Automatic Question Generation using Relative Pronouns and Adverbs

Payal Khullar   **Konigari Rachna**   **Mukul Hase**   **Manish Shrivastava**

Language Technologies Research Centre
International Institute of Information Technology, Hyderabad
Gachibowli, Hyderabad, Telangana-500032
{payal.khullar@research., konigari.rachna@research.
mukul.hase@student., m.shrivastava@}iiit.ac.in

Abstract

This paper presents a system that automatically generates multiple, natural language questions using relative pronouns and relative adverbs from complex English sentences. Our system is syntax-based, runs on dependency parse information of a single-sentence input, and achieves high accuracy in terms of syntactic correctness, semantic adequacy, fluency and uniqueness. One of the key advantages of our system, in comparison with other rule-based approaches, is that we nearly eliminate the chances of getting a wrong wh-word in the generated question, by fetching the requisite wh-word from the input sentence itself. Depending upon the input, we generate both factoid and descriptive type questions. To the best of our information, the exploitation of wh-pronouns and wh-adverbs to generate questions is novel in the Automatic Question Generation task.

1   Introduction

Asking questions from learners is said to facilitate interest and learning (Chi, 1994), to recognize problem learning areas (Tenenberg and Murphy, 2005) to assess vocabulary (Brown et al., 2005) and reading comprehension (Mitkov, 2006); (Kunichika et al., 2004), to provide writing support (Liu et al., 2012), to support inquiry needs (Ali et al., 2010), etc. Manual generation of questions from a text for creating practice exercises, tests, quizzes, etc. has consumed labor and time of academicians and instructors since forever, and with the invent of a large body of educational material available online, there is a growing need to make this task scalable. Along with that, in the recent times, there is an increased demand to cre-

ate Intelligent tutoring systems that use computer-assisted instructional material or self-help practice exercises to aid learning as well as objectively check learner's aptitude and accomplishments. Inevitably, the task of Automatic Question Generation (QG) caught the attention of NLP researchers from across the globe. Automatic QG has been defined as "the task of automatically generating questions from various inputs like raw text, database or semantic representation" (Rus et al., 2008). Apart from its direct application in the educational domain, in general, the core NLP areas like Question Answering, Dialogue Generation, Information Retrieval, Summarization, etc. also benefit from large scale automatic Question Generation.

2   Related Work

Previous work on Automatic QG has focused on generating questions using question templates (Liu et al., 2012); (Mostow and Chen, 2009); (Sneiders, 2002), transformation rules based largely on case grammars (Finn, 1975), general-purpose, transformation rules (Heilman and Smith, 2009), tree manipulation rules (Heilman, 2011); (Ali et al., 2010); (Gates, 2008), discourse cues (Agarwal et al., 2011), queries (Lin, 2008), various scopes (Mannem et al., 2010), dependency parse information (Mazidi and Nielsen, 2015), topic information (Chali and Hasan, 2015), ontologies (Alsubait et al., 2015), etc. More recent approaches also apply neural methods (Subramanian et al., 2017); (Zhou et al., 2017); (Yuan et al., 2017); (Du et al.), etc. to generate questions.

In the current paper, we fetch relative pronouns and relative adverbs from complex English sentences and use dependency-based rules, grounded in linguistic theory of relative clause syntactic structure, to generate multiple relevant questions. The work follows in the tradition of question writing algorithm (Finn, 1975) and transformation

rules based approach (Heilman and Smith, 2009). However, while Finn's work was based largely around case grammars (Fillmore, 1968), our system exploits dependency parse information using the Spacy parser (Honnibal and Johnson, 2015), which provides us with a better internal structure of complex sentences to work with. The general-purpose transformation rules in Heilman's system do not work well on sentences with a highly complex structure, as we show in the section on comparison and evaluation. Although no other state-of-the art system focuses specifically on QG from relative pronouns and relative adverbs, a more recent Minimal Recursion semantics-based QG system (Yao et al., 2012) has a sub part that deals with sentences with a relative clause, but less comprehensively. We differ from their system in that, for one, we do not decompose the complex sentence into simpler parts to generate questions. The rules are defined for the dependencies between relative pronouns and relative adverbs and the rest of the sentence as a whole. Secondly, our system generates a different set and more number of questions per sentence than their system.

