IJETCR
Open Access
ISSN: 2348 - 2117
International Journal of Engineering Technology and Computer Research (IJETCR)
Available Online at www.ijetcr.org
Volume 3; Issue 2; Page No. 140-152

An intelligent system for exam generation

A. F. Elgamal[1], H. A. Abas[2], E.A.Elayat[3]

[1]Associate Professor of Computer Science, Faculty of Specific, Education, Mansoura University, Egypt

amany_elgamal@hotmail.com

[2]professor of curriculum and teaching methods, Faculty of Specific, Education, Mansoura University, Egypt

dr_hanaa2001@yahoo.com

[3]Associate Lecturer, Computer Teacher Preparation Department, Faculty of Specific Education, Mansoura University, Egypt

ealayat@yahoo.com

ABSTRACT

This paper describes an intelligent system for exam generation. The supposed system consists of two main subsystems, namely Question Generation 'QG' and Exam Maker 'EM', which are designed to produce an exam sheet for an E-course based on modules in higher education institutions.

The first subsystem, QG depends on natural language processing techniques such as tokenization, part of speech tagging (POS) and Named Entity Recognition (NER). It also uses the template matching approach to generate two types of questions: factual questions such as WH questions (who – where – when – what and what is the percentage of), and gap filling question GFQ.

Exam maker is the second subsystem in the proposed system, and it generates an exam sheet taking into account a set of criteria which includes the relative weight of each module, the objectives set for each lesson, formal quality assurance standards of the exam paper, and diversity of the questions.

The proposed system has been implemented in the Pharmacognosy E-course taught to the first year students in the Faculty of Pharmacy, Mansoura University, Egypt. Evaluation of the proposed system is presented.

Key words: Natural Language Processing, Question Generation, Tokenization, Part Of Speech Tagging, Named Entity Recognition.

INTRODUCTION

Natural Language Processing (NLP) is an effective technology for a group of computational systems for representing and analyzing naturally occurring texts, at one or more levels of linguistic analysis for the aim of simulating human-like language processing for a set of applications [1]. NLP techniques are utilized as a part of applications that make questions to databases and extract data from texts, for example, Question Generation, retrieving relevant documents from a collection or translation, by generating text responses, or identifying spoken words then turn them into text [2].

Question Generation (QG) from a text, is a new research subject that brought to the attention from the Discourse and Dialog, Psycholinguistics, NLP searcher, Intelligent Tutoring System, and Information Retrieval (IR) communities as of late [3]. It is defined as the process of automatically generating questions from some type of data, which could vary from the data in a database to a deep semantic representation to crude text [4].

QG can possibly help instructors effectively evaluate students' acquisition of essential factual knowledge, thereby enabling teachers to focus on more learning exercises and complex tasks. Student assessment is a system intended to gather data about the information, behavior, or skillfulness of a student or gathering of students [5]. A QG framework can be useful in asking students' questions taking into account learning materials with a specific end goal to check their achievement or help students focus on the main topics in their study [6]. There are several ways to automatic QG such as QG based on syntactic and keyword modeling [7], and QG based on templates [8].

There are many researches that have been presented in this field, Liu et al. Presents a new Automatic Question Generation (AQG) approach which generates trigger questions as a kind of support for students' learning through writing. AQG first automatically extracts citations from students' compositions together with key content elements. Next, the citations are categorized utilizing a

Corresponding author: A. F. Elgamal

rule-based approach and questions are created based on a set of templates and the content elements [9].

The first Question Generation Shared Task and Evaluation Challenge (QGSTEC) -2010 is one of the efforts of the QG group it has provided two tasks. The first task concentrates on assessing the generation of questions from paragraphs(QGP), and the second on the generation of questions from sentences [10].

In this paper, an intelligent exam generation system is proposed and implemented on an academic higher education E-course. The proposed system can generate an exam paper containing two types of questions, namely factual questions which begin with (what, where, when and who, what is and what the percentage of), and (GFQ).

The rest of this paper is organized as follows: Section 2 presents the components of the proposed system; section 3 describes the experimental work; section 4 describes an evaluation of the proposed system, and finally section 5 presents the conclusion.

2- The proposed intelligent exam generation system

The proposed intelligent exam generation system contains two main subsystems, namely QG and EM, each of them consists of a lot of sub-components, as shown in figure (1).

