Automation of Question Generation From Sentences

Husam Ali, Yllias Chali, and Sadid A. Hasan

University of Lethbridge, Lethbridge, AB, Canada
{husam.ali,yllias.chali,sadid.hasan }@uleth.ca

Abstract. Question Generation (QG) and Question Answering (QA) are key challenges facing systems that interact with natural languages. The potential benefits of using automated systems to generate questions helps reduce the dependency on humans to generate questions and other needs associated with systems interacting with natural languages. In this paper we consider a system that automates generation of questions from a sentence, given a sentence, the system, will generate all possible questions which this sentence contain these questions answers. Since the given sentence may be a complex sentence, the system will generate elementary sentences, from the input complex sentences, using a syntactic parser. A part of speech tagger and a named entity recogniser are used to encode needed information. Based on the subject, verb, object and preposition the sentence will be classified, in order determine the type of questions that can possibly be generated from this sentence. We use development data provided by the Question Generation Shared Task Evaluation Challenge 2010.

Keywords: Question Generation, Syntactic Parsing, POS Tagging, Elementary Sentence, Named Entity Tagging, Recall.

1 Introduction

Learners, who actively self-regulate their learning, are often question generators [5]. Given that they recognize their knowledge deficits, learners ask questions that are either triggered by their deficits, or seeking reliable information sources to answer their questions. Unfortunately, this idealistic vision of intelligent inquiry is rarely met, except for the most skilled learners, as most learners have trouble identifying their own knowledge deficits [5].

In this paper we considered a Text-to-Question generation task using syntactic parsing, Part Of Speech (POS) tagger and Named Entity analyzer.

We had different modules to process the raw data, to generate elementary sentences from complex sentences using syntactical parser, Part of Speech tagger and Named Entity analyzer.

The elementary sentences got syntactically parsed in the next module, and using the Part of Speech tagged and Named Entity analyzed representation of the elementary sentences, we classified the sentences to determine the possible questions types that can be generated.

The questions generated got ranked based on the structure of the elementary sentence.

In the subsequent sections of this paper we will discuss the related work (section 2), the system and its structure (section 3), the evaluation of the system results (section 4) and conclusions and future plans (section 5).

2 Related Work

In the field of computational linguistics, dealing with Question Generation (QG) is getting more attention from the researchers [6]. Before the internet and electronic data storage, the time to search and find an answer for questions could extend for weeks hunting for documents in the library. Electronic books and information sources will be the mainstream in the future. In the last few years, new preoccupations appeared for automatic question generation. In ICITA'05 [1], they introduced a template-based approach to generate questions on four types of entities. The authors in ASEE/IEEE Frontiers in Education Conference [3] used WTML (Web Testing Markup Language), which is an extension of HTML, to solve the problem of presenting students with dynamically generated browser-based exams with significant engineering mathematics content.

3 Sentence to Question Generation

This section will discuss the overall framework for the Question Generation (QG) system. We considered the Question Generation from a single input sentence with a question type target (e.g. Who? Where? When? Etc,). For this purpose we used the development data provided by Question Generation Shared Task Evaluation Challenge 2010 (QGSTEC). Sentences will be selected from primary data sources for the QGSTEC (Wikipedia, Yahoo! Answers and OpenLearn) where extremely long or short sentences will be avoided. Since the sentences might have a complex structure with multiple clauses, it would be difficult to generate accurate questions from the complex sentences. Therefore, using syntactic information we simplify the process by extracting elementary sentences from the complex sentences. Based on the sentences subject, verb, object and preposition we classify the sentences to determine the possible type of questions that will be generated. Figure 1 depicts the basic structure of our QG system.

