQ.

The proposed approach is generic and expected to work across the domains. However the proposed approach necessitates a number of MCQs for creating the reference set. Hence the availability of existing MCQs is essential for the execution as well as performance of the approach. We observe that in the web a lot of MCQs are available in the sports domain. Therefore we adopt this domain for the assessment of the proposed approach. Available sports related MCQs are collected to create the reference set and the system is applied on suitable Wikipedia pages and news articles to identify candidate MCQ sentences. Next we observe that most of the questions asked in this domain deal with named entities. In order to improve the performance we incorporate a named entity recognition (NER) system and a set of named entity based rules as a post-processing phase. The quality of the system identified sentences is evaluated manually. The experimental results demonstrate the efficiency and precision of the proposed approach.

2. Related work
Development of automatic MCQ generation system has become a popular research problem in the last few years. In the literature we observe that generally automatic MCQ systems have followed three major steps: selection of sentence (or stem), selection of target word and generation of distractors. A few works on MCQ generation are discussed below.

Mitkov and Ha (2003) and Mitkov, Ha, and Karamanis (2006) developed semi-automatic systems for MCQ generation from a text book on linguistics. They used several NLP techniques like shallow parsing, term extraction, sentence transformation and computation of semantic distance for the task. They also employed natural language corpora and ontologies such as WordNet. Their system consists of three major modules: a. term extraction from the text, which is basically done by using frequency count; b. stem generation for identifying eligible clauses where a set of linguistic rules are used; c. distractors selection by finding semantically close concepts using WordNet. Brown, Frishkoff, and Eskenazi (2005) developed a system for automatic generation of vocabulary assessment questions. In this task they used WordNet for finding definition, synonym, antonym, hypernym and hyponym in order to develop the questions as well as the distractors. Aldabe, Lopez de Lacalle, Maritxalar, Martinez, and Uria (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 level of the learners and the length of the texts), marking blanks (done manually), generation of distractors, selection of distractors, evaluation with learners and item analysis. The generated questions are used for learners’ assessment in the science domain. Papasalouros, Kanaris, and Kotis (2008) proposed an ontology based approach for development of an automatic MCQ system. They have used the structure of ontology that is the concepts, instances and the relationship or properties that relates the concepts or instances. First they formed sentences from the ontology structure and then they found distractors from the ontology. Basically the distracters are related instances/classes having similar properties. Agarwal and Mannem (2011) presented a system for generating gap-fill questions, a problem similar to MCQ, from a biology text book. They also divided their work into three phases: sentence selection, key selection and distractors generation.

Next we discuss the sentence selection strategies used in various works. In the literature we found that primarily the rule based and pattern matching based approaches have been followed for sentence selection in MCQ. Toward 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, Ha, and Karamanis (2006) selected sentences if they contain at least one term, is finite and is of SVO or SV structure. Karamanis, Ha, and Mitkov (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, Heilman, and Eskenazi (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. Kurtasov (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, Agarwal, and Shah (2013) in their paper directly used a summarizer for selection of important sentences. Their system uses an extractive summarizer, MEAD to select important sentences. In some works, a set of context patterns have been extracted from a set of available stems for sentence selection. Bhatia, Kirti, and Saha (2013) used such 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, Baptista, Eskenazi, and Mamede (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.

3. Proposed approach

To test the content knowledge of the examinee it is evident to generate the MCQ from such sentence which carries information. This sentence is referred as informative
sentence in this context. The target is to select such informative sentences from an input text for MCQ stem generation. In order to identify informative sentences we propose a two-phase hybrid approach. The proposed technique which contains two distinct phases is presented in Fig. 1. The first phase consists of filtering out the under informative sentences by comparing the parse structure of the input sentence with existing MCQ. 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 pages and news articles. Most of the sentences in normal 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. More over it is often found that Wikipedia and news article sentences are long, complex and compound. For transforming a sentence into question, it is important that the sentence is in simple form. It is also found that a number of Wikipedia and news article sentences are having co-reference issues. So some pre-processing steps are required before parse structure matching, these include Reference Sentence Generation from MCQ and Simple Sentence Generation and Solving Co-reference.

