Automatic English Question Generation System Based on Template Driven Scheme

Article · November 2014

CITATIONS
23

READS
1,325

3 authors, including:

Mohammed Elmogy
Mansoura University
253 PUBLICATIONS **2,502** CITATIONS
SEE PROFILE

Shawkat Guirguis
Alexandria University
15 PUBLICATIONS **83** CITATIONS
SEE PROFILE

Automatic English Question Generation System Based on Template Driven Scheme

Hafedh Hussein[1], Mohammed Elmogy[2] and Shawkat Guirguis[3]

[1] Information Technology Department, Institute of Graduate Studies and Research, Alexandria University, Alexandria, Egypt

[2] Information Technology Department, Faculty of Computers and Information, Mansoura University, Mansoura, P.O. 35516, Egypt

[3] Information Technology Department, Institute of Graduate Studies and Research, Alexandria University, Alexandria, Egypt

Abstract

Today, automatic generation of questions is a considerable problem by many researchers. Automatic question generation from text has two major activity domains: dialogue/interactive question answering systems and educational assessment. In this paper, a proposed system is designed, implemented and tested to automate question generation process. The proposed system generates questions by selecting one sentence at a time, extracts sections of the source sentence, then applies transformation rules or patterns for constructing a question. It uses pure syntactic pattern-matching approach to generate content-related questions in order to improve the independent study of any textual material. The proposed system is powered by OpenNLP open source statistical parser to generate the questions using pattern matching strategy. The system is able to learn dynamically, ready to use in ease manner and accepts large-scale number of rules.

Keywords: *Natural Language Processing, Parsing, Question Generation, Questionnaire System.*

1. Introduction

In addition to real life discussions, questions are the significant fundamental immeasurable learning interactions from one-to-one education sessions to extensive assessments. Human has usually natural curious because of self-satisfaction of his never ending requirements of knowledge. Question is a helpful tool for performance evaluation of students. Our day-to-day lives involve asking questions in conversations and dialogues to render a meaningful co-operative society. Because humans are almost subject to ambiguous, busy, or inconsistent mind in certain situations, question generation problem was raised to be solved and researched by many researchers over the last decade years. Therefore, the potential benefits from an automated Question Generation (QG) system could assist humans in meeting their useful inquiry needs such as education, knowledge based, daily activities, and much more application(s) [1].

QG has turned into an essential element of learning environments, help systems, information seeking systems…etc. Considerable interest from the Natural Language Processing (NLP), Natural Language Generation (NLG), Intelligent Tutoring System, and Information Retrieval (IR) communities have currently identified the Text-to-Question generation task as a promising candidate for the shared task [2, 3]. In the Text-to-Question generation phase, a QG system starts by a given text and its goal would be to generate a set of questions for which the text contains answers [4].

To generate a question from a given sentence, the system must processes the sentence to get elementary items of the text. In such systems, sentence processing is a subject to NLP fields of computer science [2]. There are many researches focus on building up algorithms, techniques, and methodologies for processing natural language sentences. OpenNLP and IXA [27] pipelines are the most popular open source NLP API and software development kit that are ready to use for parsing and automatic understanding of English sentences that are powered by researchers results over years.

Question in natural language is a unit of language that can be classified into different categories as follow [10, 11]:

- Factoid Questions: They are questions that are used to ask for fact-based answers. They use one of these question tools: What, Where, When, Who, or How.
- List Questions: They are questions that are used to ask for a set of answer terms. They use "which" as a question tool.

- Other Questions: They are questions that are required interesting information about the target that is not covered by the preceding questions in the series of factoid and list questions.

Another classification can be made depends on the simplicity or the complexity of the questions. For simple question, it is defined as a question that can be answered by a word or simple sentence. Whereas, complex question has a more complicated answer rather than simple question. Question answers can be classified and decomposed into three distinct types; numeric, direction, or entity answer [10, 11].

In order to generate a question from a sentence, it must be detected and extracted into its language parts; part of speech (POS). POS consists of two major classes: open class and closed class. Different languages may have different classes, but for every language there is a class where every word should belong to [8, 9]. Using these classes in a system, the system will be able to recognize the words class and tag them accordingly, which help improving the quality of systems that deal with human software interactions [32].