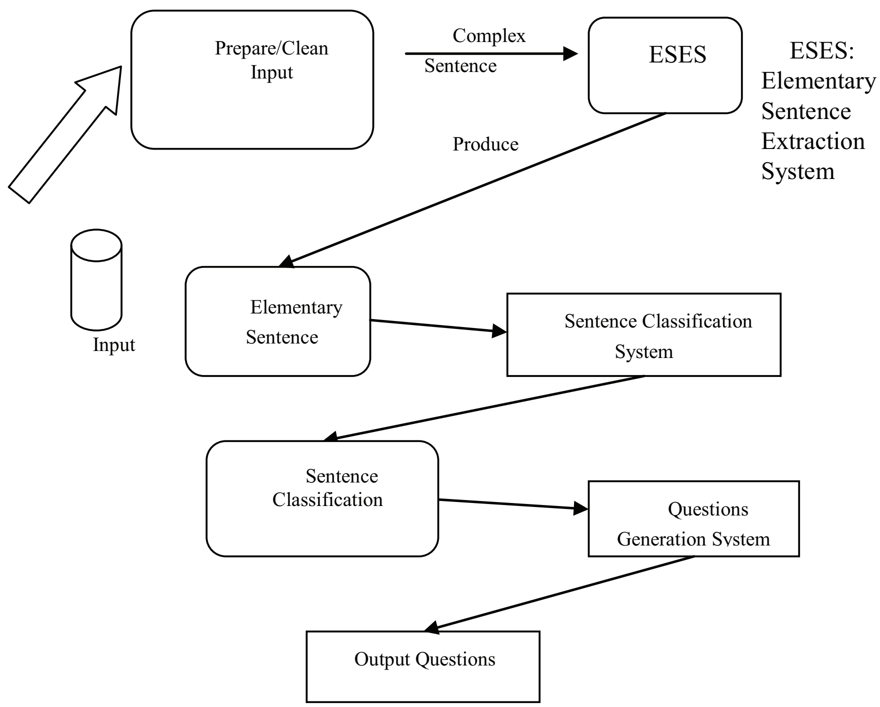

Fig. 1. Overview diagram for the QG system

We divide the work tasks into four modules:

3.1 Data Preparation

An important initial part of any NLP task is cleaning and processing the raw data. We remove the redundant tags and text from the document that has the input sentences that is provided by (QGSTEC). To tokenize the sentences we use the Oak system[1]. The system opens the file that contains the sentences, and reads the sentences. Then, we pass the sentences to the Part of Speech (POS) tagger and the Named Entity (NE) tagger. To generate the POS tagged sentence we use the Oak system. We get the information about the verbs and their tenses from the POS tagged sentences. We again use the Oak system to generate the NE tagged sentences. The Oak system provides us with 150 NEs possible types such as: PERSON, LOCATION, ORGANIZATION, DATE, TIME, MONEY, COUNT, etc. Oak system does not just recognize and label proper nouns, it also is able to recognize common nouns such as

[1] http://nlp.cs.nyu.edu/oak/

school, ship and drug. Oak also can label offences such as first-degree murder and position titles such as King, CEO and president

3.2 Elementary Sentence Construction

The provided sentences by (QGSTEC), may include complex grammatical structure with embedded clauses. Because of this to generate more accurate questions, we extracted the elementary sentences from the complex sentences. To attain this we syntactically parse each complex sentence. Syntactic parsing is analyzing a sentence using the grammar rules. We use Charniak parser[2]. This module constructs a syntactic tree representation, from the bracketed representation of the parsed sentence. While building the tree process, we construct 3 arrays, one for the Noun Phrases (NPs), one for the Verb Phrases (VPs) and one for the Prepositions (PPs) with their location in the tree, a fourth array is generated, from the tree to represent the depth first sequence of the tree nodes and leaves structure. We combine the NPs with the VPs and PPs by reading the NPs till the scope of the VPs and the PPs that are in the VPs scope and thus, we get the elementary sentences. The depth of the first sequence helps us to determine if the phrases to be joined are sequentially correct with the respect of the sentence structure. As an example, “Tom eats an apple and plants a tree”. The depth of the first sequence check will prevent the system from generating the elementary sentence, “Tom plant an apple” since the verb plant came after the noun apple.

3.3 Sentence Classification

In this module the input is the elementary sentences. Using the syntactic parser to parse the elementary sentence, and based on the associated POS and NE tagged information, we get from each elementary sentence the subject, object, preposition and verb. This information is used to classify the sentences. This module has two simple classifiers. The first classifies the sentence into fine classes (Fine Classifier) and the second classifies the sentences into coarse classes (Coarse Classifier). This approach is similar to the one that was described in the Journal of Natural Language Engineering [2] but opposite to the approach that was used by Li & Roth (2006). The second classifier will use the first classifier where candidate labels are generated by reducing the set of taken fine classes from the first into a set of coarse classes. This set is treated as the confusion set for the first classifier. OAK system has 150 named entity types that can be tagged. They are included in a hierarchy. This information is used to make candidate fine and coarse classes. We define the five major coarse classifications as:

1. Human: This will have any subject that is the name of a person.
2. Entity: This includes animals, plant, mountains and any object.

