Pseudo Ambiguous and Clarifying Questions Based on Sentence Structures Toward Clarifying Question Answering System

Yuya Nakano[1], **Seiya Kawano**[2,1], **Koichiro Yoshino**[2,1],
Katsuhito Sudoh[1] **and Satoshi Nakamura**[1]
[1]Nara Institute of Science and Technology, 8916-5, Takayama, Ikoma, Nara, 6300192, Japan
[2]Guardian Robot Project (GRP), Institute of Physical and Chemical Research (RIKEN),
2-2-2, Hikaridai, Seika, Soraku, Kyoto, 6190288, Japan
{seiya.kawano, koichiro.yoshino} at riken.jp
{nakano.yuya.nw9, sudoh, s-nakamura} at is.naist.jp

Abstract

Question answering (QA) with disambiguation questions is essential for practical QA systems because user questions often do not contain information enough to find their answers. We call this task *clarifying question answering*, a task to find answers to ambiguous user questions by disambiguating their intents through interactions. There are two major problems in building a clarifying question answering system: data preparation of possible ambiguous questions and the generation of clarifying questions. In this paper, we tackle these problems by sentence generation methods using sentence structures. Ambiguous questions are generated by eliminating a part of a sentence considering the sentence structure. Clarifying the question generation method based on case frame dictionary and sentence structure is also proposed. Our experimental results verify that our pseudo ambiguous question generation successfully adds ambiguity to questions. Moreover, the proposed clarifying question generation recovers the performance drop by asking the user for missing information.

1 Introduction

Question answering (QA) is a conventional task of natural language processing to provide answers for given user questions. The advance of neural network-based QA systems has led to a variety of benchmark datasets of the QA task (Rajpurkar et al., 2016; Yang et al., 2018). These benchmarks define the problem of QA as predicting a corresponding phrase (span) in documents to a given question when the system has both questions and target documents.

Most QA tasks defined in existing benchmark QA datasets assumes that the given questions have enough information for answering. However, real questions given by users are often ambiguous because users frequently forget to mention important terms or may hesitate. It is thus not always easy to derive clear answers for such ambiguous user questions. For example, when a user says, "What is the masterpiece drawn by Leonardo da Vinci?", the system cannot determine an answer because Leonardo da Vinci created several notable masterpieces (Figure 1; ambiguous Q). Taylor (Taylor, 1962) defined four level categories of user states in information search.

- Q1 The actual, but unexpressed request
- Q2 The conscious, within-brain description of the request
- Q3 The formal statement of the request
- Q4 The request as presented to the dialogue agent

Most existing QA systems target Q3 or Q4; however, it is required for systems to answer questions categorized into Q2. In other words, user questions do not always contain sufficient information for finding the answer; however, systems can fill in the gap by asking back users directly (Small et al., 2003; Bertomeu et al., 2006; Kato et al., 2006; Aliannejadi et al., 2020). SQuAD 2.0 (Rajpurkar et al., 2018) defined "unanswerble questions" in their dataset; however, our problem definition is that the system has potential answers but does not have enough information to reach them.

Using clarifying questions is a common method in conversational search (Radlinski and Craswell, 2017; Trippas et al., 2018; Zhang et al., 2018; Qu et al., 2020); it ascertains the user's retrieval intent with questions if the system cannot capture this from the initial request. Thus, the system can get additional information to the initial request using a clarifying question to make the user's intent clearer. In the previous example, the system can ask the user, "Which museum displays this masterpiece?" or "What is the motif?" to disambiguate possible answers to the given question (Figure 1; clarifying Q1 and Q2). Some existing work tackled this

Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering, pages 31 - 40

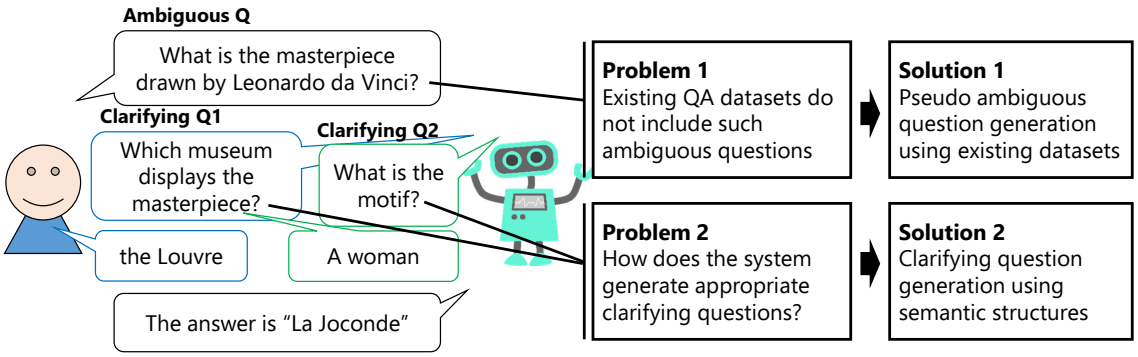

Figure 1: The problem of clarifying QA

problem on a QA system using question paraphrasing (Otsuka et al., 2019) and building ambiguous question answering datasets (Min et al., 2020).

