Retrieving Non-Redundant Questions to Summarize a Product Review

Mengwen Liu*
College of Computing and Informatics
Drexel University
Philadelphia 19104, PA, USA
ml943@drexel.edu

Yi Fang*
Department of Computer Engineering
Santa Clara University
Santa Clara 95053, CA, USA
yfang@scu.edu

Dae Hoon Park
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
dpark34@illinois.edu

Xiaohua Hu
College of Computing and Informatics
Drexel University
Philadelphia 19104, PA, USA
xh29@drexel.edu

Zhengtao Yu
Kunming University of Science and Technology
Kunming, China
ztyu@bit.edu.cn

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 retrieval and question diversification. We first propose probabilistic retrieval models to locate candidate questions that are relevant to a review. We then design a set function to re-rank the questions with the goal of rewarding diversity in the 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 for individual reviews.

*The first two authors made equal contributions to this paper

SIGIR '16, July 17-21, 2016, Pisa, Italy

DOI: http://dx.doi.org/10.1145/2911451.2911544

CCS Concepts

•**Information systems** → **Summarization;** *Information retrieval diversity;*

Keywords

Review summarization; Question retrieval; Diversification

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 extraction of useful information has become an important way to improve user experience of online shopping.

Existing techniques for review analysis include review rating prediction [33, 17], sentiment polarity classification [13, 22], and aspect-based review summarization [11, 32, 28]. 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 user's 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 the 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 & 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. Its still worse than most cameras on the market, but its 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.

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. 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, ranged from hundreds to thousands of words, while questions are much shorter. Directly matching questions to a review may lead to unsatisfactory results. Second, 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 remove redundancy in the candidate question set, we propose a set function to rerank the retrieved questions with the goal of diversifying the questions. Particularly, the set function satisfies monotone submodularity such that it can be efficiently optimized by a greedy algorithm. 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 query priors and answers.
- 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 over the past decade. Different from standard text summarization [7], which aims to generate a concise summary for a single [31] or multi-document [8], review summarization aims to integrate users' opinions for a large collection of reviews with respect to a product [23, 36]. 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 [26, 14, 21]. 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

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 quantify the similarity between newly generated

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

user questions and curated questions so that corresponding answers can be used to answer those newly generated questions. Examples of work include Zhou et al. [39] who firstly proposed a context-aware model to address the lexical gap problem between questions; and Zhou et al. [40] 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 two aspects. First, newly generated user questions and historical questions are "parallel texts", while user reviews and questions are highly asymmetric on the information they convey. Second, newly generated questions that are used to retrieve similar questions are usually short (i.e., less than 20 words), while user reviews are longer (usually more than 100 words).

Our problem also relates to automatic question generation from text data. Zhao et al., [38] developed a method to automatically generate questions from short user queries in CQA. Chali et al., [4] developed a method to generate all possible questions in regards a topic. One limitation of these studies is that questions are generated based on template, so they might not be representative of real user questions. Our study is different, as we aim to select relevant questions that can be used to summarize user reviews from real-user question archives.

2.3 Text Retrieval with Verbose Queries

As our goal is to use long reviews to find short representative questions as summaries, our problem relates to the problem of information retrieval with verbose queries [10]. Due to term redundancy, query sparsity, and difficulty in identifying key concepts, verbose queries often result in null results. In tackling these challenges, recent studies have developed techniques to re-compose queries. Examples include query reduction [15, 12], query reformulation [5, 35], and query segmentation [1, 27]. However, the errors accumulated during the query transformation process cannot be corrected during the retrieval phase. In our study, we do not split long reviews into sentences or phrases, and use the text chunks to retrieve relevant questions. Instead, we utilize a two-stage framework: 1) use the entire review as a query to retrieve relevant questions; and 2) after retrieving a set of questions, 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.

