Overview of The First Question Generation Shared Task Evaluation Challenge

Vasile Rus[1], Brendan Wyse[2], Paul Piwek[2], Mihai Lintean[1], Svetlana Stoyanchev[2] and Cristian Moldovan[2]

[1] Department of Computer Science/Institute for Intelligent Systems, The University of Memphis, Memphis, TN, 38152, USA
{vrus,mclinten,cmoldova}@memphis.edu
[2] Centre for Research in Computing, Open University, UK
bjwyse@gmail.com and {p.piwek, s.stoyanchev}@open.ac.uk

Abstract. The paper describes the First Shared Task Evaluation Challenge on Question Generation that took place in Spring 2010. The campaign included two tasks: Task A – Question Generation from Paragraphs and Task B – Question Generation from Sentences. Motivation, data sets, evaluation criteria, and guidelines for annotators are presented for both tasks.

Keywords: question generation, shared task evaluation campaign.

1 Introduction

Question Generation is an essential component of learning environments, help systems, information seeking systems, multi-modal conversations between virtual agents, and a myriad of other applications (Lauer, Peacock, and Graesser, 1992; Piwek et al, 2007).

Question Generation has been recently defined as the task (Rus & Graesser, 2009) of automatically generating questions from some form of input. The input could vary from information in a database to a deep semantic representation to raw text.

The first Question Generation Shared Task Evaluation Challenge (QG-STEC) follows a long tradition of STECs in Natural Language Processing: see various tracks at the Text REtrieval Conference (TREC; http://trec.nist.gov), e.g. the Question Answering track, the semantic evaluation challenges under the SENSEVAL umbrella (www.senseval.org), or the annual tasks run by the Conference on Natural Language Learning (CoNLL; http://www.cnts.ua.ac.be/conll/). In particular, the idea of a QG-STEC was inspired by the recent activity in the Natural Language Generation (NLG) community to offer shared task evaluation campaigns as a potential avenue to provide a focus for research in NLG and to increase the visibility of NLG in the wider Natural Language Processing (NLP) community (White and Dale, 2008). It should be noted the QG is currently perceived as a discourse processing task rather than a traditional NLG task (Rus & Graesser, 2009).

Two core aspects of a question are the goal of the question and its importance. It is difficult to determine whether a particular question is good without knowing the context in which it is posed; ideally one would like to have information about what counts as important and what the goals are in the current context. This suggests that a STEC on QG should be tied to a particular application, e.g. tutoring systems. However, an application-specific STEC would limit the pool of potential participants to those interested in the target application. Therefore, the challenge was to find a framework in which the goal and importance are intrinsic to the source of questions and less tied to a particular context/application. One possibility was to have the general goal of asking questions about salient items in a source of information, e.g. core ideas in a paragraph of text. Our tasks have been defined with this concept in mind. Adopting the basic principle of application-independence has the advantage of escaping the problem of a limited pool of participants (to those interested in a particular application had that application been chosen as the target for a QG STEC).

Besides the advantage of a larger pool of potential participants, an application-independent QG STEC would provide a more fair ground for comparison as teams already working on a certain application would not be advantaged as would be the case if the application had been the focus of a STEC. It should be noted that the idea of an application-independent STEC is not new. An example of an application-independent STEC would be generic summaries (as opposed to query-specific summaries) in summarization.

Another decision aimed at attracting as many participants as possible and promoting a more fair comparison environment was the input for the QG tasks. A particular semantic representation would have provided an advantage to groups already working with it and at the same time it would have raised the barrier-to-entry for newcomers. Instead, we have adopted a second guiding principle for the first QG-STEC tasks: no representational commitment. That is, we wanted to have as generic an input as possible. The input to both task A and B in the first QG STEC is raw text.

Task A and B described here fall in the Text-to-Question category of QG tasks identified by The First Workshop on Question Generation (www.questiongeneration.org). The first workshop identified four categories of QG tasks (Rus & Graesser, 2009): Text-to-Question, Tutorial Dialogue, Assessment, and Query-to-Question. Using another categorization, tasks A and B are part of the Text-to-text Natural Language Generation task categories (Dale & White, 2007).

