The Importance of Being Important: Question Generation

Lucy Vanderwende
Microsoft Research
Redmond, WA 98075
lucyv@microsoft.com

Abstract

We propose that the task of question generation should incorporate not only measures of grammaticality, but also a measure of the importance of a question automatically generated. Necessarily, importance of a given question can be judged only in context, so we propose that the data for a shared task be larger than a single sentence, data point or statement in a knowledge base. By focusing on the importance of questions, the community will be in a position to address the shortcomings of existing systems that employ question generation.

1 Introduction

The workshop on Question Generation Task and Evaluation seeks to define a shared task for Question Generation. In offering such a shared task, it is hoped that more members of the computational linguistics community will devote time to the area of Natural Language Generation. There is no question that the definition and evaluation of the shared task will very directly shape the direction that Natural Language Generation will take for the next few years. We have seen, for example, in the summarization community, that instead of focusing on information synthesis and coherence, the community exclusively has focused on the subtask of sentence selection, in large part because the evaluation metric Rouge only measures content.

The workshop description offers a definition of question generation that includes factual questions, yes/no-questions, why-question, from inputs that include not only text, but also raw data and knowledge bases.

We would like to offer a specification of the definition, namely, that the focus should be on generating *important* questions. In this way, question generation would encompass not only generating grammatical questions, but also deciding *which* questions should be generated. This idea was first presented in Vanderwende (2007). Including importance in the task of question generation will then draw on the community engaged in discourse planning, in addition to the community for natural language generation. Including importance should also facilitate the application of question generation technology in any of the proposed scenarios, among which information seeking systems, learning environments, etc.

In the next section, we will review some of the work that already uses question generation, highlighting the shortcomings and challenges of each. We find that deciding what text is worth asking questions about remains a great challenge for computational systems, so far. In the final section, we will propose some issues to consider while devising an evaluation methodology for question generation.

2 Automatic Question Generation

Ureel et al. (2005) describes a computer system, *Ruminator*, which learns by reflecting on the information it has acquired and posing questions in order to derive new information. *Ruminator* takes as input simplified sentences in order to focus on question generation rather than handling syntactic complexity; even so, it is reported that even a single sentence generated 2052 questions. The authors note that it is important "to weed out the easy questions as quickly as possible, and use this process to learn more refined question-posing strategies to avoid producing silly or obvious questions in the

first place" (Ureel et al. 2005). A key component that appears to be missing from the system design is an estimation of the utility, or informativeness, of an automatically generated question.

Mitkov and An Ha (2003) describe a computer system for generating multiple-choice questions automatically. Questions are only asked in reference to domain-specific terms, to ensure that the questions are relevant, and sentences must have either a subject-verb-object structure or a simple subject-verb structure. They tested this method on a linguistics textbook and found that 57% were judged worthy of keeping as test items, of which 94% required some level of post-editing.

Schwartz et al. (2004) describes a system for generating questions, in the context of learning aids, which also comprises the NLP components of lexical processing, syntactic processing, logical form, and generation. This system uses summarization as a pre-processing step as a proxy for identifying information that is worth asking a question about. Nevertheless, the authors note that "limiting/selecting questions created by Content QA Generator is difficult" (Schwartz et al. 2004).

Finally, Harabagiu et al. (2005) describes a system "to generate factoid questions automatically from large text corpora" (Harabagiu et al. 2005). User questions were then matched against these pre-processed factoid questions in order to identify relevant answer passages in a Question-Answering system. While no examples of automatically generated questions are provided, this study does report a comparison of the retrieval performance using only automatically generated questions and manually-generated questions: 15.7% of the system responses were relevant given automatically generated questions, while 84% of the system responses were deemed relevant with manually-generated questions. The discrepancy in performance indicates that significant difficulties remain.

3 Evaluating Question Generation

There are at least two questions to be asked when creating a shared task: what is the data, and how will the system output be evaluated.

Based on the discussion above, the ideal data for a shared task would be a unit of text larger than a single sentence, or a set of data, or a fragment or larger segment of a knowledge base. While automated systems might be capable of producing grammatical questions, producing sensible, or, meaningful questions is still challenging. In order to move beyond grammaticality, a larger context is required.

Evaluating automatically generated questions will involve human input. Looking at the examples provided in the studies above, however, it is easy to decide how meaningful a question is and it should not be difficult to demonstrate high inter-annotator agreement between judges. We should note that a human-in-the-loop for evaluation carries the disadvantage that the evaluation metric cannot be used during system development, generally considered necessary for training machine-learned algorithms. However, this disadvantage should be weighed against developing a task that nourishes research directed at establishing importance conveyed by text rather than treating the text as a great morass of facts all of which are equally important.

References

Sanda M. Harabagiu, Andrew Hickl, John Lehmann , and Dan Moldovan. 2005. Experiments with Interactive Question-Answering. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics* (ACL05).

Ruslan Mitkov and Le An Ha. 2003. Computer-Aided Generation of Multiple-Choice Tests. In *Proceedings of the HLT-NAACL 2003 Workshop on Building Educational Applications Using Natural Language Processing*, Edmonton, Canada, May, pp. 17 – 22

Lee Schwartz, Takako Aikawa, and Michel Pahud. 2004. Dynamic Language Learning Tools. In *Proceedings of the 2004 InSTIL/ICALL Symposium*, June 2004.

Leo C. Ureel II, Kenneth D. Forbus , Chris Riesbeck, and Larry Birnbaum. 2005. Question Generation for Learning by Reading. In *Proceedings of the AAAI Workshop on Textual Question Answering*, Pittsburgh, Pennsylvania. July 2005

Lucy Vanderwende. 2007. Answering and Questioning for Machine Reading. In *Proceedings of the 2007 AAAI Spring Symposium on Machine Reading*, Stanford, CA. March 26-28, 2007.