YORK UNIVERSITÉ UNIVERSITY

redefine THE POSSIBLE.

Context-Aware Question and Answer Generation from Car Manuals

Elnaz Delpisheh, Muath Alzghool, Aijun An, Heidar Davoudi,
Marjan Delpisheh, Emad Gohari and Sedigheh Mahdavi

Technical Report EECS-2019-01

August 15, 2019

Department of Electrical Engineering and Computer Science
4700 Keele Street, Toronto, Ontario M3J 1P3 Canada

Context-Aware Question and Answer Generation from Car Manuals

Elnaz Delpisheh, Muath Alzghool, Aijun An, Heidar Davoudi, Marjan Delpisheh,
Emad Gohari, and Sedigheh Mahdavi

Department of Electrical Engineering and Computer Science York University, Ontario, Canada
{elnaz,aan,alzghool,davoudi,immarjan,gohari,smahdavi}@eecs.yorku.ca

Abstract. We present a framework for automatically generating questions and answers (QAs) from text documents. The core of our proposed method utilizes rules on top of semantic role labels, which are easy to comprehend and maintain and effective in generating grammatically correct questions. In addition, we utilize advanced NLP techniques, such as text summarization (to avoid similar questions), word embedding (for context disambiguation), and topic modeling (to filter out irrelevant questions). We compare our method with the state-of-the-art methods for QA generation on car manuals and discuss our results.

1 Introduction

Automatically generating questions and answers is a long-term goal in many interactive intelligent assistant systems that constitute an FAQ corpus, such as intelligent tutoring systems [11] and interactive query-based environments [20]. iNAGO Inc.[1] develops interactive question-answering systems that converse with users to obtain the most appropriate answers to their questions based on a domain specific knowledge base (KB) of questions and answers (QAs). Manually creating a KB from textual sources, e.g., user manuals, requires significant human time, effort and cost [16]. Therefore, we have been working with them to automate the QA generation process. We use the car manual domain as a case study to do a proof of concept for the proposed method.

Many QA generation systems are based on the *overgenerate-and-rank* strategy that employs some rules to generate a large set of candidate QAs [8]. However, most rule-based approaches [7, 1] do not account for the context of sentences such as coreferencing, and thus can produce ambiguous or vague questions. On the other hand, many neural-network based approaches require a large set of labelled training data, which is hard to obtain, especially for a new domain.

To tackle these challenges, we propose a Context-Aware Question and Answer Generation (CAQAG) framework to automatically generate QAs from text. The framework uses advanced NLP techniques, such as semantic role labeling for rule-based question generation, text summarization and topic modeling for avoiding or filtering out irrelevant questions, co-reference resolution and word embedding for reducing the vagueness of generated questions. We apply CAQAG to car manuals, evaluate the effectiveness of each component, compare the method with the state-of-the-art methods, and report our findings.

[1] http://www.inago.com/

2 Proposed QA Generating Framework

Our CAQAG framework takes a document (e.g., a car manual) as input and outputs a set of QAs. The components of the framework are described below.

2.1 Text Summarization

A text document often contains repeated information. As a result, many generated QAs are often repeated or are not about the central theme of the text. We apply a text summarization technique, i.e., TextRank [14]. TextRank builds a graph from the text, where nodes represent the sentences and the weight on an edge represents the degree of similarity between two connecting sentences. The nodes are then ranked based on their strength that is a function of weights for the incoming and outgoing edges. Top-ranked sentences (e.g., 50% top-ranked) are selected to represent the text. We use paragraph-level summarization in our appication.

2.2 Coreference Resolution

In documents, many sentences refer to previous sentences using coreferences to avoid repetition, which can lead to vague questions, for example:

"At normal operating temperature, the needle of the engine will remain in the center section. If it enters the red section, the engine is overheating."

From the second sentence, using basic syntactic information, the following question will be generated: *"What happens if it enters the red section?"*, which is not acceptable since it is not clear what the pronoun *"it"* refers to. To solve this problem, we use a coreference resolution technique, i.e. the Stanford coreferencing tool [13], to identify the antecedents of pronouns, and replace those pronouns with their more specific antecedent phrases. As a result, the pronoun *"it"* is replaced with the word *"the needle of the engine"*:*"What happens if the needle of the engine enters the red section?"*.