On the other hand, parsing process is a very complicated process starting from lexical tokenize to constructing parsing tree passing through sentence segmentation, POS detection and tagging each part of sentence. Parsing process can be top down or bottom up process. In top down parsing, parsers seeks a parse tree by trying to build from the root node S down to the leaves. The algorithm will look for all the trees that have the root as their parent, using the information about the possible trees under the root, and the parser will build trees in parallel. 'S' represent the start of the sentence. The trees that have 'S' as the root are basically the grammar rules that are provided for the parser. Then the parser will expand the new parallel trees using the grammar into a new set of parallel trees. While, in bottom up parser starts with the words of the input, and tries to build trees from the word up, again by applying rules from the grammar one at a time. The parser is successful if the parser succeeds in building a tree rooted in the start symbol S that covers all of the input. The algorithm will look for all grammar rules that can be a root for each level of building the tree up, the parser will build trees in parallel one at a time. Each parser can be based on a different methodology to support and extract information required for expanding tree levels. Such methods are Statistical, Collins, and Syntactical.

In statistical parsers, the parser parse the grammar rules that are associated with probability information, for each non-terminal to expand to a terminal or non-terminals, this grammar rules are then given as input for the parser along with a textual entity to be parsed. The most commonly used probabilistic grammar is the Probabilistic Context-Free Grammar (PCFG). The statistical parsers get trained in a labelled data, which will help generate the probabilities for each non-terminal to expand to one of the available grammar rules provided. By considering A as non-terminal, β as list of possibilities that A can expand to, and p as the total probability for each part of the list β, [12].

$$A \rightarrow \beta[p] \tag{1}$$

The probability can be represented as

$$P(A \rightarrow \beta) \tag{2}$$

If we take all possibilities in the list β, which A can expand to, the sum of these possibilities must be 1:

$$\sum_{\beta} P(A \rightarrow \beta) = 1 \tag{3}$$

Table 1 helps to understand a sample of probabilistic grammars and their rules, which is discussed in [12].

In order to evaluate different parsers, there are different measuring methods which will discuss the most commonly used measures in the area of information retrieval. From domain to a domain these measures may vary in their importance, and there consideration. In the following points, we will discuss the definitions of these measures:

- Precision

Precision measures the percentages of system-provided a chunk that were correct [28, 31]; which is defined as:

$$\text{Precision} = \frac{Number\ of\ \ correct\ chunks\ given\ by\ system}{Total\ number\ of\ \ chunks\ given\ by\ system} \tag{4}$$

The definition of correct chunks depend on the way we want to measure the system, do we consider partially matched chunks as correct, or do we consider exact match are correct, that contributes to the level of accuracy we are looking for from the system.

- Recall

Recall measures the percentage of chunks actually present in the input that were correctly identified by the system that is defined as [28, 31]:

$$\text{Recall} = \frac{Number\ of\ \ correct\ chnks\ given\ by\ system}{Total\ number\ of\ \ actual\ chunks\ in\ the\ text} \tag{5}$$

Again the definition of correct chunks depend on the way we want to measure the system, do we consider partially matched chunks as correct, or do we consider exact match are correct, that contributes to the level of accuracy we are looking for from the system.

- F-Measure

In [29, 30], F-measure provides a way to combine the previous two measures into a single metric that is defined as:

$$F_\beta = \frac{(\beta^2+1)PR}{\beta^2 P + R} \quad (6)$$

The parameter β can be used based on the application needs [29, 30], so if β > 1 this will favor the recall while if β < 1 the favor is for the precision. However, when β = 1 that means Recall and Precision are equally balanced. Therefore, this can be called F1 that is defined as:

$$F_1 = \frac{2PR}{P+R} \quad (7)$$

This paper is organized as follows. In Section 2, the current scientific research in QA and QG is introduced. Section 3 describes the proposed QA system. In Section 4, the proposed framework is evaluated and explained with illustrative example. Finally, conclusion and future work are drawn in Section 5.

Table 1: Sample of probabilistic grammars and their rules [12].

Grammar		Lexicon
S → NP VB	0.80	Det → that [.10] | a [.30] | the [.60]
S → Aux NP VB	0.15	Noun → book [.10] | flight [.30]
S → VP	0.05	| meal [.15] | money [.05]
S → Pronoun	0.35	| flights [.30] | dinner [.10]
NP → Proper-Noun	0.30	Verb → book [.30] | include [.30]
NP → Det Nominal	0.20	| prefer;[.40]
NP → Nominal	0.15	Pronoun → I [.40] | she [.05]
Nominal → Noun	0.75	| me[.15] | you[.40]
Nominal→ Nominal Noun	0.20	Proper-Noun → Houston [.60]
Nominal→ Nominal PP	0.05	| NWA [.40]
VP → Verb	0.35	Aux → does [.60] | can [.40]
VP → Verb NP	0.20	
VP → Verb NP PP	0.10	
VP → Verb PP	0.15	Preposition → from [.30] | to [.30]
VP → Verb NP NP	0.05	| on [.20] | near [.15]
VP → VP PP	0.15	| through [.05]
PP → Prepositional NP	1.0	

2. Related Work

Generally, automatic question generation from text has two main application domains: (i) dialogue and interactive Question Answering systems and (ii) educational assessment. The question thus generated targets one part of the input sentence, e.g. the subject or an adverbial adjunct. This research domain has been the focus of increasing interest.

