

Ranking Automatically Generated Questions as a Shared Task

Michael HEILMAN, Noah A. SMITH

*Language Technologies Institute, Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA, 15213, USA*

Abstract. We propose a shared task for question generation: the ranking of reading comprehension questions about Wikipedia articles generated by a base overgenerating system. This task focuses on domain-general issues in question generation and invites a variety of approaches, and also permits semi-automatic evaluation. We describe an initial system we developed for this task, and an annotation scheme used in the development and evaluation of our system.

Keywords. question generation, human language technologies

Introduction

Questions are an important component of many educational interactions, from one-on-one tutoring sessions to large-scale assessments. Thus, the automation of generating questions would enable the efficient development of flexible and adaptive instructional technologies. For example, a teacher might use a question generation (QG) system to quickly create assessments for daily reading assignments. Or, he or she might use an automated system to generate comprehension questions about a text he or she found on the web, facilitating the use of practice reading texts that closely match students' interests.

In addition, the utility of automated QG extends beyond educational technology into fields such as web search and dialogue systems. A more extensive discussion of QG and its applications can be found in the report from the NSF-sponsored Workshop on the Question Generation Shared Task and Evaluation Challenge [1].

Each application of QG will have a slightly different set of criteria for evaluating the quality of the questions produced. For instance, in a tutoring application, the quality of questions may depend on the state of a learner model. On the other hand, in a dialogue system, quality may be determined by the efficiency and success of a commercial transaction. Nonetheless, we claim that certain characteristics of questions are domain-general, having relevance in educational applications as well as other domains. This set of general characteristics addresses issues such as grammaticality, vagueness, the use of the appropriate WH-word, and the presence of formatting errors in the output.

General characteristics of questions such as grammaticality are often quite difficult to evaluate automatically. As such, manual judgments of question quality are informative, and perhaps necessary, for effective QG evaluations. Researchers in related fields have developed semi-automatic measures of quality for their respective tasks (e.g., in machine translation [2] and text summarization [3]). While these metrics require some initial human input (e.g., reference translations, human summaries), we note that their value derives from the fact that they facilitate efficient evaluations by making the human input reusable.

In this paper, we contribute to the study of QG by proposing a shared task, the ranking of reading comprehension questions about Wikipedia articles, which focuses on application-general QG issues (§1). We describe a system we developed for this task (§2), and an initial annotation scheme and process (§3) that we used to evaluate the system. We also describe relevant previous research (§4), and conclude with a general discussion (§5).

1. Proposed Task: Ranking Questions about Wikipedia Articles

We propose as a shared QG task the ranking of automatically generated reading comprehension questions about Wikipedia articles. At least in the first version of the task, the questions would be about literal information in the articles, rather than dealing with more complex questions involving inference and prior knowledge. Additionally, the task would assume that the student has only minimal prior knowledge of the topic of the text.

It is important to emphasize that the task would not be the *generation* of questions, but rather the *ranking* of questions generated by a base system that overgenerates (i.e., produces a large amount of output in order to increase the chance of including more high-quality items, perhaps at the expense of including a higher percentage of low-quality items). For example, from the sentence *Francium was discovered by Marguerite Perey in France in 1939*,¹ an overgenerating system might produce the following questions:

- *Where was francium discovered by Marguerite Perey in 1939?*
- *When was francium discovered by Marguerite Perey in France?*
- *Was francium discovered by Marguerite Perey in France in 1939?*
- *By what was francium discovered in France in 1939?*

A system participating in the task would take as input a large number, possibly in the hundreds or more, of unranked literal comprehension questions such as the above, along with metadata formally describing how each question was produced from the source article. This metadata might include the original source sentence, the location of that sentence, the linguistics transformations performed by the overgenerator, etc. The system would then produce as output a single ranking of the input questions by their predicted levels of acceptability. For instance, for the example just presented, a system might identify the last question as unacceptable (because it uses the inappropriate WH-word) and therefore penalize it in the ranking. Acceptability would be defined for evaluation purposes as

¹From “Francium.” *Wikipedia: The Free Encyclopedia*. Retrieved Dec. 16, 2008.

the satisfaction of *all* of a set of domain-general criteria (such as those described in §3).

Researchers have found overgenerate-and-rank approaches to be successful in various domains, including natural language generation [4,5] and syntactic parsing [6]. Initially, we propose that the overgenerating system be based on lexical and syntactic transformations of sentences from the input text. In order to transform declarative sentences into questions, the overgenerator would perform linguistic transformations such as subject-auxiliary inversion, WH-movement, and replacement of answer phrases with appropriate WH-words (i.e., *who*, *what*, *where*, etc.).

