Inf Retrieval J
DOI 10.1007/s10791-017-9311-0

Product review summarization through question retrieval and diversification

Mengwen Liu[1] · Yi Fang[2] · Alexander G. Choulos[2] · Dae Hoon Park[3] · Xiaohua Hu[1]

Received: 21 December 2016 / Accepted: 11 July 2017

Abstract Product reviews have become an important resource for customers before they make purchase decisions. However, the abundance of reviews makes it difficult for customers to digest them and make informed choices. In our study, we aim to help customers who want to quickly capture the main idea of a lengthy product review before they read the details. In contrast with existing work on review analysis and document summarization, we aim to retrieve a set of real-world user questions to summarize a review. In this way, users would know what questions a given review can address and they may further read the review only if they have similar questions about the product. Specifically, we design a two-stage approach which consists of question selection and question diversification. For question selection phase, we first employ probabilistic retrieval models to locate candidate questions that are relevant to a given review. A Recurrent Neural Network Encoder–Decoder is utilized to measure the "answerability" of questions to a review. We then design a set function to re-rank the questions with the goal of rewarding diversity in the

The preliminary version of this work appeared in SIGIR 2016 entitled "Retrieving Non-Redundant Questions to Summarize a Product Review", pp.385-394.

✉ Yi Fang
yfang@scu.edu

Mengwen Liu
ml943@drexel.edu

Alexander G. Choulos
achoulos@scu.edu

Dae Hoon Park
dae.hoon.park@huawei.com

Xiaohua Hu
xh29@drexel.edu

[1] Drexel University, Philadelphia, PA, USA

[2] Santa Clara University, Santa Clara, CA, USA

[3] Huawei Research America, Santa Clara, CA, USA

Published online: 24 July 2017

final question set. The set function satisfies submodularity and monotonicity, which results in an efficient greedy algorithm of submodular optimization. Evaluation on product reviews from two categories shows that the proposed approach is effective for discovering meaningful questions that are representative of individual reviews.

Keywords Review summarization · Question retrieval · Sequence-to-sequence learning · Diversity

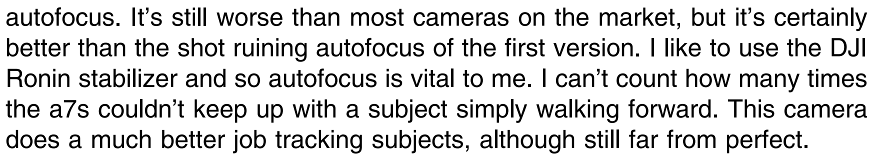
1 Introduction

With the rapid growth of online review sites, more people rely on advices from fellow users before they make purchase decisions. Unfortunately, finding relevant information from large quantities of user reviews in a short time is a huge challenge. Thus, review analysis with the goal of extracting useful information has become an important way to improve user experience of online shopping.

Existing techniques for review analysis include review rating prediction (Wang et al. 2010; Li et al. 2011), sentiment polarity classification (Jo and Oh 2011; Liu et al. 2012), and aspect-based review summarization (Hu and Liu 2004; Titov and McDonald 2008; Park et al. 2015). The first two techniques aim to predict numerical ratings and sentiment orientations of reviews. They do not summarize the main points discussed in reviews. Review summarization is beneficial for aggregating user opinions towards a product through the generation of a short summary from a set of product reviews. However, the generated summary may not be of interest to end users since it may contain little relevant information that addresses the specific questions that are in the users' mind.

In our study, we seek an approach to help customers quickly comprehend a product review through questions. Questions are often more attractive for customers to read than plain opinion sentences are. In other words, we aim to find a concise set of questions that are addressed by a given review as well as cover the main points of it. Many users have certain questions about a product in mind and want to look at online reviews to see if their questions can be answered; but examining all lengthy reviews is too time-consuming. Given a concise set of questions for a review, users can quickly understand the review and may further read it only if they have similar questions in their minds.

Directly synthesizing such questions is too intimidating. Thanks to the emergence of Community Question Answering (CQA), large e-commerce websites now offer CQA services for their products. A notable example is Amazon's Customer Questions and Answers service.[1] In this paper, our goal is to retrieve real-world user questions to summarize individual reviews. Take the following segment of a real-world review[2] from Amazon as an example:



autofocus. It's still worse than most cameras on the market, but it's certainly better than the shot ruining autofocus of the first version. I like to use the DJI Ronin stabilizer and so autofocus is vital to me. I can't count how many times the a7s couldn't keep up with a subject simply walking forward. This camera does a much better job tracking subjects, although still far from perfect.

[1] http://www.amazon.com/gp/forum/content/qa-guidelines.html.

[2] http://www.amazon.com/Sony-ILCE7SM2-Full-Frame-Mirrorless-Interchangeable/product-reviews/B0158SRJVQ/.

As we can see, this segment of review describes some personal experience with the camera's *autofocus* feature and compares it with another camera *a7s*. On the other hand, a real relevant question[3] was asked and answered on Amazon's CQA service as shown below:

> **Q:** Does it have a fast autofocus?
> **A:** Autofocus is in the middle of the pack I'd say. The a7rii has faster autofocus, (so does the a6000 for that matter, a $500 camera) but this is better than the first a7s.

This question asked about *autofocus* feature and can well represent the semantic of the segment of the above review. Meanwhile, since it is a question, users with similar questions in their minds would be very interested in further reading the review if they see this question as part of the summary of the review, and expect to find answers in the review. Thus, this question would be a good candidate to retrieve for this review. Moreover, directly retrieving this question could be challenging given the short length of the question, but we can exploit the answers of the question. For example, this particular answer also discussed the comparison with *a7s*. Using it would be helpful to measure the relevance between the question and review.

This task of summarizing a product review through user questions is a challenging task. First of all, user generated reviews are usually long, ranging from hundreds to thousands of words, while questions are much shorter. Directly matching questions to a review may lead to unsatisfactory results. Second, now that we aim to use questions to summarize a product review, the review in turn is expected to contain answers to the questions. In other words, matched questions should have the "answerability" to a review. Third, a product review often discusses multiple aspects of a product. The set of retrieved questions for a given review should cover as many aspects as possible so that customers have a comprehensive understanding of the review. Last but not the least, the questions should not be redundant.

To tackle these challenges, we develop a two-stage framework to achieve the goal of retrieving a set of non-redundant questions to represent a product review. We first employ a probabilistic retrieval model to retrieve candidate questions based on their relevance scores to a review. We further leverage answers to a question to bridge the vocabulary gap between a review and a question. To ensure the review can be used as answers to questions, we employ a sequence-to-sequence learning architecture, a Recurrent Neural Network (RNN) Encoder–Decoder, to take into consideration the answerability measurement between questions and a review. Such an architecture is designed to learn the mapping between a pair of input and output sequence with variable length, which is a natural fit to pairs of review-question data. The RNN Encoder–Decoder is first trained on a public product QA dataset, and then used to predict the answerability score of a pair of review and question data. The answerability score is then incorporated with the relevance score to determine the rank of questions. After selecting top-*k* questions as the candidate set, in the second stage, we propose a set function that is used to re-rank the retrieved questions with the goal of both perserving the relevance and answerability of the questions and diversifying the questions. Particularly, the set function satisfies monotone submodularity such that the objective function to determine the final question set for summarizing a review can be efficiently optimized by a greedy algorithm. The question set is theoretically guaranteed to be a constant-factor approximation of the optimal solution.

It is worth noting that we do not aim to replace plain sentence-based summaries with question-based summaries. Instead, our goal is to explore the possibility of using questions

[3] http://www.amazon.com/Sony-ILCE7SM2-Full-Frame-Mirrorless-Interchangeable/dp/B0158SRJVQ/.

as summaries, which can be treated as a new representation of reviews. The question-based summaries are used to reflect the main aspects discussed in a product review, but not necessarily include the author's opinions towards the aspects. According to the typology of traditional text summarization systems proposed by Hovy and Lin (1998), the usages of text summaries can be categorized in two ways, indicative and informative. An indicative summary aims to provide the gist of the input texts without including its contents; while an informative summary aims to reflect the content of a text document. In this way, a question-based summary can be regarded as an indicative summary.

The main contributions of this paper can be summarized as follows:

- We introduce a new task of summarizing a product review by real-world user questions. To the best of our knowledge, no prior work has been done, as the existing work on review summarization focuses on extracting opinion sentences from product reviews.
- We propose a two-stage approach consisting of question retrieval and question diversification. Questions are retrieved based on query likelihood language models by incorporating answers to bridge the vocabulary gaps between a review and a question, and a Recurrent Neural Network (RNN) Encoder–Decoder, a sequence-to-sequence learning model designed to measure the answerability of questions to a product review.
- Question diversification is based on submodular optimization by considering both question coverage and non-redundancy. The choice of monotone submodular functions enables an efficient greedy algorithm for question diversification.
- We create and annotate a dataset for evaluation by manually locating and editing relevant questions for reviews in two product categories. We will make the data publicly available, which can be used for similar research. We conduct thorough experiments on the dataset and demonstrate the effectiveness of our proposed approach.

2 Related work

2.1 Review summarization

Automatic review summarization has been a hot research topic in recent decades. Different from standard text summarization (Goldstein et al. 1999), which aims to generate a concise summary for a single (Svore et al. 2007) or multi-document (Goldstein et al. 2000), review summarization aims to integrate users' opinions for a large collection of reviews with respect to a product (Ly et al. 2011; Yatani et al. 2011). The key idea is to identify the key specifications of a product and opinion sentences towards each specification. Detailed analysis of state-of-the-art literature can be found in Pang and Lee (2008), Kim et al. (2011) and Liu (2012). Our problem of aligning questions to a review is similar to text summarization problem, with the goal of finding relevant and non-redundant questions (summary) for a review (document). It is also similar to review summarization, but the difference is that opinion-based summarization focuses on sentence or phrase extraction from reviews, while ours focuses on using relevant questions to represent the main points discussed in a review. By doing this, we are able to create more "relevant" summaries of reviews for potential buyers.

2.2 Question retrieval with verbose queries

As our goal is to use a lengthy review to find short representative questions as summaries, our problem relates to the problem of information retrieval with verbose queries (Gupta and Bendersky 2015). Due to term redundancy, query sparsity, and difficulty in identifying key concepts, verbose queries often result in zero matches. In tackling these challenges, recent studies have developed techniques to re-compose queries. Examples include query reduction (Kumaran and Carvalho 2009; Huston and Croft 2010), query reformulation (Dang and Croft 2010; Xue et al. 2012), and query segmentation (Bendersky et al. 2011; Parikh et al. 2013).

Our goal of finding a set of representative questions to summarize reviews is similar to question retrieval in the field of community question answering (CQA). The key problem is to find questions in the archives that are semantically similar to newly generated questions. Examples of work include Zhou et al. (2011) who proposed a context-aware model for addressing the lexical gap problem between questions; and Zhou et al. (2015) who designed an elegant study to model the question representations with metadata powered deep neural networks. However, question retrieval in CQA is different from our problem in that the queried questions and retrieved questions are usually with similar length (i.e., less than 20 words), while user reviews are longer (usually more than 100 words). Therefore, directly applying techniques for question retrieval in CQA to our problem might lead to zero results due to the verbosity of queried reviews.