In the first application context, question generation has been used for automatically producing dialogues from expository texts [13, 14]. The generated dialogues may be presented in the form of written text or to virtual agents with speech synthesis. Concrete uses of such dialogues include presenting medical information, such as patient information leaflets and pharmaceutical notices, or educational applications. In a related domain, the quality of the interactions between the system and a user can be improved if the QA system is able to predict some of the questions that the user may wish to ask.

Harabagiu et al. [15] described a question generation method for interactive QA which first identifies entities relevant for a specific topic and then applies patterns to obtain questions. Automatically generated questions are then presented to the user by selecting those which are most similar to the question asked initially.

The second application context of QG systems, question generation for educational assessment, has been investigated for many years [21]. Indeed, test writing is a very time-consuming task and the availability of a question generation system reduces the workload for instructors. Educational assessment applications rely on question generation methods for producing open or multiple-choice questions for text comprehension [16, 21]; Gates [17], or summative assessment [18]. Depending on the system, the generated questions may correspond to manually defined templates, or be less constrained [17, 19, 20].

Automatic question generation systems for the English language usually proceed according to the following steps: (i) perform a morph syntactic, syntactic, semantic and/or discourse analysis of the source sentence, (ii) identify the target phrase for the question in the source sentence, (iii) replace the target phrase with an adequate question word, (iv) make the subject agree with the verb and invert their positions, (v) post-process the question to generate a grammatical and well-formed question [16, 20, 21]. Given this procedure, question generation requires that the input sentence be at least morph-syntactically analyzed. It is often also useful to have additional information about semantics, such as named entities (person, organization,

country, and town) or the distinction between animates and in animates.

In contrast to the above systems, other approaches have an intermediate step of transforming input into some sort of semantic representation. Examples of this intermediate step can be found in Yao and Zhang [22] that used Minimal Recursive Semantics. Olney et al. [23] used concept maps. These approaches can potentially ask deeper questions due to their focus on semantics. A novel question generator by Curto et al. [24] leverages lexicon syntactic patterns gleaned from the Web with seed question-answer pairs.

Lindberg et al. [25] used semantic role labeling to identify patterns in the source text from which questions can be generated. This work most closely parallels our own with a few exceptions: our system only asks questions that can be answered from the source text, our approach is domain independent, and the patterns also identify the answer to the question.

There already exists a large body of work in automatic QG for educational purposes dating back to the Autoquest system [21], which used an entirely syntactic approach to generate Wh-Questions from individual sentences. In addition to Autoquest, several others have created systems for Wh-question generation using approaches including transformation rules [18], template based generation [12, 24], and over gene rate and rank [21]. The work in this area has largely focused on the surface form of the questions, with an emphasis on grammaticality.

Alternatively, generation of gap-fill style questions (a.k.a. cloze questions) avoids these issues of grammaticality by blanking out words or spans in a known good sentence. There is a large body of existing work that has focused on generation of this type of question, most of which has focused on vocabulary and language learning. The recent work of Agarwal and Mannem [26] is closer to our purposes; they generated fill-in-the-blank questions and distractor answers for reading comprehension tests using heuristic scoring measures and a small evaluation set.

3. The Proposed Framework:

A framework is proposed to automate the process of question generation. It facilitates the evaluation process of students in educational environments based on generating questions, ranking them and store in a back end bank question from the well formatted given lessons as rich text documents. The proposed system is a rule based model that is trained before generation process start by storing a huge amount of template rules in the data store. Training process of the proposed is semi-automatic process powered by OpenNLP framework for many benefits, listed next, which is open source statistical base parser.

The proposed system includes three major modules. The first module is training module which loads well formatted rich text documents and train the system using stated sentences. The second module is question generator engine which generates the questions, rank, review and store in the database. The third module is the user interface and exam question based on the stored bank question using a smart judgment strategy to generate exam into different hardness levels.

3.1 The Training Phase

The training phase is used to train different sentences related to certain field or application domain. Training process is easy, which can be considered as all in one training process. Training includes remarking the sentences by certain keywords, new verbs besides their major rule training about new sentence template. Remarks and keywords are special phrases that have a great role in question generation process. Such phrases includes propositions, locations, organizations, peoples and famous characters, money and currency, time and dates and custom remark or a user defined term. These remarks have a perfect major role in question generation as these remarks help the proposed system to determine the type of questions available in the given sentence. Verbs, to be and related auxiliary verbs are trained and used to determine the tens of the sentence and the generated tens of the sentence rather than pronoun conversion from the sentence to the question.