It is important to say that the two tasks offered in the first QG STEC were selected among 5 candidate tasks by the members of the QG community. A preference poll was conducted and the most preferred tasks, Question Generation from Paragraphs (Task A) and Question Generation from Sentences (Task B), were chosen to be offered in the first QG STEC. The other three candidate tasks were: Ranking Automatically Generated Questions (Michael Heilman and Noah Smith), Concept Identification and Ordering (Rodney Nielsen and Lee Becker), and Question Type Identification (Vasile Rus and Arthur Graesser).

There is overlap between Task A and B in the first QG STEC. This was intentional with the aim of encouraging people preferring one task to participate in the other. The overlap consists of the specific questions in Task A which are more or less similar with the type of questions targeted by Task B.

Overall, we had 1 submission for Task A and 4 submissions for Task B. The submissions are currently evaluated through a peer-review system for Task B. Task A is evaluted by two external judges as there was only one submission and the peer-review mechanism cannot be applied.

2 TASK A: Question Generation from Paragraphs

1.1 Task Definition

The Question Generation from Paragraphs (QGP) task challenged participants to generate a list of 6 questions from a given input paragraph. The six questions should be at three specificity/scope levels: 1 x broad (entire input paragraph), 2 x medium (one or more clauses or sentences), and 3 x specific (phrase or less). The scope is defined by the portion of the paragraph that answers the question. If multiple questions could be generated at one level, only the specified number should be submitted. That is, if the paragraph answers two broad questions then only one should be submitted at that level.

The Question Generation from Paragraphs (QGP) task has been defined such that it is *application-independent*. *Application-independent* means questions will be judged based on content analysis of the input paragraph.

For this task, questions are considered important if they ask about the core idea(s) in the paragraph. Questions are considered interesting if an average person reading the paragraph would consider them so based on a quick analysis of the contents of the paragraph.

Simple, trivial questions such as *What is X?* or generic questions such as *What is the paragraph about?* were avoided. In addition, implied questions (an example is provided later) were not allowed as the emphasis is on questions triggered and answered by the paragraph. Questions should not be compounded as in *What is … and who … ?* Questions must be grammatically and semantically correct and related to the topic of the given input paragraph. Question types (*who/what/why/…*) generated for each paragraph should be diverse, if possible. Unique question types are preferred in the set of returned questions.

1.2 Guidelines for Human Judges

We show next an example paragraph together with six interesting, application-independent questions that could be generated. We will use the paragraph and questions to describe the judging criteria.

A set of five scores, one for each criterion (specificity, syntax, semantics, question type correctness, diversity)to each question. Composite scores will also be assigned. For instance, there will be a composite score per question. That is, each question will be assigned a composite score ranging from 1 (first/top ranked, best) to 4 (lowest rank), 1 meaning the question is at the right level of specificity given its rank (e.g. the

broadest question that the whole paragraph answers will get a score of 1 if in the first position) and also it is syntactically and semantically correct as well as unique/diverse from other generated questions in the set.

Table 1. Example of input paragraph (from http://en.wikipedia.org/wiki/Abraham_lincoln).

Input Paragraph
Abraham Lincoln (February 12, 1809 – April 15, 1865), the 16th President of the United States, successfully led his country through its greatest internal crisis, the American Civil War, preserving the Union and ending slavery. As an outspoken opponent of the expansion of slavery in the United States, Lincoln won the Republican Party nomination in 1860 and was elected president later that year. His tenure in office was occupied primarily with the defeat of the secessionist Confederate States of America in the American Civil War. He introduced measures that resulted in the abolition of slavery, issuing his Emancipation Proclamation in 1863 and promoting the passage of the Thirteenth Amendment to the Constitution. As the civil war was drawing to a close, Lincoln became the first American president to be assassinated.

Table 2. Examples of questions and scores for the paragraph in Table 1.

