Question Generation: Proposed Challenge Tasks and Their Evaluation

Rodney D. Nielsen
Boulder Language Technologies
Center for Computational Language and Education Research, University of Colorado, Boulder
Rodney.Nielsen@Colorado.edu

Abstract

We propose a core task for question generation intended to maximize research activity and a subtask to identify the key concepts in a document for which questions should be generated. We discuss how these tasks are affected by the target application, discuss human evaluation techniques, and propose application-independent methods to automatically evaluate system performance.

1 Introduction

The nature of automatic question generation is different depending on the application within which it is embedded. If the purpose is educational assessment, the questions are intended to evaluate the respondent's knowledge, understanding and skills in a subject area. Whereas, if the intent of the questions is to facilitating learning, such as in a Socratic tutoring environment, then they should lead students to an "aha" moment, where they understand a concept that they previously did not. Optimally, question generation should be defined and evaluated in the context of the application requiring it. For example, in Intelligent Tutoring Systems, given the learner model, learner goals, and a context of prior interactions, the objective is to choose the next topic, question type, and surface form in a way that maximizes learning, which should then be evaluated based on learning gains. The typical types of questions generated by each of these systems are different and this must be considered when designing a question generation task. Even within a given application, the question types that are appropriate vary by many factors, such as the student's age in assessment applications.

To maximize participation a question generation challenge must constrain and define the task in a way that maximizes its relevance to the many disparate groups. Where this is not possible, there must ultimately be parallel tracks and, in the short term, these aspects of the task should focus on the largest participant group. In the following sections, we describe tasks that should be relevant to most potential participants and automatic evaluation techniques that are application independent.

2 Defining the Question Generation Tasks

Most applications utilizing question generation can be conceived of as dialog systems, where the question generated will depend not only on the text, but also the context of all previous interactions. Even assessment ultimately should adapt to the student's performance on previous questions. Given a dialog context, we view question generation as a three step process. In the first step, *Concept Selection*, the topic from which a question is to be generated is identified. In the second (not necessarily subsequent) step, *Question Type Determination*, a decision is made about the type of question to be asked. In the final step, *Question Construction*, the surface form of the question is created based on the prior steps. Since these tasks are largely separable, we propose they be run as separate challenge tasks.

Ultimately, identifying the most appropriate concept from which to construct the next question in a dialogue and deciding the question type is the most important goal of question generation. While this is a very difficult, context sensitive task, it is reasonable to identify a priori the set of key concepts from which questions are likely to be generated, similar in spirit to Vanderwende's (2007) proposal. However, even if the question types are severely constrained, the concepts selected are application dependent, since what is important to one application may not be to another, necessitating distinct challenge tracks for these tasks.

We believe the majority of applications will require generation from raw text and suggest this as the starting point for a Key Concept Identification task. *Starting with the raw text and the application track, the objective of Key Concept Identification is simply to output an annotation to identify key spans of text (snippets) for which questions are likely to be generated*.

Because the most appropriate type of question does not depend on the text alone, but also the application specific context, we propose the question type be an input to the construction task. Finally, the text itself is a common part of the context across all applications, so it too should be an input. Combined, this leads to the proposal that *the Question Construction task consists of creating a natural language question of a specified type, from specified snippets, given the full text as context*.

3 Evaluation

3.1 Key Concept Identification

Given an annotated test set, the Key Concept Identification task can be evaluated by a fully automatic method. Furthermore, the method is completely independent of the application for which the question generation is being performed.

Our evaluation weights each question snippet equally and is similar in spirit to the F-measure described by Lin and Demner-Fushman (2005) for evaluating question answering. We assume that for each application track, three experts in that area annotate a set of test documents, tagging the spans of text (*snippets*) from which they feel questions should be generated. These snippets are then adjudicated and tagged as vital or optional depending on the number of annotators that marked a similar concept and the significance of the concepts.

Our F-measure bases recall on the coverage of the vital snippets and precision on the extent to which a system tagged snippet is covered by a single human annotated snippet, vital or optional. Let k be the number of vital spans, m be the total number of annotated snippets across all human annotators, n be the total number of system-tagged snippets, $\boldsymbol{V}_i$, $\boldsymbol{A}_i$, and $\boldsymbol{S}_i$ be the set of content words in the i^{th} vital, human-annotated, and system-tagged snippets respectively, and $|\boldsymbol{X}_i|$ be the number of content words in the specified set. Calculate the instance recall for each vital snippet and instance precision for each system-tagged snippet as:

$$IR_i = \max_{j=1..n} \left|\mathbf{V}_i \cap \mathbf{S}_j\right| \Big/ \left|\mathbf{V}_i\right|$$

$$IP_j = \max_{i=1..m} \left|\mathbf{S}_j \cap \mathbf{A}_i\right| \Big/ \left|\mathbf{S}_j\right|$$

Let the overall recall and precision equal the average instance recall and precision and calculate the F-measure as usual.

The procedure described allows different span alignments when calculating *IR* versus *IP* and in some cases, multiple alignments for a single span. It could be revised to find the single alignment that maximizes the overall F-measure, but this is probably not worth the effort, as it would probably only have a significant effect on the metric for extreme cases.