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 $R^{(i)} = \{r_1^{(i)}, ..., r_{m_i}^{(i)}\}$ where m_i is the number of reviewers for product i. Each review can be represented by a bag of words. Meanwhile, we have a question database/corpus $Q = \{q^{(1)}, ..., q^{(n)}\}$ where the questions are crawled from Community Question Answering (CQA) sites. Given a review $r_j^{(i)}$ of product i, our task is to select a small subset of questions $S \subseteq Q$ to summarize the review.

Similar to other text summarization tasks [24], 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:

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

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 constraint to restrain the total number of questions in the set.

The set function $\mathcal{F}$ in Eqn.(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, Eqn.(1) would be an NP-hard problem. Fortunately, if $\mathcal{F}$ satisfies non-decreasing submodular [6], the optimization problem can be solved by efficient greedy algorithms with a close approximation. We introduce the background on submodular functions in Section 3.2.

It is worth noting that we do not solve Eqn.(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 techniques, e.g., obtaining the top 100 questions based on their relevance to a given review. We will introduce the question retrieval models in Section 4.2. Given these questions, we then apply Eqn.(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 Eqn.(1) in Section 4.3.

3.2 Submodular Functions

Submodular functions are discrete functions that model laws of diminishing returns [30]. They have been used in a wide range of applications such as sensor networks [16], information diffusion [9], and recommender systems [29]. Recently, it has been explored in multi-document summarization [19, 20]. 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* [30].

In Section 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 [25] for the optimization problem.

4. METHODS

4.1 Overview

In order 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?"*. In addition, 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 dual goal of relevancy and diversity, we propose a two-stage framework to find a set of questions that can be used to summarize a review. 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 the 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. After obtaining the top ranked relevant 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 of selected questions to the review as well as the diversity of the questions.

In the following sections, we present the query likelihood language models to generate a candidate question list (Section 4.2) and introduce our set function to re-rank candidate questions (Section 4.3) with an efficient greedy algorithm (Section 4.4) for optimization.

4.2 Question Retrieval

4.2.1 Query Likelihood Language Model

To retrieve candidate questions that are relevant to a given review, we employ query likelihood language model [2]. 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 log-likelihood of the conditional probability $P(q|r)$ of the question given the review:

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

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 Eqn.(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 [37] 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 the smoothing parameter and C denotes the whole question corpus. The MLE estimates for $P_{ml}(w|q)$ and $P_{ml}(w|Q)$ 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 Eqn.(3) denotes the prior probability of the question q regardless of review. 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 Eqn.(4) and Eqn.(8) into Eqn.(3), we can obtain the relevance scores for all questions in the question corpus.

4.2.2 Incorporating Answers

Since questions and reviews are not "parallel texts", there exists vocabulary gap between the two corpus. As shown in the real-world example in Section 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 Eqn.(5) [34]. After including all the answers a of question q, the

relevance score becomes:

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

Based on the Bayes' theorem, we have:

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

The above derivation is based on the following reasoning. Similar to Eqn.(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 $P\big(r|(q,a)\big)$:

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

where C' denotes the whole question and answe corpus, and $P_{ml}\big(w|C'\big)$ is the collection language model which is estimated based on Eqn.(7). λ is a smoothing parameter. $P_{mx}\big(w|(q,a)\big)$ 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 cofficient.

The prior probability $P(q)$ can be calculated in the same way as in Eqn.(8). By plugging $P\big(r|(q,a)\big)$ and $P(q)$ in Eqn.(10), we can obtain the relevance scores in Eqn.(9). The top-k questions are then retrieved as candidates and to be re-ranked by promoting diversity among them.

4.3 Question Diversification

Similar to other 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. In other words, we aim to achieve a dual goal in the final question set: relevancy and diversity. Mathematically, we formulate our objective function as a combinatorial optimization problem by following Eqn.(1) as follows:

$$\begin{aligned} \arg\max_{S\subseteq V} \mathcal{F}(S) &= \mathcal{L}(S) + \eta\mathcal{R}(S) \\ s.t. \sum_{q\in S} \text{length}(q) &\le b \end{aligned} \tag{13}$$