3.1. Reference sentence generation

For the purpose of reference sentence generation we convert the collected stem of MCQ into assertive form. For this conversion we replace the ‘wh’ phrase or the blank space etc. of the MCQ by the first alternative of the option set. For example:

**MCQ:** Who defeated Australia in semi-final in Twenty20 World Cup 2012?

a) England  b) West Indies  c) South Africa  d) India

**Reference Sentence:** England defeated Australia in semi-final in Twenty20 World Cup 2012.

The first alternative may not be the correct answer of the MCQ but it usefully serves our purpose to generate grammatically correct sentence. Actually here our aim is to compile a grammatically correct sentence from MCQ for generating our reference set, not to find the correct answer of it.

3.2. Simple sentence generation and solving co-reference

To convert the complex and compound sentence into simple form we use the openly available ‘Stanford CoreNLP Suit’ (http://nlp.stanford.edu/software/corenlp.shtml). The tool provides a dependency structure among the different parts of the given test sentence. We analyze the dependency structure provided by the tool in order to convert the complex and compound sentence into simple sentences. As an example we consider the following sentence.

‘The 2014 ICC World Twenty20 was the fifth ICC World Twenty20 competition, an international Twenty20 cricket tournament that took place in Bangladesh from 16 March to 6 April 2014, which was won by Sri Lanka.’

This sentence is complex in nature and having co-reference problem. Co-reference has been defined as, referring of the same object (e.g., person) by two or more expressions in a text. For generating question from such sentences the referent must be identified. In the above sentence ‘that’ and ‘which’ are referring to ‘2014 ICC World Twenty20’. We use ‘Stanford Deterministic Co-reference Resolution System’, which is
basically a module of the ‘Stanford CoreNLP Suit’, for co-reference resolution. Finally we get the following simple sentences from the aforementioned example sentence.

Fig.1. A graphical representation of the proposed technique

'Simple1: The 2014 ICC World Twenty20 is an international Twenty20 cricket tournament.'

'Simple2: The 2014 ICC World Twenty20 was the fifth ICC World Twenty20 competition.'

'Simple3: The 2014 ICC World Twenty20 was won by Sri Lanka.'

'Simple 4: An international Twenty20 cricket tournament took place in Bangladesh from 16 March 6 April 2014.'
3.3. Parse tree matching

The parse tree structure of a sentence is its important attribute. It is observed that if more than one sentences are having similar parse structures, then they are generally carrying similar type of facts. For example, the aforementioned sentence ‘Simple3’ (in Section 3.2) is defining the fact that a team has won a series. The parse structure of the sentence (similar type of parse tree structure of a reference sentence is shown in Fig. 2) is similar with many sentences which are carrying the ‘team win series’ fact. The sentences like ‘The 2014 ICC World Twenty20 was won by Sri Lanka.’ ‘1998 ICC Knock Out tournament was won by South Africa.’, ‘2006 ICC Champions Trophy was won by Australia.’ are having similar parse trees and these can be retrieved if the parse structure shown in Fig. 2, 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 test corpus. We generate the parse trees of the reference set of sentences using the openly available Stanford Parser (http://nlp.stanford.edu/software/lex-parser.shtml). In the sports domain the questions (MCQs) are on the facts embedded in the sentences. Therefore the tense information of the sentence is not important for informative sentence selection 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 a similar type of fact, but the parse structures are different due to the difference in the 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 plan to use a coarse-grain tagset where a set of subcategories of a particular word class is mapped onto one category. From the original Penn Treebank Tagset (Santorini, 1990) used in the Stanford Parser we derive the new tagset and modify the sentences according to the tagset. For that first we run the POS tagger (available in the CoreNLP suit) and replace the tags or words according to the new tagset. Then we run the Parser on the modified sentence. For example, we map ‘has been (VBZ and VBN)’ and ‘was (VBD)’ onto ‘VB’; similarly ‘NN’, ‘NNS’, ‘NNP’ and ‘NNPS’ are mapped onto ‘NN’.