3 Why Relative Clauses?

In complex sentences, relative pronouns or relative adverbs perform the function of connecting or introducing the relative clause that is embedded inside the matrix clause. Examples of these in English include *who, whom, which, where, when, how, why*, etc. An interesting thing about both relative pronouns and relative adverbs is that they carry unique information on the syntactic relationship between specific parts of the sentence. For example, consider the following sentence in English:

I am giving fur balls to John who likes cats.

In this sentence, the relative pronoun *who* modifies the object of the root verb *give* of the matrix sentence. At the same time, it acts as the subject of the relative clause *likes cats*, which it links the matrix clause with. In this paper, we aim to exactly exploit this structural relationship to generate questions, thereby adding to the pool of questions that can be generated from a given sentence. One of the key benefits of using the information from relative pronouns and relative adverbs is that we are not likely to go wrong with the wh-word, as we fetch it from the relative clause itself to generate the question. This gives our system an edge

over other QG systems.

4 System Description

We split the complete QG task into the following sub parts - the input natural language sentence is first fed into the Spacy parser. Using the parse information, the system checks for the presence of one or more relative pronouns or adverbs in the sentence. Post that, it further checks for well-defined linguistic features in the sentence, such as tense and aspect type of the root and relative clause verb, head-modifier relationship between different parts of the sentence, etc. to accordingly send the information to the rule sets. Depending upon which rule in the rule sets the information is sent to, questions are generated. We define our rule sets in the next section.

5 Rule Sets

For each of the nine relative pronouns and relative adverbs in English (*who, whom, whose, which, that, where, when, why, how*) that we took into consideration, we defined three sets of rules. Each of the three rule sets further contains a total of ten rules, backed by linguistic principles. Each relative pronoun or relative adverb in the sentence is first checked for a set of requirements before getting fed into the rules. Depending upon the relative pronoun and relative adverb and the type of relative clause (restrictive or unrestrictive), questions are generated. We present an example of one out of the ten rules from each of the three rule sets in the next section.

5.1 Rule Set 1.

We know that the relative pronoun (RP) modifies the object of the root of the matrix sentence. Before feeding the sentence to the rule, we first check the sentence for the tense and aspect of the root verb and also for the presence of modals and auxiliary verbs (aux). Based on this information, we then accordingly perform do-insertion before Noun Phrase (NP) or aux/modal inversion. For a sentence of the following form, with an optional Preposition Phrase (PP) and RP *who* that precedes a relative clause (RC),

NP (aux) Root NP (PP)+ {RC}

The rule looks like this:

RP aux NP root NP Preposition?

Hence, for the example sentence introduced in the previous section, we get the following question

using the first rule:

Who/Whom am I giving fur balls to?

In representation of the rules, we follow the general linguistic convention which is to put round brackets on optional elements and '+' symbol for multiple possible occurrences of a word or phrase.

5.2 Rule Set 2.

The next understanding about the relative pronoun or adverb comes from the relative clause it introduces or links the matrix sentence with. The relative pronoun can sometimes act as the subject of the verb of relative clause. This forms the basis for rules in the second set.

After checking for dependency of the RP, which should be noun subject (n-subj) to the verb in the relative clause, we then check the tense and aspect of the relative clause verb and the presence of modals and auxiliary verbs. Based on this information, we then accordingly perform do-insertion or modal/auxiliary inversion. For a relative clause of the following form, with the relative pronoun *who*,

{matrix } RP (aux/modals)+ RC verb (NP) (PP)+

The rule looks like this:

RP   do-insertion/aux/modal   RC verb   (NP) (Preposition)?

Taking the same example sentence, we get the following question using the second rule:

Who likes cats?