The proposed system is latency training mode but with automated processes and activities like pre-processing and eliminating unwanted phrases, sentences and punctuation marks system become more acceptable style. In order to train a system about certain sentence, you should write it either into the system text editor directly or via save into a rich text formatted document then load it into the editor.

After either activity, the system will try to generate the question(s) possible from the given sentence. From the given sentence, it would start to detect sentence, create sentence model. Next, using the sentence model, tokenize the sentence into phrases that can be tagged by using POS feature available in OpenNLP. Based on the associated tags of POS, the system connects the supplied database and try to find if having a similar stored template rule before if not the system start of saving a new un-matched statement by will prompt the user to select the best choice of question tool weather what, where, who, when or how much. Next, the user will prompt the question tags order

rather than the phrase itself in order to increase ability of the system to generate questions based on the rules not based on the sentence phrases. Another fields can be supplied during train activity is to inform the system about new verb; if exists, and store it in the database. Also, if there is a not listed remarks either of a specific type or custom user defined, it will be processed same as verb. Figure 1 shows the flowchart diagram that describes and demonstrates the major steps of training process.

3.2 Question Generation & Review and Bank Question Management Phase

When the proposed system have been trained well on a certain domain paragraphs and text documents, the system becomes ready to be used in rapid question generation process in the certain domain of education purpose.

At that moment, the user would be going in a wizard steps starting from loading a rich text formatted document into the system, next process the text exactly like train process with two main different activities:

1- Doesn't add new rule; during processes of the given text only -search the rule template store about rules that can be matched and generate a question based on it. If there is a new statement with new rule no question will be generated.

2- Question generation process includes automatic generation of two types of question:

   a- WH-question; using one of WH question tool as Who, When, Where, What or How much

   b- Complete question; based on pre-defined marks such as location, persons, date, time or organizations, a sentence would be converted into complete question up to with two complete phrases.

When the system finish the process of question generation and all is done, the system reports the user with the list of all possible questions including well-structured questions and semi well-structured questions that includes some little error. The user can edit questions, change their level of hardness to classify question during exam generation phase. Also, user can change the question phrases to be clearer, re-arrange the words even adding new hint phrases or other options. In the last step, the user would save their all level of changes and modification and store the questions into the store related to a pre-defined chapter of a course. The course and its chapters are defined in earlier stages of the workflow by using a course management screens. Next figure describes the major activities of workflow during question generation process.

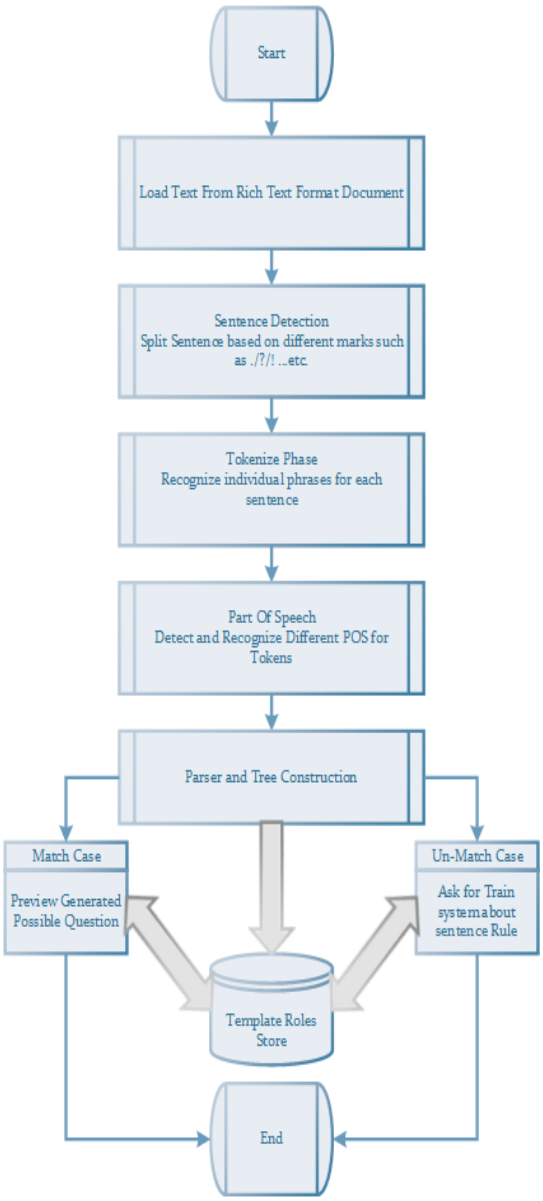

Fig. 1: The training Phase in the proposed framework.