However, it is not easy to build a dataset that covers any variation of ambiguous questions because of the diverse variety of ambiguity in questions (Figure 1; Problem 1). Moreover, even if we can define the variation of ambiguity; it is still challenging to find appropriate clarifying questions for the disambiguation to shape the system answers (Figure 1; Problem 2).

Sentence structures have an essential role in clarifying the meaning because we control the sentence clarity by modifiers in syntax. This indicates that the sentence generation system can also control sentences' clarity by focusing on sentence structures. Based on this idea, in this work, we propose a *pseudo* ambiguous question generation method for covering variations of the ambiguous question, which are derived from clear questions collected in existing QA datasets (Figure 1; Solution 1). The proposed method focuses on the syntax structures of question sentences to add ambiguity by eliminating some parts while considering grammatical roles from syntax point of view. We also propose a clarifying question generation method based on the case frame, which uses the syntax and semantic information of ambiguous questions (Figure 1; Solution 2). The clarifying question generation makes it possible to disambiguate the user's meaning by interacting with the user directly to improve the QA system performance.

We conducted two experiments to investigate the quality of proposed generation systems. Qualities of the *pseudo* ambiguous questions are evaluated by both the QA system and the human subjective test. The performance of the clarifying question generation is investigated by QA system performance using both the ambiguous questions and answers to the clarifying questions given by crowdworkers.

Section 2 sets forth our problem definition and system overview. Section 3 describes the pseudo ambiguous question generation method. Section 4 explains the proposed clarifying question generation method that uses sentence structures. Section 5 shows the evaluation setting and system performance to verify the ability of our generation system. We clarify the position of our system in relation to existing systems in Section 6, and then conclude this work in Section 7.

2 System overview

Our final goal is to build a clarifying question answering system that can ask a question back to users if the given questions do not contain sufficient information to distinguish the answer. We call such questions as *ambiguous questions*. Figure 2 shows the overall system.

We extract questions from existing QA datasets to modify them to pseudo ambiguous questions because building ambiguous question datasets is costly (Aliannejadi et al., 2019; Xu et al., 2019). Most of the existing QA datasets consist of pairs of clear questions and corresponding text spans on target documents. These questions are defined clearly to distinguish the answer terms from the document. In other words, if human experts receive these questions, they can find the answer from the documents even if it takes a lot of time. Our proposal eliminates some important parts of these questions to generate pseudo ambiguous questions using their syntax information. In the example presented in Figure 2, the system adds ambiguity to the question by removing the verbal phrase that corresponds to the verb "developed."

When the QA system receives an ambiguous question from the pseudo ambiguous question gen-

