Linguistic Considerations in Automatic Question Generation

Karen Mazidi
HiLT Lab
University of North Texas
Denton TX 76207, USA
KarenMazidi@my.unt.edu

Rodney D. Nielsen
HiLT Lab
University of North Texas
Denton TX 76207, USA
Rodney.Nielsen@unt.edu

Abstract

As students read expository text, comprehension is improved by pausing to answer questions that reinforce the material. We describe an automatic question generator that uses semantic pattern recognition to create questions of varying depth and type for self-study or tutoring. Throughout, we explore how linguistic considerations inform system design. In the described system, semantic role labels of source sentences are used in a domain-independent manner to generate both questions and answers related to the source sentence. Evaluation results show a 44% reduction in the error rate relative to the best prior systems, averaging over all metrics, and up to 61% reduction in the error rate on grammaticality judgments.

1 Introduction

Studies of student learning show that answering questions increases depth of student learning, facilitates transfer learning, and improves students' retention of material (McDaniel et al., 2007; Carpenter, 2012; Roediger and Pyc, 2012). The aim of this work is to automatically generate questions for such pedagogical purposes.

2 Related Work

Approaches to automatic question generation from text span nearly four decades. The vast majority of systems generate questions by selecting one sentence at a time, extracting portions of the source sentence, then applying transformation rules or patterns in order to construct a question. A well-known early work is Wolfe's AUTOQUEST (Wolfe, 1976), a syntactic pattern matching system. A recent approach from Heilman and Smith (2009, 2010) uses syntactic parsing and transformation rules to generate questions. Syntactic, sentence-level approaches outnumber other approaches as seen in the Question Generation Shared Task Evaluation Challenge 2010 (Boyer and Piwek, 2010) which received only one paragraph-level, semantic entry. Argawal, Shah and Mannem (2011) continue the paragraph-level approach using discourse cues to find appropriate text segments upon which to construct questions at a deeper conceptual level. The uniqueness of their work lies in their use of discourse cues to extract semantic content for question generation. They generate questions of types: *why*, *when*, *give an example*, and *yes/no*.

In contrast to the above systems, other approaches have an intermediate step of transforming input into some sort of semantic representation. Examples of this intermediate step can be found in Yao and Zhang (2010) which uses Minimal Recursive Semantics, and in Olney et al. (2012) which uses concept maps. These approaches can potentially ask deeper questions due to their focus on semantics. A novel question generator by Curto et al. (2012) leverages lexico-syntactic patterns gleaned from the web with seed question-answer pairs.

Another recent approach is Lindberg et al. (2013), which used semantic role labeling to identify patterns in the source text from which questions can be generated. This work most closely parallels our own with a few exceptions: our system only asks questions that can be answered from the source text, our approach is domain-independent, and the patterns also identify the answer to the question.

3 Approach

The system consists of a straightforward pipeline. First, the source text is divided into sentences which are processed by SENNA[1] software, de-

[1] http://ml.nec-labs.com/senna/

Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 321–326,
Baltimore, Maryland, USA, June 23-25 2014. ©2014 Association for Computational Linguistics

scribed in (Collobert et al., 2011). SENNA provides the tokenizing, pos tagging, syntactic constituency parsing and semantic role labeling used in the system. SENNA produces separate semantic role labels for each predicate in the sentence. For each predicate and its associated semantic arguments, a matcher function is called which will return a list of patterns that match the source sentence's predicate-argument structure. Then questions are generated and stored by question type in a question hash table.

Generation patterns specify the text, verb forms and semantic arguments from the source sentence to form the question. Additionally, patterns indicate the semantic arguments that provide the answer to the question, required fields, and filter condition fields. As these patterns are matched, they will be rejected as candidates for generation for a particular sentence if the required arguments are absent or if filter conditions are present. For example, a filter for personal pronouns will prevent a question being generated with an argument that starts with a personal pronoun. From: *It means that the universe is expanding*, we do not want to generate a vague question such as: *What does it mean?* Coreference resolution, which could help avoid vague question generation, is discussed in Section 5. Table 1 shows selected required and filter fields, Section 3.3 gives examples of their use.