3.3 Exam Generation Phase

The proposed system at that moment starts using the saved bank question and the questions available to generate automatic suggestion exam or parameterized suggestion exam. For the automatic mode, all questions are generated based on relative weight equation while in parameterized/manual mode the user is prompted some different values to suggest the exam perfectly as he need. Figure 2 describes different steps of exam generation using all the questions available in the question bank for a certain chapters of a course.

4. Experimental Results

The proposed system is developed using Microsoft dot.Net Technology. It depends on backing storage database Microsoft SQL Express that is used to store the learned rules and in building user friendly interface that facilitate how user interact with the system. The system requires Microsoft Windows and Microsoft dot.Net Framework as runtime environment. The system is developed and tested on a laptop with Core i5 second edition and 6 GB RAM. The system is stable and rendered well using basic graphic adapter. The computer was configured and powered by Windows 8 64 bit.

Here, a given statement for example is discussed to describe powerful of the system and ability to extend and learning functionality. The statement is "I found my books on the table". Statement is very simple contains a proposition mark "on" that let the system to predict a location in the statement so it could suggest a where question. However, question is not generated because of the total statement have a rule that doesn't exists before. Also, if there is a POS rule stored in the database there is a limitation of statistical parser OpenNLP of named entity as "the table" is not a location or organized can be detected by it so the key solution based on detecting a custom location proposition.

For the given statement "I found my books on the table", it would be parsed using OpenNLP parser as follow, PRP VBD PRP$ NNS IN DT NN, respectively. In turn, the user of the system would inform the system about three things:

a) WH-question tool and its mark; Where and on in the statement, as shown in figure 4.

b) Verbs in different tens, as shown in figure 5

c) Template rule as an ordered series of POS elements, as shown in figure 4.

The system in turn will be trained and learned so that if it faces another case of same states regardless of phrases in the statement itself, as shown in figure 5.

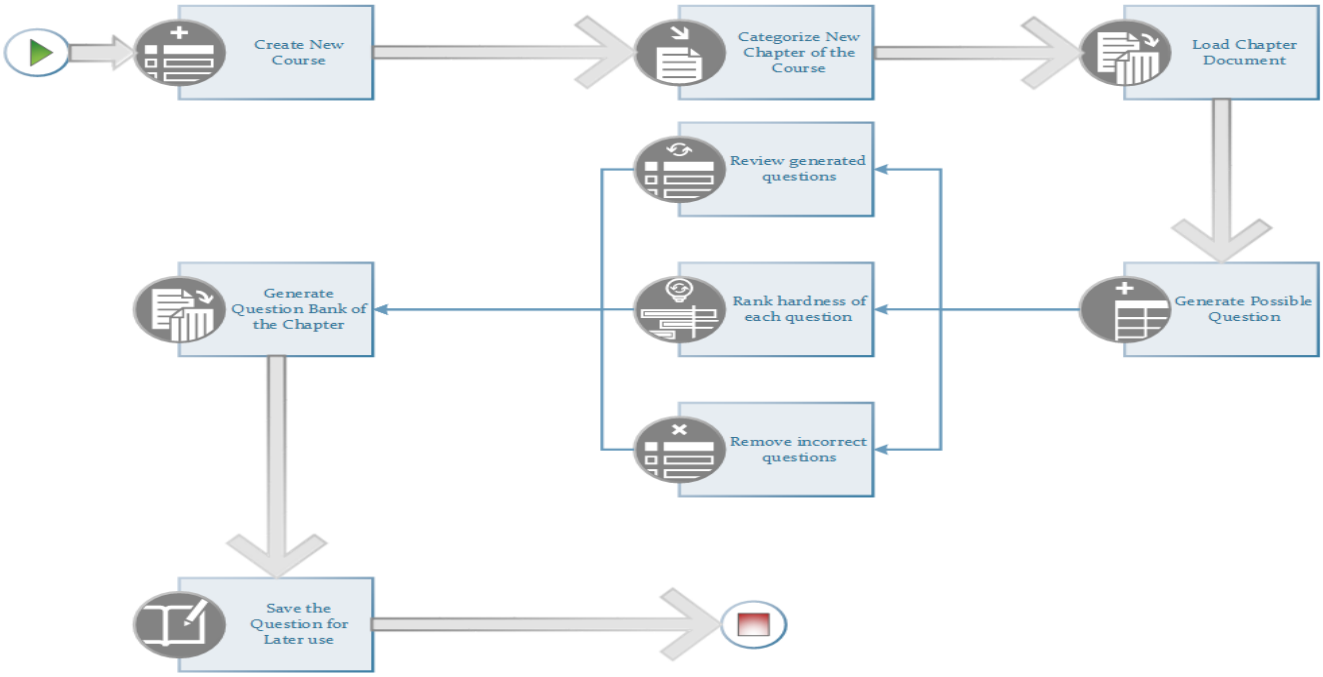

Fig. 2: The question generation workflow.

