Anaphora Reasoning Question Generation Using Entity Coreference

Kimihiro Hasegawa[1], Takaaki Matsumoto[2], and Teruko Mitamura[2]

[1]Kobe University
[2]Carnegie Mellon University

ljnjzbo417@gmail.com, {tmatsumo, teruko}@andrew.cmu.edu

Abstract

We propose an approach for Anaphora Reasoning Question Generation from a plain text: generating a question which needs anaphora resolution to answer. This is one type of Multiple Sentence Reasoning Question, a paragraph-level question which needs more than one sentence in a context to answer. We apply our system to Wikipedia articles and, based on our evaluation, our system generates more Anaphora Reasoning Question compared to the current state-of-the-art neural question generation model which intends to generate a paragraph-level question by around 30%.

1 Introduction

Reading Comprehension (RC), understanding what a text says, is one fundamental area in Natural Language Processing (NLP), and it relates to other areas of NLP, such as Dialogue, Question Answering, and Machine Translation. Although many studies in those areas have been done towards processing a single sentence as the main target, recently a context is drawing attention from researchers in those areas. Since the meaning of a sentence changes depending on a context, reading a context plays a crucial role in understanding even a single sentence. Along with this direction, there is a study about a parser, called Abstract Meaning Representation (AMR), by O'Gorman et al. (2018). They made a corpus for building a parser by retrieving the meaning of sentences and sentence relationship in a context and representing them in the AMR format.

While a machine utilizes a parser to understand a context, humans can deepen their understanding on a context by using questions, especially when a question represents a relationship between sentences. Through the process of tackling questions which need more than one sentence to answer and checking their correct answers, humans can comprehend the relationships between sentences better. Hence, question-answer pairs can be a rich source of sentence relationship: relationships between sentences in a context can be extracted from paragraph-level questions.

S15:	When *Andrew* goes home after baseball, *he* likes to eat a snack.
S16:	*He* eats carrots and bananas.
S17:	If *he* is a good boy, *his* mom, Mrs. Smith, sometimes gives *him* milk and cookies.
S18:	Afterwards, *Andrew* finishes *his* homework.
Q15:	Who likes to eat a snack?
Q16a:	Who eats carrots and bananas?
Q16b:	What does he eat?
Q16b*:	What does Andrew eat?
Q18:	Who finishes his homework?

Figure 1: Target Examples.[1]

Motivated by this idea, we propose a method to generate Anaphora Reasoning Question (ARQ) using entity coreference without human annotation. The overview of our system is the following: 1) find a sentence with at least one pronoun which refers to a specific entity in another sentence in plain text. 2) transform the sentence into a question. 3) if the pronoun still exists in the question, then replace the pronoun with the specific entity that the pronoun refers to. For instance, in Figure 1, S16 contains a pronoun, ”He”, which refers to ”Andrew” in S15. When we transform S16 into questions, we get Q16a and Q16b. Since Q16b has the pronoun, ”he”, which refers to ”Andrew”, we replace the pronoun with ”Andrew”. Then, we get Q16b* after the replacement. In order to answer Q16a or Q16b*, not only S16 but also S15 are necessary to figure out what ”He” refers to in this context. We will explain the detail of our pipeline in Chapter 3.

We conduct an evaluation of the generated questions by our system to check what proportion of them are acceptable ARQ in terms of Grammatical Correctness, Answer Existence, and The Number of Necessary Sentences. For comparison, we evaluate questions generated by a neural Question Generation (QG) model, CorefNQG, by Du and Cardie (2018), which also intends to generate questions from more than one sentence. We will describe the detail

[1] S15, 16, 17, 18 are extracted from a paragraph about children stories in the MCTest by Richardson et al. (2013). Entities in italics, *Andrew, he, He*, etc, refer to Andrew. Q15, Q16a, Q16b, Q16b*, Q18 are generated questions by our system.

of our evaluation metrics and the results in Chapter 4.