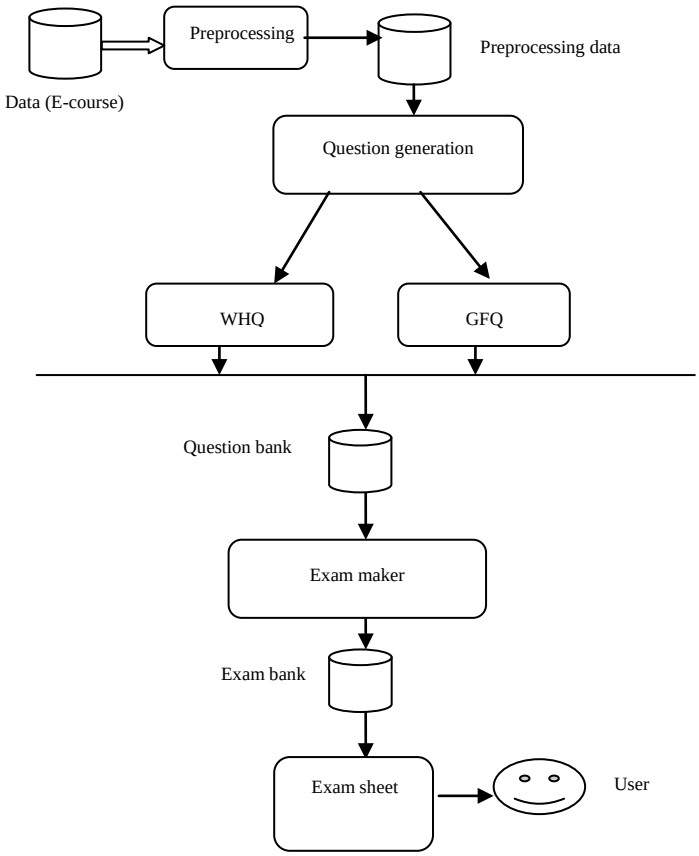

Figure 1: A general overview of the proposed intelligent exam generation system

2.1. Question generation

An AQG can be done from a paragraph or sentence, (QGP) has been defined as an application- independent which means that questions will be judged based on of the input paragraph content analysis. In (QGP), some questions will be produced at three specificity levels, the first of which is broad (whole include paragraph), the second is medium (multiple clauses or sentences), and the last is particularly (phrase or less) [11].

In Question Generation from a sentence, a gathering of sentences is given, and the user need to give possible questions to every sentence, based on predefined question types such as (WHQ, GFQ) [12].

The proposed system uses the task of generating questions from sentences, to generate all possible questions from each E- course sentences, for this purpose the process of manually E-course reorganization has been done to facilitate the questions generation process. Every paragraph of the input E-course is split into sentences for this purpose.

2.1.1. Generation of the WH question type

Figure (2) shows a general overview of the proposed QG subsystem components for the first type questions, namely the "WH".

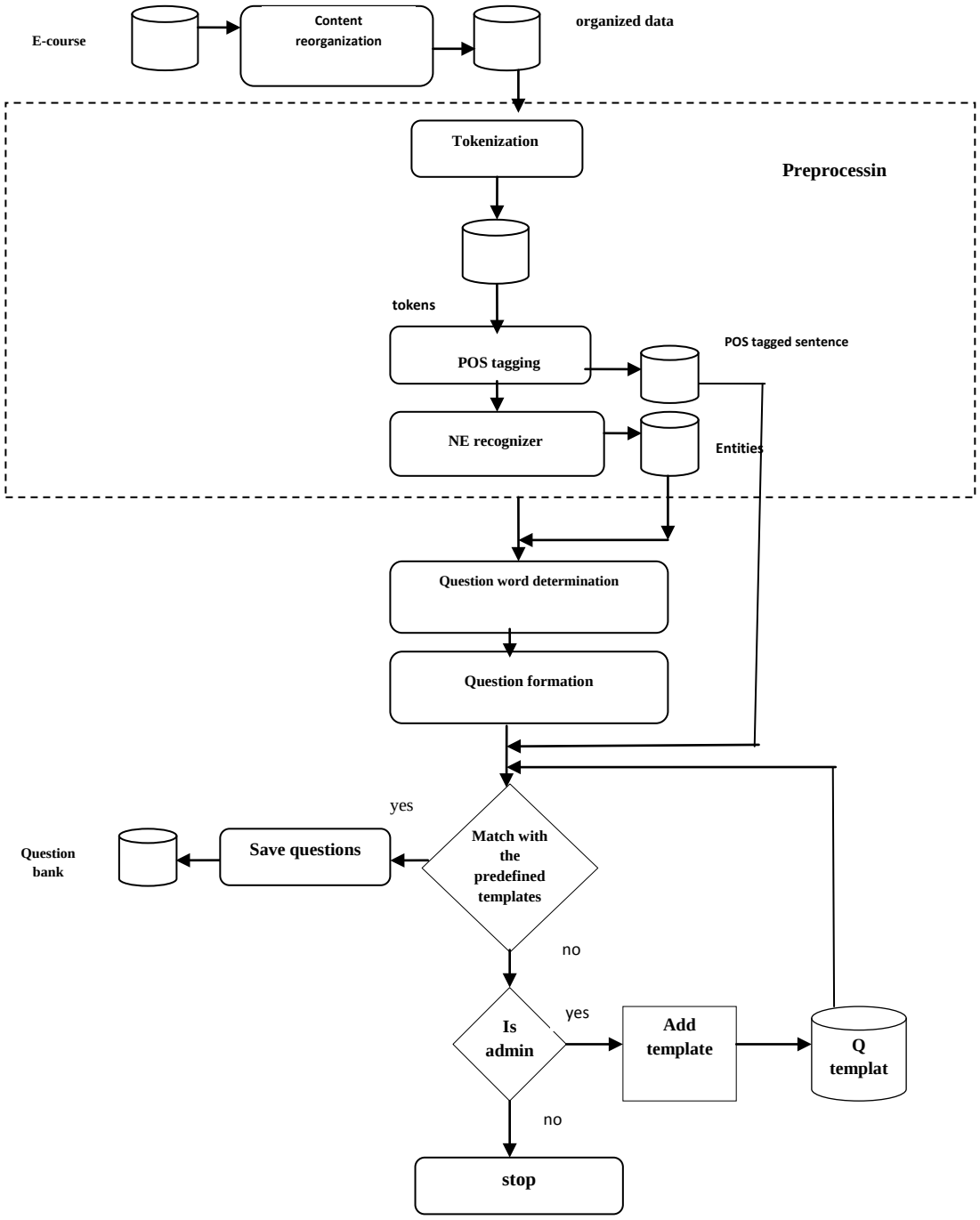

Figure 2: A general overview of the "WH" question type

The automatic question generation process for the WH question type goes through many stages, as follows:

2.1.1.1: Data preprocessing stage

Cleaning and processing raw data is an important initial part of any NLP task; the proposed subsystem uses an open NLP tool during this stage. Which is a Java library for processing natural language text based on machine learning, it includes pre-built models for some languages and annotated text resources. It is supported NLP tasks such as tokenization, part of speech tagging, named entity extraction, chunking and parsing [13].

The entered E-course is introduced in the sequence shown in figure (3). It contains a number of modules (m_1,

m_2,…, m_i), with each module containing a number of lessons (s_1, s_2, s_3,…. s_j), and the content of each lesson organized as: title, objective, introduction and contents.

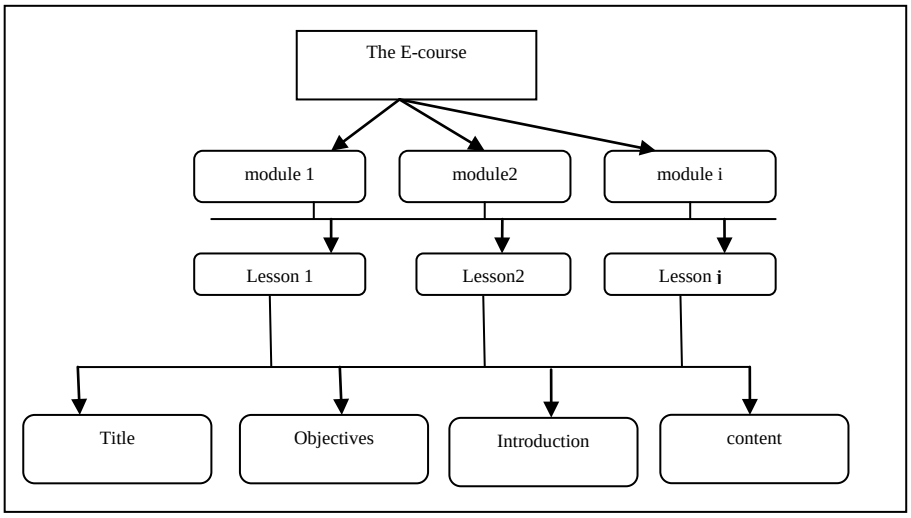

Figure 3: Sequence of the entered course

After manually reorganization of the E-course contents, the organized paragraphs are segmented into many sentences, then the preprocessing stage is applied through the open NLP tools which are included in the proposed subsystem as follows: tokenization, part of speech tagging, and named entity recognition.

I. Tokenization

The entered E-course paragraphs are segmented into sentences in separate lines. The tokenization process is defined as the task of dividing out (tokenizing) words from running text, a token can be words, numbers, punctuation marks, parentheses and quotation marks [14]. The tokenization process is implemented into each sentence of the E-course. For example: "Ali played football in the school" is converted to tokens as follows: [Ali] [played] [football] [in] [the] [school].

II. Part Of Speech Tagging (POS)

The POS process determines the verbs and their tenses, names, determinants, relational words and other types of linguistic determinants. It gives a tag for every word to recognize each token in the sentence.

A sample of these tag sets is shown in table (1), every resulting separated token is passed to the (POS) tagger to generate the POS tagged sentence.

Table 1: A sample of the tag set

Tag	Description	Example
NNP	Proper noun, singular	Ali
NN	(Common) or mass noun	Time, job, school, football
NP	Singular proper noun	Italy, FIFA
VB	Verb, base form	make, go, play
VBD	Verb, past tense	Saw, played
JJR	Comparative adjectives	Easier, taller
CC	Conjunction	And, or, but
CD	Cardinal number	Eight, one, ten
IN	Preposition or subordinating conjunction	In
JJ	Adjective	Red, big
DT	Determiner	A, the
TO	To	to

The POS tagger marks tokens with their corresponding word types taking into account both the token itself and the token context. When the tag is repeated many times in the sentence, it will be distinguished by a number to make it easy to be recognized later in the process of generating a question.

The POS tagged sentence of the previous example: [NNP] [VBD] [NN] [IN] [DT] [NN2].

III. Named Entity Recognition (NER)

Determining of named entities (e.g. People, organizations, locations, etc.) Is a basic task in many NLP applications, it is a technique of token classification [15].

A set of pre-trained models are available in the open NLP library which includes common categories such as (location, organization, person, time etc.). The library also includes methods for recognizing these categories, such as (name finder, location finder, percentage finder, and period finder). Applying the proposed subsystem to the previous example produces the following result:

Name Finder: <START: name>Ali<END> played football in the school.

Location finder: Ali played football in the <START: location> school <END>.