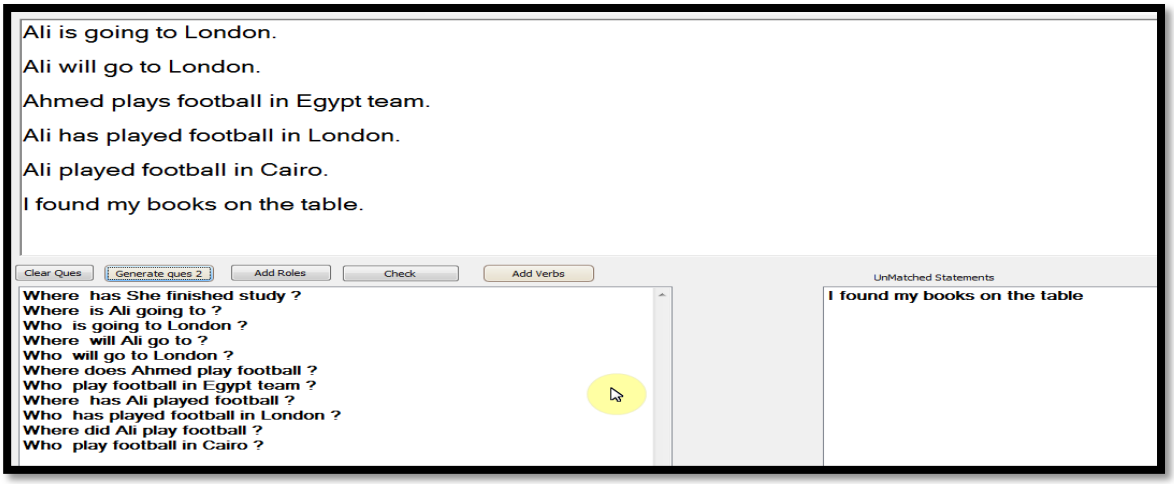

Fig. 3: Unmatched Statement [Question Training].

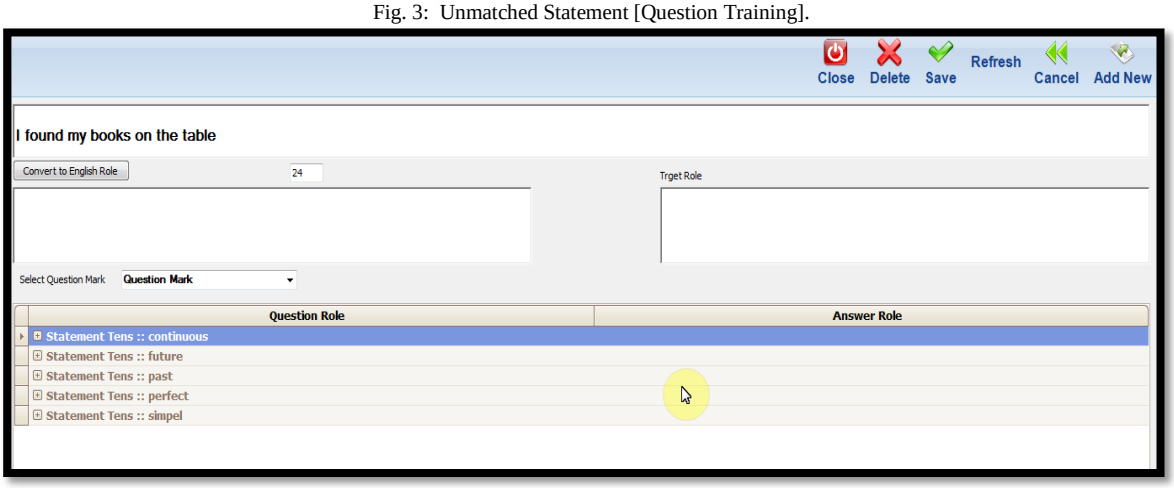

Fig. 1: Train the system using unmatched statement [Question Training].