In our study, we first use the entire review as a query to retrieve relevant questions. In order to incorporate the answerability measurement between a question and a review, we split a review into sections, and score each pair of question and review section. After determining a set of candidate questions based on their relevance and answerability, we employ a diversity objective function to encourage question diversity. To the best of our knowledge, no existing work attempts to retrieve non-redundant questions to summarize a product review.

2.3 Question generation

Our problem also relates to automatic question generation (AQG) from text data. It is a challenging task as it involves natural language understanding and generation (Rus and Arthur 2009). Most AQG approaches focus on generating factoid questions for supporting domain-specific tasks. One of the applications is to generate questions to facilitate reading practice and assessment. For example, Heilman and Smith (2010) proposed rule-based question transformations from declarative sentences. The transformed questions are then ranked and selected by a logistic regression model. Zhao et al. (2011) developed a method to automatically generate questions from short user queries in community question and answering (CQA). Chali and Hasan (2012) developed a method to generate all possible questions with regards to a topic. Liu et al. (2014) proposed a learning-to-rank based system which ranks generated questions based on citations of articles. One limitation of these aforementioned studies is that questions are generated based on templates, which require lots of manual work and domain knowledge.

Nowadays, with the explosive amount of data available on the web, deep learning techniques have shown great success in various domains, such as image recognition (Krizhevsky et al. 2012), speech recognition (Hinton et al. 2012), and natural language processing (Bengio et al. 2003; Mikolov et al. 2013; Socher et al. 2013). Among the various deep learning architectures, sequence-to-sequence learning architecture is a representative one to learn the mapping between a pair of data, one is an input sequence, and the other one is an output sequence. Examples of sequential data include but are not limited to text, image, voice; and the input/output sequences are not necessary the same type of modality. A notable example of sequence-to-sequence learning architecture is Recurrent Neural Network (RNN) Encoder–Decoder (Cho et al. 2014). A trained RNN Encoder–Decoder can be used to generate sequences of outputs given new sequences of inputs, or to score a pair of input/output sequences. Such a model has been successfully applied to machine translation (Sutskever et al. 2014; Bahdanau et al. 2014), image caption generation (Mostafazadeh et al. 2016), and document summarization (Nallapati et al. 2016). The advantage of using an RNN Encoder–Decoder over template-based AQG approaches is that an RNN Encoder–Decoder requires little manually-coded features or templates. The semantic and syntactic alignment between the pair of input/output sequence can be automatically learned from this architecture. In this study, we explore the usage of an RNN Encoder–Decoder to measure whether a review can be used to answer a set of questions.

2.4 Diversified ranking

As our goal aims to find a set of non-redundant questions to summarize a product review, our problem relates to search result diversification (Harman 2002; Soboroff and Harman 2003; Soboroff et al. 2004). The approaches can be categorized into implicit and explicit approaches. Implicit approaches assume similar documents cover similar aspects. Carbonell and Goldstein (1998) proposed the Maximal Marginal Relevance (MMR) which intuitively selects a result that maximizes an objective function until a given cardinality constraint is met (e.g., the number of results). Dang and Croft (2012) proposed PM-2 that iteratively selects documents that maintain the proportionality of topics. Explicit approaches, on the other hand, explicitly select documents that cover different aspects. Examples of work include xQuAD (Santos et al. 2010), which examines different aspects underlying original query in the form of sub-queries, and estimates the relevance of retrieved documents to each sub-queries. In our study, we use an implicit approach by designing a monotone submodular objective function to determine a set of non-redundant questions. Different from the aforementioned implicit approaches, the submodular objective function can be maximized approximately by an efficient greedy algorithm that results in a constant-factor approximation of the optimal solution.

3 Problem characterization

3.1 Problem statement

Our task is to use a set of questions to summarize a product review. The review in turn is supposed to contain the answers to those questions. Introducing this feature to e-commerce platforms is beneficial for customers who want to quickly capture the main idea of lengthy reviews before reading the details. Consider a product database with m products. Each product i is associated with a set of reviews $R^{(i)} = \{r_1^{(i)}, \ldots, r_{m_i}^{(i)}\}$ where m_i is the number of reviews for product i. Each review can be represented by a bag of words. Meanwhile, we have a question database/corpus $Q = \{q^{(1)}, \ldots, q^{(n)}\}$ where the questions are crawled from community question answering (CQA) sites. Given a review $r_j^{(i)}$ of product i where $j \in \{1, \ldots, m_i\}$, our task is to select a small subset of questions $S \subseteq Q$ to summarize the review.

Similar to traditional text summarization tasks (Mani and Maybury 1999), the quality of selected questions can be quantified by a set function $\mathcal{F} : 2^Q \rightarrow \mathbb{R}$. In addition, the selected subset S should satisfy certain constraints. Formally, our task is to find the optimal question subset S^* defined as the following combinatorial optimization problem:

$$\begin{aligned} S^* &= \arg\max_{S \subseteq Q} \mathcal{F}(S) \\ s.t. : &\sum_{q \in S} c(q) \leq b, \end{aligned} \tag{1}$$

where $c(\cdot)$ is a constraint function defined on q, and $b \geq 0$ is a constant threshold. For example, if we want to enforce that the total length of all the selected questions should not exceed 50 words, we can define $c(\cdot)$ as a function to calculate the length of each question and set $b = 50$. Similarly, we can define a constraint to restrain the total number of questions in the set.

The set function $\mathcal{F}$ in Eq. (1) measures the quality of the selected question subset S. The choice of $\mathcal{F}$ depends on the property of the questions that we desire. In general, Eq. (1) would be an NP-hard problem. Fortunately, if $\mathcal{F}$ satisfies non-decreasing submodular (Fujishige 2005), the optimization problem can be solved by efficient greedy algorithms with a close approximation. We introduce the background on submodular functions in Sect. 3.2.

It is worth noting that we do not solve Eq. (1) directly over all the possible questions in the database. Otherwise, it would be too time-consuming given the sheer size of all available questions on CQA. Instead, we retrieve a set of potentially relevant questions first by using information retrieval and sequence-to-sequence learning techniques, e.g., obtaining the top 100 questions based on their relevance to a given review and whether they can be answered by the review. We will introduce the question retrieval models and a sequence-to-sequence learning model in Sect. 4.2. Given these questions, we then apply Eq. (1) to select a few questions (e.g., 5) as the final results by considering both question coverage and diversity. Thus, this module can be viewed as re-ranking for achieving diversified results. We present our formulation of Eq. (1) in Sect. 4.3.

3.2 Submodular functions

Submodular functions are discrete functions that model laws of diminishing returns (Shephard and Färe 1974). They have been used in a wide range of applications such as sensor networks (Leskovec et al. 2007), information diffusion (Gomez Rodriguez et al. 2010), and recommender systems (Qin and Zhu 2013). Recently, it has been well-explored in text summarization (Lin and Bilmes 2010, 2011). Following the notations introduced in the previous section, some basic definitions of submodular functions are given as follows.

Definition 1 A set function $\mathcal{F} : 2^Q \rightarrow \mathbb{R}$ is submodular if for any subset $S, T \subseteq Q$,

$$\mathcal{F}(S) + \mathcal{F}(T) \geq \mathcal{F}(S \cap T) + \mathcal{F}(S \cup T).$$

Definition 2 A set function $\mathcal{F} : 2^Q \rightarrow \mathbb{R}$ is modular if for any subset $S, T \subseteq Q$,

$$\mathcal{F}(S) + \mathcal{F}(T) = \mathcal{F}(S \cap T) + \mathcal{F}(S \cup T).$$

Modular set functions also satisfy submodularity according to Definition 1.

Definition 3 A set function $\mathcal{F} : 2^Q \rightarrow \mathbb{R}$ is monotone, if for any subset $S \subseteq T \subseteq Q$,

$$\mathcal{F}(S) \leq \mathcal{F}(T).$$

The class of submodular functions enjoys a good property with concave functions as follows.

Theorem 1 *If $\mathcal{F} : 2^Q \rightarrow \mathbb{R}$ is a submodular function, $g(S) = \phi(\mathcal{F}(S))$, where $\phi(\cdot)$ is a concave function, is also a submodular function* (Shephard and Färe 1974).

In Sect. 4.3, we discuss the construction of $\mathcal{F}(S)$ and demonstrate that it is submodular and monotone based on Theorem 1. These properties enable efficient greedy approximation algorithms that generate provably near-optimal solutions (Nemhauser et al. 1978) for the optimization problem introduced in Eq. (1).

4 Methods

4.1 Overview

In order to use a few questions to provide customers with “hints” of a review, the questions should be representative of the review. For example, if a review discusses *image quality* and *battery life* of a camera, relevant questions would be related to these two features, e.g., “Does the camera take high quality macro images?” or “How many days of battery life can you get with this camera?”. Second, the answers to the questions are expected to be included in the review. For example, the review segments “I’ve included a few un-edited examples using nature macro, sunset, and iAuto because, wow! Color quality is amazing even straight off the camera. I can’t imagine how great this camera would be with a good photo-editing program. The possibilities are endless.” and “Battery life is OK but an all-day shoot requires a second battery.” can be used to answer the aforementioned two questions. Moreover, the questions are expected to be dissimilar to each other such that there is little redundant information covered in the question set. For example, the question “How is the battery life?” is redundant as it contains similar semantic information with the aforementioned question related to *battery life*.

With the multiple goals of relevancy, answerability, and diversity, we design a two-stage framework to find a set of questions that can be used to summarize a review. The first stage of the framework is used to rank a list of questions based on the relevancy and answerability between questions and a review, while the second stage is used to promote the diversity of questions. Specifically, we first utilize a probabilistic retrieval model to select a smaller set of candidate questions that are relevant to a given review from a large pool of questions crawled from a community question and answering (CQA) website. Considering the possible semantic mismatch between the review and question corpus, we incorporate answers into the retrieval model to resolve the vocabulary gap between them. To measure the answerability of questions to a review, we employ a sequence-to-sequence learning model, a Recurrent Neural Network (RNN) Encoder–Decoder. After obtaining the top-k relevant and answerable questions, we design a set function to re-rank questions in the candidate list with the goal of removing redundant questions. The final question set is derived through the measurement of a trade-off between the relevance and answerability of selected questions to the review as well as the diversity of the questions.

In the following sections, we first present the query likelihood language models and a Recurrent Neural Network (RNN) Encoder–Decoder to rank a list of questions given a review (Sect. 4.2), and introduce our set function to re-rank candidate questions (Sect. 4.3) with an efficient greedy algorithm (Sect. 4.4) for optimization.