where V is the candidate question set obtained by the question retrieval component. $\mathcal{L}(S)$ measures the relevance of the final question set S with respect to the review. $\mathcal{R}(S)$ measures the diversity of the final question set. η is a constant for diversity regularization. The constraint $\sum_{q\in S}$ length$(q) \le b$ 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,

$$\mathcal{L}(S) = \log\big(\sum_{q\in S} \text{score}(q) - c\big) \tag{14}$$

where score(q) is the relevance score of question q. It can be calculated based on the query likelihood language models without (Eqn.(2)) or with (Eqn.(9)) incorporating answers (for convenience of presentation, we omit argument r and a). $c = \min_{q\in V}(\text{score}(q))$ 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,

$$\mathcal{R}(S) = \sum_{i=1}^{T} \log\big(\epsilon + \sum_{q\in P_i\cap S} r_q\big), \tag{15}$$

where $P_i, i = 1,...,T$ 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 [20]. 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. The logarithm function is non-decreasing concave function. The functions inside each logarithm function are non-negative modular functions (see Definition 2), so they are monotone (see Definition 3). Applying the logarithm function, which is a concave function, to non-decreasing modular functions yield submodular functions (see Theorem 1). For $\mathcal{R}(S)$, the summation of submodular functions results in a submodular function as well. Hence, the set function $\mathcal{F}(S)$ satisfies monotonicity and submodularity. □

4.4 Greedy Algorithm

The submodular optimization problem in Eqn.(13) is still NP-hard, but Nemhauser et al. [25] 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 Eqn.(13). 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

Algorithm 1: The Greedy Algorithm

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

of the questions in the candidate set V satisfy the length threshold constraint $l + \text{length}(q) < b$.

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 meaningless questions whose lengths are shorter than 3 words. 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 many 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 2,000. 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. To ensure the generated questions are representative for real-user questions, 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 generate up to 10 questions. The maximum of total 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 100 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 both relevancy and diversity. We apply the same preprocessing steps (as we did for the QA data) to process the annotated review data. The averaged review length for camera dataset is 814.976 and the averaged review length for TV dataset is 582.932.

Table 1: Statistics of Question Data for Camera and TV Category

	Camera	TV
Number of Products	331	226
Number of Questions	8,781	12,926
Average Question Length	11.898	11.179
Vocabulary Size of Questions	1,196	1,318
Vocabulary Size of Answers	2,948	2,541
Vocabulary Size in Total	2,987	2,668

5.2 Retrieval/Summarization Systems

In order to evaluate the performance of our proposed approach, we implement the following six summarization systems based on the variant of our approach:
(1) Query Likelihood Model: The query generation probability is estimated based on question corpus (Eqn.(3)).
(2) Combined Query Likelihood Model: The query generation probability is estimated based on question and answer corpus (Eqn.(10)).
(3) Query Likelihood Model with Maximal Marginal Relevance (MMR): re-rank retrieved questions by query likelihood model (system (1)) using MMR [3], 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 [20].
(4) Combined Query Likelihood Model with Maximal Marginal Relevance: re-rank retrieved questions by combined query

[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

likelihood model (system (2)) using MMR.
(5) Query Likelihood Model with Submodular Function: re-rank retrieved questions by query likelihood model (system (1)) using submodular function (Eqn.(13)).
(6) Combined Query Likelihood Model with Submodular Function: re-rank retrieved questions by combined query likelihood model (system (2)) using submodular function.

For system (1) and (2), we choose the Jelinek-Mercer smoothing parameter λ between 0.1 and 0.3 (Eqn.(5)). For system (3) and (4), we choose the trade-off parameter σ between 0.1 and 1.0. For system (5) and (6), we set the number of questions in the candidate set V (Eqn.(13)) as 100, the length threshold b as 50, 75, and 100, and the number of clusters (Eqn.(15)) as 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 (Eqn.(12)) and diversity regularizer η (Eqn.(13)), which will be shown in Section 6.3.

5.3 Evaluation Metrics

We follow the evaluation of conventional summarization systems to measure the performance of the aforementioned six systems for finding questions to summarize a product review. Specifically, we rely on ROUGE [18] (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. In our experiment, we compare unigram and bigram-based ROUGE scores.