The relative simplicity of this ranking task allows researchers to focus on domain-general issues that are relevant to QG in various application domains. In contrast, a QG task in the context of a tutoring system might focus too much on the interactions of question quality with students' prior knowledge for its results to be relevant to a QG researcher working on a dialogue system for commercial transactions. On the other hand, a QG task in the context of a dialogue system for flight reservations might focus too much on the step-by-step communicative process for its results to be relevant to a QG researcher working on tutoring. The issues that the ranking task focuses on, however, are important in many domains.

Of course, participants' ranking systems would to some extent depend on the specific evaluation criteria used for evaluating this task. However, since the criteria are intended to be domain-general, they would likely constitute a subset of the criteria for many specific QG applications. Therefore, to build a ranking system for a specific application, a participant could *extend* rather than replace the ranking system—for example, by adding additional features for a machine learning algorithm to consider when learning to rank.

Another benefit of the ranking task is that human judgments could be reused, making evaluation a semi-automatic process as in machine translation and summarization. Only a single round of human annotation of the questions from the base overgenerating system would be necessary. Once this single set of ratings is acquired, new question-ranking systems could be evaluated automatically by examining the original human judgments of the questions that they rank the highest (e.g., in the top-10).

In subsequent versions of the task, the overgenerator might be revised, replaced, or augmented to address more complex semantic and discourse phenomena. Altering the overgenerator would also deal with the potential problem of focusing too much on the idiosyncrasies of a particular overgenerating approach. An additional shared task for building overgenerating systems also seems possible. For such a task, the overgenerating systems might be evaluated by combining them with various question ranking systems developed for the ranking task.

2. Description of an Implemented System for this Task

We have developed an initial system that overgenerates and ranks questions about English Wikipedia and Simple English Wikipedia articles. The system works in three stages, the first two of which correspond to the overgenerating system de-

scribed previously, and the third of which corresponds to the ranking system that participants in the shared task would build.

In stage 1, sentences from the text are transformed into declarative sentences by optionally altering or transforming lexical items, syntactic structure, and semantics. Many existing NLP transformations may be exploited in this stage, including extractive summarization, sentence compression, sentence splitting, sentence fusion, paraphrase generation, textual entailment, and lexical semantics for word substitution. In our system, we have included a small set of transformations based on previous work in headline generation [7] and summarization [8], such as the removal of appositives and adverbial modifiers.

In stage 2, the derived declarative sentence is turned into a question by executing a set of well-defined syntactic transformations (WH-movement, subject-auxiliary inversion, etc.). The system explicitly encodes well-studied linguistic constraints on WH-movement such as noun phrase island constraints [9]. The transformation rules were implemented by automatically parsing the input into phrase structure trees with the Stanford Parser [10] and using hand-written rules in the Tregex and Tsurgeon tree searching and tree manipulation languages [11] to modify those trees. Note that both the automatic statistical parsing and manually defined transformation operations introduce errors, which further motivate the introduction of a ranking component to identify situations in which errors are likely to have occurred.

In stage 3, which corresponds to the proposed ranking task, the questions are scored and ranked using a statistical classifier based on various features of the questions from stage 2. The system will be described in detail in a forthcoming technical report.

3. Annotation for Evaluation Questions According to General Criteria

In this section, we describe the annotation scheme we used to evaluate our rating system. The annotation process was imperfect, both in the categorization of question deficiencies and the execution of the manual annotation process, but a few key alterations would lead to a more effective and reliable annotation scheme.

We defined a set of question deficiencies, listed and described in Table 1, which focus on characteristics of questions that are reasonably domain- and task- independent, such as grammaticality and appropriate specificity. Since these aspects of questions are not mutually exclusive, the annotator makes a binary decision for each category as to whether a given question is acceptable or not according to that factor. For example, a question might be both vague and ungrammatical, and, in such a case, an annotator would mark both deficiencies. Note that the “Acceptable” category is for questions that are not deficient according to any of the other factors, and *is* therefore mutually exclusive with the others.

The set of questions used in our evaluation test set included 644 questions generated from 8 articles. Four articles, two each, came from the English Wikipedia and the Simple English Wikipedia. The others were two pairs of articles from the Wall Street Journal section of the Penn Treebank, with one of each pair including automatic syntactic parses from the Stanford Parser [10], and the other including

the human-annotated gold-standard syntactic parses from the Treebank. We used the Penn Treebank texts to explore the effects of parser output quality on the generated questions, but we will not go into detail on those results here. Three persons annotated the questions from each article.

Inter-rater agreement, measured by Fleiss’s κ , was fairly low across the categories. For the most important distinction, between questions that are acceptable and those that possess any deficiency, the κ value was .42, corresponding to “moderate agreement” [12]. From observations of the data and comments by annotators, it appears that a few alterations to the scheme would improve agreement. For instance, the categories of “Ungrammatical” and “Does not make sense” might be merged. While the first focuses on syntactic errors and the second on semantic errors, this distinction was not readily apparent to annotators. Additionally, these two types of deficiency can often be attributed to similar causes, such as syntactic parsing errors. The infrequent “Missing Answer” category could also be merged into “Other”. Of course, it would be possible to merge all of the categories and simply rate questions as good or bad. However, the identification of the causes of erroneous output likely has value, both from a scientific point of view, for understanding which aspects of QG are most challenging and interesting, as well as from an engineering point of view, in that a statistical ranking model may be able to leverage this information when ranking questions.