The resulted POS tagged sentence, and named entities are considered as the input to the next stage named question word determination.

2.1.1.2 Question Word Determination stage

The QG proposed subsystem generates factual questions, where the QG subsystem detects the question word automatically, using the identified entity type contained in the sentence (s), through some production rules as follows:

(NN ∃ S) → (people category ^ (QW ≡ who))
(NNP ∃ S) →((location category) ᵛ (organization category)) ^ (QW ≡ where)
(NNP ∃ S) ^ ((NNP ¬ location category) ᵛ (NNP ¬ organization category)) →((no entity category)^ (QW ≡ what))
(CD ∃ S) → (period category (QW ≡ when))
(CD ∃ S) ^ (CD ¬ period category) → (QW ≡ what is the percentage of)

The system dictionary is an embedded component in the QG proposed subsystem to improve the system performance. It supports the identification of most entities in the input data; and it consists of a predefined category contained of a lot of words with their alternative such as (differentiate, different), and the suitable question word for each category. When the NER recognizer identifies any predefined entity, it detects the appropriate question word. But when it cannot recognize the entity, the administrator adds this entity to the system dictionary.

2.1.1.3. Question Formation stage

At this stage, distinguishing the verb complex (VC) and its tense is fundamental for the syntactic transformations that need to be performed to transform the declarative sentence into a question. The VC contains of the main verb alongside any auxiliaries (AUX). A (AUX) is a verb that determines the mood, tense, or part of another verb in a verb phrase. It could be modal (can, may, will, shall, must, ought) or non-modal (be, have, do) [16]. The following pseudo code illustrated how to deal with VC to formulate a correct question.

Question Formation pseudo code

```
Assign POS tagged sentence to P_O_S
Assign verb with its AUX to VC
Assign AUX to  A_U_X
Determine the VC  tense according to following nested if
 // Check the existence of an A_U_X  in VC
   If VC contained A_u_x  then
   Go to  1
   Else
   Applying do mapping support rules to identify the Aux.
   1: move the Aux into first position in the sentence
   Insert question word (WH) before the Aux
  //Lemmatization
  Connect verbs DB to get VB (verb in stem form)
    If VB exists in verbs DB then go to 2
    Else
    Add the missing verb with its three conjunctions  in verbs DB
```

```
2:  Put VB after the Aux
    // Check the existence of pronouns in P_ O_S
    If P_ O_S contains pronouns then
   Call sub pronoun_ treatment ()
   End if
```

Nested if to determines the verb tense

```
If (p_o_s  Contains ("VBG")) then
Verb tense = "continuous"
Else if
((p_o_s  Contains ("VBD")) AND NOT (p_o_s  Contains ("VBZ") OR ("VBP")) AND NOT (( p_o_s. Contains ("VBN") AND A_U_X  Contains ("had")) AND (p_o_s  Contains ("VBN") AND A_U_X  Contains ("has")) AND (p_o_s  Contains ("VBN") AND A_U_X  Contains ("have")))) then
Tense = "past"
 Else if
((p_o_s. Contains ("VBZ") OR p_o_s. Contains ("VBP")) AND NOT (p_o_s. Contains ("VBG") AND NOT (( p_o_s. Contains ("VBN") AND A_U_X. Contains ("has")) OR (p_o_s. Contains ("VBN") AND A_U_X. Contains ("have"))))) then
Tense = "simple"
Else if
((p_o_s. Contains ("VBZ")) OR (p_o_s. Contains ("VBN") AND A_U_X. Contains ("has")) OR (p_o_s. Contains ("VBP") AND  p_o_s. Contains ("VBN") AND  A_U_X. Contains ("have")) OR (p_o_s. Contains ("VBD") AND p_o_s. Contains ("VBN"))) then
Tense = "perfect";
End if
Return tense
```

If the VC does not include an AUX, some rules need to be connected to bring an AUX into the question. The question transducer breaks down the main verb into the proper form of "do" by do-support through the mapping rules shown in table (2) [17].

Table 2: do-support in the question

POS	AUX
VBD	Did
VBZ	Does
VBP	Do

The main verb is lemmatized according to its tense through an embedded database" verbs DB" in the QG proposed subsystem for the three conjunctions (stem verb, past simple, past participle) of the irregular and non- irregular verbs.

Sub Pronouns _ treatment ()

```
//  Converting between the pronouns
Assign the original sentence words  to word ()
If (word. Contains ("I")) then
   {int I = word. IndexOf ("I");
    Word. Remove_At (I);
    Word. Insert (I, "you");}
End if
If (word. Contains ("My")) then
     {int I = word. IndexOf ("My");
      Word. Remove _At (I);
      Word. Insert (I, "your")}
End if
If (word. Contains ("am")) then
   {int I = word. IndexOf ("am");
   Word. Remove_At (I);
   Word. Insert (I, "are")}
  End if
```

The sentence " Ahmed played football in the garden at 6 am." Is a good example of the question formation stage as it is shown in table (3)

Table 3: an example of question formation stage

Sentence	POS tagged sentence	VC	Is Aux exists	Applying nested if	Applying do mapping rules	Question reorganized
Ahmed played football in the garden at 6 am	NNP VBD IN DT NN1 IN CD	VBD (PLAYED)	No	Tense = past	Played → AUX = DID	QW & DID & NN & PLAY (VB) &……

The question formation stage is considered the most important one, where it is integrated with the question template which will be illustrated soon to reach the correct final question form.