One limitation of ROUGE score is that it assumes all words play equally important roles in a document. However, the words related to product aspects such as "image" or "screen" are more important than stopwords such as "does" or "it", which are frequently occurred in questions. Therefore, we also use TFIDF cosine similarity, which rewards important words by inverse document frequency. The definition of cosine similarity function can be found in [20].

6. RESULTS

6.1 Qualitative Analysis

We first show the feasibility of our method to retrieve non-redundant questions that can be used to summarize a review. We take one review[8] 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:

> ...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 2 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 3 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 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 4 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 differ-

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

Table 2: Human Annotation

(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 3: Questions Retrieved by Query Likelihood Model

(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 α7s?
(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?

Table 4: Questions Reranked by Submodular Function

(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??

ent 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, due to that 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 reranked 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:

> ...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 reranking 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 Section 5.1) achieved by different summarization systems (introduced in Section 5.2) are shown in Table 5 and 6. We set the total length threshold 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, and ‡ indicates the corresponding method outperforms all the other methods significantly.

On the TV dataset, the combined query likelihood language model ($\text{QL}(Q, A)$) yields better results than simple query likelihood language model ($\text{QL}(Q)$) does in terms of all evaluation metrics for different length threshold settings. Using MMR to rerank questions achieves competitive results against $\text{QL}(Q, A)$ and $\text{QL}(Q)$ do. Using the submodular function to re-rank the questions retrieved by simple and combined query likelihood language model (denoted as $\text{QL}(Q)$ +sub and $\text{QL}(Q, A)$ + sub, respectively) show better results over corresponding retrieval models for all evaluation metrics. $\text{QL}(Q, A)$ + sub achieve significant better results than all the other systems do at 0.01 level for all evaluation metrics, except for bigram-ROUGE precision score when $b = 50$ and TFIDF cosine similarity score when $b = 100$.

On the camera dataset, unfortunately, incorporating answer corpus in the query likelihood language model does not bring improvement on the ROUGE and TFIDF cosine similarity scores. One possible reason is that the vocabulary size of answer collections for the camera category is bigger 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 $\text{QL}(Q, A)$ contain more noises than that by $\text{QL}(Q)$. After promoting diversity in the retrieved question set using MMR, $\text{QL}(Q)$ + MMR is able to achieve slightly higher or competitive results against $\text{QL}(Q)$ except for bigram ROUGE scores when $b = 100$; but $\text{QL}(Q, A)$ + MMR yields slightly inferior results against $\text{QL}(Q, A)$.

Even though the combined retrieval model does not help increase the ROUGE and TFIDF cosine similarity scores, $\text{QL}(Q, A)$ + sub yields the highest unigram-ROUGE scores, in which the precision and F_1 scores are significantly higher than that by $\text{QL}(Q)$ ($p < 0.01$). $\text{QL}(Q)$ + sub achieves the best TFIDF cosine similarity scores without significant difference with that by $\text{QL}(Q)$. The results on bigram-ROUGE scores are mixed. The highest bigram-ROUGE scores achieved by either $\text{QL}(Q)$ or $\text{QL}(Q, A)$ are significantly better than the score achieved by simple query likelihood at level 0.01, except the bigram-ROUGE recall score and F_1 score when b is set to 100.

In summary, query likelihood model incorporating answers is able to yield better summarization performance when the vocabulary size of the answer collection is moderate. The results achieved by query 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 both ROUGE and TFIDF cosine similarity metrics. This model also shows the potential ability to correct the order of a question list by promoting diversified results on the camera dataset.