2.3 Semantic Role Labeling

Semantic Role Labeling (SRL) assigns labels to words or phrases in a sentence that indicate their semantic role in the sentence. We use the model from Propositional semantics [4] which is based on thematic proto-roles and argument selection in Linguistics [4]. Each sentence is represented by one or more propositions. Each proposition consists of a predicate (usually a verb) and its semantic arguments. The arguments are the phrases from the sentence carrying the semantic roles of the predicate. For example, given a sentence *"ABS is activated during braking under certain road or stopping conditions"*, a semantic role labeler may recognize "activate" as representing the predicate, "ABS" as an argument of the predicate representing the *patient* of the predicate, and "during braking under certain road or stopping conditions" as another argument of the predicate representing the *time*. The goal of semantic role labeling is to help catch the meaning of a sentence.

The top block of Table 1 shows examples of semantic-role-labeled sentences from a car manual. A predicate is denoted as TARGET, and its various arguments are denoted as A0=PAG (representing agent), A1=PPT (representing patient), AM-TMP (representing time, i.e., when), AM-LOC (representing location, i.e., where), etc.

Sample Sentences with SRL tags: S1. [ABS (A1=PPT)] is [activated (TARGET)] [during braking under certain road or stopping conditions (AM-TMP)]. S2. [The Odometer (A1=PPT)] [is located (TARGET)] [in the bottom of the information display (AM-LOC)]. S3. [Engine Coolant Temperature (A0=PAG)] [illuminates (TARGET)] [when the engine coolant temperature is high (AM-TMP)]. S4. [Low Tire Pressure Warning (A0=PAG)] [will illuminate(TARGET)] [when (R-AM-TMP)] [your tire pressure (A1=PPT)] is [low (A2=PRD)]. S5. [Cruise control (A0=PAG)] [disengages (TARGET)] [if the vehicle speed decreases more than 16 km/h below the set speed (AM-ADV)].
Sample rules to generate QAs: R1. (Replace [A1=PPT] with *what*) Q: What is [TARGET][AM-TMP]? A:[A1=PPT]. R2. (Replace [AM-TMP] with *when*) Q: When is [A1=PPT][TARGET]? A:[AM-TMP]. R3. (Replace [AM-LOC] with *where*) Q: Where is [A1=PPT]? A:[AM-LOC]. R4: (Replace [AM-TMP] with *why*) Q: Why does [A0=PAG] [TARGET]? A:[AM-TMP]. R5: (Replace [A0=PAG][TARGET] before [R-AM-TMP] with *how do we know if*) Q: How do I know if [A1=PPT] is [A2=PRD]? A:[A0=PAG][TARGET]. R6: (Replace [AM-ADV] with *In what circumstances*) Q: In what circumstances [A0=PAG][TARGET]? A:[AM-ADV].
Generated QAs: QA1: What is activated during braking under certain road or stopping conditions? ABS. (Generated from S1 using R1) QA2: When is ABS activated? During braking under certain road or stopping conditions. (Generated from S1 using R2) QA3: Where is the odometer? In the button of the information display. (Generated from S2 using R3) QA4: Why does engine coolant temperature illuminate? When the engine coolant temperature is high. (Generated from S3 using R4) QA5: How do I know if your tire pressure is low? Low tire pressure will illuminate. (Generated from S4 using R5) QA6: In what circumstances cruise control disengages? If the vehicle speed decreases more than 16 km/h below the set speed.(Generated from S5 using R6)

Table 1. Sample sentences with semantic role labels, rules and generated QAs.

2.4 Generating Questions

We design a set of 75 general purpose rules to transform the semantic role labeled sentences to QAs. We chose the rule-based method because it was the most effective method in the industry [2] and rules are comprehensible and easy to maintain. The middle block of Table 1 shows some sample rules, and the bottom block shows the generated QAs using these rules from SRL-tagged sentences in the top block.