This task is similar to Automatic Summarization (AS) in that both seek to identify critical information in the source text. However, AS evaluations, such as ROUGE (Lin, 2004), are not adequate, in part because they operate over the full summary, not weighting snippets equally, and they do not differentiate between vital and optional concepts.

3.2 Question Construction

Optimally, the Question Generation task would be evaluated differently depending on the application. Questions for educational assessment might be evaluated according to their discriminating power (Lin and Miller, 2005), tutoring questions for their effect on learning gains, etc.

In the educational assessment track, where most prior work has taken place, we propose a two part human evaluation. First, judges filter out questions that do not match the specified type or topic. Then, the remaining questions are distributed across tests, and the final evaluation would be based on the average discriminating power of the questions, assigning questions filtered out due to type or content errors the lowest power possible, -1.0. It is currently impractical to optimally evaluate tracks, such as tutoring. Here, we suggest evaluating the systems based on average question ratings from appropriate experts, (assessment questions could also be evaluated in this fashion by, e.g., experts from the Educational Testing Service).

If the question type and source text snippet are provided as an input, then questions are likely to look very similar regardless of the application, especially in the early years of question generation. Therefore, we propose an automatic evaluation

technique that compares the system-constructed question to one or more gold standard questions written by application experts. This form of evaluation, which is consistent with the proposal of Rus et al. (2007) and common in other areas such as Machine Translation (MT) and Automatic Summarization (AS), generally involves comparing overlap in n-grams. Soricut and Brill (2004) provide a unified framework for automatic evaluation using n-gram co-occurrence statistics, which in part relates evaluation factors (faithfulness, compactness, precision and recall) to the size of n-grams. MT typically utilizes up to 4-grams to ensure fluency; whereas, AS, which usually comprises selecting already syntactically sound key sentences is often evaluated strictly by unigrams, since fluency is essentially guaranteed. Question Generation might best be evaluated by bigram overlap, since it often involves the use of syntactically valid question stems and or the extraction of syntactically valid key phrases from the text, but does involve more syntactic composition than AS. This conjecture must be tested empirically.

Nielsen et al. (2008) propose a *facet*-based representation, derived from dependency parses, that effectively factors out much valid syntactic alternation and focuses near the bigram level. We propose the use of this representation and the corresponding entailment system to automatically evaluate the extent to which a system question is a paraphrase of a gold standard question. Specifically, we propose to use an average F-measure over questions, where a constructed question's F-measure is based on the most similar expert question, with its recall calculated using the probabilities for each facet of the expert question being entailed by the constructed question and its precision calculated from the probabilities of each facet of the generated question being entailed by the expert question. The metric must also penalize questions that include reference answer facets not in the expert question, perhaps by multiplying by one minus the probability the answer is entailed. Otherwise, questions that give away the answer or that simply repeat the source text could result in a very high score.

Precedent for this evaluation includes: Owczarzak et al. (2007) showing a dependency-based metric correlates higher with human judgment on fluency than n-gram metrics, Turian et al. (2003) finding that an F-measure can outperform current precision-focused metrics in similar evaluations, Perez and Alfonseca's (2005) result that MT n-gram-based metrics fall far short in recognizing textual entailment, and Lin and Demner Fushman's (2005) finding that macro-averaging over answers is more appropriate than micro-averaging over answer nuggets in Question Answering evaluation.

A nice compromise between human and automatic evaluation is to have a number of expert questions evaluated by humans and then weight automatic metrics by the quality of the expert question involved in the entailment.

A downside to automatic evaluation is that it will inappropriately penalize the best problem solving questions, all of which are unique. However, this can be addressed in the future, when such question generation becomes more feasible.

Acknowledgments

We thank Wayne Ward, Steve Bethard, Philipp Wetzler and the anonymous reviewers for helpful suggestions. This work was partially funded by IES Award R305B070434.

References

Lin, J and Demner-Fushman D. (2005) Automatically Evaluating Answers to Definition Questions. In *Proc. HLT/EMNLP*.

Lin, CY. (2004). ROUGE: a Package for Automatic Evaluation of Summaries. In *Proc. of the Workshop on Text Summarization Branches Out.*

Lin, RL and Miller, MD. (2005). *Measurement and Assessment in Teaching*. Prentice Hall.

Nielsen, R, Ward, W and Martin, J. (2008). Automatic Generation of Fine-Grained Representations of Learner Response Semantics. In *Proc. ITS*.

Owczarzak, K, van Genabith, J and Way, A. (2007). Dependency-Based Automatic Evaluation for Machine Translation. In *Proc. NAACL/HLT Workshop on Syntax and Structure in Statistical Translation*.

Perez, D and Alfonseca, E. (2005). Application of the Bleu algorithm for recognising textual entailments. *Proc. PASCAL WS Recognizing Textual Entailment*.

Rus, V, Cai, Z and Graesser, AC. (2007). Evaluation in Natural Language Generation: The Question Generation Task. In *Workshop on Shared Tasks and Comparative Evaluation in Natural Language Generation*.

Turian, J, Shen, L and Melamed, ID. (2003). Evaluation of Machine Translation and its Evaluation. In *Proc. of the MT Summit IX*.

Vanderwende, L. (2007). Answering and Questioning for Machine Reading. In *Proc of the 2007 AAAI Spring Symposium on Machine Reading*.