2 Related Works

In the situation that neural network models have been gaining attention from many NLP fields, there have been increasing demands for rich and large-scale question-answer-pair datasets. Rajpurkar et al. (2016) created Stanford Question Answering Dataset (SQuAD) which consists of more than 100k question-answer pairs with various types of reasoning, which encourages researchers in NLP communities to build neural network models. This dataset is created by posing a paragraph from Wikipedia to crowd workers and asking them to create questions about the paragraph. According to the paper, Multiple Sentence Reasoning Question (MSRQ), which needs more than one sentence to answer, accounts for around 13%. Using a similar method, Khashabi et al. (2018) created Multi-Sentence Reading Comprehension (MultiRC) corpus from selected paragraphs in various sources: News articles, Wikipedia articles, and Fictions, where one of the Fictions is from MCTest by Richardson et al. (2013). The MultiRC corpus consists of paragraphs, questions, and several choices for answer-options, where MSRQ accounts for around 60% according to the paper.

Corresponding to these datasets creation by human annotation, which gets expensive and time-consuming as the scale of a dataset becomes large, there have been researches on Question Generation by machines. Two approaches are chosen for machine QG: neural QG and rule-based QG. Du and Cardie (2018) built a neural network model incorporating with coreference knowledge for paragraph-level question generation. The model, CorefNQG, generates a dataset with over 1 million question-answer pairs from 10,000 top-ranking Wikipedia articles. Whereas, Heilman and Smith (2010) built a rule-based QG model with overgenerating and scoring, which transforms a single sentence into multiple types of questions. Satria and Tokunaga (2017) built a rule-based QG model which split a sentence with non-restrictive clause into two independent sentences, generating reference questions with multiple answer choices.

Also, there is close research with ours: Araki et al. (2016) proposed an approach for rule-based QG from a human-annotated text utilizing specific inference steps over multiple sentences. They generate not only ARQ but also event coreference questions and paraphrase questions.

3 Approach

In this section, we delineate the detail of our pipeline for generating Anaphora Reasoning Question. Figure 2 shows an overview of our pipeline. A box is a data on that step, and an arrow is a process to get the next data. Paragraphs below explains each process, an arrow in Figure 2.

Paragraph Selection: We retrieve articles from the dataset generated by CorefNQG, in order to make a comparison between CorefNQG and our system. Referring to the condition of article selection used in MultiRC, i.e., the number of sentences in an article is more than 5 and less than 19, we select articles which have more than 7 and less than 16 sentences. When an article is very short, the probability of detecting entity coreference gets low. In addition, when an article is very long, the accuracy of entity coreference detection decreases. In general, it is difficult to find entity coreference between two entities when those two locate far away from each other.

Entity Corefernce Detection and Sentence Segmentation: In order to get entity coreference clusters, we apply to those selected articles NeuralCoref[2], which detects entity coreference in an article using a neural network, extended on Spacy[3]. To split an article into sentences, we use Spacy's sentence segmentation which uses its dependency parse to determine sentence boundaries. As a supplement to the tool, we also use Segtok[4] which is a pattern based segmentation tool.

Entity Coreference Selection: Among the detected clusters, we pick clusters based on the most representative entity in a cluster: we discard a cluster if the most representative entity is either a pronoun or determiner/possessive determiner + noun/noun phrase. This is because the target of our system is a cluster which has a proper noun as the most representative entity.

Source Sentence Selection: For each candidate entity in a selected cluster, we check the original sentence where the candidate entity comes from. We discard a sentence when it contains the most representative entity because a question generated from that type of sentence does not require multiple sentences to answer. For instance, suppose we generated a question from S15 in Figure 1 to ask what "he" refers to. One generated question could be Q15, but this question can be answered by looking at S15 even though it asks about a pronoun which refers to a proper noun.

Question Generation: We apply the Heilman's question generation tool[5] to generate questions from each selected sentence. The tool could generate both Yes/No questions and Wh/How questions, but we only generate Wh/How questions considering that the dataset by CorefNQG contains only Wh/How questions.