[2] Available at ftp://ftp.cs.brown.edu/pub/nlparser/

3. Location: This will be the words that represent locations, such as country, city, school, etc.
4. Time: This will be any time, date or period such as year, Monday, 9 am, last week, etc.
5. Count: This class will hold all the counted elements, such as 9 men, 7 workers, measurements like weight and size, etc.

Organizations which include companies, institutes, government, market, etc are all a type of category Entity in our classification. Once the sentence words have been classified to coarse classes, we consider the relationship between the words in the sentence. As an example, if the sentence has the structure “Human Verb Human”, it will be classified as “whom and who” question types. If it is followed by a preposition that represents date, then we add the “When” question type to its classification.

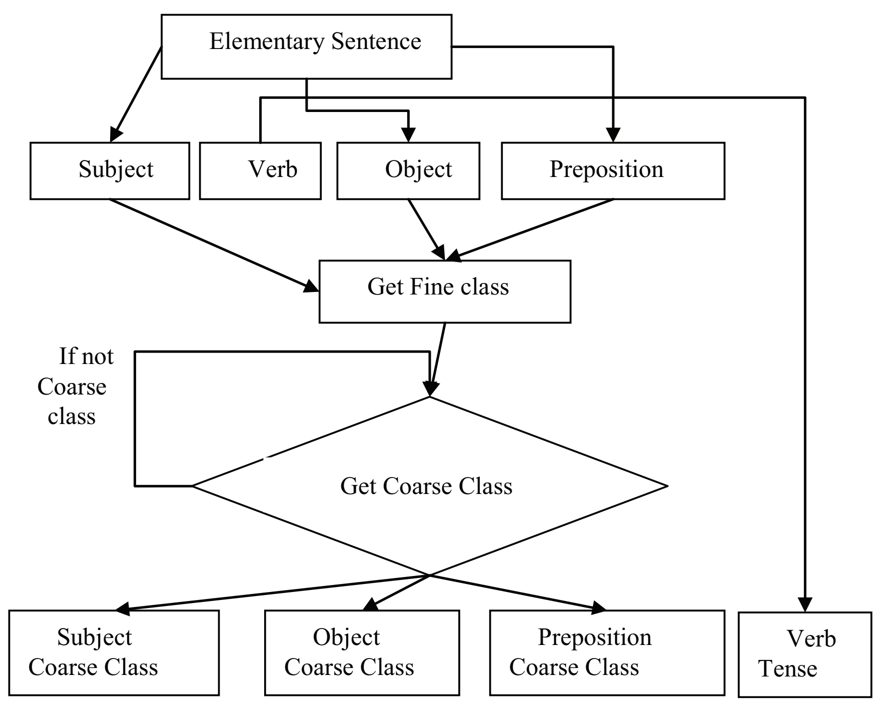

Fig. 2. Coarse Classes and Fine Classes classification diagram

In Fig. 3 we show a sample of the process of classifying a Named Entity type into coarse class Entity.