Patterns specify whether verbs should be included in their lexical form or as they appear in the source text. Either form will include subsequent particles such as: The lungs *take in* air. The most common use of the verb as it appears in the sentence is with the verb *be*, as in: What *were* fused into helium nuclei? This pattern takes the copular *be* as it appears in the source text. However, most patterns use the lexical form of the main verb along with the appropriate form of the auxiliary *do* (do, does, did), for the subject-auxiliary inversion required in forming interrogatives.

3.1 Pattern Authoring

The system at the time of this evaluation had 42 patterns. SENNA uses the 2005 PropBank coding scheme and we followed the documentation in (Babko-Malaya, 2005) for the patterns. The most commonly used semantic roles are A0, A1 and A2, as well as the ArgM modifiers. [2]

[2]Within PropBank, the precise roles of A0 - A6 vary by predicate.

Field	Meaning
Ax	Sentence must contain an Ax
!Ax	Sentence must not contain an Ax
AxPER	Ax must refer to a person
AxGER	Ax must contain a gerund
AxNN	Ax must contain nouns
!AxIN	Ax cannot start with a preposition
!AxPRP	Ax cannot start with per. pronoun
V=*verb*	Verb must be a form of *verb*
!be	Verb cannot be a form of *be*
negation	Sentence cannot contain negation

Table 1: Selected required and filter fields *(Ax is a semantic argument such as A0 or ArgM)*

3.2 Software Tools and Source Text

The system was created using SENNA and Python. Importing NLTK within Python provides a simple interface to WordNet from which we determine the lexical form of verbs. SENNA provided all the necessary processing of the data, quickly, accurately and in one run.

In order to generate questions, passages were selected from science textbooks downloaded from www.ck12.org. Textbooks were chosen rather than hand-crafted source material so that a more realistic assessment of performance could be achieved. For the experiments in this paper, we selected three passages from the subjects of biology, chemistry, and earth science, filtering out references to equations and figures. The passages average around 60 sentences each, and represent chapter sections. The average grade level is approximately grade 10 as indicated by the on-line readability scorer read-able.com.

3.3 Examples

Table 2 provides examples of generated questions. The pattern that generated Question 1 requires argument A1 (underlined in Table 2) and a causation ArgM (italicized). The pattern also filters out sentences with A0 or A2. The patterns are designed to match only the arguments used as part of the question or the answer, in order to prevent over generation of questions. The system inserted the correct forms of *release* and *do*, and ignored the phrase *As this occurs* since it is not part of the semantic argument.

The pattern that generated Question 2 requires A0, A1 and a verb whose lexical form is *mean* (V=mean in Table 1). In this pattern, A1 (itali-

Question 1: Why did potential energy release? **Answer:** *because the new bonds have lower potential energy than the original bonds* **Source:** As this occurs, potential energy is released *because the new bonds have lower potential energy than the original bonds.*
Question 2: What does an increased surface area to volume ratio indicate? **Answer:** *increased exposure to the environment* **Source:** An increased surface area to volume ratio means *increased exposure to the environment*.
Question 3: What is another term for electrically neutral particles? **Answer:** *neutrons* **Source:** The nucleus contains positively charged particles called protons and electrically neutral particles called *neutrons*.
Question 4: What happens if you continue to move atoms closer and closer together? **Answer:** *eventually the two nuclei will begin to repel each other* **Source:** If you continue to move atoms closer and closer together, *eventually the two nuclei will begin to repel each other.*

Table 2: Selected generated questions with source sentences

cized) forms the answer and A0 (underlined) becomes part of the question along with the appropriate form of *do*. This pattern supplies the word *indicate* instead of the source text's *mean* which broadens the question context.

Question 3 is from the source sentence's 3rd predicate-argument set because this matched the pattern requirements: A1, A2, V=call. The answer is the text from the A2 argument. The ability to generate questions from any predicate-argument set means that sentence simplification is not required as a preprocessing step, and that the sentence can match multiple patterns. For example, this sentence could also match patterns to generate questions such as: *What are positively charged particles called?* or *Describe the nucleus*.