Entity Coreference Resolution: The Heilman's tool generates multiple questions from one sentence. In some questions, a pronoun which can be replaced with the most representative entity is replaced with an interrogative. In other questions, where a target pronoun remains, we replace the pronoun with the most representative entity utilizing Spacy's dependency parser: checking if a remain-

[2] https://github.com/huggingface/neuralcoref
[3] https://spacy.io/
[4] https://github.com/fnl/segtok
[5] https://github.com/sumehta/question-generation

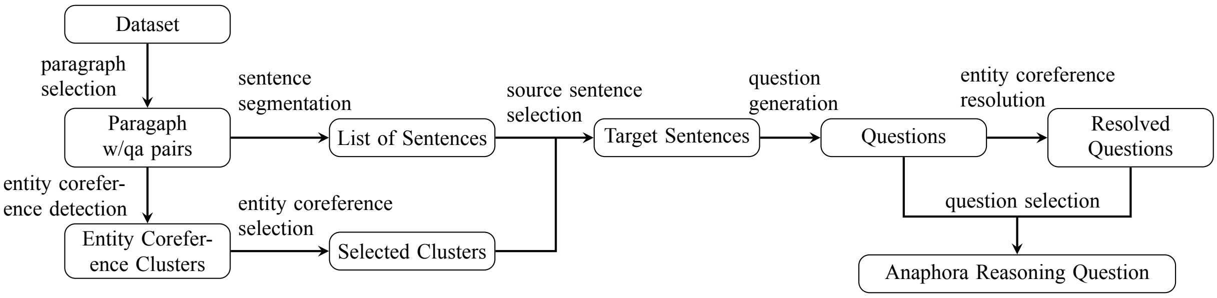

Figure 2: Pipeline of Our Anaphora Reasoning Question Generation.

<Grammatical Correctness>
check if the question is syntactically well-formed.
1 (best) : the question has no grammatical error.
2 : the question has 1 or 2 grammatical errors.
3 (worst) : the question has 3 or more grammatical errors.
<Answer Existence>
check if the answer to the question can be inferred from the article.
1 : the answer to the question can be inferred from the article.
2 : the answer to the question cannot be inferred from the article.
<The Number of Necessary Sentences>
check how many sentences are required to answer the question. This metric is used only when Answer Existence is 1.

Figure 3: Evaluation Metrics.

ing pronoun has the same head in the question's dependency as that in the original sentence's dependency. For instance, in Figure 1, "He" is replaced with "Who" in Q16a, while "he" still remains in Q16b. In order to make Q16a more specific and clear, we replace "he" with "Andrew" by checking if "he" in Q16a has the same head-dependent relationship with "eat" as the relationship between "He" and "eats" in S16.

Question Selection: Even after Entity Coreference Resolution is performed, some questions still have pronouns. In order to make a question less ambiguous, we discard a question with pronouns by utilizing Spacy's part-of-speech tag. Suppose we get Q18 from S18 in Figure 1. The pronoun, "his", makes Q18 ambiguous, because "his" could refer to another boy's name in the context.

4 Experiment

We conducted a human evaluation on 400 question-answer pairs derived from 57 articles: 200 QA pairs by our system from 40 out of 57 articles, and 200 QA pairs by the neural QG system from 57 out of 57 articles. Based on the evaluation metrics by Araki et al. (2016), our metrics for evaluating generated questions are Grammatical Correctness, Answer Existence, and The Number of Necessary Sentences. The details of our metrics are described in Figure 3.

Paragraph: The flower may consist only of these parts, as in (3) *willow*, where each flower comprises only (4) *a few stamens or two carpels*. ... The individual members of these surrounding structures are known as (5) *sepals and petals* (or tepals in flowers such as "Magnolia" where sepals and petals are not distinguishable from each other). The outer series (calyx of sepals) is usually (6) *green and leaf-like*, and functions to protect the rest of the flower, especially the bud. (S1) The inner series (corolla of petals) is, in general, white or brightly colored, and is more delicate in structure. (S2) It functions to attract insect or bird pollinators. ...
Questions by Our System:
Q1a: What is the inner series more delicate in?
Q1b: What is, in general, white or brightly colored?
Q2a: What functions to attract insect or bird pollinators?
Q2b: What the outer series (calyx of sepals) functions to attract ?
Questions by CorefNQG:
Q3: In what language is the flower of the flower located ?
Q4: What does each flower have ?
Q5: What are the individual members of these surrounding structures called ?
Q6: What is calyx of sepals ?