Questions	Scope
Who is Abraham Lincoln?	*General*
What major measures did President Lincoln introduce?	*Medium*
How did President Lincoln die?	*Medium*
When was Abraham Lincoln elected president?	*Specific*
When was President Lincoln assassinated?	*Specific*
What party did Abraham Lincoln belong to?	*Specific*

The specificity scores are assigned primarily based on the answer span in the input paragraph. The broadest question is the one whose answer spans the entire paragraph. The most specific question is the one whose answer is less than a sentence: a clause, phrase, word, or collocation. Scores will be assigned based on the following rubric: 1 – input paragraph, 2 – multiple sentences, 3 – a clause or less, 4 – trivial/generic, implied, no question (empty question), or undecided, e.g. a semantically wrong question may not be understood well enough to judge its scope. As we expect six questions as output, if one level is missed we encourage participants to generate questions of a narrower scope. For instance, if a broad-scope question cannot be

generated then a multiple-sentence or a specific question should be submitted. This assures that each participant submits as many as six questions for each input paragraph.

Best question specificity scores for six questions corresponding to an input paragraph would be 1, 2, 2, 3, 3, 3. The best configuration of scores (1, 2, 2, 3, 3, 3) would only be possible for paragraphs that could trigger the required number of questions at each scope level, which may not always be the case.

While the initial plan was for the judges to look at the question itself and select themselves the portion of the paragraph that may have triggered the question we opted instead, for practical reasons, to allow the judges to see the span of text submitted by participants for each question and decide based on the span the specificity of the question. The advantage of the initial plan is that the judges' selected text span could be automatically compared to participants' for an automated scoring process.

The syntactic correctness will be judged using the following scores: 1 – grammatically correct and idiomatic/natural, 2 – grammatically correct, 3 – some grammar problems, 4 – grammatically unacceptable.

The semantic correctness will be judged using the following scores: 1 – semantically correct and idiomatic/natural, 2 – semantically correct and close to the text or other questions, 3 – some semantic issues, 4 – semantically unacceptable.

Correctness of question type means the specified type by a participant is agreed upon by the judge. This is a binary dimension: 0 – means the judge agrees with the specified answer type, 1 – means the judge disagrees.

Diversity of question types was also be evaluated. At each scope level, ideally, each question will have a different question type. A question type is loosely defined as being formed by the question word (e.g., *wh*-word or auxiliary) and by the head of the immediately following phrase. For instance, in *What U.S. researcher* …. ? the head of the phrase *U.S. researcher that* follows the question word *What* indicates a person which means the question is actually a *Who* question and not a *What* question. Preference will be given to diversity of question words though. Full question types, i.e. including the head of the phrase following the question word, will be considered in special cases when the use of diverse question words is constrained by the input paragraph, that is, when different question words are hard to employ in order to generate different question types. For instance, some paragraphs may facilitate the generation of true *What* questions, i.e. *What* question types, but not *When* questions. For diversity ratings, we will use the following rubric: 1 – diverse in terms of question type and main body, 2 – diverse in terms of main body, 3 – paraphrase of a previous question, and 4 – similar-to-identical to a previous question.

While full diversity would be ideal, it can be quite challenging for some input paragraphs. For pragmatic reasons, we relaxed the diversity criteria. We assigned the highest score of diversity if at least 50% of the question types in the whole 6-question set are different and the distribution of types is balanced, e.g. 2-*Who*, 2-*What*, and 2-*Where* would be scored higher than 4-*Who*, 1-*What*, and 1-*Where*.

Diversity of body will be evaluated also in terms of answer scope by asking judges to highlight in the input paragraph the fragments that constitute the answer to the question according to their opinion.

We also defined composite scores. In general, composite scores are the average of individual composite scores. An individual composite score summarizes the scores

along a dimension, e.g. syntactic correctness, and is computed by taking the average of individual scores shown by the formula below where Q is the number of questions.

$$Syntactic-overall-score = \frac{\sum_{|Q|} individual\_score}{|Q|}$$

The composite score for specificity is more challenging to define. The goal would be to have a summative score with values from 1 to 4. 1 should be assigned to perfect system that generates 6 questions for each paragraph with the required distribution of specificity levels: 1 general, 2 medium, and 3 specific. For instance, if there are 4 specific questions in a set of 6 questions, then when judging the fourth specific question it will be penalized because the number of expected specific questions (3) have been exhausted and another question at general or medium scope has not been generated. As of this writing, we are refining our composite score for specificity.