2.1.1.4. Question generation stage

This is the final stage in generating WH-questions with template-based approach, as shown in figure (2). Question templates offers the capacity to ask questions that are not as firmly coupled with the accurate wording of the source content. A question template contains predefined order tags with placeholder variables to be replaced with content from the source POS tagged sentence [18].

The template matching process is applied according to the integration between the extracted syntactic features (POS tagged sentence) and the extracted semantic features (entities) which identifies the (WH) question word, this integration decreases the processing time. Where the connection to the DB is concerned with specific tense and question word, the template matching produces the suitable template of question to generate the appropriate question. Table (4) illustrates some examples of sentences which passed all stages to get the WHQ type.

Table 4: Examples of sentences which passed all stages described to get the WHQ type

No	Sentence	Extracted syntactic features	Extracted Semantic features			Question word	Question template	The generated questions
1	Ali went to the school.	NNP VBD TO DT NN.	NNP	Ali	Person	Who	VBD TO DT NN.	Who went to the school?
			NN	School	Location	Where	NNP VBD TO	Where did Ali go?
2	Ahmed plays football in Egypt team.	NNP VBZ NN IN NP NN1.	NNP	Ahmed	Person	Who	VBZ NN IN NP NN.	Who plays football in Egypt team?
			NN	Football	No entity	What	NNP VBZ IN NP NN.	What does Ahmed play in Egypt team?
			NP	Egypt	Location	Where	NNP VBZ NN	Where does Ahmed play football?
3	My son will travel to Egypt in 02/02/2018.	PRP NN NN1 VB TO NP IN CD.	CD	02/02/2010	Date	When	VB PRP NN NN1 TO NP	When will your son travel to Egypt?
4	In Arabic exam I get over than 75%.	IN JJ NN PRP VB RB IN2 CD NN3.	CD	75%	Percentage	What percentage	PRP VB IN JJ NN	What percentage do you get in Arabic exam?
5	I found my books on the table	PRP VBD PRP$ NN IN DT NN1	NN1	Table	Location	Where	PRP VBD PRP$ NN	Where did you find your books?

For the unmatched sentences, the QG proposed subsystem creates new question templates to generate appropriate questions by the administrator. The QG subsystem helps the administrator to build a new question template, by recalling the POS tagged sentence, selecting the question word manually, and reorganizing the POS tagged sentence to be a question.

The QG subsystem previews the question after building its template. If it is true, the template will be saved in the question template repository; but if it is false, the administrator rebuilds the template and tests it until it becomes correct. By continuing application of the proposed system, the template repository is increased which allowing match of more POS tagged sentences to generate more appropriate questions.

2.1.2. Generating the gap filling question type (GFQ)

It is usually known as objective questions in which one or more words are removed from a sentence/paragraph, and they require the utilization of the target word in a particular context. A good GF sentence ought to be informative. An informative sentence in a document is

one which has relevant knowledge that is useful in the context of the document [19].

The QG subsystem generates GFQ by replacing the entities identified from the NER step with empty place according to their importance in the context of the E-course, which is determined from the context objectives.

After generating the two types of question (WH and GF) for each module of the E-course, these questions are saved in a question bank.

2.2. Exam Maker

The question bank contains questions which are organized according to the modules and lessons, to facilitate the task of exam maker subsystem. It is concerned with producing exam sheets for the E- course.

There are sets of criteria to be taken into account in the exam making process, including the relative weight of each module, the objectives set for each lesson, formal quality assurance standards of the exam paper, and diversity of the questions.

Creating an exam in the exam maker subsystem can be performed manually or automatically. In both the automatic or manual exam making process, the user should enter the number of questions included in the exam sheet. Selecting the appropriate number of questions for each module to achieve the relative weight is based on equation (1).

$$\text{No. question of each module} = \frac{\text{module's total page no}}{\text{total page no. of the whole course}} * \text{total exam questions no} \quad \ldots\ldots\ldots\ldots (1)$$

After determining the number of questions for each module, the EM subsystem should determine the number of each type of questions that have been generated for each module.

Question words such as " who, where, when, what, what the percentage of and GFQ" have a ratio of 20% for the total number of questions for each module, because they are so close in the level of mental maturity required to answer questions with these question words, and the knowledge level is the basis of knowledge in any science, taking into account the other levels.

Question word " what is" has a ratio of 80% of the total number of questions for each module, because it requires mental maturity levels higher than the previous type of questions.

According to the previous steps of determining the question numbers of each module and the percentage of each question word type, the EM subsystem presents the exam sheet to the user for preview and selects the date of the exam, the college logo and the exam duration. Finally, the exam sheet is ready for printing.

In the manual exam making process, the questions are displayed to the user to choose from them. The displayed questions and their lessons are organized according to their modules, taking into account the same determinants in the automatic exam making process; which are illustrated to the user in the help menu. In such a way, the exam sheet final form is ready for printing, the printed exam paper contains the name, date, duration of the exam, and the questions with their degrees.