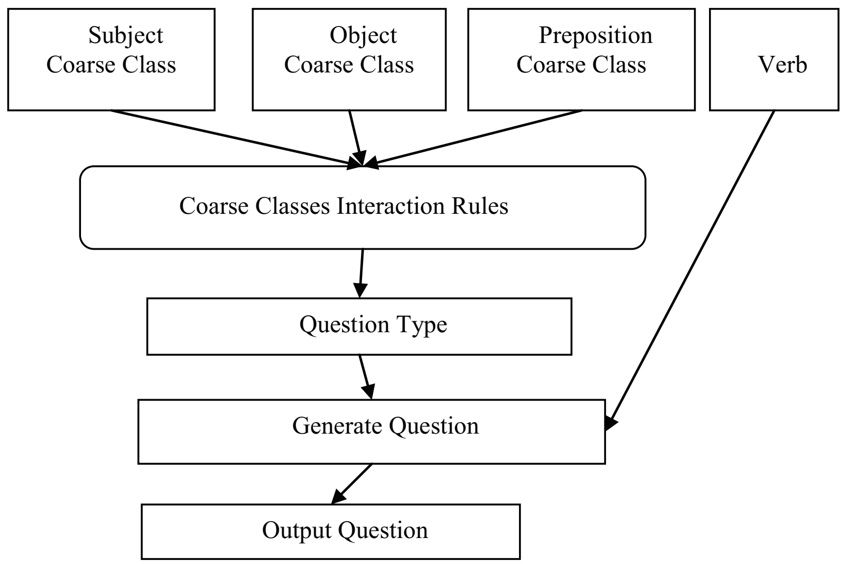

Fig. 3. Question Generation process diagram

3.4 Question Generation

This module takes the elements of the sentences with their coarse classes, the verbs (with its stem) and the tense information. Based on a set of 90 predefined interaction rules, we check the coarse classes according to the word to word interaction.
For example:

Tom ate an orange at 7 pm

Tom is a subject of coarse class Human
An orange is an object of type Entity
At 7 pm is a preposition of type Time

Sample generated questions based on the rule "Human Entity Time" will be:

Who ate an orange?
Who ate an orange at 7 pm?
What did Tom eat?
When did Tom eat an orange?

A sample of an interaction rules is shown in Table 1

Core classes: H = Human E= Entity L= Location T=Time C=Count

Relations			Question type	Example Questions
Subject	Object	Preposition		
H	H	-	Who	Who teach Tom?
			Whom	Whom Sam teaching?
			What	What did Sam do to Tom?
H	H	L	Who	Who teach Tom?
			Whom	Whom Sam teaching?
			What	What did Sam do to Tom?
			Where	Where did Sam teach Tom?
H	L	T	Who	Who study at U of L?
L	H		Where	Where does Sam study?
			When	When did Sam study at U of L?
C	C	-	How many	How many farmers plant 10 trees?
			How many	How many trees did the 10 farmers plant?
E	E	L	Who	Who bought IBM?
			What	What the rabbit eat?
			Where	Where did the rabbit eat the carrot?

Table 1. Sample interaction rules

4 Evaluation Results

For the evaluation of the system, we ran it against the development data provided by (QGSTEC). The development data were provided in the format of a single sentence and a specific target question type (E.g. WHO? WHY? HOW? WHEN? etc). Sentences were selected from the primary data sources for the QGSTEC (Wikipedia, Yahoo! Answers and OpenLearn). In warrants mentioning that extremely long or short sentences were avoided. We used the questions provided by (QGSTEC) for each sentence from different types of possible questions for the sentence. We then employed the widely used evaluation measure: Recall. We define Recall as follows:

$$Recall = \frac{\text{Qg} \cap \text{Qa}}{\text{Qa}}$$

Where, Qg is the number of question generated by our QG system and Qa is the number of actual questions provided by (QGSTEC). With consideration for the fact that humans sometimes generate questions that are worded differently than the sentence structure, results can be seen in Table 2 which represents the recall for the different types of questions that the system was able to generate. We also included the overall recall for the system results:

Type	Qg	Qa	Recall
What	14	36	0.389%
Which	3	30	0.100%
Who	15	25	0.600%
Where	7	23	0.304%
When	11	29	0.379%
How	3	21	0.143%
Why	2	8	0.250%
Yes/No	2	9	0.222%
Over all Recall	57	181	0.315%

Table 2. Evaluation Results

We found that type “Who” had the highest recall, while types “What, Who, Where and When” were in a closed range between above 0.300.

However the other types were not as good, since they had a recall below 0.300%.

The overall recall for the results was 0.315%, which can be improved if we included the semantically parsed sentences, in the process of generating the elementary sentences, and the generation of the questions to adopt the possibilities of wording the questions differently while preserving the meaning.

The noise that was generated was high considering the provided sample questions. The noise was due to both grammatical incorrectness and questions that were generated, which are not included in the dataset provided, but the grammatically correct generated questions were high.

We also used the other widely used method of evaluation which is precision. We calculated the precision for the factoid questions types, and we found that it is better to use the other widely used method of evaluation measure: Precision. We define Precision as follow:

$$Precision = \frac{\mathrm{Qg} \cap \mathrm{Qr}}{\mathrm{Qr}}$$

We conducted experiment with 20 sentences for the precision evaluation due to the human effort needed for this kind of evaluation method.