5.3 Rule Set 3.

The relative pronoun modifies or gives more information on the head of the noun phrase of the preceding sentence. This forms the basis to rules in the third set. Before feeding the sentence to this rule, we first check the tense of the relative clause verb along with its number agreement. We do this because English auxiliaries and copula carry tense and number features and we need this information to insert their correct form. For a sentence of the following form:

NP (aux/modals) Root NP RP (aux)+ RC verb (NP) (PP)+

The rule looks like this:

RP   aux   Head of NP?

Taking the first example sentence, from the previous sections, we get the following question using the fourth rule:

Who is John?

In a similar fashion, we define rules for all other relative pronouns and adverbs that we listed in the previous section.

6 Evaluation Criteria

There is no standard way to evaluate the output of a QG system. In the current paper, we go with manual evaluation, where 4 independent human evaluators, all non-native English speakers but proficient in English, give scores to questions generated from the system. The scoring schema is similar to one used by (Agarwal et al., 2011) albeit with some modifications. To judge syntactic correctness, the evaluators give a score of 3 when the questions are syntactically well-formed and natural, 2 when they have a few syntactic errors and 1 when they are syntactically unacceptable. Similarly, for semantic correctness, the raters give a score of 3 when the questions are semantically correct, 2 when they have a weird meaning and 1 when they are semantically unacceptable. Unlike (Agarwal et al., 2011), we test the fluency and semantic relatedness separately. The former tells us how natural the question reads. A question with many embedded clauses and adjuncts is syntactically acceptable, but disturbs the intended purpose of the question and, hence, should be avoided. For example, a question like *Who is that girl who works at Google which has its main office in America which is a big country?* is syntactically and semantically fine, but isn't as fluent as the question *Who is that girl who works at Google?* which is basically the same question but is more fluent. The evaluators give a score of 1 for questions that aren't fluent and 2 to the ones that are. Lastly, evaluators rate the questions for how unique they are. Adding this criteria is important because questions generated for academic purposes need to cover different aspects of the sentence. This is why if the generated questions are more or less alike, the evaluators give them a low score on distribution or variety. For a well distributed output, the score is 2 and for a less distributed one, it is 1. The evaluators give a score of 0 when there is no output for a given sentence. The scores obtained separately for syntactic correctness, semantic adequacy, fluency and distribution are used to compare the performance of the two systems.
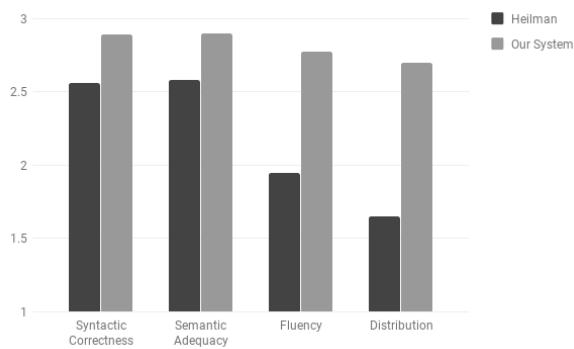
7 Evaluation

We take sentences from the Wikipedia corpus. Out of a total of 25773127 sentences, 3889289 sen-

tences have one or more relative pronoun or relative adverb in them. This means that sentences with relative clauses form roughly 20% of the corpus. To conduct manual evaluation, we take 300 sentences from the set of sentences with relative clauses, and run ours and Heilman's system on them. We give the questions generated per sentence for both the systems to 4 independent human evaluators who rate the questions on syntactic correctness, semantic adequacy, fluency and distribution.

8 Results

The results of our system and comparison with Heilman's is given in Figure 1. The ratings presented are average of ratings of all the evaluators. Our system gets 2.89/3.0 on syntactic correctness, 2.9/3.0 on semantic adequacy, 1.85/2.0 in fluency and 1.8/2.0 in distribution. On the same metrics, Heilman's system gets 2.56, 2.58, 1.3 and 1.1. The Cohen's kappa coefficient or the inter evaluator agreement is 0.6, 0.7, 0.7 and 0.7 on syntactic correctness, semantic adequacy, fluency and distribution respectively, which indicate reliability. The overall rating of our system is 9.44 out of 10 in comparison of Heilman's which is 7.54.