Question 4 requires A1 and an ArgM that includes the discourse cue *if*. The ArgM (underlined) becomes part of the question, while the rest of the source sentence forms the answer. This pattern also requires that ArgM contain nouns (AxNN from Table 1), which helps filter vague questions.

4 Results

This paper focuses on evaluating generated questions primarily in terms of their linguistic quality, as did Heilman and Smith (2010a). In a related work (Mazidi and Nielsen, 2014) we evaluated the quality of the questions and answers from a pedagogical perspective, and our approach outperformed comparable systems in both linguistic and pedagogical evaluations. However, the task here is to explore the linguistic quality of generated questions. The annotators are university students who are science majors and native speakers of English. Annotators were given instructions to read a paragraph, then the questions based on that paragraph. Two annotators evaluated each set of questions using Likert-scale ratings from 1 to 5, where 5 is the best rating, for grammaticality, clarity, and naturalness. The average inter-annotator agreement, allowing a difference of one between the annotators' ratings was 88% and Pearson's r=0.47 was statistically significant ($p<0.001$), suggesting a high correlation and agreement between annotators. The two annotator ratings were averaged for all the evaluations reported here.

We present results on three linguistic evaluations: (1) evaluation of our generated questions, (2) comparison of our generated questions with those from Heilman and Smith's question generator, and (3) comparison of our generated questions with those from Lindberg, Popowich, Nesbit and Winne. We compared our system to the H&S and LPN&W systems because they produce questions that are the most similar to ours, and for the same purpose: reading comprehension reinforcement. The Heilman and Smith system is available online;[3] Lindberg graciously shared his code with us.

4.1 Evaluation of our Generated Questions

This evaluation was conducted with one file (Chemistry: Bonds) which had 59 sentences, from which the system generated 142 questions. The

[3] http://www.ark.cs.cmu.edu/mheilman/questions/

purpose of this evaluation was to determine if any patterns consistently produce poor questions. The average linguistics score per pattern in this evaluation was 5.0 to 4.18. We were also interested to know if first predicates make better questions than later ones. The average score by predicate position is shown in Table 3. Note that the Rating column gives the average of the grammaticality, clarity and naturalness scores.

Predicate	Questions	Rating
First	58	4.7
Second	35	4.7
Third	23	4.5
Higher	26	4.6

Table 3: Predicate depth and question quality

Based on this sample of questions there is no significant difference in linguistic scores for questions generated at various predicate positions. Some question generation systems simplify complex sentences in initial stages of their system. In our approach this is unnecessary, and simplifying could miss many valid questions.

4.2 Comparison with Heilman and Smith

This task utilized a file (Biology: the body) with 56 source sentences from which our system generated 102 questions. The Heilman and Smith system, as they describe it, takes an over-generate and rank approach. We only took questions that scored a 2.0 or better with their ranking system,[4] which resulted in less than 27% of their top questions. In all, 84 of their questions were evaluated. The questions again were presented with accompanying paragraphs of the source text. Questions from the two systems were randomly intermingled. Annotators gave 1 - 5 scores for each category of grammaticality, clarity and naturalness.

As seen in Table 4, our results represent a 44% reduction in the error rate relative to Heilman and Smith on the average rating over all metrics, and as high as 61% reduction in the error rate on grammaticality judgments. The error reduction calculation is shown below. Note that $rating^*$ is the maximum rating of 5.0.

$$\frac{rating_{system2} - rating_{system1}}{rating^* - rating_{system1}} \times 100.0 \quad (1)$$

[4]In our experiments, their rankings ranged from very small negative numbers to 3.0.

System	Gram	Clarity	Natural	Avg
H&S	4.38	4.13	3.94	4.15
M&N	4.76	4.26	4.53	4.52
Err. Red.	61%	15%	56%	44%

Table 4: Comparison with Heilman and Smith

System	Gram	Clarity	Natural	Avg
LPN&W	4.57	4.56	4.55	4.57
M&N	4.80	4.69	4.78	4.76
Err. Red.	54%	30%	51%	44%

Table 5: Comparison with Lindberg et al.