Figure 4: Example Questions by Our System and CorefNQG. [6]

4.1 Results

Table 1 shows the result of the human evaluation on generated questions about Grammatical Correctness (GC), Answer Existence (AE) and The Number of Necessary Sentences (Num) by one evaluator. As for GC, CorefNQG is better than our system, though the sum of GC 1 and 2 of our system is the same as that of CorefNQG. From the evaluation of AE, our system generates more answerable questions than the neural model by about 25%. About Num, we label "None" for a question whose AE is 2. Our system generates questions which need more than one sentence than the neural model by around 30%.

4.2 Analysis

Since our system uses multiple tools to generate ARQ, the errors of each tool accumulate, which results in mistakes in the final results. Especially, Entity Coreference detection and Heilman's QG tool are the main reasons, for instance, in Figure 4, Q2b is generated from S2 with

[6] Wavy lines are sentences used to generate questions by our system. Words in italics are answers picked by CorefNQG.

	Grammatical Correctness			Answer Existence		The Number of Necessary Sentences			
Models	1 (best)	2	3 (worst)	1 (Exist)	2 (Not Exist)	1	2	>	None
Our System	115	68	17	**159 (0.795)**	41	96 (0.48)	**62 (0.31)**	1	41
CorefNQG	**127**	56	17	110 (0.55)	90	107 (0.535)	3 (0.015)	0	90

Table 1: Evaluation Results.

replacing “It” with “the outer series (calyx of sepals)”, which is a fault of entity coreference detection. As it is seen in Q1b, Heilman’s QG tool tends to generate not-accurate questions from a complex-structured sentence, such as a sentence with a restrictive clause or a parenthesis. Whereas, as it is seen in Q3, CorefNQG tends to generate questions with using an identical phrase multiple times. As for AE, one possible reason why our system generates more answerable questions than CorefNQG is that our system transforms sentences into questions by replacing an entity with an interrogative, for instance, ”It” with ”What” in Q2a. Whereas, CorefNQG uses phrases in the context to generate questions, which results in an inconsistency between an interrogative in a question and the rest of it, represented in Q3: ”what language” for ”flower” is ”located”. According to our evaluation, ARQ, even MSRQ, is rarely found in generated questions by CorefNQG, which does not correspond closely to the evaluation results by Du and Cardie (2018): 36.42% examples of the original test set to train the neural model require coreference knowledge to answer.

5 Conclusion and Future Work

This paper presents a rule-based approach without human annotation to generate a question which needs to look more than one sentence to answer. Specifically, we generate a question from a sentence with a pronoun which refers to a proper noun in a context in the way that the question asks the pronoun. Our experiment shows that our system outperforms the state-of-the-art neural QG model which intends to generate MSRQ.

Possible future work includes the improvement of entity coreference detection and to widen the variety of questions by generating other types of questions which also need more than one sentence to answer, such as event coreference questions and paraphrase questions. Although we employ all questions generated from one sentence except a question with a pronoun, we need a method to pick one question from them. Based on the idea that a question could represent a relationship between sentences, our next step would be to extract sentence relationship in MSRQ from a corpus, such as MultiRC, in ordet to improve machine Reading Comprehension. In addition, although in this study one evaluator judged the generated questions, we can employ more evaluators and compare the results to each other, in order to minimize the subjective opinions of evaluators.

References

Jun Araki, Dheeraj Rajagopal, Sreecharan Sankaranarayanan, Susan Holm, Yukari Yamakawa, and Teruko Mitamura. Generating questions and multiple-choice answers using semantic analysis of texts. In *COLING*, 2016.

Xinya Du and Claire Cardie. Harvesting paragraph-level question-answer pairs from wikipedia. In *Association for Computational Linguistics (ACL)*, 2018.

Michael Heilman and Noah A Smith. Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–617. Association for Computational Linguistics, 2010.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. Looking beyond the surface:a challenge set for reading comprehension over multiple sentences. In *Proceedings of North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018.

Tim O’Gorman, Michael Regan, Kira Griffitt, Ulf Hermjakob, Kevin Knight, and Martha Palmer. Amr beyond the sentence: the multi-sentence amr corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3693–3702. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/C18-1313.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. URL https://aclweb.org/anthology/D16-1264.

Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203. Association for Computational Linguistics, 2013. URL http://aclweb.org/anthology/D13-1020.

Arief Yudha Satria and Takenobu Tokunaga. Automatic generation of english reference question by utilising nonrestrictive relative clause. In *CSEDU*, 2017.