Automatic Generation of Multiple Choice Questions Using Wikipedia

Arjun Singh Bhatia, Manas Kirti, and Sujan Kumar Saha

Department of Computer Science and Engineering
Birla Institute of Technology, Mesra, Ranchi, India - 835215
{arjunbhatia2304,manasmikku21,sujan.kr.saha}@gmail.com

Abstract. In this paper we present a system for automatic generation of multiple choice test items using Wikipedia. Here we propose a methodology for potential sentence selection with the help of existing test items in the web. The sentences are selected using a set of pattern extracted from the existing questions. We also propose a novel technique for generating named entity distractors. For generating quality named entity distractors we extract certain additional attribute values on the key from the web and search the Wikipedia for the entities having similar attribute values. We run our experiments in sports domain. The generated questions and distractors are evaluated by a set of human evaluators using a set of parameters. The evaluation results demonstrate that the system is reasonably accurate.

1 Introduction

Multiple choice question (MCQ) is a very popular form of assessment in which respondents are asked to select the best possible answer out of a set of choices. A MCQ is composed of three elements: stem, target word and distractors. The *stem* (also known as *item*) is the sentence from which the question is formed, *target word* (also named as *key*) is the correct answer of the question and *distractors* are the set of wrong answers.

Development of automatic MCQ generator has become a popular research problem in the last few years. In the literature we observe, generally automatic MCQ systems have been followed three major steps: selection of sentences (or stem), selection of target word and generation of distractors. Mitkov and Ha (2003) and Mitkov et al. (2006) proposed a NLP-based methodology for generating MCQ semi-automatically from an electronic text, a textbook on linguistics. They used several NLP techniques, natural language corpora and WordNet. They have used various post editing phases for betterment of the system. Aldabe et al. (2006) and Aldabe and Maritxalar (2010) developed another system to generate MCQ in Basque language in the science domain. They divided the task into six phases: selection of text (based on level of the learners and the length of the texts), marking of blanks (done 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

P. Maji et al. (Eds.): PReMI 2013, LNCS 8251, pp. 733–738, 2013.

MCQ system. They used the structure of an ontology - the concepts, instances and the relationship or properties that relates the concepts or instances - to generate the MCQs. First they formed sentences from the ontology structure and then they found distractors from the ontology. Agarwal and Mannem (2011) presented a system for generating gap-fill questions, a problem similar to MCQ, from a biology text book. For sentence selection they 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. Similarly for key selection and distractor generation they used sets of relevant features.

In this paper we present a novel technique for MCQ generation. Our system does not require any ontology or WordNet; rather it is using the web, specially the Wikipedia, as the source of information. Therefore it can be easily transferred to any domain or language. Currently we run the experiments in sports domain. We have divided the task into several subtasks which are summarized below and discussed in the remaining sections of the paper.

- Find some available MCQs on the domain of interest from the web. Form sentences from these collected MCQs. This set is called as 'reference set' in our experiments.
- Search for potential sentences from the Wikipedia. For that we extract patterns from the reference set sentences and find sentences that are containing these patterns.
- Select *key* from the potential sentences. Most of the keys are *named entity* (NE) in this domain.
- Form question from the sentences.
- Generate *distractors* using Wikipedia. A technique is proposed for generating interesting and relevant NE distractors.

2 Sentence Selection and Question Generation

We have already mentioned that we have generated the MCQs from Wikipedia text. A typical Wikipedia page contains hundreds of sentences and thousands of words. For example, the Wikipedia page on *Sachin Tendulkar* contains more than 650 sentences and about 11000 words. MCQ can not be formed from all the sentences. Therefore first of all we have to select the potential MCQ sentences. Again a selected potential sentence contains several words; we need to select the word (or word n-gram) which can become the key. Next we form the question. These phases of the task are described below.

Reference Set Generation: For selecting the potential sentences we have taken the help of existing MCQs on this domain. MCQ based assessment is quite popular and multiple choice question papers are often generated (manually) for assessment purpose. A number of such MCQ papers are available in the web. We collect such available MCQs from different sources.

Next we form sentences from the MCQs. MCQs contain the stem and a few options. For sentence generation we replace the 'Wh phrases' (who, where, which,

when, which of the following etc.) in the stem by the first option. The first option might not be the correct answer of the question, but our intention is to sentences that are syntactically correct. We refer this set of generated sentences as 'reference set'.

Sentence Extraction Using Patterns: We extract a set of context patterns from the reference set sentences for finding potential sentences. For extracting the patterns we first run stemmer (for getting the root words for the inflected words) and the Stanford NER system[1]. The identified NEs are replaced by variables (for example, *PER* for person NEs, *LOC* for location entities). Then find the most frequent n-grams (where n is taken upto seven and minimum length is three) that are not occurring in general domain pages. These are the patterns. Some example patterns are, *PER be the captain of LOC*, *PER be the middle name of PER*, *PER word* the man of the tournament*, (word* can be replaced by zero or more words) *maximum number of one day ducks*, *fastest one day (or, word*) century*, *most successful bowler of the tournament*, *opening ceremony was held in* etc.