4.3 Comparison with Lindberg et al.

For a comparison with the Lindberg, Popowich, Nesbit and Winne system we used a file (Earth science: weather fronts) that seemed most similar to the text files for which their system was designed. The file has 93 sentences and our system generated 184 questions; the LPN&W system generated roughly 4 times as many questions. From each system, 100 questions were randomly selected, making sure that the LPN&W questions did not include questions generated from domain-specific templates such as: *Summarize the influence of the maximum amount on the environment.* The phrases *Summarize the influence of* and *on the environment* are part of a domain-specific template. The comparison results are shown in Table 5. Interestingly, our system again achieved a 44% reduction in the error rate when averaging over all metrics, just as it did in the Heilman and Smith comparison.

5 Linguistic Challenges

Natural language generation faces many linguistic challenges. Here we briefly describe three challenges: negation detection, coreference resolution, and verb forms.

5.1 Negation Detection

Negation detection is a complicated task because negation can occur at the word, phrase or clause level, and because there are subtle shades of negation between definite positive and negative polarities (Blanco and Moldovan, 2011). For our purposes we focused on negation as identified by the NEG label in SENNA which identified *not* in verb phrases. We have left for future work the task of

identifying other negative indicators, which occasionally does lead to poor question/answer quality as in the following:

Source sentence: In Darwin's time and today, many people incorrectly believe that evolution means humans come from monkeys.

Question: What does evolution mean?

Answer: that humans come from monkeys

The negation in the word *incorrectly* is not identified.

5.2 Coreference Resolution

Currently, our system does not use any type of coreference resolution. Experiments with existing coreference software performed well only for personal pronouns, which occur infrequently in most expository text. Not having coreference resolution leads to vague questions, some of which can be filtered as discussed previously. However, further work on filters is needed to avoid questions such as:

Source sentence: Air cools when it comes into contact with a cold surface or when it rises.

Question: What happens when it comes into contact with a cold surface or when it rises?

Heilman and Smith chose to filter out questions with personal pronouns, possessive pronouns and noun phrases composed simply of determiners such as *those*. Lindberg et al. used the emPronoun system from Charniak and Elsner, which only handles personal pronouns. Since current state-of-the-art systems do not deal well with relative and possessive pronouns, this will continue to be a limitation of natural language generation systems for the time being.

5.3 Verb Forms

Since our focus is on expository text, system patterns deal primarily with the present and simple past tenses. Some patterns look for modals and so can handle future tense:

Source sentence: If you continue to move atoms closer and closer together, eventually the two nuclei will begin to repel each other.

Question: Discuss what the two nuclei will repel.

Light verbs pose complications in NLG because they are highly idiosyncratic and subject to syntactic variability (Sag et al., 2002). Light verbs can either carry semantic meaning (*take* your passport) or can be bleached of semantic content when combined with other words as in: *make* a decision, *have* a drink, *take* a walk. Common English verbs that can be light verbs include give, have, make, take. Handling these constructions as well as other multi-word expressions may require both rule-based and statistical approaches. The catenative construction also potentially adds complexity (Huddleston and Pullum, 2005), as shown in this example: As the universe expanded, it became less dense and *began* to *cool*. Care must be taken not to generate questions based on one predicate in the catenative construction.

We are also hindered at times by the performance of the part of speech tagging and parsing software. The most common error observed was confusion between the noun and verb roles of a word. For example in: *Plant roots and bacterial decay use carbon dioxide in the process of respiration,* the word *use* was classified as NN, leaving no predicate and no semantic role labels in this sentence.

6 Conclusions

Roediger and Pyc (2012) advocate assisting students in building a strong knowledge base because creative discoveries are unlikely to occur when students do not have a sound set of facts and principles at their command. To that end, automatic question generation systems can facilitate the learning process by alternating passages of text with questions that reinforce the material learned.

We have demonstrated a semantic approach to automatic question generation that outperforms similar systems. We evaluated our system on text extracted from open domain STEM textbooks rather than hand-crafted text, showing the robustness of our approach. Our system achieved a 44% reduction in the error rate relative to both the Heilman and Smith, and the Lindberg et al. system on the average over all metrics. The results shows are statistically significant ($p<0.001$). Our question generator can be used for self-study or tutoring, or by teachers to generate questions for classroom discussion or assessment. Finally, we addressed linguistic challenges to question generation.