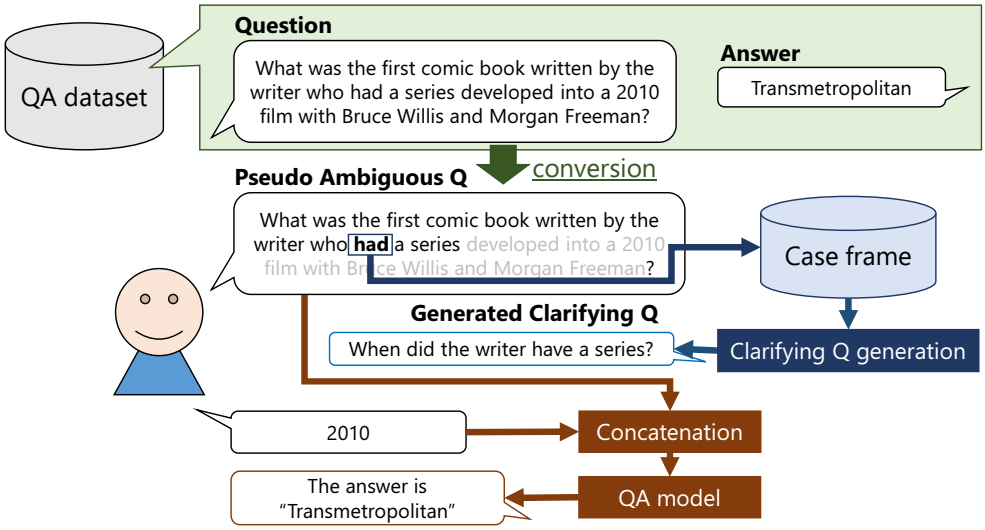

Figure 2: System overview

erator, the QA system needs to generate a clarifying question. We focus on predicates in the ambiguous question and their missing cases on the syntax to generate the clarifying question. We used the case frame dictionary to estimate the missing case of the extracted predicates. In the example in Figure 2, the system generates the clarifying question "When did the writer have a series?"[1] because the system found that the adverbial modifier of "had" in the ambiguous question is missing. The system receives the answer to the clarifying question and then runs the QA model using both the ambiguous question and the answer to the clarifying question. Technical details are described in the following sections.

3 Pseudo ambiguous question generation

It is not realistic to collect all possible varieties of ambiguous questions because possible ambiguous questions given to the QA system are diverse and depend on the situation that the users are facing. In this paper, we present a method to generate pseudo ambiguous questions by modifying questions in existing QA datasets. We apply syntax parsing to question sentences to focus on modifiers, which have a role in clarifying the question's intent, and then eliminate them from the questions to make the sentences ambiguous. This section describes the generation process and its evaluations.

[1]Formally, this question should be "When did the write have *the* series," but here we explain the system process with our system outputs.

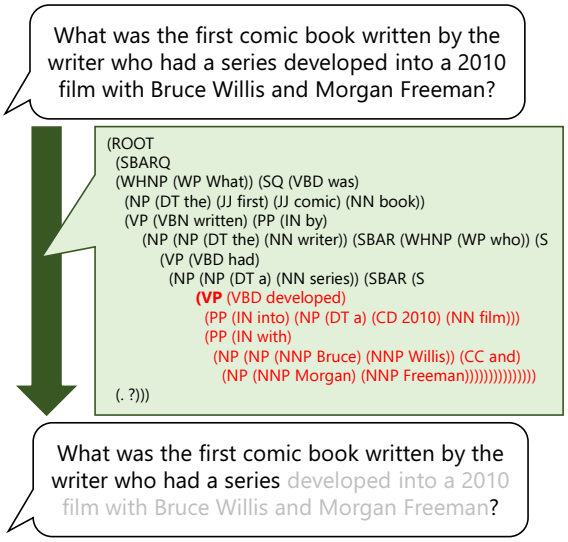

Figure 3: Generation of ambiguous question with removal of verbal phrase (VP)

3.1 Question generation using syntax information

A generation example is shown in Figure 3. In this example, the system generates an ambiguous question "What was the first comic book written by the writer who had a series?" while eliminating the verbal phrase indicated by "developed" because the phrase describes the detail of the antecedent "a series." We use the Stanford parser (Manning et al., 2014)[2] to get the syntax. Our system focuses on a verbal phrase (VP) and prepositional phrase (PP) as chunks to be removed.

[2]https://nlp.stanford.edu/software/lex-parser.shtml

	EM	F1
Original (w/o modification)	55.92	70.15
VP	10.88	28.70
PP	13.73	34.41
Mixed	13.69	33.73

Table 1: Evaluation scores of QA system given ambiguous questions

	Total	Normal	Irregular
#questions	200	71	129
VP	1.928	2.008	1.9001
PP	2.351	2.492	2.265
Mixed	2.371	2.479	2.292

Table 2: Human evaluation of sentence quality

3.2 Evaluation of pseudo ambiguous questions

We evaluated the proposed pseudo ambiguous question generation from two viewpoints: increased ambiguity and sentence quality, measured by QA system accuracy and human subjective evaluation, respectively. In the experiment, we used the HotpotQA dataset (Yang et al., 2018)[3], which consists of training and development sets. Note that the test set is not distributed to be used on their leaderboard; we used the development set as our test set. We used the training set to train the QA model to be used for the first evaluation. We modified all 7,405 sentences in the development set to pseudo ambiguous questions. As the QA model, we used a BERT-based model with the same setting (Devlin et al., 2019), which predicts a span in the given document set. Our system generated one ambiguous question for each original question in this evaluation by eliminating the shortest phrase. We tried three elimination strategies: removing a VP, removing a PP, and removing a VP and PP's shortest phrase (Mixed).