6.3 Parameter Analysis

In order to examine the impact of the smoothing parameter α of the answer collection (Eqn.(12)) and diversity regularizer η for the sumbodular function (Eqn.(13)), we examine the summarization performance achieved by system (2) and (6) (introduced in Section 5.2) with different settings of α and η on the TV and camera datasets. Figure 1 shows the unigram ROUGE F_1 scores achieved by different α between 0 and 1 with an interval of 0.1 when the length threshold is set to 50. The ROUGE curves achieved with other threshold settings follow similar patterns so we leave them out. 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 between 0 and 0.3, the unigram ROUGE F_1 scores increase with the benefit of the integration of the answer collection. After that, the scores decrease when α is getting larger, mean-

Table 5: Summarization Results on TV Dataset

L	Method	COSINE	ROUGE1-R	ROUGE1-P	ROUGE1-F_1	ROUGE2-R	ROUGE2-P	ROUGE2-F_1
50	QL(Q)	0.191	0.248	0.177	0.192	0.0440	0.0281	0.0313
	QL(Q, A)	0.211	0.267	0.190	0.205	0.0447	0.0303	0.0329
	QL + MMR	0.191	0.250	0.181	0.195	0.0443	0.0290	0.0320
	QL(Q, A) + MMR	0.207	0.263	0.189	0.204	0.0414	0.0292	0.0312
	QL(Q) + sub	0.219	0.268	0.190	0.206	0.0440	0.0302	0.0330
	QL(Q, A) + sub	**0.241**‡	**0.288**‡	**0.209**‡	**0.225**†	**0.0601**‡	**0.0409**	**0.0446**‡
75	QL(Q)	0.190	0.324	0.157	0.199	0.0590	0.0261	0.0335
	QL(Q, A)	0.199	0.334	0.161	0.203	0.0605	0.0273	0.0347
	QL(Q) + MMR	0.188	0.326	0.158	0.200	0.0580	0.0260	0.0330
	QL(Q, A) + MMR	0.199	0.336	0.162	0.205	0.0630	0.0290	0.0370
	QL(Q) + sub	0.208	0.332	0.161	0.203	0.0612	0.0274	0.0352
	QL(Q, A) + sub	**0.222**‡	**0.353**‡	**0.175**‡	**0.220**‡	**0.0797**‡	**0.0361**‡	**0.0462**‡
100	QL(Q)	0.179	0.372	0.137	0.190	0.0696	0.0237	0.0333
	QL(Q, A)	0.190	0.380	0.140	0.194	0.0746	0.0255	0.0355
	QL(Q) + MMR	0.177	0.376	0.139	0.192	0.0700	0.0240	0.0340
	QL(Q, A) + MMR	0.191	0.386	0.142	0.196	0.0800	0.0270	0.0370
	QL(Q) + sub	0.200	0.382	0.140	0.194	0.0757	0.0254	0.0357
	QL(Q, A) + sub	**0.206**†	**0.401**‡	**0.150**‡	**0.207**‡	**0.0921**‡	**0.0315**‡	**0.0441**‡

Table 6: Summarization Results on Camera Dataset

L	Method	COSINE	ROUGE1-R	ROUGE1-P	ROUGE1-F_1	ROUGE2-R	ROUGE2-P	ROUGE2-F_1
50	QL(Q)	0.111	0.218	0.260	0.227	0.0463	0.0520	0.0467
	QL(Q, A)	0.091	0.211	0.258	0.223	0.0406	0.0497	0.0427
	QL(Q) + MMR	0.111	0.218	0.263	0.229	0.0469	0.0531	0.0474
	QL(Q, A) + MMR	0.090	0.210	0.259	0.223	0.0401	0.0491	0.0422
	QL(Q) + sub	**0.112**	0.223	0.231	0.236	**0.0484**†	0.0585	0.0507
	QL(Q, A) + sub	0.093	**0.225**	**0.275**†	**0.238**†	0.0477	**0.0605**†	**0.0511**†
75	QL(Q)	0.109	0.286	0.231	0.245	0.0626	0.0474	0.0516
	QL(Q, A)	0.090	0.277	0.228	0.240	0.0530	0.0433	0.0455
	QL(Q) + MMR	0.110	0.288	0.234	0.248	0.0634	0.0482	0.0523
	QL(Q, A) + MMR	0.090	0.277	0.229	0.241	0.0528	0.0435	0.0456
	QL(Q) + sub	**0.116**	0.295	0.241	0.254	**0.0648**†	**0.0511**†	**0.0546**†
	QL(Q, A) + sub	0.102	**0.297**	**0.242**†	**0.256**†	0.0617	0.0509	0.0532
100	QL(Q)	0.110	0.342	0.209	0.249	0.0785	0.0447	0.0545
	QL(Q, A)	0.094	0.333	0.207	0.246	0.0661	0.0410	0.0484
	QL(Q) + MMR	0.110	0.344	0.211	0.251	0.0773	0.0445	0.0541
	QL(Q, A) + MMR	0.094	0.331	0.207	0.245	0.0656	0.0406	0.0481
	QL(Q) + sub	**0.116**	0.350	0.216	0.257	**0.0786**	0.0467	**0.0562**
	QL(Q, A) + sub	0.106	**0.352**	**0.217**†	**0.258**†	0.0759	**0.0474**†	0.0558