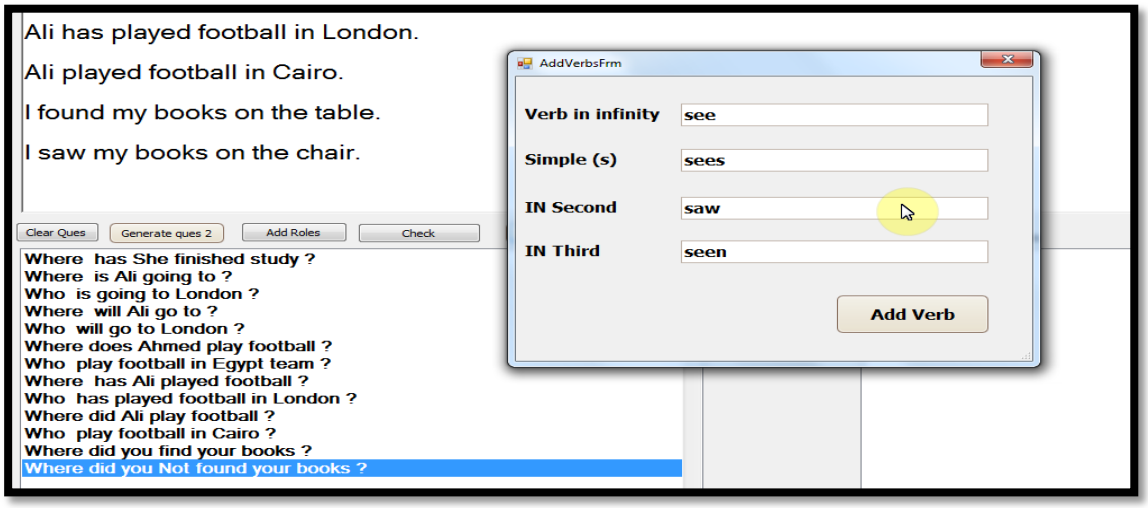

Fig. 5: Insert new verb [Question Training].

Based on the experimental results and the testing, the proposed system can be used in educational environment such as school and colleagues because of its high ability to adapt to different levels of topics, articles or even lectures. The system is capable to be trained more and more and hold thousands to millions of rules in the database. Also, the system is able to expand more and more of its application area by more training about the new

application area topics and documents. The system is powerful tool to facilitate the process of student evaluation and test his score on different levels reducing the hard effort on the tutor or the teacher of the class to make questions over days. The system is fast computing program as it is based on multi-threaded program that utilize the time and resources usage.

5. Conclusions

Question generation process is a hard activity in the educational environment that became one of the most popular research problems in modern days. The proposed system is a solution to question generation based on a powerful statistical open source parser call OpenNLP. The proposed system is flexible, ready to use, ease to use and applicable in different levels of education environments. The proposed system is large scale ready with ability to learn more and more about different styles of topics and fields. The system can be extended in future to be self-learning and able to recognize more intelligently about new statements with auto suggestion without needs to user actions.

References

[1] M. Heilman, "Automatic Factual Question Generation from Text", Ph.D. Dissertation, Carnegie Mellon University, 2011.

[2] X. Yao, "Question Generation with Minimal Recursion Semantics", Master Thesis, Saarland University and University of Groningen, 2010.

[3] T. W. Lauer, E. Peacock, and A. C. Graesser, "Questions and Information Systems", Lawrence Erlbaum, Hillsdale, 1992.

[4] A. C. Graesser, K. Vanlehn, C. Rose, P. Jordan, and D. Harter, "Intelligent Tutoring Systems with Conversational Dialogue", AI Magazine, Vol. 22, No. 4, 2001, pp. 39–51.

[5] V. Rus, and A. C. Graesser, Workshop Report: "The Question Generation Shared Task and Evaluation Challenge", Institute for Intelligent Systems, Memphis, TN, ISBN: 978-0-615-27428-7, 2009.

[6] L. Bednarik, and L. Kovacs, "Implementation and Assessment of the Automatic Question Generation Module", In Proceedings of the IEEE 3rd International Conference on Cognitive Infocommunications (CogInfoCom), 2012, pp. 687–690.

[7] H. T. Dang, D. Kelly, and J. Lin, "Overview of the TREC 2007 Question Answering Track", In Proceedings of the 16th Text Retreival Conference, Gaithersburg, Maryland, 2007.

[8] A. Andrenucci, and E. Sneiders, "Automated Question Answering: Review of the Main Approaches", In Proceedings of the 3rd International Conference on Information Technology and Applications (ICITA'05), 2005, Vol.1, pp.514-519.

[9] J. McGough, J. Mortensen, J. Johnson, and S. Fadali, "A Web-Based Testing System with Dynamic Question Generation", In IEEE Frontiers in Education Conference, 2001,Vol. 3, pp. S3C - 23-8.