3.2.1 Evaluation on QA accuracy

We used exact matching (EM) and F1 scores to evaluate the QA accuracy. EM indicates the exact matching accuracy of the extracted answer from the target documents. QA answers often consist of several words; thus, the harmonic mean of precision and recall of word matching is also used (F1).

Table 1 shows the result, which indicates that the accuracy of QA systems decreased in any condition; even our system removed the shortest phrase for each question. VP had the most significant impact on decreasing the score; this is probably because VPs are more widespread than PPs.

3.2.2 Evaluation of sentence quality

In the human subjective evaluation, we hired three annotators who have comparable English reading skills to natives and asked them to evaluate sentences using the following three grades.

- 3: Fluent English sentence
- 2: Grammatically correct English sentence
- 1: Incorrect English sentence

We randomly sampled 200 sentences from the generated 7,405 sentences for the evaluation.

Table 2 shows the result. # indicates frequencies. We categorized the selected 200 sentences into "Normal" and "Irregular" forms with their interrogative position. The "Normal" form sentences start from the interrogative. The "Irregular" has the interrogative on other parts. These results verified that the "Mixed" strategy achieved a suitable naturalness score of 2.371. However, the "VP" strategy has lower scores because it eliminates widespread spans and often removes necessary parts of questions. The "Normal" form had better scores than the "Irregular" form. Their sentence structures probably cause this; interrogatives in the "Irregular" form are sometimes placed on the leaves of syntax trees.

4 Clarifying question generation

We built clarifying question generation system toward a clarifying question answering system, asking a question back to the questioners. The proposed system generates clarifying questions using predicate-argument structures; it finds predicates in ambiguous questions and generates questions to clarify their arguments. We used the case frame dictionary (Kawahara and Kurohashi, 2006; Kawahara et al., 2014) for the generation, which consists of frequencies of cases and arguments depending on predicates. This section describes the technical details of clarifying question generation.

4.1 Case frame

Words or phrases that have specific roles to predicates on dependency structures are called arguments, with their semantic/syntactic roles (cases). For example, in the sentence "I saw a girl," "see

[3]https://hotpotqa.github.io/

Predicate sense	case	argument	Freq.
eat:1	-	-	12,645
	nsubj	-	9,682
		they	1,036
		I	944
		you	896
		...	...
eat:2	-	-	12,073
	dobj	-	9,366
		lunch	3,443
		meal	3,265
		breakfast	2,081
		...	...

Table 3: Examples in case frame

Case	Freq.	Case	Freq.
nmod	81,442	amod	951
nsubj	60,702	parataxis	452
dobj	49,679	acl:relcl	444
nsubjpass	23,910	acl	285
advmod	17,991	cc:preconj	282
dep	6,817	csubjpass	218
conj	5,335	nmod:poss	177
cc	5,152	nummod	175
advcl	4,943	csubj	143
xcomp	4,521	expl	108
ccomp	4,461	iobj	100
compound	1,740	neg	83
cop	1,554	mwe	62
case	1,529	appos	37
compound:prt	1,344	nmod:npmod	27
nmod:tmod	1,132	discourse	6

Table 4: Frequency of each case in the training data

(saw)" is a predicate, and "I" and "a girl" have roles to the predicate as "nsubj (noun subject)" and "dobj (direct object)." The case frame is a statistically collected dictionary consisting of cases, arguments, and frequencies (case frame frequency) for each predicate. Kawahara et al., (2014) is distributing a case frame dictionary, which is based on parsing results of the Stanford parser to a billion-sentences English corpus. An example of the case frame dictionary is shown in Table 3. Each predicate entry has a corresponding predicate sense with its usage (see numbers after predicates in Table 3).

4.2 Generation and selection process

Our clarifying question generation outputs clarifying questions to a given ambiguous question sentence by the following four steps.

1. Predicate identification
2. Missing case extraction
3. Target case decision
4. Interrogative word decision

Figure 4 illustrates the generation and selection process. We used the Stanford parser in predicate identification, using verbal tags: VB, VBD, VBG, VBN, VBP, and VBZ. We extracted triples of a predicate, an argument, and its case of these identified predicates.

In the missing case extraction, the system extracts missing cases (possible but unseen cases) of identified predicates. The system generates clarifying questions for filling these missing cases. In the example of Figure 4, the "adverbial modifiers (adv-mods)" of "write" and "have" are extracted.

Target case decision prioritizes missing cases with case frequency and the relative position of predicates; frequent cases and predicates on postposed places have higher priority because frequent cases in questions probably contain essential information. Case frequencies are calculated from the QA system's training data, in our case, the training set of HotpotQA. Any questions in the training set are parsed to count the case frequency as shown in Table 4.