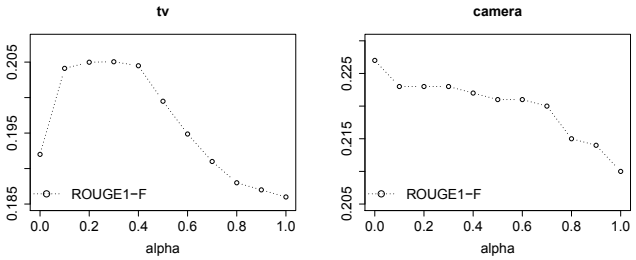

Figure 1: ROUGE-1 F_1 Scores on TV and Camera Datasets with different Weights of Answer Collection when $b = 50$

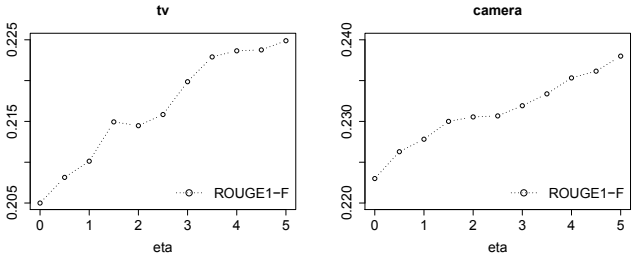

Figure 2: ROUGE-1 F_1 Scores on TV and Camera Datasets with Different Diversity Regularizer when $b = 50$

ing that imposing too much weights on the estimates from the answer collection is harmful to the performance of retrieval model. 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 getting smaller.

Figure 2 shows the impact of diversity regularizer η on the combined query likelihood language model. With the increasing η values, the unigram ROUGE F_1 scores increase on both datasets. These figures 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.

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. 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 retrieval and question diversification. 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 experiments 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.

8. ACKNOWLEDGMENTS

The authors would like to thank Miao Jiang for data collection, anonymous reviewers for valuable comments, and TCL Research America for generous funding support.

9. REFERENCES

[1] M. Bendersky, W. B. Croft, and D. A. Smith. Joint annotation of search queries. In *ACL*, pages 102–111, 2011.

[2] A. Berger and J. Lafferty. Information retrieval as statistical translation. In *SIGIR*, pages 222–229, 1999.

[3] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, pages 335–336, 1998.

[4] Y. Chali and S. A. Hasan. Towards automatic topical question generation. In *COLING*, pages 475–492, 2012.

[5] V. Dang and B. W. Croft. Query reformulation using anchor text. In *WSDM*, pages 41–50, 2010.

[6] S. Fujishige. *Submodular functions and optimization*, volume 58. Elsevier, 2005.

[7] J. Goldstein, M. Kantrowitz, V. Mittal, and J. Carbonell. Summarizing text documents: sentence selection and evaluation metrics. In *SIGIR*, pages 121–128, 1999.

[8] J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitz. Multi-document summarization by sentence extraction. In *NAACL-ANLP Workshop on Automatic Summarization*, pages 40–48, 2000.

[9] M. Gomez Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. In *SIGKDD*, pages 1019–1028, 2010.