4.2 Question selection

4.2.1 *Query likelihood language model*

To retrieve candidate questions that are relevant to a given review, we employ query likelihood language model (Berger and Lafferty 1999). We assume that before drafting a review, a user would think about what questions he/she would like to answer. Therefore, the relevance score of a question q retrieved by a review r is computed as the conditional probability $P(q|r)$ of the question given the review:

$$\text{score}(r, q) = P(q|r) \tag{2}$$

Similar to other text retrieval tasks, a review can be regarded as a sample drawn from a language model built on a question pool. Formally, using the Bayes' theorem, the conditional probability can be calculated by:

$$\begin{aligned} P(q|r) &= \frac{P(r|q)P(q)}{P(r)} \\ &\propto P(r|q)P(q) \end{aligned} \tag{3}$$

In Eq. (3), $P(r)$ denoted the probability of the review r, which can be ignored for the purpose of ranking questions because it is a constant for all questions. Thus, we only need to compute $P(r|q)$ and $P(q)$. $P(r|q)$ represents the conditional probability of review r given question q. We can apply the unigram language model to calculate $P(r|q)$:

$$P(r|q) = \prod_{w \in r} P(w|q) \tag{4}$$

where $P(w|q)$ is the probability of observing word w in a question q. The word probability can be estimated based on maximum likelihood estimation (MLE) with Jelinek-Mercer smoothing (Zhai and Lafferty 2004) to avoid zero probabilities of unseen words in q:

$$P(w|q) = (1 - \lambda)P_{ml}(w|q) + \lambda P_{ml}(w|C) \tag{5}$$

where λ is a smoothing parameter and C denotes the whole question corpus. The MLE estimates for $P_{ml}(w|q)$ and $P_{ml}(w|C)$ are:

$$P_{ml}(w|q) = \frac{\text{count}(w, q)}{|q|} \tag{6}$$

$$P_{ml}(w|C) = \frac{\text{count}(w, C)}{|C|} \tag{7}$$

where $\text{count}(w, q)$ and $\text{count}(w, C)$ denote the term frequency of w in q and C, respectively. $|\cdot|$ denotes the total number of words in q or C.

$P(q)$ in Eq. (3) denotes the prior probability of the question q regardless of review r. It can encode our prior preference about questions. In order to summarize a review, we prefer shorter questions so that users can digest information faster. Hence, we reward shorter questions by making the prior probability inversely proportional to the length of the question as follows:

$$P(q) \propto \frac{1}{|q|} \tag{8}$$

$P(q)$ can also be computed by other ways. For example, if there exists rating information of the questions on the CQA website, we can use it to prefer questions with higher ratings.

By plugging Eqs. (4) and (8) into Eq. (3), we can obtain the relevance scores for all questions in the question corpus.

4.2.2 Incorporating answers

Due to the length difference, reviews and questions are highly asymmetric on the information they convey. Thus, there exists vocabulary gap between the two corpus. As shown in the real-world example in Sect. 1, directly retrieving this question could be challenging given the short length of the question. To address this issue, we incorporate the corresponding answers of the question corpus to estimate the parameters in the language model defined in Eq. (5) (Xue et al. 2008). After including all the answers a of question q, the relevance score becomes:

$$\text{score}(r, (q, a)) = P((q, a)|r). \tag{9}$$

Based on the Bayes' theorem, we have:

$$\begin{aligned} P((q,a)|r) &= \frac{P(r|(q,a))P(q,a)}{P(r)} \\ &\propto P(r|(q,a))P(q,a) \\ &= P(r|(q,a))P(a|q)P(q) \\ &\propto P(r|(q,a))P(q) \end{aligned} \tag{10}$$

The above derivation is based on the following reasoning. Similar to Eq. (3), $P(r)$ is a constant for all the questions, and thus it can be ignored. We further assume the probability of answers a given a question q is uniform, and thus $p(q, a)$ is proportional to $p(q)$.

We then leverage both question and answers to estimate $P(r|(q,a))$:

$$\begin{aligned} P(r|(q,a)) &= \prod_{w \in r} P(w|(q,a)) \\ &= \prod_{w \in r} (1-\lambda)P_{mx}(w|(q,a)) + \lambda P_{ml}(w|C') \end{aligned} \tag{11}$$

where C' denotes the whole question and answer corpus, and $P_{ml}(w|C')$ is the collection language model which is estimated based on Eq. (7). λ is a smoothing parameter. $P_{mx}(w|(q,a))$ denotes the word probability estimated from the question and answers. It takes a weighted average of maximum-likelihood estimates from question and answers, respectively:

$$\begin{aligned} P_{mx}(w|(q,a)) &= (1-\alpha)P_{ml}(w|q) + \alpha P_{ml}(w|a) \\ &= (1-\alpha)\frac{\text{count}(w,q)}{|q|} + \alpha\frac{\text{count}(w,a)}{|a|} \end{aligned} \tag{12}$$

where $\alpha \in [0, 1]$ is a trade-off coefficient.

The prior probability $P(q)$ can be calculated in the same way as in Eq. (8). By plugging $P(r|(q, a))$ and $P(q)$ in Eq. (10), we can obtain the relevance scores in Eq. (9).

4.2.3 Incorporating answerability of reviews

Now that we aim to use questions to summarize a review, we expect the review could include answers to the questions. The aforementioned query likelihood language model has not yet taken into consideration of such an answerability measurement between a review and questions. In tackling the issue, we enrich the ranking score between a review and a question defined in Eq. (2) by integrating the answerability of the question to the review:

$$\mathrm{score}(q, r) = (1 - \gamma)\,\mathrm{score}_r(q, r) + \gamma\,\mathrm{score}_a(q, r) \tag{13}$$

where $\mathrm{score}_r(q, r)$ denotes the relevance score between the question q and r, $\mathrm{score}_a(q, r)$ denotes the answerability of r to q, and $\gamma \in [0, 1]$ is a trade-off parameter to balance the relevancy and answerability. When $\gamma = 0$, the scoring functions is the same as Eq. (2); when $\gamma = 1$, the scoring function is fully dominated by the answerability measurement between questions and a review.

We expect the answers to questions could be addressed by some parts of a review, but not necessarily the entire review. Thus, the answerability measurement $\mathrm{score}_a(q, r)$ in Eq. (13) could be calculated based on the summation of the answerability scores of each section of a review and a question:

$$\begin{aligned}\mathrm{score}_a(q|r) &= \sum_s P_a(q, s|r)\\ &= \sum_s P_a(q|s)P(s|r)\end{aligned} \tag{14}$$

where s denotes a section of r, and $P(s|r)$ denotes the importance of s in r, which could be estimated based on the proportion of the length of a section to that of a review: $P(s|r) = |s|/|r|$. We assume q is only dependent on a section of review s instead of the entire review r.

$P_a(q|s)$ in Eq. (14) denotes likelyhood that a review section $s = \{s_1, \ldots, s_{s_m}\}$ with s_m words generates a question $q = \{w_1, \ldots, w_{q_m}\}$ with q_m words:

$$P_a(q|s) = P(q_1, \ldots, q_{q_m}|s_1, \ldots, s_{s_m}) \tag{15}$$

To obtain the answerability score defined in Eq. (15), we employ a sequence-to-sequence learning model, a Recurrent Neural Network (RNN) Encoder–Decoder, which can learn semantic relations between a pair of sequences of data. The model allows variable-length of input and output sequence and is initially applied to machine translation tasks (Sutskever et al. 2014; Cho et al. 2014). Since a review is relatively long and a question is short, an RNN Encoder–Decoder is a natural fit to pairs of review and question data.

Following the notation in Eq. (15), an RNN Encoder–Decoder consists of two RNNs: one RNN encodes a review section s tokens into a fixed-length vector representation c which summarizes the information of the input sequence. Figure 1 depicts the architecture of sequence-to-sequence learning with Gated Recurrent Units (GRUs) by treating answer/review as input and question as output. Mathematically,

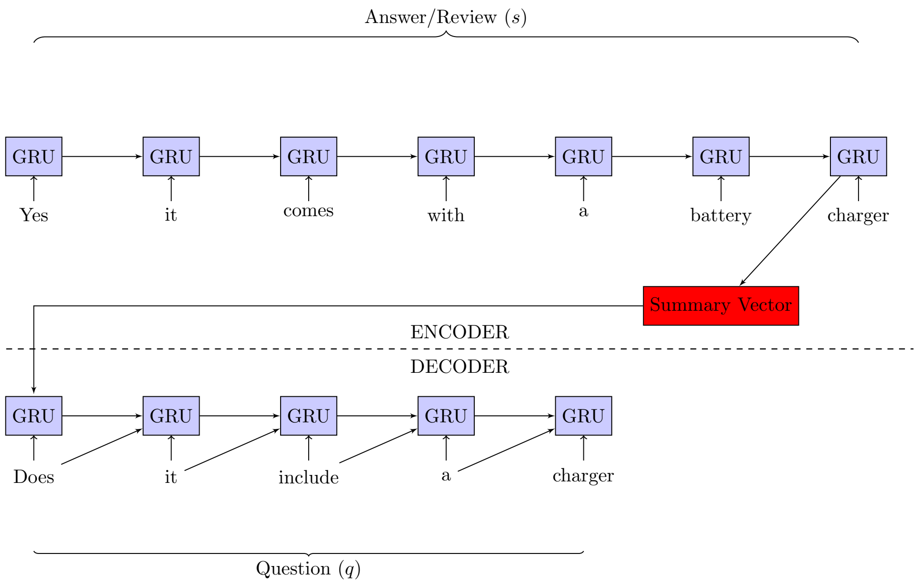


Fig. 1 The architecture of the sequence-to-sequence learning model with Gated Recurrent Units (GRUs) for learning semantic relations between answer/review (input) and question (output)

$$\begin{aligned} h_t &= f(s_t, h_{t-1}) \\ c &= u(h_1, \ldots, h_{s_m}) \end{aligned}$$

where $h_t \in \mathbb{R}^k$ is a hidden state at position t, and c is a summary vector generated from the sequence of the hidden states. f is the RNN. There are several network architectures for each RNN, such as long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997), bidirectional neural networks (BRNN) (Schuster and Paliwal 1997), and gated recurrent neural networks (GRU) (Chung et al. 2014). u is a nonlinear function. For instance, Sutskever et al. (2014) uses LSTM for f and $c = u(h_1, \ldots, h_{s_m}) = h_{s_m}$.

Another RNN decodes the representation into a question q tokens. Specifically, the decoder is trained to predict the next word w_t given the summary vector c and all the previously predicted words $w_1, \ldots, w_{t-1}$. In other words, the decoder defines a probability over the question q by decomposing the joint probability into the ordered conditionals. Hence, Eq. (15) is calculated as:

$$P_a(q|s) = \prod_{t=1}^{q_m} P(w_t|w_1, \ldots, w_{t-1}, c)$$

where $q = (w_1, \ldots, w_{q_m})$. With an RNN, each conditional probability is modeled as

$$P(w_t|w_1, \ldots, w_{t-1}, c) = f(w_{t-1}, v_t, c)$$

where f is an RNN that outputs the probability of w_t and v_t is the hidden state of the RNN.