3. Experiment work

The proposed intelligent exam generation system is implemented using the c# programming language, and SQL Server 2010. The system is applied to the Pharmacognosy E - course as a sample of experimental work. The course contents, sequence are shown in figure (4).

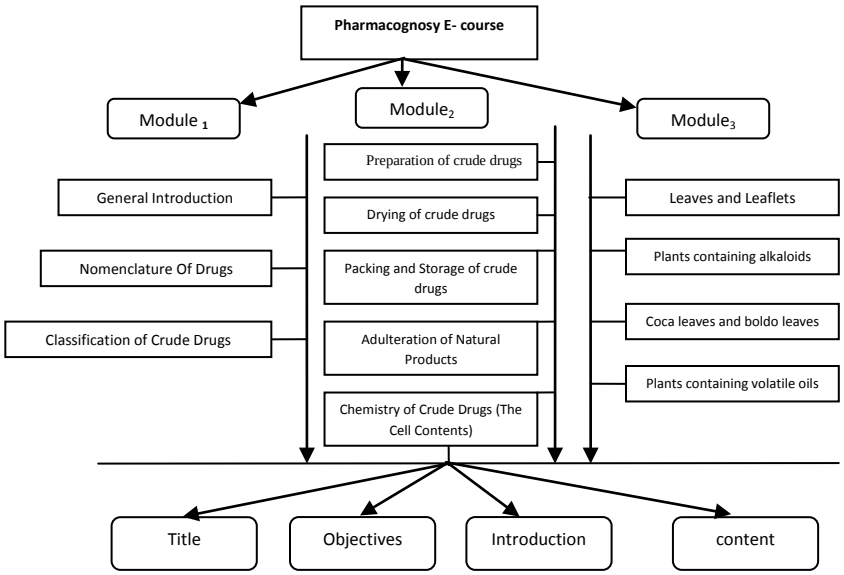

Figure 4: The organization sequence of the " Pharmacognosy " E- course "

Table (5) illustrates the application of the previous steps into a sample of the entered E-course.

Table (5) Examples of the generated WHQ type

No	Sentence	Tokenization	POS tagging	NER			Question word	Question template	The generated questions
1	Crude drug means the collected and usually dried or fresh plant or animal product of medicinal use without any further processing	[Crude] [drug] [means] [the] [collected] [and] [usually] [dried] [or] [fresh] [plant] [or] [animal] [product] [of] [medicinal] [use] [without] [any] [further] [processing].	JJ NN VBZ DT VBN CC RB VBN1 CC2 JJ3 NN4 CC5 JJ6 NN7 IN JJ8 NN9 IN10 DT11 JJ12 NN13.	JJ NN	Crude drug	No entity	What	JJ NN VBZ	What does crude drug mean?
2	Recognize the damage using of wild plants in the pharmaceutical industry.	[Recognize] [the] [damage] [using] [wild] [plants] [in] [the] [pharmaceutical] l [industry].	VB DT NN VBG JJ NNS IN DT1 JJ2 NN3	JJ NN S	Wild plants	No entity	What is	DT NN VBG JJ NNS IN DT1 JJ2 NN3	What is the damage using of wild plants in the pharmaceutical industry?
3	Buchu is cultivated in South Africa	[Buchu] [is] [cultivated] [in] [South] [Africa]	NNP VBZ VBN IN NNP1 NNP2 .	NN P1 NN P2	South Africa	Location	Where	NNP NN NNS VBP RB	Where is Buchu cultivated?
4	Pharmacognosy was introduced by Sydler in 1815	[Pharmacognosy] [was] [introduced] [by] [Sydler] [in] [1815]	NN VBZ VBN IN NNP IN1 CD.	CD	1815	Date	When	NNP VBN NN	When did Sydler introduce Pharmacognosy?

After saving the generated questions, the next step is generating the exam. The user chooses the exam parameters such as name, course name, modules, and full number of exam questions. The final output of the proposed system is an exam sheet shown in figure (5) " preview mode".