[10] M. Gupta and M. Bendersky. Information retrieval with verbose queries. In *SIGIR*, pages 1121–1124, 2015.

[11] M. Hu and B. Liu. Mining opinion features in customer reviews. In *AAAI*, volume 4, pages 755–760, 2004.

[12] S. Huston and W. B. Croft. Evaluating verbose query processing techniques. In *SIGIR*, pages 291–298, 2010.

[13] Y. Jo and A. H. Oh. Aspect and sentiment unification model for online review analysis. In *WSDM*, pages 815–824, 2011.

[14] H. D. Kim, K. Ganesan, P. Sondhi, and C. Zhai. Comprehensive review of opinion summarization. *UIUC Technical Report*, 2011.

[15] G. Kumaran and V. R. Carvalho. Reducing long queries using query quality predictors. In *SIGIR*, pages 564–571, 2009.

[16] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *SIGKDD*, pages 420–429, 2007.

[17] F. Li, N. Liu, H. Jin, K. Zhao, Q. Yang, and X. Zhu. Incorporating reviewer and product information for review rating prediction. In *IJCAI*, volume 11, pages 1820–1825, 2011.

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

[19] H. Lin and J. Bilmes. Multi-document summarization via budgeted maximization of submodular functions. In *NAACL*, pages 912–920, 2010.

[20] H. Lin and J. Bilmes. A class of submodular functions for document summarization. In *ACL*, pages 510–520, 2011.

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

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

[23] D. K. Ly, K. Sugiyama, Z. Lin, and M.-Y. Kan. Product review summarization from a deeper perspective. In *ACM/IEEE joint conference on Digital ibraries*, pages 311–314, 2011.

[24] I. Mani and M. T. Maybury. *Advances in automatic text summarization*, volume 293. MIT Press, 1999.

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

[26] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

[27] N. Parikh, P. Sriram, and M. Al Hasan. On segmentation of ecommerce queries. In *CIKM*, pages 1137–1146, 2013.

[28] D. H. Park, H. D. Kim, C. Zhai, and L. Guo. Retrieval of relevant opinion sentences for new products. In *SIGIR*, pages 393–402, 2015.

[29] L. Qin and X. Zhu. Promoting diversity in recommendation by entropy regularizer. In *IJCAI*, pages 2698–2704. AAAI Press, 2013.

[30] R. W. Shephard and R. Färe. *The law of diminishing returns*. Springer, 1974.

[31] K. M. Svore, L. Vanderwende, and C. J. Burges. Enhancing single-document summarization by combining ranknet and third-party sources. In *EMNLP-CoNLL*, pages 448–457, 2007.

[32] I. Titov and R. T. McDonald. A joint model of text and aspect ratings for sentiment summarization. In *ACL*, volume 8, pages 308–316, 2008.

[33] H. Wang, Y. Lu, and C. Zhai. Latent aspect rating analysis on review text data: a rating regression approach. In *SIGKDD*, pages 783–792, 2010.

[34] X. Xue, J. Jeon, and W. B. Croft. Retrieval models for question and answer archives. In *SIGIR*, pages 475–482, 2008.

[35] X. Xue, Y. Tao, D. Jiang, and H. Li. Automatically mining question reformulation patterns from search log data. In *ACL*, pages 187–192, 2012.

[36] K. Yatani, M. Novati, A. Trusty, and K. N. Truong. Analysis of adjective-noun word pair extraction methods for online review summarization. In *IJCAI*, volume 22, page 2771, 2011.

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

[38] S. Zhao, H. Wang, C. Li, T. Liu, and Y. Guan. Automatically generating questions from queries for community-based question answering. In *IJCNLP*, pages 929–937, 2011.

[39] G. Zhou, L. Cai, J. Zhao, and K. Liu. Phrase-based translation model for question retrieval in community question answer archives. In *ACL*, pages 653–662, 2011.

[40] G. Zhou, T. He, J. Zhao, and P. Hu. Learning continuous word embedding with metadata for question retrieval in community question answering. In *ACL*, pages 250–259, 2015.