The model is jointly trained to maximize the conditional log-likelihood of the questions given the answers:

$$\max_{\theta} \frac{1}{N} \sum_{i=1}^{N} \log P(q^{(i)} | s^{(i)})$$

where θ is the set of the model parameters and each $(s^{(i)}, q^{(i)})$ is an (answer, question) pair from the training set and N is the total number of such pairs. We can use a gradient-based algorithm to estimate the model parameters. The resultant model will be used to score a given pair of a review section and a question based on Eq. (15). We will discuss the implementation details of the model in Sect. 5.2.

In addition to the incorporation of answerability measurement between a review and a question, the RNN Encoder–Decoder is also beneficial for capturing the semantic matching between the two types of texts by incorporating an attention mechanism (Bahdanau et al. 2014). Such an mechanism aims to discover the semantic alignment of positions of tokens between input (a review section) and output (a question) sequences. The query likelihood language model [Eq. (3)] for matching a review with questions is based on keyword matching, even though we leverage answers as external resources to expand the query likelihood language model [Eq. (10)]. For example, “Would you recommend for gymnastics photos?” would be a good candidate summary for the review section “That’s a big deal if you shoot sports/action/aviation.”. However, it would not rank high by the query likelihood language model, as the question and a review section do not share any common words. Using the RNN Encoder–Decoder with an attention mechanism would be able to capture the semantic alignment between “gymnastics” and “sports/action/aviation” if they co-occur in the input/output sequences in the training data, and thus to rate the question higher.

It is noted that it is not necessary to pair all questions with each of the review sections due to the large size of a question pool. Instead, we only pair each review section with the most relevant questions based on $\text{score}_r(q, r)$ in Eq. (13). In accordance with the way of calculating the answerability of a review to a question, we first partition a review into sections and then use each of the sections to retrieve relevant questions:

$$\begin{aligned} \text{score}_r(q, r) &= P(q|r) \\ &= \sum_s P(q, s|r) \\ &= \sum_s P(q|s)P(s|r) \\ &= \sum_s \frac{P(s|q)P(q)}{\sum_{q'} P(s|q')P(q')} P(s|r) \end{aligned} \tag{16}$$

The above derivation is based on the following reasoning. We assume that the relevance score between a question q and a review r is dependent on the summation of relevance scores $P(q|s)$ between each section and the question. We further derive $P(q|s)$ based on Bayes’ theorem, similar to Eq. (3). $P(q)$ represents the question prior, which can be calculated from Eq. (8). $P(s|r)$ represents the probability that a review r generates a section s, and it could be calculate as the proportion of section length to review length similar to Eq. (15). Based on Eq. (16), the top-k questions are then retrieved as candidates and to be re-ranked by promoting diversity among them. Unlike query likelihood language models introduced in Eq. (3) and Eq. (10), which find the relevant questions that match to an entire review, Eq. (16) rewards questions that match with multiple review sections.

4.3 Question diversification

Similar to traditional text summarization tasks, the final questions presented to users should avoid redundancy as much as possible. At the same time, these questions are still relevant to the review and can convey the main information in the review; and answers of the questions could be addressed in the review. In other words, we aim to achieve three goals in the final question set: relevancy, answerability and diversity. Mathematically, we formulate our objective function as a combinatorial optimization problem by following Eq. (1) as follows:

$$\begin{aligned} \arg\max_{S \subseteq V} \mathcal{F}(S) = \mathcal{L}(S) + \eta \mathcal{R}(S) \\ s.t. \sum_{q \in S} \text{length}(q) \leq b \end{aligned} \tag{17}$$

where V is the candidate question set obtained by the question retrieval component. $\mathcal{L}(S)$ measures the relevance and answerability of the review and final question set S. $\mathcal{R}(S)$ measures the diversity of the final question set. η is a constant for diversity regularization. The constraint $\sum_{q \in S} \text{length}(q) \leq b$ requires that the word count of all the questions is less than a threshold b, which is usually a small number because a concise summary is desirable for users.

The set function $\mathcal{L}(S)$ is defined to encourage the selection of questions with high relevance scores. Specifically, we use the logarithm of sum of offset relevance scores of questions in the final question set S. Formally,

$$\mathcal{L}(S) = \log\left(\sum_{q \in S} \text{score}(q) - c\right) \tag{18}$$

where $\text{score}(q)$ is the relevance score of question q. It can be calculated based on the simple query likelihood language model [Eq. (2)] or the query likelihood language model [Eq. (9)] incorporating answers (for convenience of presentation, we omit argument r and a), or with answerability measurement [Eq. (13)]. $c = \min_{q \in V}(\text{score}(q))$ is a constant to ensure the argument of $\log(\cdot)$ is always positive.

The set function $\mathcal{R}(S)$ is designed to select as "diverse" questions as possible. The function will score a set of questions high if those questions do not semantically overlap with each other. Formally,

$$\mathcal{R}(S) = \sum_{i=1}^{T} \log\left(\epsilon + \sum_{q \in P_i \cap S} r_q\right), \tag{19}$$

where $P_i, i = 1, \ldots, T$ indicates a partition of the candidate question set V into T disjoint clusters, and r_q indicates the reward of selecting question q in the final summary set. Specifically, $r_q = \frac{1}{|V|} \sum_{v \in V} w_{qv}$, where w_{qv} is the similarity score between question q and v (Lin and Bilmes 2011). Applying the logarithm function will make one cluster have diminishing gain if one question has been chosen from it. In this way, $\mathcal{R}(S)$ rewards question selection from a cluster in which none of the questions have been selected. Addition of a small positive value ϵ to the argument of the logarithm function guarantees the argument is positive.

Theorem 2 *Both $\mathcal{L}(S)$ and $\mathcal{R}(S)$ are monotone submodular functions.*

Proof For $\mathcal{L}(S)$, the function inside the logarithm function $\sum_{q \in S} \text{score}(q) - c$ is a modular function according to Definition 2 and it satisfies submodularity according to

Definition 1. The function is monotone according to Definition 3. Applying the logarithm function, which is a concave function, to the submodular function yields a submodular function $\log(\sum_{q\in S}\text{score}(q)-c)$ according to Theorem 1. Hence, $\mathcal{L}(S)$ is a monotone submodular function.

Similarly, for $\mathcal{R}(S)$, the function inside the logarithm function $\epsilon+\sum_{q\in P_i\cap S} r_q$ in $\mathcal{R}(S)$ is a modular function according to Definition 2, which satisfies submodularity according to Definition 1. The function is monotone based on Definition 3. Applying the concave logarithm function to the submodular function yields a submodular function $\log(\sum_{q\in P_i\cap S} r_q)$ according to Theorem 1. The summation of this submodular function results in a submodular function as well. Hence, $\mathcal{R}(S)$ is also a monotone submodular function.

The set function $\mathcal{F}(S)$ defined in Eq. (17) satisfies monotonicity and submodularity as it is the summation of two monotone submodular functions $\mathcal{L}(S)$ and $\mathcal{R}(S)$. □

4.4 Greedy algorithm

The monotone submodular optimization problem in Eq. (17) is still NP-hard, but Nemhauser et al. (1978) has proven that the approximated solution achieved by a greedy algorithm is guaranteed to be within $(1-1/e)$ of the optimal solution. It is worth noting that this is a worst case bound, and in most cases the quality of the solution obtained would be much better than this bound suggests. Hence, we describe an efficient approximation algorithm by utilizing monotone submodular properties of $\mathcal{F}(S)$. Algorithm 1 shows a greedy algorithm that finds approximation solution to the optimization problem in Eq. (17). The algorithm selects the best question q^* that brings maximum increase in $\mathcal{F}(S)$ at stage i, as long as the total length of questions l in the selected question set S does not exceed the threshold b. It terminates when none of the questions in the candidate set V satisfy the length threshold constraint $l+\text{length}(q)<b$.

ALGORITHM 1: The Greedy Algorithm

```
input  : candidate question set $V$ with relevance scores, length threshold $b$, diversity
         trade-off $\eta$
output: selected question set $S$, total length $l$
initialization $S \leftarrow \emptyset$, $A \leftarrow \emptyset$, $\mathcal{L}(S_q) \leftarrow \emptyset$
$l \leftarrow 0$
for $i = 1$ to $|V|$ do
    for $q \in V \setminus S$ do
        if $l + \text{length}(q) < b$ then
            $S_q \leftarrow S \cup \{q\}$
            $\mathcal{L}(S_q) \leftarrow \log\left(\sum_{q\in S_q} \text{score}(q) - c\right)$
            $\mathcal{R}(S_q) \leftarrow \sum_{t=1}^{T} \log\big(\epsilon + \sum_{q\in P_t\cap S_q} \frac{1}{|V|} \sum_{v\in V} w_{qv}\big)$
            $\mathcal{F}(S_q) \leftarrow \mathcal{L}(S_q) + \eta\mathcal{R}(S_q)$
            $A \leftarrow A \cup \{q\}$
        end
    end
    if $A = \emptyset$ then
        return $S$, $l$
    end
    $q^* \leftarrow \arg\max_{q\in A} \mathcal{F}(S_q)$
    $S \leftarrow S \cup \{q^*\}$
    $l \leftarrow l + \text{length}(q^*)$
    $A \leftarrow \emptyset$
end
return $S$, $l$
```

5 Experiments

5.1 Data collection and annotation

One of the fundamental challenges is the lack of ground-truth data available for evaluating the quality of retrieved questions. Since the proposed task is a document summarization problem, we follow the same evaluation method and metric that are used for text summarization task in NIST Document Understanding Conferences (DUC).[4]

We choose to focus on products from Amazon,[5] as it displays various kinds of products with associated reviews and question and answering (QA) data contributed by real end users. We first decide on which product category to focus in our experiment. We select products from two categories, camera and TV, and download their QA data. We rely on NLTK[6] to preprocess the content of the data, including sentence segmentation, word tokenization, lemmatization and stopword removal. We remove questions whose lengths are shorter than 3 words, as we assume there is little information conveyed in very short questions. We also discard questions that are longer than 25 words, which are supposed to convey detailed information, as they might not be general to summarize product reviews. The preprocessing step yields 331 products in the digital camera category and 226 in the TV category. Table 1 summarizes the questions and answers of products for each category.

After obtaining the QA data, we need to create a review dataset for evaluation. We first select the top 100 products retrieved from the two product categories, each for 50 products. For each product, we select the top 5 reviews ranked by Amazon's *Helpfulness* voting system, and retain only reviews whose length is between 200 and 2000 words. After obtaining the 500 reviews for the two product categories, we follow the guidelines for summary generation of NIST DUC.[7] Specifically, we request 10 graduate students to read the reviews and generate questions for each of them. The questions, which is regarded as a summary, should cover all the product features that are discussed in a product review, but not overlap with each other with respects to product features.