Once the target predicate and the target case are decided, the case frame dictionary is used again to determine the interrogative word. The system looks up the entry of the decided predicate and case in the dictionary. Then the system picks up the most frequent interrogative word corresponding to them (interrogative word decision). The system generates clarifying questions using the decided interrogative word, predicate, and depending phrase to the predicate.

5 Experiments

We evaluated the proposed clarifying question generation system. We gave the pseudo ambiguous question generated by the method presented in Section 3 to the clarifying question generation described in Section 4.

5.1 Experimental setting

We used the HotpotQA dataset as the original QA dataset of our system. The HotpotQA dataset records many complicated sentences with several modifiers because the dataset was built for QA systems with multi-hop reasoning. As the QA model, we used a BERT-based model with the same setting (Devlin et al., 2019), which predicts a span in the given document set. Specifically, we used the BERT-Base-Uncased model as the pre-trained model. In the fine-tuning, the batch size was 12, the training rate was $3e^{-5}$, and the number of epochs was 2.

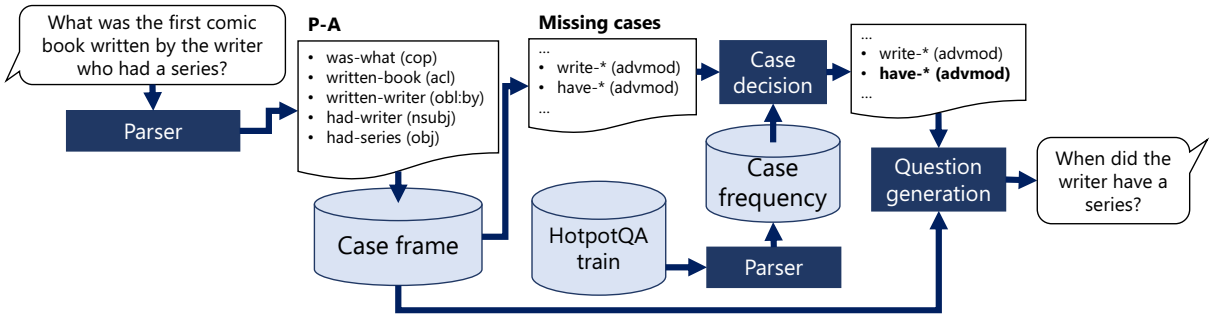

Figure 4: Procedure to generate clarification questions

As indicated in Figure 2, the pseudo ambiguous question is given to the system and then the system generates a clarifying question to the ambiguous question. The system receives the user's reply to the clarifying question in the evaluation. In our evaluation, we allowed only one clarification for each question.

We generated pseudo ambiguous questions from the development set of the HotpotQA dataset as described in Section 3. In this experiment, we generated several pseudo ambiguous questions from one sentence with the following conditions.

1. Eliminated words are less than 50% of the original question.
2. Eliminated words do not contain any interrogative words.
3. Eliminated parts are selected from both VPs and PPs.
4. QA system results are changed from correct to incorrect by the modification.

The first and second points are necessary to generate interrogative sentences. For the fourth point, we input both the original question and the pseudo ambiguous question with the elimination to a QA model and compared their results as shown in Figure 5. This is because our focus in this experiment is whether the clarifying question can recover important information by asking a question back to the user. We finally selected 850 sentences that match the above conditions.

We generated clarifying questions to these 850 pseudo ambiguous questions. We used crowdsourcing to add the answer to the clarifying question. We showed the original question as "intent," the pseudo ambiguous question as "your question," and the clarifying question as "clarification question" to the crowdworkers and gave them the following instructions:

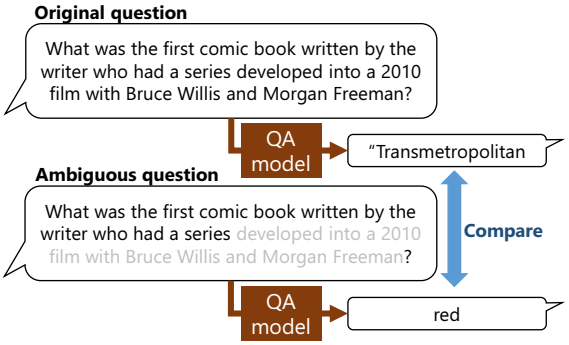

Figure 5: Comparison of QA results

> Assume that you are talking with a chat assistant. "Intention" indicates what you wanted to ask, and "your question" indicates what you said to the system. The system says a "clarification question" as a response to your question. First, select Yes/No according to whether the "clarification question" correctly specifies missing information of your "intention" or not. Then, write your answer for the "clarification question" in the shortest terms. Do not write the original question itself.