For example, we replace [ABS (A1=PPT)] in S1 with the word *What* and generate a question as: *"What is activated during braking under certain road or stopping conditions?"*. Moreover, we use [ABS (A1=PPT)] to present the answer. These rules are general purpose rules and can be used to generate many types of questions (e.g. what, when, where, why, how, in what circumstances).

2.5 Filtering the Generated QAs

Although text summarization was used earlier to remove unimportant sentences before the question generation step, too many questions are generated in the above step, some of which are not related to the main theme of the document and thus need to be removed. We use Top Mine [5] to extract the important topics from the input document [2]. In topic modeling, each topic is a distribution over terms (e.g., words or phrases). We choose the most frequent terms from the most likely unigrams and merge these terms into a list of topical terms. Then, we filter out questions that do not contain a topical term on the list. The resulting QAs contain at least one topical term, and are thus related to the main theme of the document.

2.6 Context Disambiguation

Pronouns are not the only cause of ambiguities in QAs. Noun phrases that refer to terms in previous sentences can lead to vague questions, e.g.,:

"Switching the adaptive cruise control sets a speed to maintain a safe distance. You can override the system by applying the brakes."

[2] We selected the number of topics to be 20, since this number results in the lowest perplexity score [6].

Generating a question from the second sentence above leads to the following ambiguous question:

"What should I do to override the system?"

It is not clear what *"the system"* refers to.

We utilize the word embedding[3] [15] to compute the semantic similarity between a concept h_i in a domain ontology[4] and question q as follows:

$$Sim(q, h_i) = \frac{(\sum_{m=1}^{|q|} \sum_{n=1}^{|h_i|} Cosine(\boldsymbol{w}_m, \boldsymbol{t}_{i,n}))}{|q| \times |h_i|}, \tag{1}$$

where $t_{i,n}$ is the word vector of the n'th term of concept h_i and w_m is the word vector of the m'th non-stop-word of the question q. We select the concept with the highest score as the *context* of question q and use it to augment the question. For example, the most similar concept to the previous ambiguous question is: *"adaptive cruise control* = {*smart, cruise, control, override, adaptive, cruise, ascc, system*}*"*. Thus, concept *"Adaptive cruise control"* is used to augment the question resulting in: *"What should I do to override the system, regarding the adaptive cruise control?"*

3 Evaluations

The proposed CAQAG framework is an unsupervised method for question generation. To see the usefulness of each of its components, we conduct an ablation test that removes one component from the method at a time, resulting in **5 variations of our method** (Tables 2 and 3). In addition, we compare CAQAG with two state-of-the-art supervised learning methods: (1) **Transformer**: This is a neural network based sequence-to-sequence model based on the attention mechanism and positional encoding [19]. We train a 6-layer model with the same hyperparameters as in [19]. (2) **Pointer-Generator Networks (PGN)**: This is a sequence-to-sequence model determining whether to copy words from the input sentence (answer) based on the attention distribution or generate them from the vocabulary distribution [18]. A 2-layer model is trained with the Adam optimizer [10].

Our dataset consists of 4672 QAs created by human annotators from two car manuals (Ford and GM). We divided the dataset into training (4360 QAs) and testing sets (312 QAs). The supervised methods are trained on the training set. All the methods are evaluated on the test set.

3.1 Human Evaluation

We randomly selected 25 automatically generated QAs for each method and asked five English speakers to rate the quality of each QA based on the three criteria: *Grammar*, *Vagueness*, and *Q and A Relatedness)*. Furthermore, we asked the human evaluators to

[3] The pre-trained GloVe vectors [17] are used in our experiments.

[4] The ontology, provided by iNAGO Inc., consists of a set of concepts, each of which is described by a set of terms.

QA System	Score	Bad Grammar (%)	Vague (%)	Unrelated (%)
CAQAG (All Features)	3.53	**19**	**21**	29
CAQAG without Summarization	3.47	24	29	38
CAQAG without Coreferene Resolution	3.22	23	39	45
CAQAG without Context Disambiguation	3.16	21	46	37
CAQAG without Filtering	3.07	23	45	50
Transformer	2.41	55	65	26
Pointer Generator Network (PGN)	**3.67**	26	22	26

Table 2. Human evaluation results on car manuals.