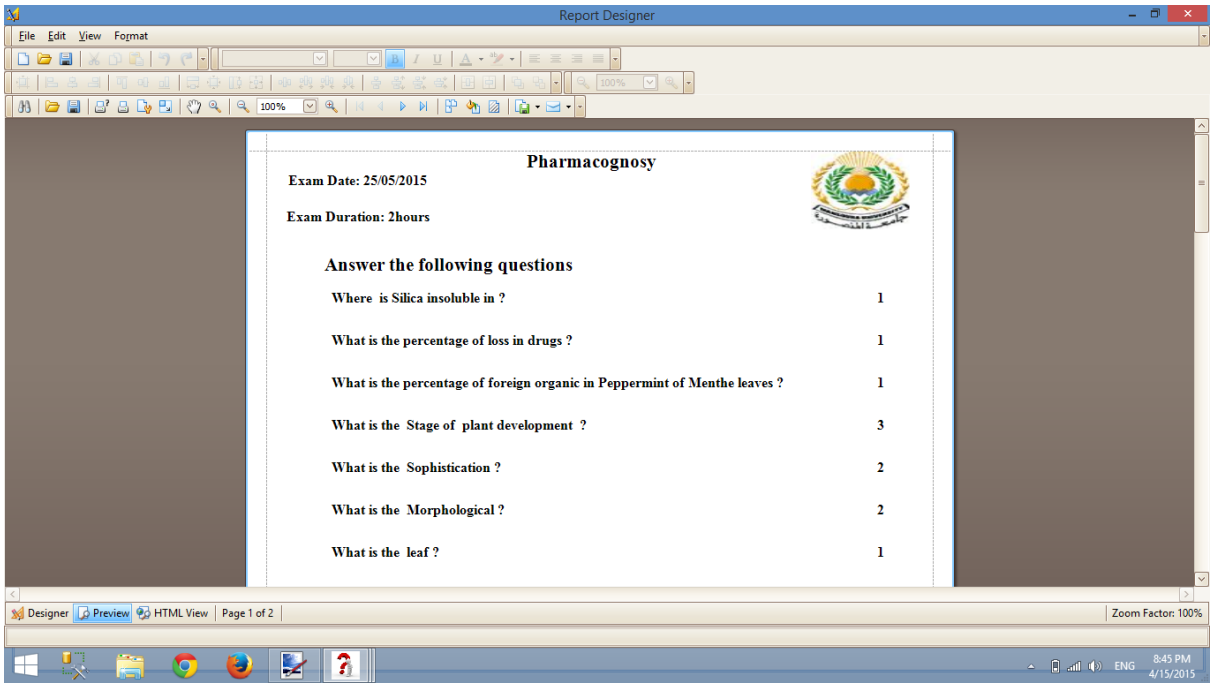

Figure 5: Exam sheet in the preview mode

The graphical user interface in figure (6) illustrates the exam sheet in design mode, which enables users to control each of the following modify the design of the exam sheet (font type, color, effects and size), Watermark type and existence, Printing preview the whole exam sheet, and Save the exam sheet in multiple type of files such as (Pdf, rtf, image, and Html).

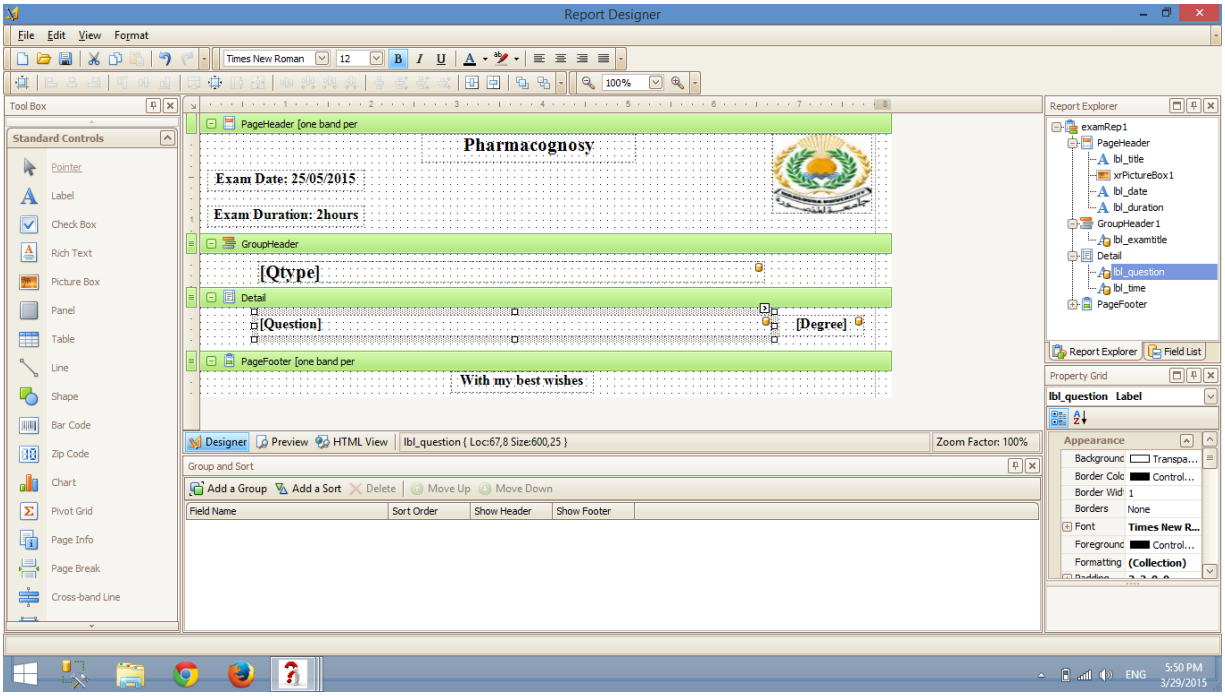

Figure 6: Exam sheet in design mode

4. Evaluation

There is no standard approach to evaluate the output of a QG system. On the other hand, we use recall, precision and F-score, which is defined as follows:

$$\Pr ecesion = \frac{correct}{correct + spurious} \quad .....(2)$$

$$\mathrm{Re}\, call = \frac{correct}{correct + missed} \quad .....(3)$$

$$F - score = 2 * \frac{precision * recall}{precision + recall} \quad .....(4)$$

Where correct = the number of questions generated by the QG subsystem and present in the questions generated manually from a document by the human judges; spurious = the number of questions generated by the QG subsystem, however not present in the manually generated questions; and missed = the number of questions present in the manually generated questions but not in the questions generated by the QG subsystem. [20]

To evaluate the factoid questions which generated by the proposed system, three independent judges are asked to generate all possible questions from the Pharmacognosy E-course, also the question bank contained in the Pharmacognosy E-course at the E- learning solutions at Mansoura university which is considered as the fourth judges. The four groups of questions are compared with the generated questions about the proposed system, the results are shown in table (6).