The crowdworkers thus evaluate the correctness of clarifying questions and then input the answer to the clarifying question. We assigned five crowdworkers for each sample and then determined the correctness label by the majority. We used all responses to clarifying questions to calculate the QA model accuracy. In other words, our evaluation score is calculated from $850 \times 5 = 4,250$ samples. We concatenated the received answers to the ambiguous questions to be used as the input of the QA model. We used the same QA model as in Section 3.2, the BERT-based fine-tuned model.

Category	EM	F1	#q	#eval
Yes+No	49.52%	57.28%	850	4,250
Yes	50.21%	57.82%	486	2,430

Table 5: Evaluation scores of the QA system given both ambiguous questions and answers to the clarifying questions. Category means the added correctness of the clarifying questions. #q and #eval indicate the numbers of used questions and evaluation samples.

5.2 Experimental results

For the correctness of clarifying questions, the ratio of samples evaluated as "Yes" was $486/850 = 0.572$. This indicates that our clarifying question generation method based on sentence structure and the case frame dictionary successfully generated clarifying questions to major questions; however, we still need to refine the method by focusing on the content words of questions.

Table 5 shows the accuracy of the QA system by inputting both ambiguous questions and generated clarifying questions. Note that scores are 0.0% if we give only ambiguous questions and 100.0% if we give the original question before adding the ambiguity. These results show that our clarifying question recovers 50% of lost information through interactions, which is lost in the modification process of a pseudo ambiguous question.

5.3 Analysis

Table 6 shows examples from the evaluation. In example 1, the pseudo ambiguous question generation removed the term "Jerry Goldsmith" and the clarifying question successfully got the word to recover the information. In example 2, the system also succeeded in recovering the removed information, but the QA system failed to output the correct answer by a small difference. In examples 3 and 7, the system's clarifying question is not appropriate, but the system output the correct answer. In examples 6 and 7, users may misunderstand their task and put a new question to clarify their original question. Recent search system interfaces probably cause this; the users usually give a new query to the system if their first search fails. We can improve the clarification quality in some cases; however, the system could get additional information to recover the information, even if the system failed to ask questions back to the users correctly. In general, when the ambiguous question was generated by eliminating PPs, our clarifying question successfully worked in many cases to ask back the phrase. Recovering VPs was more difficult for the system.

6 Related works

We built a generation system that clarifies user's requests by clarifying questions when the user's questions are ambiguous. There are two major approaches for building a QA system that can withdraw additional information to the initial ambiguous user query. One approach is based on paraphrasing, which paraphrases ambiguous sentences to clear sentences. The other major approach is using clarifying or confirmation questions, which is similar to our system. This section describes relationships to these works.

6.1 Paraphrasing approach

The paraphrasing approach's critical idea is converting given user questions to other forms (McKeown, 1983; Buck et al., 2017; Dong et al., 2017). This idea is similar to query expansion, which is used in the information retrieval area. It is often difficult for users to express their questions in clear language. This difficulty often causes ambiguous questions. This kind of works tackled this problem by presenting possible paraphrases of the given ambiguous question with their answers. However, such approaches do not work well if paraphrased questions do not contain the appropriate question for the user. Moreover, the system needs paraphrasing datasets to learn the paraphrasing models, which requires enormous annotation costs in the open domain (Min et al., 2020).

Otsuka et al., (2019) used syntactic structures to generate pseudo training examples for the paraphrasing approach. Our approach is similar to their works; however, we also used statistical information from the case frame to distinguish the clarified point to realize a dialogue-based system. The dialogue-based approach has an advantage in decreasing user interaction costs if the system can predict the clarifying point appropriately.