However, human-generated questions are expected to have very different words compared with system-selected questions. In order to mitigate such a problem, we ask students to first select questions from the question pool obtained through the crawling process. If no question can be selected, they are allowed to write their own questions. For each review, a student can select or generate up to 10 questions. The maximum length of all questions is 100. In order to accomplish the annotation task, 10 students are equally divided into two groups. The students from the first group select or write questions for reviews, and the students from another group examine the quality of questions. The students from the two groups will do one more round of annotation together to resolve any conflicts. It usually takes 50 minutes to finish question generation and examination for a single review, which is a very time-consuming process since the annotators should consider relevancy, answerability, and diversity. Even so, it is challenging to evaluate the performance of system-generated summaries. Our results shown in Sect. 6 demonstrate such an issue. We apply the same preprocessing steps (as we did for the QA data) to process the annotated

[4] http://duc.nist.gov/duc2004/.

[5] http://www.amazon.com/.

[6] http://www.nltk.org/.

[7] http://duc.nist.gov/duc2004/t1.2.summarization.instructions.

Table 1 Statistics of question data for camera and TV category

	Camera	TV
Number of products	331	226
Number of questions	8781	12,926
Average question length	11.898	11.179
Vocabulary size of questions	1196	1318
Vocabulary size of answers	2948	2541
Vocabulary size in total	2987	2668

review data. The averaged review length for camera dataset is 814.976 and the averaged review length for TV dataset is 582.932.

In Sect. 4.2.3, we introduce the Recurrent Neural Network (RNN) Encoder–Decoder, which is used to measure the answerability of a review to a question. As mentioned before, the answers to a question are not necessarily addressed by an entire review, so it is not wise to pair the entire review with each of the relevant questions. Thus, we pair each review section, which is created by authors, with questions. Considering encoding a long input takes more steps of computation than a short input does, we split long review sections whose length is longer than 150 into 2 or 3 sections. The statistics of the review data is summarized in Table 2.

5.2 Retrieval/summarization systems

In order to evaluate the performance of our proposed approach, we implement the following eight summarization systems based on the variant of our approach:

(1) Query Likelihood Model: The query generation probability is estimated based on question corpus [Eq. (3)].
(2) Combined Query Likelihood Model: The query generation probability is estimated based on question and answer corpus [Eq. (10)].
(3) Query Likelihood Model with Answerability Measurement: The likelihood of a question is calculated based on a combination of its relevance score and answerability measurement to a review section [Eqn. (13)].
(4) Query Likelihood Model with Maximal Marginal Relevance (MMR): re-rank retrieved questions by query likelihood model [system (1)] using MMR (Carbonell and Goldstein 1998), which is designed to remove redundancy while preserving the relevance by using a trade-off parameter σ. Note that MMR is non-monotone submodular, so a greedy algorithm is not theoretically guaranteed to be a constant factor approximation algorithm (Lin and Bilmes 2011).
(5) Combined Query Likelihood Model with Maximal Marginal Relevance: re-rank retrieved questions by combined query likelihood model [system (2)] using MMR.
(6) Query Likelihood Model with Submodular Function: re-rank retrieved questions by query likelihood model [system (1)] using submodular function [Eq. (17)].
(7) Combined Query Likelihood Model with Submodular Function: re-rank retrieved questions by combined query likelihood model [system (2)] using submodular function.
(8) Query Likelihood Model with Answerability Measurement and Submodular Function: re-rank retrieved questions by query likelihood model with answerability measurement [system (3)] using submodular function.

Table 2 Statistics of review data for camera and TV category

	Camera	TV
Minimal number of review sections	2	1
Maximal number of review sections	40	42
Average number of review sections	13.728	10.296
Minimal review length per section	10	10
Maximal review length per section	150	150
Average review length per section	56.591	52.711

We experiment with different parameter settings on both camera and TV datasets. For system (1), (2) and (3), we empirically choose the Jelinek-Mercer smoothing parameter λ between 0.1 and 0.3 [Eq. (5)]. For system (4) and (5), we choose the trade-off parameter σ between 0 and 1.0. For system (6), (7) and (8), the number of questions in the candidate set V [Eq. (17)] is set to 100, and the number of clusters [Eq. (19)] is set to 10. We rely on K-means clustering algorithm to partition V, which leverages IDF-weighted term vector for each question. We also experiment with different settings of smoothing parameter α [Eq. (12)] and diversity regularizer η [Eq. (17)], which will be shown in Sect. 6.3.

For system (3) and system (8), we employ an attentional Recurrent Neural Network (RNN) Encoder–Decoder (Bahdanau et al. 2014) to score a pair of question and a review section. We choose gated recurrent neural network (GRU) as the RNN architecture. Each GRU has 2 hidden layers, and each layer has 128 hidden units. The RNN Encoder–Decoder is trained on a large-scale public product Q&A dataset[8] (McAuley and Yang 2016) with around 1.4 million answered questions. We select the electronics category for training the RNN as the evaluation data is obtained from the same category. We treat the answers of products as the input sequence, and the questions as the output sequence. The maximum length of answer sequence is 160 and the maximum length of question sequence is 30. We remove sequences whose length is below 5. The total number of training pairs is 300k. We use NLTK to perform lemmatization for each sentence. The vocabulary size for question data is 7,839 and the vocabulary size for answer data is 12,009. The training is run using the TensorFlow library (Abadi et al. 2015). We set the batch size to 32, the learning rate of stochastic gradient descent (SGD) to 0.1, and the number of training epochs to 10. We tune the trade-off parameter γ in Eq. (13) between 0.1 and 1.0.

5.3 Evaluation metrics

We follow the evaluation of conventional summarization systems to measure the performance of the aforementioned eight systems for finding questions to summarize a product review. Specifically, we rely on ROUGE[9] (Recall-Oriented Understudy for Gisting Evaluation), which measures how well a system-generated summary matches the content in a human-generated summary based on n-gram co-occurrence (Lin 2004). In our experiment, we compare unigram and bigram-based ROUGE scores.

[8] http://jmcauley.ucsd.edu/data/amazon/qa/.

[9] http://www.rxnlp.com/rouge-2-0/.

6 Results

6.1 Qualitative analysis

6.1.1 The impact of answerability

We first show the feasibility of our method to retrieve questions whose answers can be addressed by a review. We set the length threshold of a question-based summary as 100. Table 3, 4 and 5 show the questions annotated by human, retrieved by simple query likelihood language model, and selected by query likelihood language model incorporating answerability measurement for a camera review.[10] The author mainly discussed the experience of using a Nikon D3300 camera and its comparison with other Nikon models, D3200 and D5200, which correspond to the first and second questions in human generated summary. The third question, which is more detailed, is asking the size and weight of the camera.

All but the seventh question selected by simple query likelihood language model (shown in Table 5) contain the name of several camera models (e.g., D3200, D3300, and D5200), as they frequently occurred in this review. However, the questions in the 1st, 2nd, 3rd, and 5th positions are not semantically aligned with the main points discussed in the review, the strengths and weaknesses of D3300, even they contain the name of the camera model D3300. Those questions are answerable to reviews that discuss about other camera models, e.g., d750 for the 2nd question, and rebel t5 for the 5th question. The last two questions make more senses as they are addressed by the review segment shown below:

> Although the D3300 is the eventual replacement for the D3200, I purchased the D3300 in anticipation of replacing my D5200 assuming that this newer camera would have improved image quality over last year's models. I was actually somewhat disappointed as I preferred the image quality of the older D5200. That is not to say that the D3300 is not an excellent camera because actually it is.

After incorporating answerability measurement to the query likelihood language model, the final question-based summary contains more detailed questions asking color mode, comparison between D5200 and D5300, image quality, weight and size, and iso setting (shown in Table 5). The answers of the 1st, 2nd, 5th, and 7th questions are addressed in multiple sections of the review (shown in Table 6). They could be used as good candidate questions to summarize the review, even though they are not selected by the annotator, who provides a more general and abstract summary for the review.

6.1.2 The impact of diversity

In this section, we show the feasibility of our method to retrieve non-redundant questions that can be used to summarize a review. We take one review[11] from the digital camera category from Amazon as an example. The review length is around 700 tokens after preprocessing. The following segment shows the main aspects that the author talks about:

[10] https://www.amazon.com/gp/customer-reviews/R1LK357AT7ZJ3D?ASIN=B00HQ4W1QE.

[11] http://www.amazon.com/gp/customer-reviews/R360W96STA0KUI?ASIN=B0158SRJVQ.

Table 3 Human annotation (Nikon D3300)

(1) What is the biggest physical change of the Nikon D3300?

(2) What is the reason I prefer the D5200 to the D3300?

(3) I am seeking a small and light DSLR,Can you help me?

Table 4 Questions retrieved by query likelihood model (Nikon D3300)

(1) Nikon D3300 or this camera? Which has better image quality and features?

(2) I have nikon d3300 w 70200 2.8 but image quality is terrible in night football games. will d750 vastly improve that?

(3) I really like this camera but still confuse between D3300

(4) Is the nikon d3300 dslr camera with 18-55mm and 55-200mm lenses kit available with the red camera? and if so, is it the same price?

(5) Which would be a better beginner dslr for the price, nikon d3300 or this rebel t5?

(6) What are the essential differences between the D5200 and D3200?

(7) Anything actually affecting image quality?

Table 5 Questions retrieved by query likelihood model incorporating answerability (Nikon D3300)

(1) Dose it have selective color mode like D5200?

(2) What are the essential differences between the D5200 and D3200? Anything actually affecting image quality?

(3) Are all the settings manual like aperture and iso?

(4) Is it true that you cannot program video record to one of the customization button (i.e., C1)?

(5) Due to its light weight and small size, is SL1 balanced w/ a standard macro or zoom lens? Is it front-heavy? Hard to keep steady?

(6) How is the lowlight video? and how high is the iso not usable for client use? Thank you!

(7) What is the highest iso setting?

> ...Highlights: 14 bit uncompressed RAW, 4k video internal recording, new 50% quiet shutter rated at 500,000 cycles, 5-axis stabilization, better EVF, better signal to noise ratio...

Table 7 shows the questions edited by a human annotator. The first five questions are selected from the question corpus, while the last two are created by the annotator. Basically, the questions correspond to the top features highlighted in the review segment, and covers all the aspects that are discussed in the review, including RAW files, 4K recording, shutter, stabilization, EVF, low light performance, and sensor. The last two aspects are not mentioned in the segment but are discussed in the main body. Table 8 shows the top-10 questions retrieved by query likelihood language model smoothed by answers. They cover the following aspects, camera's performance in low light (the 1st, 3rd, 5th, and 7th question), comparison between different camera models (the 2nd question), lens adaption (the 4th question), video recording (6th question), shutter (the 9th and 10th

Table 6 Questions retrieved by query likelihood model incorporating answerability and their answers from review (Nikon D3300)

Q: Dose it have selective color mode like D5200?

A: It seemed like the D3300 colors needed to be manually re-adjusted for many different lighting situations. Each of these cameras benefited from shooting raw with the JPGs of each camera being a bit too warm and under-sharpened. However, the JPGs rendered by the D5200 resulted in more pleasing colors than the D3300 (to me anyway).

Q: What are the essential differences between the D5200 and D3200? Anything actually affecting image quality?