Table 6: the obtained Precision, Recall and F-Scores

Judges	Precision	Recall	F-score
Human judge1	0.39	0.71	0.52
Human judge2	0.27	0.57	0.36
Human judge3	0.28	0.69	0.41
E- learning solutions	0.32	0.83	0.46
The average	0.32	0.7	0.44

the previous table illustrates that, the recall is higher than precision because the generated questions from the proposed system are always greater than the questions generated by the human judges, this is because the proposed system passes over each sentence in the E-course and generates many questions on it .

5. Conclusion

In this paper, an intelligent system for exam generation was produced, which contained two main subsystems, "Question Generation 'QG' using NLP tools for generating questions, and the Exam Maker to produce an exam sheet for an E-course in higher education institutions.

Experimental work is done through the application of the proposed system in the Pharmacognosy E-course, which is taught to first Year students in the College of Pharmacy at Mansoura university, in the first semester. Evaluation of the experimental work illustrates that the proposed system is promising.

The proposed system does not contain questions with the question word "why or whom" which need more recognition of entities and sentence simplification. Also, question answering systems can implement and integrated with the proposed system.

6. REFERENCES:

1. D. Liddy, "Natural Language Processing for information retrieval and knowledge discovery ", In *Proc. COLING '94*, volume 2, 1998.
2. Fredrik Olsson, "*A literature survey of active machine learning in the context of natural language processing*", SICS Technical Report T2009:06, ISSN: 1100-3154,2009.
3. Kellaghan, Thomas, Vincent, Greaney. Using Assessment to Improve the Quality of Education. IIEP, *International institute for educational planning,* p. 19. Paris 2001. Paris
4. Rus, V. And Graesser, A.C, Workshop Report: The Question Generation Task and Evaluation Challenge, Institute for Intelligent Systems, Memphis, TN, ISBN: 978-0-615- 27428-7,2009, available at *http://www.questiongeneration.org*, viewed on 12-1-2015.
5. Rus, V. And Graesser, A.C., op.
6. M. Heilman. Automatic Factual Question Generation from Text. PhD thesis, Language Technologies

Institute, School of Computer Science, Carnegie Mellon University, 2011.

7. Saidalavi Kalady, Ajeesh Elikkottil, and Rajarshi Das. Natural language question generation using syntax and keywords. In Kristy Elizabeth Boyer and Paul Piwek, editors, Proceedings of QG2010:The Third Workshop on Question Generation, pages 1 {10, Pittsburgh, June 2010. Question generation. Org. URLhttp://oro.open.ac.uk/22343/.
8. David Lindbergh, "Automatic question generation from text for self-directed learning", *MASTER thesis*, School of Computing Science. Faculty of Applied Sciences. Simon Fraser University, 2013.
9. M. Liu, R.A. Calvo and V. Rus, &ldquo, Automatic Question Generation for Literature Review Writing Support, &rdquo, *Intelligent Tutoring Systems,* V. Aleven, J. Kay, and J. Mostow, Eds., Vol. 1, Springer, 2010.
10. Available at: http://www.cnts.ua.ac.be/conll, (Boyer, K. E. And Piwek, P, 2010), viewed on 20-1-2015.
11. V. Rus B. Wyse P. Piwek M. Lintean and C. Moldovan," Overview of The First Question Generation Shared Task Evaluation Challenge", Proceedings of *QG2010: The Third* Workshop on Question Generation shared task and evaluation challenge, 2010. p.47.
12. HUSAM Ali, " AUTOMATIC QUESTION GENERATION: A SYNTACTICAL APPROACH TO THE SENTENCE-TO-QUESTION GENERATION CASE", *MASTER thesis*, Bachelor of Science in Computer Science, University of Lethbridge, p. 58, 2012.
13. Available at: http://opennlp.sourceforge.net/projects.html, viewed on 25-12-2014.
14. Daniel Jurafsky and James H. Martin, "*Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*, PRENTICE HALL SERIES IN ARTIFICIAL INTELLIGENCE, Englewood Cliffs, New Jersey. P. 296, 2009.
15. David Nadeau and Satoshi Sekine," A survey of names entity recognition and classification". In Proc. Of the national research council Canada/ New York university, p. 3, 2009.
16. Available at:http://dictionary.cambridge.org/grammar/british-grammar/verb-phrases, viewed on 17-10-2014
17. Mannem, P., Prasad, R., And Joshi, A., "Question generation from paragraphs at UPenn: QGSTEC system description". In Proc. Of the Third Workshop on Question Generation, 2010.
18. David Lindbergh, OP,P.14,2013.
19. Manish Agarwal," Cloze and Open Cloze Question Generation Systems and their Evaluation Guidelines", *MASTER thesis,* Language Technology and Research Center (LTRC) International Institute of Information Technology Hyderabad - 500032, INDIA p. 2,2012.
20. Saidalavi Kalady, Ajeesh Elikkottil, and Rajarshi Das, OP,P.8,2011.