Figure 1: Evaluation scores: Our system performs better than Heilman's system on all of the given criteria; syntactic correctness, semantic adequacy, fluency and uniqueness.

9 Discussion

On all the four evaluation criteria that we used for comparison, our system performs better than Heilman's state-of-the-art rule based system, while generating questions from complex English sentences. Let us take a look at some specific input example cases to analyze the results. First of all, by fetching and modifying the wh-word

from the sentence itself, we nearly eliminate the possibility of generating a sentence with a wrong wh-word. From the example comparison in Figure.2, we can see that the output of both the systems is the same. However, our system generates the correct wh-word for the generated question.

Sentences	Heilman	Our System
Robert's elder brother plays badminton with the little girl who my mother teaches Mathematics to in the school which is state operated	Which is state operated?	What is state operated

Figure 2: Wh-Word: Our system performs better than Heilman's system at fetching the correct Wh-word for the given input.

By exploiting the unique structural relationships between relative pronouns and relative adverbs with the rest of the sentence, we are able to cover different aspects of the same sentence. Also, by eliminating unwanted dependencies, we ensure that the system generates fluent questions. See Figure 3 for a reference example.

Sentences	Heilman	Our System
Mary gave dark chocolates and flowers to Bart who works at Amazon which was known to be the biggest online store in Africa	Who did Mary give dark chocolates and flowers to?	Who did Mary give dark chocolates and flowers to?
	Who gave dark chocolates and flowers to Bart who works at Amazon which was known to be the biggest online store in Africa?	Who works at Amazon?
		Who is Bart?
	What did Mary give to Bart who works at Amazon which was known to be the biggest online store in Africa?	What is the biggest online store in Africa?
	Did Mary give dark chocolates and flowers to Bart who works at Amazon which was known to be the biggest online store in Africa?	What is Amazon?

Figure 3: Fluency: As compared to Heilman's system, our system generates more fluent questions for the given input.

Since Heilman's system does not look deeper into the internal structural dependencies between different parts of the sentence, it fails to generate reasonable questions for most cases of complex sentences. Our system, on the other hand, exploits such dependencies and is, therefore, able to handle complex sentences better. See Figure 4 for

a reference example of this case. Lastly, there is a restriction put on the length of the input sentence in Heilman's system. Due to this, there is zero or no output at all for complex sentences that are very long. Our system, however, works well on such sentences also and gives reasonable output.

Sentences	Heilman	Our System
Using the findings , men can get a rough idea of their personal odds by answering questions that also include weight and race , experts said	Who said? Did experts say?	What also include weight and race? What are questions that also include weight and race? What can men get a rough idea of their personal odds by answering questions that also include weight and race?

Figure 4: Complex Sentences: Our system is able to handle the given highly complex sentence better than Heilman's system.

10 Conclusion

This paper presented a syntax, rule-based system that runs on dependency parse information from the Spacy parser and exploits dependency relationship between relative pronouns and relative adverbs and the rest of the sentence in a novel way to automatically generate multiple questions from complex English sentences. The system is simple in design and can handle highly complex sentences. The evaluation was done by 4 independent human evaluators who rated questions generated from our system and Heilman's system on the basis of how syntactically correct, semantically adequate, fluent and well distributed or unique the questions were. Our system performed better than Heilman's system on all the aforesaid criterion. A predictable limitation of our system is that it is only meant to generate questions for sentences that contain at least one relative clause. Such sentences form about 20% of the tested corpus.

11 Acknowledgments

References

Manish Agarwal, Rakshit Shah, and Prashanth Mannem. 2011. Automatic question generation using discourse cues. *Proceedings of the Sixth Workshop on Innovative Use of NLP for Building Educational Applications*, page 19.

Husam Ali, Yllias Chali, and Sadid A. Hasan. 2010. Automation of question generation from sentences. *Proceedings of QG2010: The Third Workshop on Question Generation*, pages 58–67.