![Fig. 2. Example of a reference parse tree](image-url)
Once we get the parse trees of the reference sentences and Wikipedia sentences, we need to find the similarity among them. In order to find the similarity in these parse trees we have proposed the following algorithm named Parse Tree Matching (PTM) Algorithm.

![Parse Tree Matching Example](image)

**Fig. 3. Example of another reference parse tree**

The algorithm is basically trying to find whether the sentences are having similar structure. The sentences that we are targeting here normally contain some domain specific words which play major role in the sentence matching. These words are very frequent in this domain but rare in other domains and represent the domain. With the help of word frequency count, inverse domain frequency of the tokens and our knowledge about the domain we compile a list containing such words. The list is containing 29 words, like, ‘series’, ‘tournament’, ‘trophy’, ‘run’, ‘batsman’, ‘bowler’, ‘umpire’, ‘wicket’, ‘captain’, ‘win’, ‘defeat’ etc. The parse tree matching algorithm considers only the non-leaf nodes and these domain words during matching. All other words that occur as leaf of the tree are not playing any role in the matching process.

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

Once the reference structures are finalized, we use 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 Wikipedia or news article 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.
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. D_Word: list of domain specific words;
2. T1 and T2 are using the coarse-grain tagset;
3. Set Cnode1 as root of T1 and Cnode2 as root of T2;
4. if (label(Cnode1) = label(Cnode2) and number of children(Cnode1) = number of children(Cnode2)) then
5. n = number of children of Cnode1;
6. for (i= 1 to n) do
7. if both Cnode1_child_i and Cnode2_child_i are non-leaf then
8. if label(Cnode1_child_i) != label(Cnode2_child_i) then return 0 and exit;
9. end
10. end
11. if both Cnode1_child_i and Cnode2_child_i are leaf then
12. if (both Cnode1_child_i and Cnode2_child_i are both belong to D_word but different or only one belongs to D_word) then
13. return 0 and exit;
14. end
15. if Only one of Cnode1_child_i and Cnode2_child_i is leaf then
16. return 0 and exit;
17. end
18. end
19. Increase level by 1, update Cnode1 and Cnode2, and Go to Line 4;
20. return 1;
21. else
22. return 0 and exit;
23. end

Fig. 4. Example of a test parse tree
Fig. 2 and Fig. 3 are showing two example reference structures and Fig. 4 and Fig. 5 are showing parse trees of two test sentences. When the PTM Algorithm compares the first parse tree of test sentence, shown in Fig. 4 with the reference structures (with Fig. 3 in this case) a match is found. The other parse tree in the Fig. 5 does not have similarity with any of the reference trees.

The proposed PTM algorithm retrieves a set of sentences from Wikipedia and news article corpus having parse structure similarity with the reference set structures. Upon analyzing these sentences we have found that a portion of these sentences are not carrying sufficient information to generate MCQs. For example, the following sentences retrieved are not able to produce good MCQ.

‘The opening ceremony was held on 17 February 2011 at Bangabandhu National Stadium, Dhaka.’

‘The final was between India and Sri Lanka at Wankhede Stadium, Mumbai.’

‘The fairytale ended for the Kenyan team.’

‘Sachin Tendulkar became the first person in history to achieve this feat.’

These sentences are under informative. The first two sentences are not specifying the name of the tournament of which ‘opening ceremony’ and ‘final’ are talking about. In the third sentence what is referred as ‘fairytale’ for the Kenyan team is not clear. From the context we get that the fourth sentence is discussing on the 100 international century of Sachin Tendulkar but the phrase ‘this feat’ is not carrying that information.