QA System	ROUGE-L	METEOR
CAQAG (All Features)	0.39	0.22
CAQAG without Summarization	0.38	0.21
CAQAG without Coreferene Resolution	0.25	0.15
CAQAG without Context Disambiguation	0.35	0.20
CAQAG without Filtering	0.32	0.18
Transformer	0.52	0.18
Pointer Generator Network (PGN)	**0.63**	**0.30**

Table 3. Automatic evaluation results on car manuals.

score the generated QAs based on the five-point rating scale [9] [5]. Table 2 shows the average score and the percentage of the generated questions having each problem for each method.

The results show that CAQAG with all features leads to the best score and least problems among its variations. Among the 4 auxiliary components of CAQAG, filtering with topic modeling is the most effective feature, followed by context disambiguation, coreference resolution and text summarization. Without filtering, 50% of the generated questions are unrelated to their answers. Context disambiguation is the most effective in lowering the vagueness of the generated questions. All the features help lower the vagueness and relatedness of generated questions, and lead to questions with better grammar although the grammar improvement is not as significant as the ones in vagueness and unrelatedness. The reason is that the grammatical structures of generated QAs highly depend on the semantic role labels. It is worth noting that the reason for the least effectiveness of summarization is that the paragraphs in car manuals are often very short, containing only a few sentences.

Table 2 also shows that CAQAG with all features significantly outperforms the Transformer model in human evaluation, and is comparable to the Pointer-Generator Network (PGN). The questions generated by CAQAG have least grammatical errors and vagueness, but the questions generated by PGN are more related to their answers, resulting in the highest overal score for PGN. This is not surprising because PGN is a supervised learning method (which requires a large set of labelled training data) while the rule-based method is unsupervised. The reason for the worst performance of Transformer is that it has many unknown words in the generated questions. PGN solves this problem by copying words from input sentences.

[5] 1= Bad: the question has major problems; 2= Unacceptable: the question definitely has a minor problem (Grammar, Vagueness, Awkwardness); 3= Borderline: the question might have a problem, but I'm not sure; 4= Acceptable: The question does not have problems; and 5= Good: the question is as good as one that a human asks.

3.2 Automatic Evaluation

We use ROUGE [12] and METEOR [3] to evaluate the questions generated from the whole test data set. The Transformer and PGN are trained on the training data. We report the F_1 scores for ROUGE-L (longest common subsequence text overlap). METEOR improves ROUGE by taking into account word re-ordering, stemming, synonyms, and paraphrase matching. The results in Table 3 show that CAQAG with all features obtains the best scores compared to its variations and coreference resolution is the most effective feature in this evaluation. Compared to the neural network based methods, both Transformer and PGN have much higher ROUGE scores. But in terms of METEOR, CAQAG is better than Transformer, but not as good as PGN. This result indicates that METEOR is better in line with human judgment and is more reliable evaluation metric than ROUGE.

3.3 Evaluation on Recall

To see how the features of CAQAG affects the coverage of generated QAs, we compute recall values in the ablation test. The recall for CAQAG is 0.44, while the recalls after removing summarization, co-referencing, context disambiguation and filtering are 0.46, 0.31, 0.46, and 0.45 respectively. This shows that summarization, context disambiguation and filtering do not significantly influence the recall, meaning that mostly irrelevant QAs are filtered out, while removing co-referencing greatly reduces the recall. The reason is that co-referencing affects semantic rule labels.

4 Conclusions

We presented a rule-based unsupervised method for automatically generating QAs from text and its application to car manuals. A number of NLP techniques (i.e., semantic role labeling, topic modeling, coreference resolution and context disambiguation) are used. Our results show that these techniques are effective in improving the quality of generated QAs. Compared to the supervised neural network based methods, our method is significantly better in terms of grammatical correctness of the generated QAs. It outperforms the Transformer significantly according to human evaluation and the METERO metric. Although it does not beat the Pointer-Generator Network, it has the benefit of not requiring a large set of labelled training data, which makes it more applicable, especially in new domains. The use of CAQAG by iNAGO reported significant reduction in human effort and cost in QA generation, comparing to a purely manual QA generation process. Moreover, the rules are easy to understand and can be easily used or adapted to other domains. As future work, we will investigate how to combine the rule-based with neural network based methods to improve the quality of generated QAs.