Key Identification: Key identification is the next phase where we select the word (or n-gram) that has the potential to become the key. We have identified the potential sentences using the patterns. A pattern is likely to extract sentences containing a particular type of entities. For example, "fastest one day century" pattern should retrieve sentences containing the name of the cricketer having the fastest century. Therefore the key for this pattern should be the person name in the retrieved sentence. Similarly, "the man of the tournament" pattern will extract sentences having the name of the player who got the man of the tournament in a particular tournament. The key for the pattern should be the person name. The pattern "opening ceremony was held in" is expected to retrieve the location (city name or ground name) where the opening ceremony of a tournament was held; therefore the corresponding key will be the location entity. For each of the extracted patters we identify the entity type which is having the potential to become the key. The sentences are tagged using the NER system and the corresponding entity is selected as the key.

Question Formation: For question formation also we have taken the help of the patterns. The patterns give us the information regarding the type of the key. Depending on the key category we replace the key by a proper *wh-word*. The parse structure of the sentence is also consulted to bring the wh-word in the beginning of the question. If the category is person then the wh-word is *who*; similarly, for location *where*, for date *when*, for number *how many* etc.

3 Distractor Generation

Distractor is a concept semantically close to the keyword. Quality of the distractors plays an important role in MCQ. In this article we have focussed in the

[1] `http://nlp.stanford.edu/software/CRF-NER.shtml`

sports domain where most of the distractors are named entity. Here selection of distractors is more challenging. For example, consider the question "Who was the Indian wicketkeeper in the final match of Cricket World Cup 2011?". And if the generated options are, Ricky Ponting, Adam Gilchrist, Chris Gayle and Mahendra Singh Dhoni then quality of the MCQ is somehow degraded. Because among the options Mahendra Singh Dhoni is only from India and is becoming the obvious choice. For selecting a set of good and interesting distractors we use certain additional information extracted from the web regarding the particular stem.

For distractor generation WordNet, domain ontologies or related knowledge base for finding similar or related words (synonyms, hypernyms, hyponyms, antonyms etc.) are often used in the literature. Mitkov et al. (2009) experimented on several similarity methods for distractor generation: collocation pattern, four different methods of WordNet based semantic similarity, distributional similarity and phonetic similarity. Brown et al. (2005) also used WordNet for finding distractors. Correia et al. (2010) studied random, graphemic and phonetic approaches for automatic distractor generation. Aldabe and Maritxalar (2010) used Latent Semantic Analysis (LSA), dictionary and WordNet to generate distractors. Papasalouros et al. (2008) used the ontology to generate distractors. They have chosen distracters as the related instances or classes having similar properties with the key. Agarwal and Mannem (2011) generated distractors using three features namely, contextual similarity, sentence similarity and term frequency, along with the parts-of-speech information.

3.1 Our Approach for Distractor Generation

In this cricket domain the major category of key (or distractors) are: person name (cricketer, bowler, batsman, wicketkeeper, captain, man-of-the-match etc.), name of team (country name, club or franchise name etc.), location name (name of cricket ground, name of city etc.), date (year, match date etc.), numerical (number of matches, batting average, number of wicket, highest score etc.) tournament name (cup, trophy, championship etc.) etc. We develop a framework for handling all these categories. Basically all the categories are handled using similar approach, only the *attribute set* representing the category differs. In the following we have discussed in detail the strategy for generating person name distractors.

For generating the distractors for a person name we find a set of additional attributes for the key from the Wikipedia. The attribute set for the cricketer contains: date of birth (or born), role (bowler, batsman etc.), batting style, bowling style, national side, other team, last match played (ODI/Test/T20), is captain (binary value, current or former captain). These information are extracted from the tabular or structured data (information box on the title) present in the right hand side top of an Wikipedia page or the first sentence of the content.

Next we search the Wikipedia for a list of related candidates from the same category. For example, for cricketer key we feed a search query "list of <national side> cricketers"; if the *is-captain* attribute value is one then the query is "List

Table 1. Performance of the MCQ generator

Evaluator	Domain	Key	Question formation	Info	Distractors	Close
Evaluator 1	100	76	70	20	81	80
Evaluator 2	100	84	72	27	85	84
Evaluator 3	100	81	74	18	96	85
Evaluator 4	100	77	68	20	89	88
Evaluator 5	100	79	73	21	89	92
Average	100	79.4	72.6	21.2	88	85.8

of <national side> national cricket captains". Wikipedia search engine retrieves page containing the lists given in tabular form. The table consists of a set of fields. For example, the list of cricketer contains the fields like, span, number of matches played, batting, bowling, and fielding summary. From the table we extract a set of cricketers having attribute values similar as the key. This set acts as *candidate distractors*.