Fig. 5. Example of another test parse tree

While analyzing the sentences retrieved by the PTM algorithm we found that the sentences containing some specific numbers of named entities (NEs) following a few patterns are informative enough to be the stems of MCQs. Examples are listed in the following:
A sentence having NEs like Series/Tournament name (SNE) along with Year (YNE) is highly informative so that it can be used to form MCQ stem.

‘For the first time the ICC Cricket World Cup 2011 was broadcast in high definition format.’

A sentence having NEs like Person Name (Player Name: PNE), SNE (Series/Tournament name) is selected as an informative sentence if it has any other NE tag, like Team name (TNE), as in the following sentence.

‘Sachin Tendulkar of India scored the most number of runs in the Cricket World Cup.’

A sentence having at least three different NEs like OSNE (Official Song), SNE, is selected for MCQ generation if it has YNE.

‘The official song for the 2007 World Cup was “The Game of Love and Unity.”’

A sentence having three different NEs like MNE (Stumpy), LNE (Colombo) and TNE (Sri Lanka) is selected as informative sentence.

‘Stumpy was unveiled at a function in Colombo, Sri Lanka.’

A sentence having minimum four different NEs is selected as an informative sentence.

‘Stumpy’s name was revealed on 2 August 2010 after an online competition conducted by the ICC in the last week of July.’

As a result of this observation, we plan to identify these NEs from the potential sentences, identified after using the PTM Algorithm. In the cricket domain the primary NEs are player name, team name, series/tournament name, ground name, location name, tournament mascot and official song, organization name, year and number etc. In order to identify the NEs we considered developing a NER system. For the purpose we opted machine learning based technique and chose Conditional Random Field (CRF) as the classification algorithm. Machine learning technique requires annotated training corpus. As openly available training data in cricket domain is not available, we created our own training data by annotating the sentences formed from existing MCQs based on the above mention NE categories as follows.

1. Year/Date of a series/tournament/match: #YNE
2. Tournament/Series name: #SNE (BSNE and ISNE)
3. Team name: #TNE (BTNE and ITNE)
4. Person name: #PNE (BPNE and IPNE)
5. Ground name: #GNE (BGNE and IGNE)
6. Location/Place name: #LNE (BLNE and ILNE)
7. Mascot name: #MNE
8. Official song: #OSNE (BOSNE and IOSNE)
9. Organization name: #OGNE (BOGNE and IOGNE)
10. Number: #NNE
11. Other: #O
3.4. Named entity recognition based post-processing

We were unable to find any NER system working in this cricket domain to identify the name classes of our interest. Therefore we plan to develop a NER system. The NER system is based on the conditional random fields (CRF) classifier (Lafferty, McCallum, & Pereira, 2001). In order to train the system we manually annotated the sentences which are generated from existing MCQs using the aforementioned NE tags. For the purpose of CRF’s training data generation we used about 380 sentences generated from existing cricket related multiple choice questions. These sentences are annotated using the popular BIO format where ‘B’ indicates the first token of a NE, ‘I’ indicates the rest of the tokens and ‘O’ refers to the words that are not a name.

To learn a classifier we need to identify a set of features. For this reason we use a simple and easily derivable feature set containing the words (current and surrounding words of a target word), affix (variable length and list of suffix and prefixes of the current word), capitalization information (initial character is capital, all characters are capital), numeric information (the token is a number or word denoting a numerical value), parts-of-speech information and parse information. These features are adopted according to Borthwick (1999). We have been experimenting with various combinations of these features (like, word window and affixes of various lengths) to choose the best feature set. Using the best feature set the system achieved an F-Measure of 90.32 with 95.25% Precision and 85.88% Recall.