Also, some of the categories could benefit from more detailed descriptions, and though we provided a few examples with each category, more positive and negative examples for each would likely increase agreement. The “Obvious Answer” category was particularly problematic in this regard.

We also note that many of the guidelines and methods for training annotators and improving annotation schemes (cf. [13]) will likely be useful for QG evaluation.

In addition to the set of categories used to annotate questions, improving the *process* of annotation—i.e., annotator training, spot-checking, and redundancy—would likely lead to higher agreement. Three novice annotators rated each article in the test set for our experiments, and because of the relatively low agreement, questions were judged to be “acceptable” only when no annotator annotated it with any deficiency. Other annotation processes might involve either a smaller number of extensively trained annotators, or a much larger number of novice annotators rating questions redundantly through a facility such as Amazon.com’s Mechanical Turk (cf. [14]).

For the evaluation metric for this task, we propose using the percentage of questions labeled “Acceptable” in the top N questions, or precision-at- N . This metric is appropriate because a typical user would likely consider only a limited number of questions. Precision at a single N value (e.g., $N = 10$) would serve as the primary metric, and precision at other N ranging from 1 to 30 would provide additional detail.

4. Prior Work

In this section, we describe some of the prior work relevant to the proposed shared task on ranking questions.

Deficiency	Description	%	κ
Ungrammatical	The question does not appear to be a valid English sentence.	36.7	.29
Does not make sense	The question is grammatical but indecipherable. (e.g., <i>Who was the investment?</i>)	39.5	.29
Vague	The question is too vague to know exactly what it is asking about, even after reading the article (e.g., <i>What did Lincoln do?</i>).	40.2	.40
Obvious answer	The correct answer would be obvious even to someone who has not read the article (e.g., the answer is obviously the subject of the article, or the answer is clearly <i>yes</i>).	14.7	.13
Missing answer	The answer to the question is not in the article.	5.1	.14
Wrong WH word	The question would be acceptable if the WH phrase were different (e.g., <i>in what</i> versus <i>where</i>). WH phrases include <i>who</i> , <i>what</i> , <i>where</i> , <i>when</i> , <i>how</i> , <i>why</i> , <i>how much</i> , <i>what kind of</i> , etc.	12.2	.40
Formatting	There are minor formatting errors (e.g., with respect to capitalization, punctuation)	18.3	.50
Other	There are other errors in the question that are not covered by any of the categories	10.8	.03
Acceptable	None of the above deficiencies were present.	12.8	.42

Table 1. The categories used in our evaluation to describe the possible deficiencies a generated question may exhibit. On the right are, for each category, the percentages of questions that at least one annotator annotated with that category as well as the Fleiss’s κ value for inter-annotator agreement.

Overgenerate-and-rank and reranking approaches have been applied successfully in various domains. In particular, Walker et al. [4] rank potential output from a natural language generation system based on features that correlate with human judgments of quality. Langkilde and Knight [5] also rank output of a generation system, but using corpus evidence rather than human judgments. Similar ranking approaches have been employed for natural language processing tasks. Many state-of-the-art approaches to syntactic parsing currently re-rank their output based on features that cannot be easily incorporated into the pre-ranking component [6].

Most previous evaluations of QG systems have been domain- or task-specific and relatively small-scale (e.g., [16], [17]). Humans rated the output of a particular system. As such, if further developments were made in QG, even for the same task, another round of human judgments would be required.

Evaluations for similar domains are also informative. In summarization [3] and machine translation [2], initial human input in the form of reference translations and summaries have led to semi-automatic metrics. For a given input sentence or document, these metrics compute the surface similarity of system output, of which there are infinitely many possibilities, to output produced by a human. However, it is unclear whether such an approach could be directly adapted for QG. Furthermore, criticisms of metrics based on surface similarity are well-known [18].

Human evaluations are often used in such tasks, particularly for final evaluations rather than ongoing development. The rating schemes often focus on aspects of the output that are relevant to QG, such as grammaticality (e.g., [19]).

In tasks such as recognizing textual entailment [20], paraphrase identification [21], and question answering [22], semi-automatic measures have also been developed based on initial human input. Unlike the above tasks, the output in these cases is a classification decision (e.g., paraphrase or not, correct or incorrect answer). This makes the assessment of correctness of system output straightforward.