References

1. Chali, Y., Hasan, S.A.: Towards topic-to-question generation. Computational Linguistics **41**(1), 1–20 (2015). https://doi.org/10.1162/COLI_a_00206

2. Chiticariu, L., Li, Y., Reiss, F.R.: Rule-based information extraction is dead! long live rule-based information extraction systems! In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. pp. 827–832. Association for Computational Linguistics, Seattle, Washington, USA (October 2013), `http://www.aclweb.org/anthology/D13-1079`
3. Denkowski, M., Lavie, A.: Meteor universal: Language specific translation evaluation for any target language. In: In Proceedings of the Ninth Workshop on Statistical Machine Translation (2014)
4. Dowty, D.: Thematic proto-roles and argument selection. language pp. 547–619 (1991)
5. El-Kishky, A., Song, Y., Wang, C., Voss, C.R., Han, J.: Scalable topical phrase mining from text corpora. CoRR **abs/1406.6312** (2014), `http://arxiv.org/abs/1406.6312`
6. Griffiths, T.L., Steyvers, M.: Finding scientific topics. Proceeding of the National Academy of Sciences of the United States of America **101**, 5228–5235 (2004). https://doi.org/10.1073/pnas.0307752101
7. Heilman, M.: Automatic Factual Question Generation from Text. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, USA (2011), aAI3528179
8. Heilman, M., Smith, N.A.: 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. pp. 609–617. HLT '10, Association for Computational Linguistics, Stroudsburg, PA, USA (2010), `http://dl.acm.org/citation.cfm?id=1857999.1858085`
9. Heilman, M., Smith, N.A.: Rating computer-generated questions with mechanical turk. In: Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk. pp. 35–40. CSLDAMT '10, Association for Computational Linguistics, Stroudsburg, PA, USA (2010), `http://dl.acm.org/citation.cfm?id=1866696.1866701`
10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), `http://arxiv.org/abs/1412.6980`
11. Kuyten, P., Bickmore, T., Stoyanchev, S., Piwek, P., Prendinger, H., Ishizuka, M.: Fully automated generation of question-answer pairs for scripted virtual instruction. In: Proceedings of the 12th International Conference on Intelligent Virtual Agents. pp. 1–14. IVA'12, Springer-Verlag, Berlin, Heidelberg (2012)
12. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. p. 10 (01 2004)
13. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: Association for Computational Linguistics (ACL) System Demonstrations. pp. 55–60 (2014), `http://www.aclweb.org/anthology/P/P14/P14-5010`
14. Mihalcea, R., Tarau, P.: Textrank: Bringing order into texts. In: Lin, D., Wu, D. (eds.) Proceedings of EMNLP 2004. pp. 404–411. Association for Computational Linguistics, Barcelona, Spain (July 2004)
15. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. CoRR **abs/1310.4546** (2013), `http://arxiv.org/abs/1310.4546`
16. Mitkov, R., An Ha, L., Karamanis, N.: A computer-aided environment for generating multiple-choice test items. Natural Language Engineering **12**(2), 177194 (2006). https://doi.org/10.1017/S1351324906004177
17. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543 (2014), `http://www.aclweb.org/anthology/D14-1162`

18. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27, pp. 3104–3112. Curran Associates, Inc. (2014), http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf
19. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. CoRR **abs/1706.03762** (2017), http://arxiv.org/abs/1706.03762
20. Xue, X., Jeon, J., Croft, W.B.: Retrieval models for question and answer archives. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 475–482. SIGIR '08, ACM, New York, NY, USA (2008). https://doi.org/10.1145/1390334.1390416, http://doi.acm.org/10.1145/1390334.1390416