Next we studied the NE annotated sentences in order to define a set of NE based rules for refinement of sentence selection. These rules are:

1. A sentence having only the following two types of NEs (SNE & YNE) together is selected.
2. A sentence having at least three different types of NEs like TNE, PNE and any other NE (except number NE: NNE) is selected.
3. A sentence having at least three different types of NEs like PNE, SNE is selected if it has any other NE (except number NE: NNE).
4. A sentence having at least three different types of NEs like GNE/LNE, SNE is selected if it has any other NE (except number NE: NNE).
5. A sentence having at least three different types of NEs like GNE/LNE, PNE is selected if it has any other NE (except number NE: NNE).
6. A sentence having at least three different types of NEs like MNE/OSNE, and other two NEs (except PNE and NNE).
7. A sentence having minimum four different types of NEs is selected.

All the sentences, selected by the parse-tree similarity based module, are now used as input to the NE rule based post-processing module. The post-processing module first recognizes the named entities from an input sentence and then verifies whether these rules are satisfied. The sentences which satisfy any of the aforementioned rules are considered as the final set of informative sentences for MCQ stems generation.

4. Result and discussion

The developed system has been tested using cricket related Wikipedia pages and news articles. A number of relevant documents were given as input to the system for identification of informative sentences from these. The system identified sentences that
were then manually assessed to measure the accuracy of the system. For the assessment we used the feedback of five human evaluators, all associated with a technical institute Birla Institute of Technology Mesra, India. Two human evaluators are faculty members, two of them are students and the remaining one is a staff member of the institute. The primary metric of the evaluation is the quality of the retrieved sentences – whether a MCQ can be generated from the selected sentence. Additionally we also counted the percentage of sentences extracted from the total amount of informative sentences. As the system is able to access any web-page or text document as input, this metric is not much essential for the evaluation. However to check the selection efficiency of the system we measured the amount of sentence extraction.

There is no gold-standard data in this task. Therefore we have chosen our test data by ourselves. But for the sake of fair assessment we have used openly available web text as input. To compute the accuracy of the system we consider six cricket related Wikipedia pages namely, ICC Cricket World Cup 2003, 2007 and 2011, 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, ‘Lankans Lord Over India’, ‘When Yuvi cut a sorry figure’, ‘Kohli the lone man standing for India’ and ‘Lanka talismans Jayawardene, Sangakkara bow out of T20s on a high’. Only the text portions of these pages are taken as input that contains a total of about 795 simple sentences. The parse tree matching algorithm selects 302 sentences as the candidate set. Then we processed these sentences through the NER rule based post-processing module. Finally the system selects 95 sentences as potentially relevant for the MCQ stem generation. These sentences are examined by five human evaluators. They consider 90, 89, 87, 90 and 89 sentences respectively as a correct identification. Therefore the accuracy of the system is 93.684%. Table 1 summarizes the evaluation results of the system.

Table 1
Performance of the developed system

<table>
<thead>
<tr>
<th>No. of Simple Sentence</th>
<th>Sentence After PTMA</th>
<th>Sentence After NER Rules</th>
<th>Evaluators Judgment</th>
<th>% Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>~795</td>
<td>~302</td>
<td>95</td>
<td>Evaluator1: 90</td>
<td>93.684%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Evaluator2: 89</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Evaluator3: 87</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Evaluator4: 90</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Evaluator5: 89</td>
<td></td>
</tr>
</tbody>
</table>

5. Conclusion

This paper presented a novel technique for the selection of informative sentences that are potential to act as the basis of a MCQ. The technique is based on a parse tree matching algorithm, surrounded by a set of pre-processing and post-processing steps. The evaluation results indicate that the proposed technique is effective.

The system requires a set of existing MCQs in the same domain from which the reference set will be created. So availability of sufficient MCQs is a prerequisite, but in many domains adequate MCQs are not easily available. This can be a restraint to use the system in such domains. To overcome this, domain portability of parse structures may be studied. Also, the system requires an appropriate post-processing phase to refine the
candidate set of sentences. In this task we have manually prepared the rules. Finding some automatic approach for defining the rules can be another direction of future work. We also consider the Keyword selection and Distractors generation as a future scope to finally develop an automated MCQ generation system to boost up next generation educational technology that can effectively meet the increased needs for knowledgeable graduates and skilled workforce in Modern Age and Smart Cities.

References