A: The Nikon D3300 is smaller and lighter than its predecessors, the D3200 and D3100. It is also considerably smaller and lighter than the D5200, the somewhat more advanced entry level Nikon DSLR. The reason I prefer the D5200 to the D3300 is white balance and color rendition.

Q: Due to its light weight and small size, is SL1 balanced w/ a standard macro or zoom lens? Is it front-heavy? Hard to keep steady?

A: The reduced size and weight of the D3300 appears to be Nikon's response to Canon's 100D/SL1. Although the SL1 and D3300 are about the same size and weight, the D3300 has a better/larger grip and is more comfortable (to me anyway) than the SL1.

Q: What is the highest iso setting?

A: Both cameras delivered excellent high ISO results with similar ISO performance through ISO 3200 (I really do not like shooting past ISO 3200). High ISO performance on the D3300 was better than its predecessor, the D3200. On the D3300 and D5200, ISO 800 is really indistinguishable from ISO 100. ISO 1600 is also very good on both cameras with some graininess/noise creeping in. ISO 3200 is usable but there is a definite degradation in image quality.

question), and a general one (the 8th question). It shows that three of the top-5 results are redundant with respect to low light performance, and the last two questions overlap with each other with respect to shutter noise.

Table 9 shows the top-10 questions selected by the submodular function. The re-ranked questions cover the following aspects: camera's performance in low light (the 1st, 6th, 8th, and 10th question), comparison between different camera models (the 2nd question), shutter (the 3rd and 9th question), video recording (4th question), lens adaption (the 5th question), and RAW files (7th question). Compared with questions retrieved by query likelihood model, even though there still exist four questions that are relevant to low light performance, three of the related questions are demoted from the top due to their redundancy with the top-1 question. The questions asking camera model comparison and shutter noise are promoted because they are semantically dissimilar to the top-1 question. There are non-redundant questions in top-5 positions of the re-ranked list. The re-ranking function is able to promote one question related to RAW files, which is not included in the candidate question set retrieved by query likelihood model. In addition, it also demotes the general question which was ranked at the 8th position, probably because it is not representative of questions asking product aspects.

By comparing the human annotation with retrieved/ranked question set, there are overlaps such as low light performance, RAW files, 4K video recording, and shutter noise. Still, there are three aspects annotated by annotator that are not covered in the re-ranked question list: image stabilization, sensor, and EVF. It is not surprising that the retrieved questions do not cover the last two aspects, sensor and EVF, as the annotator does not select relevant questions from the question pool either. Meanwhile, the questions related to comparison between different models and adaption of lenses are not selected by annotator. However, if we take a close look at the review, we can find some relevant sentences that can be used to answer the retrieved questions regarding the two questions:

Table 7 Human annotation (Sony a7S II)

(1) How does this camera take videos in low light?
(2) Does this camera provide RAW Image format?
(3) Does this camera record 4K internally?
(4) Does this camera have image stabilization?
(5) How would you describe the shutter noise?
(6) Does the EVF work well in bright conditions?
(7) Is there much of a difference in term of sensor?

Table 8 Questions retrieved by query likelihood model (Sony a7S II)

(1) What were the improvements to the low light capabilities of the sensor?
(2) What are the key differences between the a7, the a7r and the a7s?
(3) How is the camera for indoor low light? I've had Sony point and shoots in the past and the interior shots had so much noise.
(4) What lens adapter would allow someone to use canon ef lenses on the a7s and a7s ii with reasonable autofocus performance?
(5) One review claims the camera has very poor low light performance for video, lots of video noise. Comments from videographers?
(6) Do you need a special external recorder for 4k video like it is with a7s?
(7) Very curious to see how it does in low light. did sony really solve the noise problem??
(8) Where is it better? or is it?
(9) Does the a7II have a silent electronic shutter like the a7s?
(10) Is the shutter noise less pronounced than the a7?

> ...Sony, having already introduced 2nd gen versions on the A7 and A7R, is now applying the same treatment to the A7S. The A7S II blends and combines a variety of features from the two aforementioned cameras... The 7S II can record internally, thus eliminating the additional cost of an external recorder which in turn can allow one to spend the money on additional lenses...

Considering the nature that summarizing a review is highly subjective, the questions generated by the proposed automatic retrieval and re-ranking method are reasonable and cover most of the aspects discussed in a product review.

6.2 Quantitative analysis

The results on the two datasets (introduced in Sect. 5.1) achieved by different summarization systems (introduced in Sect. 5.2) are shown in Tables 10, 11, 12 and 13. We set the total length threshold b in Eq. (1) as 50, 75, and 100, respectively. Boldface stands for the best performance per column with respect to each length threshold. We conduct paired t-test for all comparisons of results achieved by two different methods. † indicates the corresponding method outperforms the simple query likelihood baseline statistically significantly at the 0.05 level, and ‡ indicates the corresponding method outperforms all the other methods significantly at the 0.05 level.

On the TV dataset, the combined query likelihood language model (QL(Q, A)) and the query likelihood language model with answerability measurement (QL + answerability)

Table 9 Questions re-ranked by submodular function (Sony a7S II)

(1) What were the improvements to the low light capabilities of the sensor?

(2) What are the key differences between the a7, the a7r and the a7s?

(3) Is the shutter noise less pronounced than the a7?

(4) Does sony a7r ii have the maximum aperture of f3.5 when video recording as other sony camera?

(5) What lens adapter would allow someone to use canon ef lenses on the a7s and a7s ii with reasonable autofocus performance?

(6) How is the camera for indoor low light? I've had Sony point and shoots in the past and the interior shots had so much noise.

(7) Raw files, Would I see higher noise in the raw files?

(8) One review claims the camera has very poor low light performance for video, lots of video noise. Comments from videographers?

(9) Does the a7II have a silent electronic shutter like the a7s?

(10) Very curious to see how it does in low light. did sony really solve the noise problem??

yields better results than simple query likelihood language model (QL(Q)) does in terms of all evaluation metrics for different length threshold settings. Using MMR to re-rank questions achieves higher ROUGE scores against QL(Q) does except for bigram-ROUGE scores when b is set to 75. The results achieved by QL$(Q, A) + MMR$ are higher than QL(Q, A) does for ROUGE scores when b is set to 75 and 100. Using the submodular function to re-rank the questions retrieved by simple and combined query likelihood language model (denoted as QL(Q) +sub and QL(Q, A) + sub, respectively) show better results over corresponding retrieval models for all evaluation metrics. QL(Q, A) + sub achieves better results than the first five systems do for all evaluation metrics. QL(Q) + answerability + sub yield better ROUGE scores than all the other systems without diversity promotion except for unigram-ROUGE score when b is set to 50.

On the camera dataset, unfortunately, incorporating answer corpus in the query likelihood language model does not bring improvement on the ROUGE scores. One possible reason is that the vocabulary size of answer collections for the camera category is larger than that of the TV category according to Table 1. Incorporating an answer collection might add many irrelevant words to the language model, such that the results retrieved by QL(Q, A) contain more noises than that by QL(Q). Incorporating answerability measurement helps improve unigram-ROUGE scores achieved by QL(Q) except when b is set to 50. After promoting diversity in the retrieved question set using MMR, QL(Q) + MMR is able to achieve competitive results against QL(Q) except for bigram ROUGE scores when $b = 100$; but QL(Q, A) + MMR does not consistently yields better results than QL(Q, A) across different length settings.

Even though the combined retrieval model does not help increase the ROUGE scores, QL(Q) + sub and QL(Q, A) + sub yields competitive ROUGE scores than retrieval models without diversity promotion do. QL(Q) + answerability + sub achieves the highest ROUGE scores for all evaluation metrics. It even significantly outperforms all the other systems for unigram-ROUGE scores.

In summary, query likelihood model incorporating answers is able to yield better summarization performance when the vocabulary size of the answer collection is moderate. Incorporating answerability measurement helps improve ROUGE scores on the TV dataset and competitive ROUGE scores on the camera dataset. The results achieved by query

Table 10 Summarization results (Unigram-ROUGE Scores) on TV dataset

Length	Method	ROUGE1-R	ROUGE1-P	ROUGE1-F_1
50	$\mathrm{QL}(Q)$	0.248	0.177	0.192
	$\mathrm{QL}(Q,A)$	0.267	0.190	0.205
	QL(Q) + answerability	0.258	0.190	0.203
	QL + MMR	0.250	0.181	0.195
	$\mathrm{QL}(Q,A)$ + MMR	0.263	0.189	0.204
	$\mathrm{QL}(Q)$ + sub	0.268	0.190	0.206
	$\mathrm{QL}(Q,A)$ + sub	**0.288**†	0.209	0.225
	$\mathrm{QL}(Q)$ + answerability + sub	0.287	**0.212**†	**0.226**†
75	$\mathrm{QL}(Q)$	0.324	0.157	0.199
	$\mathrm{QL}(Q,A)$	0.334	0.161	0.203
	QL(Q) + answerability	0.329	0.164	0.206
	$\mathrm{QL}(Q)$ + MMR	0.326	0.158	0.200
	$\mathrm{QL}(Q,A)$ + MMR	0.336	0.162	0.205
	$\mathrm{QL}(Q)$ + sub	0.332	0.161	0.203
	$\mathrm{QL}(Q,A)$ + sub	0.353	0.175	0.220
	$\mathrm{QL}(Q)$ + answerability + sub	**0.357**†	**0.177**†	**0.222**†
100	$\mathrm{QL}(Q)$	0.372	0.137	0.190
	$\mathrm{QL}(Q,A)$	0.380	0.140	0.194
	QL(Q) + answerability	0.387	0.145	0.199
	$\mathrm{QL}(Q)$ + MMR	0.376	0.139	0.192
	$\mathrm{QL}(Q,A)$ + MMR	0.386	0.142	0.196
	$\mathrm{QL}(Q)$ + sub	0.382	0.140	0.194
	$\mathrm{QL}(Q,A)$ + sub	0.401	0.150	0.207
	$\mathrm{QL}(Q)$ + answerability + sub	**0.411**†	**0.152**†	**0.210**†

likelihood models with the submodular function are promising compared with conventional diversity promotion technique. The combined query likelihood model with submodular function yields significantly better performance on the TV dataset for ROUGE metrics. This model also shows the potential ability to promote relevant questions by rewarding diversified results on the camera dataset. With diversity promotion, the query likelihood model with answerability measurement achieves the highest ROUGE scores on both camera dataset.