6.2 Clarifying approach

The second approach is giving clarifying questions to users, which is closer to our approach. The clarifying strategy has been used widely in conventional spoken dialogue systems because the systems sometimes fail the task by ambiguity caused by speech recognition or natural language understanding errors (Misu and Kawahara, 2006; Stoy-

#	Methods	sentence
1	**(O)** original	What is the name of the executive producer of the film that has a score composed by Jerry Goldsmith?
	(A) ambiguous	What is the name of the executive producer of the film that has a score composed?
	(C) clarifying	which composed?
	(R) reply to C	Jerry Goldsmith
	(G) gold	Ronald Shusett
	(QA w/ A)	Jerry Goldsmith
	(QA w/ A+R)	Ronald Shusett
2	**(O)** original	The lamp used in many lighthouses is similiar to this type of lamp patented in 1780 by Aime Argand?
	(A) ambiguous	The lamp used in many lighthouses is similiar to this type?
	(C) clarifying	what was similiar?
	(R) reply to C	lamp patented in 1780 by Aime Argand
	(G) gold	Argand lamp
	(QA w/ A)	oil lamp
	(QA w/ A+R)	Lewis lamp
3	**(O)** original	Lt Col. Stewart Francis Newcombe was a British army officer and associate of a military officerthat was given what title?
	(A) ambiguous	Lt Col. Stewart Francis Newcombe and associate of a military officerthat was given what title?
	(C) clarifying	which was the Newcombe and associate given?
	(R) reply to C	a military officer
	(G) gold	Lawrence of Arabia
	(QA w/ A)	British archaeologist, military officer, diplomat, and writer
	(QA w/ A+R)	Lawrence of Arabia
4	**(O)** original	According to the 2001 census, what was the population of the city in which Kirton End is located?
	(A) ambiguous	According, what was the population of the city in which Kirton End is located?
	(C) clarifying	where was the End located?
	(R) reply to C	population of the city in which Kirton End is located
	(G) gold	35,124
	(QA w/ A)	66,900
	(QA w/ A+R)	66,900
5	**(O)** original	Hatyapuri was a novel by the filmmaker of what nationality?
	(A) ambiguous	Hatyapuri was a novel of what nationality?
	(C) clarifying	what was novel?
	(R) reply to C	Hatyapuri
	(G) gold	Indian
	(QA w/ A)	Bengali
	(QA w/ A+R)	Bengali
6	**(O)** original	Which other Mexican Formula One race car driver has held the podium besides the Force India driver born in
	(A) ambiguous	Which other Mexican Formula One race car driver has held the podium besides the Force India driver?
	(C) clarifying	where did the car hold?
	(R) reply to C	When was the force India driver born?
	(G) gold	Pedro Rodriguez
	(QA w/ A)	1990/1/26
	(QA w/ A+R)	Pedro Rodriguez
7	**(O)** original	What relationship does Fred Gehrke have to the 23rd overall pick in the 2010 Major League Baseball Draft?
	(A) ambiguous	What relationship does Fred Gehrke have overall pick in the 2010 Major League Baseball Draft?
	(C) clarifying	when did the Gehrke have?
	(R) reply to C	What is the number of the overall pick?
	(G) gold	great-grandfather
	(QA w/ A)	Miami Marlin
	(QA w/ A+R)	23rd

Table 6: Examples of clarifying question answering. **O**, **A**, and **C** indicate an original question, ambiguous question generated from the original question, and the generated clarifying question, respectively. Crowdworkers saw these contexts and input "(R) reply to **C**". **G** is the correct answer to question **O** and **QA w/ A** is the output of the QA model given only the ambiguous question. **QA w/ A+R** uses both the ambiguous question and the reply to the clarifying question given by the crowdworkers.

anchev et al., 2014). Our system uses this idea to tackle a problem of question ambiguity in the QA system caused by the user's ability or lack of knowledge. In recent QA systems, there is a study to learn the re-ranking function of clarifying questions by deep neural networks (Rao and Daumé III, 2018). They also proposed a model based on a generative neural network to generate clarifying questions (Rao and Daumé III, 2019). These studies require triples of an ambiguous question, a clarifying question, and a corresponding fact. Building a large dataset to cover open-domain QA is costly. Our system does not require such data preparation cost and uses a general syntactic parser and the case frame dictionary built without specified annotations. The system can work on any QA datasets already developed in the existing work of QA systems.

Question generation is also widely researched by using generative models (Duan et al., 2017; Du et al., 2017; Sasazawa et al., 2019) or syntactic rules (Heilman and Smith, 2010). Our clarifying question generation is motivated by them.

7 Conclusion

In this paper, we worked on building a clarifying question answering system for ambiguous questions, questions with some necessary information

dropped. We proposed two-generation methods toward the clarifying question answering system: pseudo ambiguous question generation based on syntax and clarifying question generation based on sentence structures and case frame dictionaries. Our experimental results revealed that these generation methods worked to drop and to regain the important information in the original clear questions. The system used domain-independent syntactic and semantic information of questions; thus, the method can be applied to various QA domains. Moreover, our method does not require data annotation; we can extend existing QA datasets for the clarifying QA task.