Then we check whether there is any special feature in the stem; for example, man-of-the-match of a particular match, highest wicket taker of a tournament. If any such is there then we extract the list of players played the match or tournament, most run getter or most wicket taker statistics of the tournament etc. These entities get higher preference to become the distractors. Otherwise randomly pick the required distractors from the candidate set.

4 System Evaluation and Discussion

In order to evaluate the quality of the MCQs generated by the system, we define a set of parameters. Item analysis has been used for MCQ evaluation in a few articles which consists of three parameters (Gronlund, 1982): difficulty of the item, discriminating power and usefulness of each distractor. In this paper we have extracted the items from Wikipedia and majority of the distractors are NE, therefore here we have used a extended set parameters. These are, whether the question is relevant to the domain of interest (referred in table as *domain*), the key is chosen properly, question is formed properly, question is over-informative or under-informative (*info*), whether the distractors are related with the key (*distractor*), at least one distractor is close to key (*close*). *Question is over-informative or under-informative* is a special parameter required when the sentences are extracted from the web using patterns. From the Wikipedia page on ICC World Cup 2011 our system form a MCQ as "who was the man of the tournament?" But for which tournament? This information was not there in that particular sentence which is not required to mention repeatedly as the whole article is on World Cup 2011. But the question is under-informative and cannot be answered. Closeness it is another special parameter required in this task. For example, in question related to man-of-the-match at least one of the distractors should have played that match and also performed well.

The quality of the generated MCQs is evaluated by five human evaluators using the aforementioned parameters. We have generated 100 MCQs by the

system and given to the human evaluators. In Table 1 we present the accuracy of the system. From the table we observe that the system is generating good quality distractors, the average accuracy is 88%. The system also selected the key with high accuracy (about 80%). But in question generation the accuracy is 72.6%. Around 28% questions are not formed properly. We have analyzed those questions - the errors occurred mainly because of lengthy, compound or complex sentences. Also 21.2% questions are over-informative or under-informative.

5 Conclusion

In this paper we have presented a system for automatic MCQ generation in sports domain. The system is using web information, specially Wikipedia, for generating questions and distractors. We have attempted to find the potential MCQ sentences with the help of existing questions of this domain. Our system generates named entity distractors of good quality. Finally the system is tested by a set of human evaluators using a proposed set of parameters. Evaluation results demonstrates that the system produces the MCQs and distractors with reasonable accuracy.

References

Agarwal, M., Mannem, P.: Automatic Gap-fill Question Generation from Text Books. In: Proceedings of the Sixth Workshop on Innovative Use of NLP for Building Educational Applications, pp. 56–64 (2011)

Aldabe, I., de Lacalle, M.L., Maritxalar, M., Martinez, E., Uria, L.: ArikIturri: An Automatic Question Generator Based on Corpora and NLP Techniques. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) ITS 2006. LNCS, vol. 4053, pp. 584–594. Springer, Heidelberg (2006)

Aldabe, I., Maritxalar, M.: Automatic Distractor Generation for Domain Specific Texts. In: Loftsson, H., Rögnvaldsson, E., Helgadóttir, S. (eds.) IceTAL 2010. LNCS (LNAI), vol. 6233, pp. 27–38. Springer, Heidelberg (2010)

Brown, J.C., Frishkoff, G.A., Eskenazi, M.: Automatic question generation for vocabulary assessment. In: Proceedings of HLT/EMNLP, pp. 819–826 (2005)

Correia, R., Baptista, J., Mamede, N., Trancoso, I., Eskenazi, M.: Automatic Generation of Cloze Question Distractors. In: Second Language Studies: Acquisition, Learning, Education and Technology (2010)

Gronlund, N.: Constructing achievement tests. Prentice-Hall Inc., New York (1982)

Mitkov, R., Ha, L.A.: Computer-aided generation of multiple-choice tests. In: Proceedings of the HLT/NAACL 2003 Workshop on Building educational applications using Natural Language Processing, pp. 17–22 (2003)

Mitkov, R., An, L.A., Karamanis, N.: A computer-aided environment for generating multiple-choice test items. Journal of Natural Language Engineering 12(2), 177–194 (2006)

Mitkov, R., Ha, L.A., Varga, A., Rello, L.: Semantic similarity of distractors in multiple-choice tests: extrinsic evaluation. In: Proceedings of the EACL 2009 Workshop on GEometical Models of Natural Language Semantics, pp. 49–56 (2009)

Papasalouros, A., Kanaris, K., KotisAutomatic, K.: Generation of multiple-choice questions from domain ontologies. IADIS e-Learning (2008)