6.3 Parameter analysis

In order to examine the impact of the smoothing parameter α of the answer collection [Eq. (10)], the trade-off between relevancy and answerability [Eq. (13)], diversity regularizer η and number of question clusters T for the sumbodular function [Eq. (17)], we examine the summarization performance (unigram-ROUGE F_1 scores) achieved by system (2), (3), (7), and (8) (introduced in Sect. 5.2) with different settings of α, γ, η, and T respectively on the TV and camera datasets. All the length threshold is set to 50. The

Table 11 Summarization results (Bigram-ROUGE Scores) on TV dataset

Length	Method	ROUGE2-R	ROUGE2-P	ROUGE2-F$_1$
50	QL(Q)	0.0440	0.0281	0.0313
	QL(Q, A)	0.0447	0.0303	0.0329
	QL(Q) + answerability	0.0449	0.0296	0.0319
	QL + MMR	0.0449	0.0296	0.0319
	QL(Q, A) + MMR	0.0414	0.0292	0.0312
	QL(Q) + sub	0.0440	0.0302	0.0330
	QL(Q, A) + sub	0.0601	0.0409	0.0446
	QL(Q) + answerability + sub	**0.0623**†	**0.0429**†	**0.0460**†
75	QL(Q)	0.0590	0.0261	0.0335
	QL(Q, A)	0.0605	0.0273	0.0347
	QL(Q) + answerability	0.0592	0.0269	0.0341
	QL(Q) + MMR	0.0580	0.0260	0.0330
	QL(Q, A) + MMR	0.0630	0.0290	0.0370
	QL(Q) + sub	0.0612	0.0274	0.0352
	QL(Q, A) + sub	0.0797	0.0361	0.0462
	QL(Q) + answerability + sub	**0.0813**†	**0.0370**†	**0.0465**†
100	QL(Q)	0.0696	0.0237	0.0333
	QL(Q, A)	0.0746	0.0255	0.0355
	QL(Q) + answerability	0.0784	0.0260	0.0363
	QL(Q) + MMR	0.0700	0.0240	0.0340
	QL(Q, A) + MMR	0.0800	0.0270	0.0370
	QL(Q) + sub	0.0757	0.0254	0.0357
	QL(Q, A) + sub	0.0921	0.0315	0.0441
	QL(Q) + answerability + sub	**0.0978**†	**0.0321**†	**0.0449**†

ROUGE curves achieved with other threshold settings follow similar patterns so we leave them out.

Figure 2 shows the unigram-ROUGE F_1 scores achieved by different α between 0 and 1 with an interval of 0.1. The Jelinek-Mercer(JM) smoothing parameter λ for combined query likelihood language model is set to 0.3 for both datasets. For the TV dataset, as shown in the previous section, incorporating answers benefits the simple query likelihood language model estimated on the question collection. When α is greater than zero, the unigram-ROUGE F_1 scores increase with the benefit of the integration of the answer collection. For the camera dataset, results have shown that the answer collection does not help increase the unigram-ROUGE F_1 scores. With larger α values, the scores are consistently lower than that achieved by the query likelihood language model without the incorporation of answers. We set $\alpha = 0.3$ for both datasets.

Figure 3 shows the impact of answerability measurement on the question retrieval model. The JM smoothing parameter is set to 0.3 for both datasets. When $\gamma = 0.0$, the ranking function itself is a simple query likelihood model; and when $\gamma = 1.0$, the ranking function is dominated by the answerability of a question to a review. On both datasets, adding answerability measurement achieves higher unigram-ROUGE F_1 scores, which is

Table 12 Summarization results (Unigram-ROUGE Scores) on camera dataset

Length	Method	ROUGE1-R	ROUGE1-P	ROUGE1-F_1
50	QL(Q)	0.218	0.260	0.227
	QL(Q,A)	0.211	0.258	0.223
	QL(Q) + answerability	0.217	0.266	0.229
	QL(Q) + MMR	0.218	0.263	0.229
	QL(Q,A) + MMR	0.210	0.259	0.223
	QL(Q) + sub	0.223	0.273	0.236
	QL(Q,A) + sub	0.225	0.275	0.238
	QL(Q) + answerability + sub	**0.236**‡	**0.291**‡	**0.250**‡
75	QL(Q)	0.286	0.231	0.245
	QL(Q,A)	0.277	0.228	0.240
	QL(Q) + answerability	0.291	0.240	0.253
	QL(Q) + MMR	0.288	0.234	0.248
	QL(Q,A) + MMR	0.277	0.229	0.241
	QL(Q) + sub	0.295	0.241	0.254
	QL(Q,A) + sub	0.297	0.242	0.256
	QL(Q) + answerability + sub	**0.310**‡	**0.255**‡	**0.268**‡
100	QL(Q)	0.342	0.209	0.249
	QL(Q,A)	0.333	0.207	0.246
	QL(Q) + answerability	0.352	0.219	0.259
	QL(Q) + MMR	0.344	0.211	0.251
	QL(Q,A) + MMR	0.331	0.207	0.245
	QL(Q) + sub	0.350	0.216	0.257
	QL(Q,A) + sub	0.352	0.217	0.258
	QL(Q) + answerability + sub	**0.267**‡	**0.229**‡	**0.271**‡

consistent with the analysis in Sect. 6.2. With the increasing values of γ, the ROUGE scores on the TV dataset slightly decrease, but are still higher that that by simple query likelihood language model. The unigram-ROUGE F_1 scores achieved on the camera dataset slightly increase until $\gamma = 0.5$. When the ranking score is dominated by answerability measurement, the ROUGE scores are inferior than that by simple query likelihood, which demonstrates the necessity of retrieval models for the selection of high-quality question candidates. The values between 0.1 and 0.4 would be good choices for both datasets as the ROUGE scores are consistently higher than that when $\gamma = 0.0$. In our experiments, we set $\gamma = 0.2$ for both datasets.

Figure 4 shows the impact of diversity regularizer η on the combined query likelihood language model. The JM smoothing parameters for TV and camera datasets are set to 0.2 and 0.3 respectively. With the increasing η values, the unigram-ROUGE F_1 scores increase on both datasets. These numbers are consistent with previous findings that adding submodular function to retrieval models will improve the summarization results. It shows that $\eta = 5.0$ is a good choice for both datasets.

Figure 5 shows the impact of number of question clusters T on the query likelihood language model with answerability measurement and diversity promotion. The JM

Table 13 Summarization results (Bigram-ROUGE Scores) on camera dataset

Length	Method	ROUGE2-R	ROUGE2-P	ROUGE2-F$_1$
50	QL(Q)	0.0463	0.0520	0.0467
	QL(Q,A)	0.0406	0.0497	0.0427
	QL(Q) + answerability	0.0424	0.0511	0.0443
	QL(Q) + MMR	0.0469	0.0531	0.0474
	QL(Q,A) + MMR	0.0401	0.0491	0.0422
	QL(Q) + sub	0.0484	0.0585	0.0507
	QL(Q,A) + sub	0.0477	0.0605	0.0511
	QL(Q) + answerability + sub	**0.0510**	**0.0651**†	**0.0547**†
75	QL(Q)	0.0626	0.0474	0.0516
	QL(Q,A)	0.0530	0.0433	0.0455
	QL(Q) + answerability	0.0593	0.0474	0.0503
	QL(Q) + MMR	0.0634	0.0482	0.0523
	QL(Q,A) + MMR	0.0528	0.0435	0.0456
	QL(Q) + sub	0.0648	0.0511	0.0546
	QL(Q,A) + sub	0.0617	0.0509	0.0532
	QL(Q) + answerability + sub	**0.0692**	**0.0576**†	**0.0602**†
100	QL(Q)	0.0785	0.0447	0.0545
	QL(Q,A)	0.0661	0.0410	0.0484
	QL(Q) + answerability	0.0778	0.0463	0.0557
	QL(Q) + MMR	0.0773	0.0445	0.0541
	QL(Q,A) + MMR	0.0656	0.0406	0.0481
	QL(Q) + sub	0.0786	0.0467	0.0562
	QL(Q,A) + sub	0.0759	0.0474	0.0558
	QL(Q) + answerability + sub	**0.0835**	**0.0516**†	**0.0612**

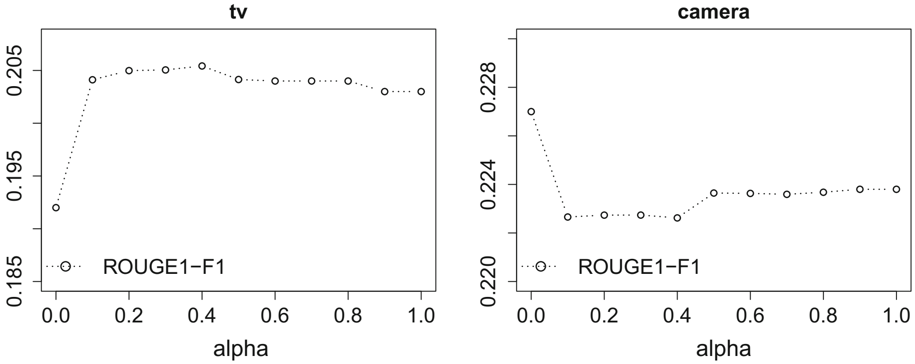


Fig. 2 ROUGE-1 F_1 scores on TV and camera datasets by combined query likelihood language model with different weights of answer collection

smoothing parameters for both datasets are set to 0.3. The number of candidate questions is set to 100. When $T = 0$, no re-ranking function is applied to candidate question set. For both datasets, applying diversity function help increase the unigram-ROUGE F_1 when T is

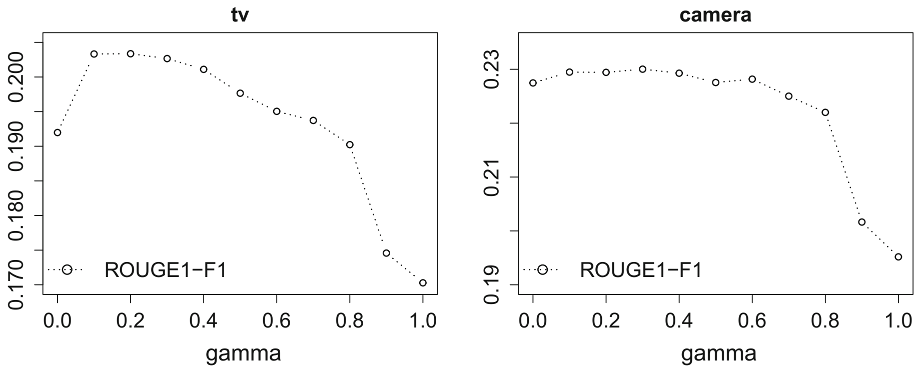


Fig. 3 ROUGE-1 F_1 scores on TV and camera datasets by query likelihood language model with different weights of answerability measurement

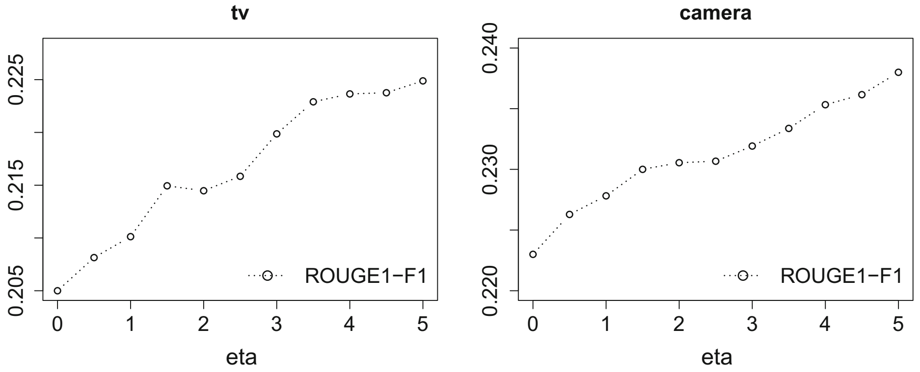


Fig. 4 ROUGE-1 F_1 scores on TV and camera datasets by combined query likelihood language model with different diversity regularizer