1.3 Data Sources and Annotation

The primary source of input paragraphs were: Wikipedia, OpenLearn, Yahoo!Answers. We collected 20 paragraphs from each of these three sources. We collected both a development data set (65 paragraphs) and a test data set (60 paragraphs). For the development data set we manually generated and scored 6 questions per paragraph for a total of 6 x 65 = 390 questions.

Paragraphs were selected such that they are self-contained (no need for previous context to be interpreted, e.g. will have no unresolved pronouns) and contain around 5-7 sentences for a total of 100-200 tokens (excluding punctuation). In addition, we aimed for a diversity of topics of general interest.

We decided to provide minimal annotation for input in order to allow individual participants to choose their own preprocessing tools. We did not offer annotations for lemmas, POS tags, syntactic information, or PropBank-style predicate-argument structures. This linguistic information can be obtained with acceptable levels of accuracy from open-source tools. Furthermore, this favors comparison of full systems in a black-box manner as opposed to more specific components. We only provided discourse relations based on HILDA, a freely available automatic discourse parser (duVerle & Prendinger, 2009).

2 TASK B: Question Generation from Sentences

2.1 Task Definition

Participants were given a set of inputs, with each input consisting of:

- a single sentence and

- a specific target question type (e.g., WHO?, WHY?, HOW?, WHEN?; see below for the complete list of types used in the challenge).

For each input, the task was to generate 2 questions of the specified target question type.

Input sentences, 60 in total, were selected from OpenLearn, Wikipedia and Yahoo! Answers (20 inputs from each source). Extremely short or long sentences were not included. Prior to receiving the actual test data, participants were provided with a development data set consisting of sentences from the aforementioned sources and, for one or more target question types, examples of questions. These questions were manually authored and cross-checked by the team organizing Task B.

The following three examples are taken from the development data set, one example from OpenLearn, Wikipedia and Yahoo! Answers. Each instance has a unique identifier and information on the source it was extracted from. The <text> element contains the input sentence and the <question> elements contain possible questions. The <question> element has the type attribute for specification of the target question type.

```
<instance id="3">
        <id>OpenLearn</id>
        <source>A103_5</source>
        <text>
                The poet Rudyard Kipling lost his only son
                in the trenches in 1915.
        </text>
        <question type="who">
                Who lost his only son in the trenches in 1915?
        </question>
        <question type="when">
                When did Rudyard Kipling lose his son?
        </question>
        <question type="how many">
                How many sons did Rudyard Kipling have?
        </question>
</instance>

<instance id="46">
        <id>Wikipedia</id>
        <source>Igneous_rock</source>
        <text>
                Two important variables used for the classification of
                igneous rocks are particle size, which largely depends
                upon the cooling history, and the mineral composition
                of the rock.
        </text>
        <question type="which">
                Which two important variables are used for the
```

classification of igneous rocks?
</question>

<instance id="77">
<id>**YahooAnswers**</id>
<source>
http://answers.yahoo.com/question/index;_ylt=AoWV
WMdwLigujQeRxf0LHWCD6xR.;_ylv=3?qid=201002200
15521AA0slZo
</source>
<text>
In Australia you no longer can buy the ordinary
incandescent globes, as you probably already know.
</text>
<question type="where">
Where can you no longer buy the ordinary incandescent globes?
</question>
<question type="yes/no">
Can you buy the ordinary incandescent globes in Australia?
</question>
<question type="what">
What can you no longer buy in Australia?
</question>
</instance>

Note that input sentences were provided as raw text. Annotations were not provided. There are a variety of NLP open-source tools available to potential participants and the choice of tools and how these tools are used was considered a fundamental part of the challenge.