Question Type	Qr	Qg ∩ Qr	Precision
What	49	19	0.388
Which	31	7	0.226
Who	70	45	0.643
Where	55	31	0.564
When	53	27	0.509
How many/ How Much	43	17	0.395

Table 3. Evaluation results for the different types of factoid questions that were generated by our system using Precision

Question Type	Qg	Qg ∩ Qr	Precision
Overall Factoid	301	146	0.485

Table 4. Overall evaluation results for the different types of factoid questions that were generated by our system using Precision

5 Conclusion and Future Work

In this paper, we proposed an approch to automatically generate questions given a sentence. We used the development data given by (QGSTEC) and evaluated our system using Recall. We used human effort to evaluate the system. We extracted elementary sentences from complex sentences using syntactic information and classified the elementary sentences. We generated questions based on the subject, verb, object and prepostion using a predefined interaction rules. We plan to extend the number of interaction rules. We will also focus on the sentence classification module to make it more robust. Since human generated questions tend to have words with different meanings and senses, the system can be improved with the inclusion of semantic information and word sense disambiguation. We hope to develop further mechanisms to QG based on the dependency features of the answer and its finding [2][4].

6 References

1. Andrenucci, A. & Sniders, E. (2005). Automated Question Answering: Review of the Main Approaches. In Proceedings of the 3rd International Conference on Information Technology and Applications (ICITA'05), Sydney, Australia.
2. Li, X. & Roth, D. (2006). Learning Question Classifiers: The Role of Semantic Information. Jurnal of Natural Language Engineering, 12(3), 229-249
3. Mcgough, J., Mortensen, J., Johnson, J. & Fadali, S. (2001). A Web-based Testing System with Dynmic Question Generation. In ASEE/IEEE Frontiers in Education Conference.

4. Pinchak, C. & Lin, D. (2006). A Probabilistic Answer Type Model. In Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics, p. 393-400.
5. Rus, V. & Grasser, A. C. (2009). The Question Generation Shared Task and Evaluation Challenge. In Workshop on the Question Generation Shared Task and Evaluation Challenge, Final Report, The University of Memphis: National Science Foundation, http://www.cs.memphis.edu/~vrus/questiongeneration/TheQuestionGenerationSharedTaskAndEvaluationChallenge.pdf.
6. Leung, H., Li, F. & Lau, R. Advances in Web Based Learning - ICWL 2007: 6th International Conference

Question Generation with Minimal Recursion Semantics

Xuchen Yao† and Yi Zhang†‡

†Department of Computational Linguistics & Phonetics, Saarland University
‡LT-Lab, German Research Center for Artificial Intelligence
Saarbrücken, Germany
{xuchen,yzhang}@coli.uni-saarland.de

Abstract. This paper proposes a semantics-based approach to question generation by transforming the Minimal Recursion Semantics representation of declarative sentences to that of interrogative sentences. Sentences are first decomposed into smaller units on the semantic level. Then the semantic representation of target words are mapped to that of question words. Finally generation is realized through a linguistically deep English grammar. A prototype system is developed to verify the feasibility.

Key words: question generation, deep linguistic grammar, Minimal Recursion Semantics.

1 Introduction

Question Generation (QG) is the task of generating reasonable questions from a text. In terms of target complexity, the types of QG can be divided into *deep* QG (with deep questions, such as *why, what-if, how* questions) and *shallow* QG (with shallow questions, such as *who, what, when, where, which, how many/much, yes/no* questions) ([12]).

Different systems have been proposed or implemented to facilitate QG research and applications. These systems can be divided into mainly three categories: template-based ([9]), syntax-based ([14], [8]) and semantics-based ([13]). Template-based approaches are mostly suitable for applications with a special purpose, which sometimes comes within a closed-domain. The tradeoff between coverage and cost is hard to balance because human labors must be involved to produce high-quality templates. Syntax-based approaches are rather effective, especially for short sentences. The whole generation is based on tree nodes matching and operation. All operations are straight-forward from a syntactic point of view. However, the computer does this without knowing any underlying meaning of the transformed sentence. Also, it sometimes generates ungrammatical questions, which come from the surface realization of transformed trees thus do not always guarantee grammaticality.

While the first two kinds have already been applied, the third approach is theoretically more interesting and practically challenging. Following [13], we propose a semantics-based method of transforming the Minimal Recursion Semantics (MRS, [4]) representation of declarative sentences to that of interrogative