Tahani Alsubait, Bijan Parsia, and Ulrike Sattler. 2015. Ontology-based multiple choice question generation. *KI - Knstliche Intelligenz*, 30(2):183188.

Jonathan C. Brown, Gwen A. Frishkoff, and Maxine Eskenazi. 2005. Automatic question generation for vocabulary assessment. *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing - HLT 05*.

Yllias Chali and Sadid A. Hasan. 2015. Towards topic-to-question generation. *Computational Linguistics*, 41.

M Chi. 1994. Eliciting self-explanations improves understanding,. *Cognitive Science*, 18(3):439477.

Xinya Du, Junru Shao, and Claire Cardie. Learning to ask: Neural question generation for reading comprehension. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 1.

Charles J. Fillmore. 1968. The case for case. *Emmon W. Bach and Robert T. Harms, editors, Universals in Linguistic Theory. Holt, Rinehart & Winston*, page 188.

Patrick J Finn. 1975. A question writing algorithm. *Journal of Reading Behavior*, 7(4).

Melinda D. Gates. 2008. Generating reading comprehension look back strategy questions from expository texts. *Masters thesis, Carnegie Mellon University*.

Michael Heilman. 2011. Automatic factual question generation from text.

Michael Heilman and Noah A. Smith. 2009. Question generation via overgenerating transformations and ranking.

Matthew Honnibal and Mark Johnson. 2015. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1378, Lisbon, Portugal. Association for Computational Linguistics.

Hidenobu Kunichika, Tomoki Katayama, Tsukasa Hirashima, and Akira Takeuchi. 2004. Automated question generation methods for intelligent english learning systems and its evaluation. *Proc. of ICCE2001*, page 11171124.

Chin-Yew Lin. 2008. Automatic question generation from queries.

Ming Liu, Rafael Calvo, and Vasile Rus. 2012. G-asks: An intelligent automatic question generation system for academic writing support. *Dialogue & Discourse*, 3(2):101124.

Prashanth Mannem, Rashmi Prasad, and Aravind Joshi. 2010. Question generation from paragraphs at upenn: Qgstec system discription. *Proceedings of QG2010: The Third Workshop on Question Generation.*

Karen Mazidi and Rodney D. Nielsen. 2015. Leveraging multiple views of text for automatic question generation. *Lecture Notes in Computer Science Artificial Intelligence in Education*, page 257266.

R. Mitkov. 2006. Comtuter-aided generation of multiple-choice tests. *International Conference on Natural Language Processing and Knowledge Engineering, 2003. Proceedings. 2003.*

J. Mostow and W Chen. 2009. Generating instruction automatically for the reading strategy of self questioning. *Proceeding of the Conference on Artificial Intelligence in Education*, pages 465–472.

Vasile Rus, Zhiqiang Cai, and Arthur Graesser. 2008. Question generation : Example of a multi-year evaluation campaign. *Online Proceedings of 1st Question Generation Workshop.*

Eriks Sneiders. 2002. Automated question answering using question templates that cover the conceptual model of the database. *Natural Language Processing and Information Systems Lecture Notes in Computer Science*, page 235239.

Sandeep Subramanian, Tong Wang, Xingdi Yuan, Saizheng Zhang, Adam Trischler, and Yoshua Bengio. 2017. Neural models for key phrase detection and question generation.

Josh Tenenberg and Laurie Murphy. 2005. Knowing what i know: An investigation of undergraduate knowledge and self-knowledge of data structures. *Computer Science Education*, 15(4):297315.

Xuchen Yao, Gosse Bouma, and Yi Zhang. 2012. Semantics-based question generation and implementation. *Dialogue & Discourse*, 3(2).

Xingdi Yuan, Tong Wang, Caglar Gulcehre, Alessandro Sordoni, Philip Bachman, Saizheng Zhang, Sandeep Subramanian, and Adam Trischler. 2017. Machine comprehension by text-to-text neural question generation. *Proceedings of the 2nd Workshop on Representation Learning for NLP.*

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2017. Neural question generation from text: A preliminary study. *Lecture Notes in Computer Science*, pages 662–671.