Participants were also provided with the following list specifying the target question types:

- WHO?: The answer to the generated question is a person (e.g. Abraham Lincoln) or group of people (e.g. the American people) named in the input sentence.
- WHERE?: The answer to the generated question is a placename (e.g. Dublin, Mars) or location (North-West, to the left of) which is contained in or can be derived from the input sentence.
- WHEN?: The answer is a specific date (e.g. 3rd July 1973, 4th July), time (e.g. 2:35, 10 seconds ago), era or other representation of time.
- WHICH?: The answer will be a member of a category (e.g. Invertebrate or Vertebrate) or group (e.g. Colours, Race) or a choice of entities (e.g. Union or Confederacy) given in the input sentence.
- WHAT?: The question might describe a specific entity mentioned in the input sentence and ask what it is. The question may also ask the purpose, attributes or relations of an entity as described in the input sentence.

- WHY?: The question asks the reasoning behind some statement made in the input sentence
- HOW MANY/LONG?: The answer will be a duration of time or range of values (e.g. 2 days) or a specific count of entities (e.g. 32 counties) within the input sentence.
- YES/NO: The generated question should ask whether a fact contained in the input sentence is either true or false (e.g. Are mathematical co-ordinate grids used in graphs?).

2.2 Evaluation criteria for System Outputs and Human Judges

The evaluation criteria fulfilled two roles. Firstly, they were provided to the participants as a specification of the kind of questions that their systems should aim to generate. Secondly, they also played the role of guidelines for the judges of system outputs in the evaluation exercise.

For this task, five criteria were identified:

- Relevance
- Question Type
- Syntactic Correctness and Fluency
- Ambiguity
- Variety

All criteria are associated with a scale from 1 to *N* (where *N* is 2, 3 or 4), with 1 being the best score and *N* the worst score.

The criteria are defined as follows:

Relevance

Questions should be relevant to the input sentence. This criterion measures how well the question can be answered based on what the input sentence says.

Table 3. Scoring rubric for relevance.

Rank	***Description***
1	The question is completely relevant to the input sentence.
2	The question relates mostly to the input sentence.
3	The question is only slightly related to the input sentence.
4	The question is totally unrelated to the input sentence.

Question Type

Questions should be of the specified target question type.

Table 4. Scoring rubric for Question Type.

Rank	***Description***
1	The question is of the target question type.
2	The type of the generated question and the target question type are different.

Syntactic Correctness and Fluency

The syntactic correctness is rated to ensure systems can generate grammatical output. In addition, those questions which read fluently are ranked higher.

Table 5. Scoring rubric for syntactic Correctness and Fluency.

Rank	***Description***	***Example***
1	The question is grammatically correct and idiomatic/natural.	In which type of animals are phagocytes highly developed?
2	The question is grammatically correct but does not read as fluently as we would like.	In which type of animals are phagocytes, which are important throughout the animal kingdom, highly developed?
3	There are some grammatical errors in the question.	In which type of animals **is** phagocytes, which are important throughout the animal kingdom, highly developed?
4	The question is grammatically unacceptable.	**On** which type of animals **is** phagocytes, which are important throughout the animal kingdom, developed?

Ambiguity

The question should make sense when asked more or less out of the blue. Typically, an unambiguous question will have one very clear answer.

Table 6. Scoring rubric for Ambiguity.

Rank	*Description*	*Example*
1	The question is unambiguous.	Who was nominated in 1997 to the U.S. Court of Appeals for the Second Circuit?
2	The question could provide more information.	Who was nominated in 1997?
3	The question is clearly ambiguous when asked out of the blue.	Who was nominated?

Variety

Pairs of questions in answer to a single input (i.e., with the same target question type) are evaluated on how different they are from each other. This rewards those systems which are capable of generating a range of different questions for the same input.

Table 7. Scoring rubric for Variety.

Rank	*Description*	*Example*
1	The two questions are different in content.	Where was X born?, Where did X work?
2	Both ask the same question, but there are grammatical and/or lexical differences.	What is X for?, What purpose does X serve?
3	The two questions are identical.	

The procedure for applying these criteria is as follows:

- Each of the criteria is applied *independently* of the other criteria to each of the generated questions (except for the stipulation provided below).

We need some specific stipulations for cases where no question is returned in response to an input. For each target question type, two questions are expected. Consequently, we have the following two possibilities regarding missing questions:

- *No question is returned for a particular target question type*: for each of the missing questions, the worst score is recorded for all criteria.
- *Only one question is returned*:[1] For the missing question, the worst score is assigned on all criteria. The question that is present is scored following

[1] This includes cases where exactly the same question is returned twice for a target question type. In that case, the second identical question is treated the same way as a missing question.

the criteria, with the exception of the VARIETY criterion for which the lowest possible score is assigned.