Acknowledgments

This research was supported by the Institute of Education Sciences, U.S. Dept. of Ed., Grant R305A120808 to UNT. The opinions expressed are those of the authors.

References

Agarwal, M., Shah, R., and Mannem, P. 2011. Automatic question generation using discourse cues. In *Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications*, Association for Computational Linguistics.

Babko-Malaya, O. 2005. Propbank annotation guidelines. *URL: http://verbs.colorado.edu*

Blanco, E., and Moldovan, D. 2011. Some issues on detecting negation from text. In *FLAIRS Conference.*

Boyer, K. E., and Piwek, P., editors. 2010. In *Proceedings of QG2010: The Third Workshop on Question Generation.* Pittsburgh: questiongeneration.org

Carpenter, S. 2012. Testing enhances the transfer of learning. In *Current directions in psychological science,* 21(5), 279-283.

Charniak, E., and Elsner, M. 2009. EM works for pronoun anaphora resolution. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics.* Association for Computational Linguistics.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12, 2493-2537.

Curto, S., Mendes, A., and Coheur, L. 2012. Question generation based on lexico-syntactic patterns learned from the web. *Dialogue & Discourse*, 3(2), 147-175.

Heilman, M., and Smith, N. 2009. *Question generation via overgenerating transformations and ranking.* Technical Report CMU-LTI-09-013, Language Technologies Institute, Carnegie-Mellon University.

Heilman, M., and Smith, N. 2010a. Good question! statistical ranking for question generation. In *Proceedings of NAACL/HLT 2010*. Association for Computational Linguistics.

Heilman, M., and Smith, N. 2010b. Rating computer-generated questions with Mechanical Turk. In *Proceedings of the NAACL-HLT Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk.* Association for Computational Linguistics.

Huddleston, R. and Pullum, G. 2005. *A Student's Introduction to English Grammar*, Cambridge University Press.

Lindberg, D., Popowich, F., Nesbit, J., and Winne, P. 2013. Generating natural language questions to support learning on-line. In *Proceedings of the 14th European Workshop on Natural Language Generation*, (2013): 105-114.

Mannem, P., Prasad, R. and Joshi, A. 2010. Question generation from paragraphs at UPenn: QGSTEC system description. In *Proceedings of QG2010: The Third Workshop on Question Generation.*

Mazidi, K. and Nielsen, R.D. 2014. Pedagogical evaluation of automatically generated questions. In *Intelligent Tutoring Systems.* LNCS 8474, Springer International Publishing Switzerland.

McDaniel, M. A., Anderson, J. L., Derbish, M. H., and Morrisette, N. 2007. Testing the testing effect in the classroom. *European Journal of Cognitive Psychology,* 19(4-5), 494-513.

Olney, A., Graesser, A., and Person, N. 2012. Question generation from concept maps. *Dialogue & Discourse,* 3(2), 75-99.

Roediger III, H. L., and Pyc, M. 2012. Inexpensive techniques to improve education: Applying cognitive psychology to enhance educational practice. *Journal of Applied Research in Memory and Cognition*, 1.4: 242-248.

Sag, I. A., Baldwin, T., Bond, F., Copestake, A., and Flickinger, D. 2002. Multiword expressions: A pain in the neck for NLP. In *Computational Linguistics and Intelligent Text Processing*, (pp. 1-15). Springer Berlin Heidelberg.

Sternberg, R. J., & Grigorenko, E. L. 2003. Teaching for successful intelligence: Principles, procedures, and practices. *Journal for the Education of the Gifted*, 27, 207-228.

Wolfe, J. 1976. Automatic question generation from text-an aid to independent study. In *Proceedings of ACM SIGCSE-SIGCUE.*

Yao, X., and Zhang, Y. 2010. Question generation with minimal recursion semantics. In *Proceedings of QG2010: The Third Workshop on Question Generation.*