[10] H. Ali, Y. Chali and S. Hasan, "Automatic Question Generation from Sentences: a Preliminary Approach", in Proceedings of the Conference on Traitement Automatique de la Langue Naturelle, Montreal, Canada, 2010.

[11] H. Ali, Y. Chali and S. Hasan, "Automation of Question Generation From Sentence", In Proceedings of the Third Workshop on Question Generation, Pittsburgh, Pennsylvania, United States of America, 2010.

[12] W. Chen ,G. Aist, and J. Mostow, "Generating Questions Automatically from Informational Text", In Proceedings of AIED Workshop on Question Generation, 2009, pp. 17–24.

[13] H. Prendinger, P. Piwek, and M. Ishizuka, "Automatic Generation of Multi-Modal Dialogue from Text Based on Discourse Structure Analysis", In Proceedings 1st IEEE International Conference on Semantic Computing (ICSC-07), 2007, pp. 27–36.

[14] P. Piwek and S. Stoyanchev, "Generating Expository Dialogue from Monologue: Motivation, Corpus and Preliminary Rules", In Proceedings of the 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2010, pp. 333–336.

[15] S. Harabagiu, A. Hickl, J. Lehmann, and D. Moldovan. "Experiments with Interactive Question-Answering", In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, 2005, pp. 205–214.

[16] M. Heilman and N. A. Smith. "Question Generation via Overgenerating Transformations and Ranking", Online:http://www.cs.cmu.edu/mheilman/papers/heilman-smith-qg-tech-report.pdf, Last visit on 2014.

[17] D. M. Gates, "Automatically Generating Reading Comprehension Look-back Strategy Questions from Expository Texts", Master's thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania , 2008.

[18] K. Nikiforos, L. Ha, and M. Ruslan, "Generating Multiple-Choice Test Items from Medical Text: A Pilot Study", In Proceedings of the Fourth International Natural Language Generation Conference, 2006, pp. 111–113.

[19] J. C. Brown, G. A. Frishkoff, and M. Eskenazi, "Automatic Question Generation for Vocabulary Assessment", In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing(HLT/EMNLP), 2005, pp. 819–826

[20] W. Weiming, H. Tianyong, and L. Wenyin, "Automatic Question Generation for Learning Evaluation in Medicine". In Advances in Web Based Learning – ICWL '07, of Lecture Notes in Computer Science, 2008, pp. 242–251.

[21] J. H. Wolfe, "Automatic Question Generation from Text - an Aid to Independent Study SIGCUE Outlook", In proceedings of the SIGCSE-SIGCUE technical symposium on Computer science education, 1976, pp.104–112.

[22] X. Yao, and Y. Zhang, "Question Generation with Minimal Recursion Semantics", In Proceedings of QG2010: The Third Workshop on Question Generation, 2010, pp. 68-75.

[23] A. Olney, A. Graesser, and N. Person, "Question Generation from Concept Maps", Dialogue and Discourse, Vol. 3, No. 2, 2012, pp. 75-99.

[24] S. Curto, A. Mendes, and L. Coheur, “Question Generation Based on Lexico-Syntactic Patterns Learned from the Web” Dialogue & Discourse, Vol. 3, No. 2, 2012, pp. 147-175.
[25] D. Lindberg, F. Popowich, J. Nesbit, and P. Winne, “Generating Natural Language Questions to Support Learning on-line”, In Proceedings of the 14th European Workshop on Natural Language Generation, 2013, pp. 105-114.
[26] M. Agarwal, R. Shah, and P. Mannem, “Automatic Question Generation using Discourse Cues”, In Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications, Association for Computational Linguistics, 2011, pp. 1-9.
[27] R. Agerri, J. Bermudez, and G. Rigau, "IXA Pipeline: Efficient and Ready to Use Multilingual NLP tools", in: Proceedings of the 9th Language Resources and Evaluation Conference (LREC2014), 2014, pp. 26-31.
[28] M. W. David, "Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness& Correlation". International Journal of Machine Learning Technologies, Vol. 2, No. 1, 2011, pp. 37–63.
[29] P. Perruchet, R. Peereman, "The Exploitation of Distributional Information in Syllable Processing", Journal of Neurolinguistics, Vol. 17, No. 2, 2004, pp. 97−119.
[30] M. W. David, "The Problem with Kappa", Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, 2012, pp. 345-355.
[31] F. Tom., "An Introduction to ROC Analysis", Pattern Recognition Letters, Vol. 27, No. 8, 2006, pp. 861–874.
[32] L. O. David, and D. Dursun, Advanced Data Mining Techniques, Springer, 2008.