The result of applying these criteria to a set of questions *including* the missing questions $q_1,\ldots, q_n$ is a score for each (missing) question q_k ($1 \leq k \leq n$) for each of the criteria:

- ScoreRelevance(q_k)
- ScoreQuestionType(q_k)
- ScoreCorrectness(q_k)
- ScoreAmbiguity(q_k)
- ScoreVariety(q_k)

We compute the overall score on a specific criterion C given $q_1,\ldots, q_n$ as follows:

$$\text{OverallScoreC}(q_1,\ldots, q_n) = \sum_{k=1}^{n} \text{ScoreC}(q_k)$$

This way we can, for example, rank systems on the Relevance criterion: the system with the lowest OverallScoreRelevance is ranked first.

We can also compute a score which aggregates the overall scores for the criteria. Here we, however, need to be careful since some criteria are more important than others. For example, if a generated question is irrelevant or of the wrong question type, the scores for correctness, ambiguity and variety don't really matter. Otherwise, a system could achieve high scores through the trivial strategy of always generating the *same* pair of correct/fluent, unambiguous and different questions.

Thus, we propose to calculate the aggregate score as follows:

- For each individual question, if the score for Relevance is 4 or the score for Question Type is 2, revise the score for all the other criteria to the worst possible score.
- Now, compute the OverallScoreC for each criterion C and add these together to obtain the aggregate score.

Of course, we acknowledge that the computation of the aggregate score does have an element of arbitrariness. For instance, we could, alternatively, have assigned weights to the different criteria and used those in the calculation of the aggregate score (then again, the precise choice of the weights would be a further contentious issue). To take this into account, when reporting the Task B results, we will not just return a single total score for each system, but rather provide a profile for each of the evaluated systems and systematic comparisons between them. This way we hope to provide a better insight into which aspects of QG each of the systems is good at.

Conclusions

The submissions to the first QG STEC are being evaluated as of this writing using peer-review mechanism in which participants blindly evaluate their peers questions.

At least two reviews per submissions are performed with the results to be made public at the 3rd Workshop on Question Generation that will take place in June 2010.

Acknowledgments. We are grateful to a number of people who contributed to the success of the First Shared Task Evaluation Challenge on Question Generation: Rodney Nielsen, Amanda Stent, Arthur Graesser, Jose Otero, and James Lester. Also, we would like to thank the National Science Foundation who partially supported this work through grants RI-0836259 and RI-0938239 (awarded to Vasile Rus) and the Engineering and Physical Sciences Research Council who partially supported the effort on Task B through grant EP/G020981/1 (awarded to Paul Piwek). The views expressed in this paper are solely the authors'.

References

1. Lauer, T., Peacock, E., & Graesser, A. C. (1992) (Eds.). *Questions and information systems*. Hillsdale, NJ: Erlbaum.
2. Rus, V. and Graesser, A.C. (2009). *Workshop Report: The Question Generation Task and Evaluation Challenge*, Institute for Intelligent Systems, Memphis, TN, ISBN: 978-0-615-27428-7.
3. Piwek, P., H. Hernault, H. Prendinger, M. Ishizuka (2007). T2D: Generating Dialogues between Virtual Agents Automatically from Text. In: Intelligent Virtual Agents: Proceedings of IVA07, LNAI 4722, September 17-19, 2007, Paris, France, (Springer-Verlag, Berlin Heidelberg) pp.161-174
4. Dale, R. & M. White (2007) (Eds.). *Position Papers of the Workshop on Shared Tasks and Comparative Evaluation in Natural Language Generation*.
5. duVerle, D. and Prendinger, H. (2009). A novel discourse parser based on Support Vector Machines. Proc 47th Annual Meeting of the Association for Computational Linguistics and the 4th Int'l Joint Conf on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP'09), Singapore, Aug 2009 (ACL and AFNLP), pp 665-673.