5. Discussion

An ideal QG system would be broad-domain (e.g., would work for tutoring in science or history, and for elementary school or college), scalable to large datasets, and generate questions involving deep inference about the source material. Such a solution is not likely to be feasible with current semantic representations and automatic natural language processing technologies.

The feasible approaches to QG fall along a range with respect to their domain-generality, scalability, and the depth of the linguistic and cognitive issues that they address. At one end of this range are approaches that focus on particular narrow-domain QG applications. While such approaches are able to consider deep aspects of linguistics (e.g., semantics, discourse processing) and human knowledge in their specific domains, they make use of resources (e.g., complex ontologies) that may be expensive and not useful across systems.

At the other end of this range are broad-domain, scalable approaches—such as the ranking task proposed here—that focus on more general (though perhaps more shallow) aspects of language, questions, and human knowledge. Focusing on such issues would spur innovation on generalizable and scalable techniques that would be relevant to a variety of specific QG applications. Even very narrow-domain applications would likely benefit because they must still address issues such as grammaticality and vagueness.

In conclusion, the domain-generality of the proposed task and the existence of a straightforward automated evaluation method, as discussed in §1, are the primary benefits that make it likely to succeed in furthering research on QG.

Acknowledgements

We thank the anonymous reviewers for their helpful comments. This work was supported by an NSF Graduate Research Fellowship, Institute of Education Sciences grant R305B040063, and DARPA grant NBCH-1080004.

References

- [1] V. Rus and A. Graesser, Eds. *The Question Generation Shared Task and Evaluation Challenge*, ISBN:978-0-615-27428-7 (2009).

- [2] K. Papineni and S. Roukos and T. Ward and W.-J. Zhu, BLEU: A method for automatic evaluation of machine translation, *Proc. of ACL* (2002).
- [3] C. Lin, ROUGE: A package for automatic evaluation of summaries, *Proc. of Workshop on Text Summarization* (2004).
- [4] M. A. Walker and O. Rambow and M. Rogati, SPoT: A trainable sentence planner, *Proc. of NAACL* (2001), 1–8.
- [5] I. Langkilde and K. Knight, Generation that exploits corpus-based statistical knowledge, *Proc. of ACL* (1998).
- [6] M. Collins, Discriminative reranking for natural language parsing, *Proc. of ICML* (2000).
- [7] B. Dorr and D. Zajic, Hedge Trimmer: A parse-and-trim approach to headline generation, *Proc. of Workshop on Automatic Summarization* (2003).
- [8] K. Toutanova and C. Brockett and M. Gamon and J. Jagarlamudi and H. Suzuki and L. Vanderwende, The PYPHY summarization system: Microsoft Research at DUC 2007, *Proc. of DUC* (2007).
- [9] N. Chomsky, On wh-movement, In P. W. Culicover and T. Wasow and A. Akmajian (Eds.), *Formal Syntax*, New York: Academic Press (1977).
- [10] D. Klein and C. D. Manning, Fast exact inference with a factored model for natural language parsing, *Advances in NIPS 15* (2003).
- [11] R. Levy and G. Andrew, Tregex and Tsurgeon: Tools for querying and manipulating tree data structures, *Proc. of LREC* (2006).
- [12] J. R. Landis and G. G. Koch, The measurement of observer agreement for categorical data, *Biometrics* **33** (1977) 159–174.
- [13] K. Krippendorff, *Content Analysis: An Introduction to Its Methodology, 2nd Edition*, Sage (2004).
- [14] R. Snow, B. O’Connor, D. Jurafsky and A. Ng, Cheap and fast – but is it good? Evaluating non-expert annotations for natural language tasks, *Proc. of EMNLP* (2008).
- [15] L. von Ahn and L. Dabbish, Labeling images with a computer game, *Proc. of the SIGCHI conference on Human factors in computing systems* (2004).
- [16] R. Mitkov and L. A. Ha and N. Karamanis, A computer-aided environment for generating multiple-choice test items, *Natural Language Engineering* **12** (2006), 177–194.
- [17] H. Kunichika and T. Katayama and T. Hirashima and A. Takeuchi, Automated question generation methods for intelligent English learning systems and its evaluation, *Proc. of ICCE* (2004).
- [18] C. Callison-Burch and M. Osborne, Re-evaluating the role of BLEU in machine translation research, *Proc. of EACL* (2006), 249–256.
- [19] K. Knight and D. Marcu, Statistics-based summarization - step one: sentence compression, *Proc. of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence* (2000).
- [20] O. Glickman and I. Dagan and M. Koppel, A probabilistic classification approach for lexical textual entailment, *Proc. of AAAI-05* (2005).
- [21] W. B. Dolan and C. Brockett, Automatically constructing a corpus of sentential paraphrases, *Proc. of IWP2005* (2005).
- [22] E. M. Voorhees, Overview of the TREC 2003 question answering track, *Proc. of TREC 2003* (2004).