As future work, we can integrate our model with other generative models. Another approach is to use pseudo ambiguous questions as training data of QA-related modules such as discriminative systems to predict or score given questions. Improving the model architecture is another issue, for example, network design to feed the whole dialogue history to the QA network.

References

Mohammad Aliannejadi, Julia Kiseleva, Aleksandr Chuklin, Jeff Dalton, and Mikhail Burtsev. 2020. Convai3: Generating clarifying questions for open-domain dialogue systems (clariq). *arXiv preprint arXiv:2009.11352*.

Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval*, pages 475–484.

Núria Bertomeu, Hans Uszkoreit, Anette Frank, Hans-Ulrich Krieger, and Brigitte Jörg. 2006. Contextual phenomena and thematic relations in database qa dialogues: results from a wizard-of-oz experiment. In *Proceedings of the Interactive Question Answering Workshop at HLT-NAACL 2006*, pages 1–8.

Christian Buck, Jannis Bulian, Massimiliano Ciaramita, Wojciech Gajewski, Andrea Gesmundo, Neil Houlsby, and Wei Wang. 2017. Ask the right questions: Active question reformulation with reinforcement learning. *arXiv preprint arXiv:1705.07830*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 875–886.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352.

Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 866–874.

Michael Heilman and Noah A. Smith. 2010. 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, Los Angeles, California. Association for Computational Linguistics.

Tsuneaki Kato, Jun'ichi Fukumoto, Fumito Masui, and Noriko Kando. 2006. Woz simulation of interactive question answering. In *Proceedings of the Interactive Question Answering Workshop at HLT-NAACL 2006*, pages 9–16.

Daisuke Kawahara and Sadao Kurohashi. 2006. Case frame compilation from the web using high-performance computing. In *LREC*, pages 1344–1347.

Daisuke Kawahara, Daniel Peterson, Octavian Popescu, and Martha Palmer. 2014. Inducing example-based semantic frames from a massive amount of verb uses. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 58–67.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Kathleen McKeown. 1983. Paraphrasing questions using given and new information. *American Journal of Computational Linguistics*, 9(1):1–10.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. Ambigqa: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797.

Teruhisa Misu and Tatsuya Kawahara. 2006. Dialogue strategy to clarify user's queries for document retrieval system with speech interface. *Speech Communication*, 48(9):1137–1150.

Atsushi Otsuka, Kyosuke Nishida, Itsumi Saito, Hisako Asano, and Junji Tomita. 2019. Specific question generation for reading comprehension. *Proceedings of the AAAI 2019 Reasoning and Complex QA Workshop*, pages 12–20.

Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W Bruce Croft, and Mohit Iyyer. 2020. Open-retrieval conversational question answering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 539–548.

Filip Radlinski and Nick Craswell. 2017. A theoretical framework for conversational search. In *Proceedings of the 2017 conference on conference human information interaction and retrieval*, pages 117–126.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. 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.

Sudha Rao and Hal Daumé III. 2018. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2737–2746.

Sudha Rao and Hal Daumé III. 2019. Answer-based adversarial training for generating clarification questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 143–155.

Yuichi Sasazawa, Sho Takase, and Naoaki Okazaki. 2019. Neural question generation using interrogative phrases. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 106–111.

Sharon G Small, Nobuyuki Shimizu, Tomek Strzalkowski, and Ting Liu. 2003. Hitiqa: A data driven approach to interactive question answering: A preliminary report. In *New Directions in Question Answering*, pages 94–104.

Svetlana Stoyanchev, Alex Liu, and Julia Hirschberg. 2014. Towards natural clarification questions in dialogue systems. In *AISB symposium on questions, discourse and dialogue*, volume 20.

Robert S Taylor. 1962. The process of asking questions. *American documentation*, 13(4):391–396.

Johanne R Trippas, Damiano Spina, Lawrence Cavedon, Hideo Joho, and Mark Sanderson. 2018. Informing the design of spoken conversational search: Perspective paper. In *Proceedings of the 2018 Conference on Human Information Interaction & Retrieval*, pages 32–41.

Jingjing Xu, Yuechen Wang, Duyu Tang, Nan Duan, Pengcheng Yang, Qi Zeng, Ming Zhou, and SUN Xu. 2019. Asking clarification questions in knowledge-based question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1618–1629.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W Bruce Croft. 2018. Towards conversational search and recommendation: System ask, user respond. In *Proceedings of the 27th acm international conference on information and knowledge management*, pages 177–186.