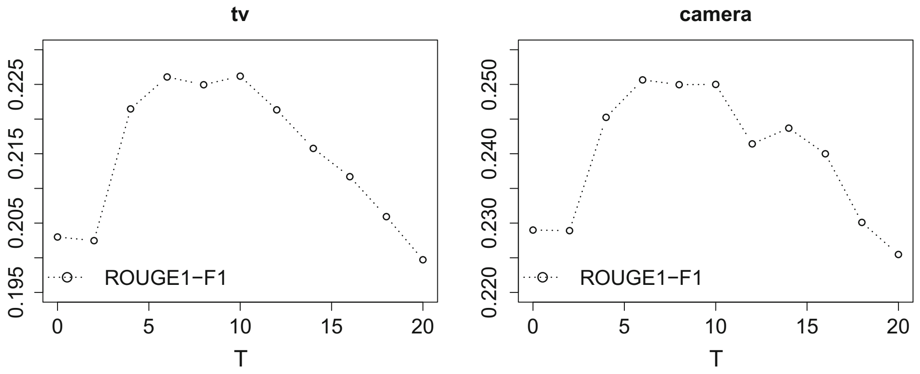


Fig. 5 ROUGE-1 F_1 scores on TV and camera datasets by query likelihood language model with answerability measurement with different number of question clusters

set to between 6 and 10. The score decrease when T is greater than 10. We set $T = 10$ for both datasets.

7 Conclusions and future work

This paper addresses a new task: summarizing a review through questions. Questions are often more attractive for customers to read than plain opinion sentences are. They can serve as “hints” for customers to decide whether they want to further read the review. To the best of our knowledge, no prior work has studied this task. We propose a two-stage approach consisting of question selection and question diversification. Question selection is based on relevancy and answerability measurement between questions and a review. Submodular optimization is used to consider both question coverage and non-redundancy. To evaluate the proposed approach, we create and annotate a dataset by manually locating and editing questions for reviews in two product categories. The experimental results demonstrate the proposed approach can effectively find relevant questions for review summarization.

This work is an initial step towards a promising research direction. In future work, we will utilize more information about products such as product specifications and question ratings to enrich the proposed question retrieval component. Regarding question diversification, we will explore other submodular functions. We also would like to deploy the proposed method to a real-world review system and measure the satisfaction of real users. Last but not the least, we plan to apply the proposed approach to other information retrieval tasks. In fact, the problem tackled in this paper can be formulated as an inverse question-answering task by using given answers to find relevant questions, which may have numerous applications in the real world. We will also explore fully generative approaches to generate questions given answers instead of retrieving from a predefined candidate question set by utilizing sequence-to-sequence learning.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., & Devin, M., et al. (2015). Tensorflow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow org 1.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv:14090473

Bendersky, M., Croft, W. B., Smith, D. A. (2011). Joint annotation of search queries. In *ACL* (pp. 102–111).

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, *3*(Feb), 1137–1155.

Berger, A., & Lafferty, J. (1999). Information retrieval as statistical translation. In *SIGIR* (pp. 222–229).

Carbonell, J., & Goldstein, J. (1998). The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR* (pp. 335–336).

Chali, Y., & Hasan, S. A. (2012). Towards automatic topical question generation. In *COLING* (pp. 475–492).

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv:14061078

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv:14123555

Dang, V., & Croft, B. W. (2010). Query reformulation using anchor text. In *WSDM* (pp. 41–50).

Dang, V., & Croft, W. B. (2012). Diversity by proportionality: An election-based approach to search result diversification. In *Proceedings of the 35th international ACM SIGIR conference on research and development in information retrieval* (pp. 65–74). ACM.

Fujishige, S. (2005). *Submodular functions and optimization* (Vol. 58). Amsterdam: Elsevier.

Goldstein, J., Kantrowitz, M., Mittal, V., & Carbonell, J. (1999). Summarizing text documents: Sentence selection and evaluation metrics. In *SIGIR* (pp. 121–128).

Goldstein, J., Mittal, V., Carbonell, J., & Kantrowitz, M. (2000). Multi-document summarization by sentence extraction. In *NAACL-ANLP workshop on automatic summarization* (pp. 40–48).

Gomez Rodriguez, M., Leskovec, J., & Krause, A. (2010). Inferring networks of diffusion and influence. In *SIGKDD* (pp. 1019–1028).

Gupta, M., & Bendersky, M. (2015). Information retrieval with verbose queries. In *SIGIR* (pp. 1121–1124).

Harman, D. (2002). Overview of the trec 2002 novelty track. In *TREC*.

Heilman, M., & Smith, N. A. (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, association for computational linguistics* (pp. 609–617).

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Ar, Mohamed, Jaitly, N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, *29*(6), 82–97.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hovy, E., & Lin, C. Y. (1998). Automated text summarization and the summarist system. In *Proceedings of a workshop on held at Baltimore, Maryland: October 13–15, 1998, Association for Computational Linguistics* (pp. 197–214).

Hu, M., & Liu, B. (2004). Mining opinion features in customer reviews. *AAAI*, *4*, 755–760.

Huston, S., & Croft, W. B. (2010). Evaluating verbose query processing techniques. In *SIGIR* (pp. 291–298).

Jo, Y., & Oh, A. H. (2011). Aspect and sentiment unification model for online review analysis. In *WSDM* (pp. 815–824).

Kim, H. D., Ganesan, K., Sondhi, P., & Zhai, C. (2011). Comprehensive review of opinion summarization. UIUC Technical Report.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Kumaran, G., & Carvalho, V. R. (2009). Reducing long queries using query quality predictors. In *SIGIR* (pp. 564–571).

Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., & Glance, N. (2007). Cost-effective outbreak detection in networks. In *SIGKDD* (pp. 420–429).

Li, F., Liu, N., Jin, H., Zhao, K., Yang, Q., & Zhu, X. (2011). Incorporating reviewer and product information for review rating prediction. *IJCAI*, *11*, 1820–1825.

Lin, C. Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop* (Vol. 8).

Lin, H., & Bilmes, J. (2010). Multi-document summarization via budgeted maximization of submodular functions. In *NAACL* (pp. 912–920).

Lin, H., & Bilmes, J. (2011). A class of submodular functions for document summarization. In *ACL* (pp. 510–520).

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, *5*(1), 1–167.

Liu, C. L., Hsaio, W. H., Lee, C. H., Lu, G. C., & Jou, E. (2012). Movie rating and review summarization in mobile environment. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, *42*(3), 397–407.

Liu, M., Calvo, R. A., & Rus, V. (2014). Automatic generation and ranking of questions for critical review. *Educational Technology and Society*, *17*(2), 333–346.

Ly, D. K., Sugiyama, K., Lin, Z., & Kan, M. Y. (2011). Product review summarization from a deeper perspective. In *ACM/IEEE joint conference on digital ibraries* (pp. 311–314).

Mani, I., & Maybury, M. T. (1999). *Advances in automatic text summarization* (Vol. 293). Cambridge: MIT Press.

McAuley, J., & Yang, A. (2016). Addressing complex and subjective product-related queries with customer reviews. In *Proceedings of the 25th international conference on World Wide Web, International World Wide Web Conferences Steering Committee* (pp. 625–635).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Mostafazadeh, N., Misra, I., Devlin, J., Mitchell, M., He, X., & Vanderwende, L. (2016). Generating natural questions about an image. arXiv:160306059

Nallapati, R., Zhou, B., Gulçehre, Ç., & Xiang, B. (2016). Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of the 20th SIGNLL conference on computational natural language learning (CoNLL)* (pp. 280–290).

Nemhauser, G. L., Wolsey, L. A., & Fisher, M. L. (1978). An analysis of approximations for maximizing submodular set functions-i. *Mathematical Programming*, *14*(1), 265–294.

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, *2*(1–2), 1–135.

Parikh, N., Sriram, P., & Al Hasan, M. (2013). On segmentation of ecommerce queries. In *CIKM* (pp. 1137–1146).

Park, D. H., Kim, H. D., Zhai, C., & Guo, L. (2015). Retrieval of relevant opinion sentences for new products. In *SIGIR* (pp. 393–402).

Qin, L., & Zhu, X. (2013). Promoting diversity in recommendation by entropy regularizer. In *IJCAI* (pp. 2698–2704).

Rus, V., & Arthur, C. G. (2009). *The question generation shared task and evaluation challenge*. Citeseer: The University of Memphis, National Science Foundation.

Santos, R. L., Macdonald, C., & Ounis, I. (2010). Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th international conference on World wide web* (pp. 881–890), ACM.

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*(11), 2673–2681.

Shephard, R. W., & Färe, R. (1974). *The law of diminishing returns*. New York: Springer.

Soboroff, I. (2004). Overview of the trec 2004 novelty track. In *TREC*.

Soboroff, I., & Harman, D. (2003). Overview of the trec 2003 novelty track. In *TREC*.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP), Citeseer* (Vol. 1631, p. 1642).

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104–3112).

Svore, K. M., Vanderwende, L., & Burges, C. J. (2007). Enhancing single-document summarization by combining ranknet and third-party sources. In *EMNLP-CoNLL* (pp. 448–457).

Titov, I., & McDonald, R. T. (2008). A joint model of text and aspect ratings for sentiment summarization. *ACL*, *8*, 308–316.

Wang, H., Lu, Y., & Zhai, C. (2010). Latent aspect rating analysis on review text data: A rating regression approach. In *SIGKDD* (pp. 783–792).

Xue, X., Jeon, J., & Croft, W. B. (2008). Retrieval models for question and answer archives. In *SIGIR* (pp. 475–482).

Xue, X., Tao, Y., Jiang, D., & Li, H. (2012). Automatically mining question reformulation patterns from search log data. In *ACL* (pp. 187–192).

Yatani, K., Novati, M., Trusty, A., & Truong, K. N. (2011). Analysis of adjective-noun word pair extraction methods for online review summarization. In *IJCAI* (Vol. 22, p. 2771).

Zhai, C., & Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)*, *22*(2), 179–214.

Zhao, S., Wang, H., Li, C., Liu, T., & Guan, Y. (2011). Automatically generating questions from queries for community-based question answering. In *IJCNLP* (pp. 929–937).

Zhou, G., Cai, L., Zhao, J., & Liu, K. (2011). Phrase-based translation model for question retrieval in community question answer archives. In *ACL* (pp. 653–662).

Zhou, G., He, T., Zhao, J., & Hu, P. (2015). Learning continuous word embedding with metadata for question retrieval in community question answering. In *ACL* (pp. 250–259).