Question-based Text Summarization

A Thesis

Submitted to the Faculty

of

Drexel University

by

Mengwen Liu

in partial fulfillment of the

requirements for the degree

of

Doctor of Philosophy

December 2017

Dedications

This thesis is dedicated to my family

Acknowledgments

I would like to thank to my advisor, Prof. Xiaohua Tony Hu for his generous guidance and support in the past five and a half years. Prof. Hu provided a flexible environment for me to explore different research fields. He encouraged me to work with intellectual peers and attend professional activities, and gave immediate suggestions on my studies and career development.

I am also grateful to Prof. Yuan An, whom I worked closely in the first two years of my doctoral study. I thank to him for offering guidance about research, paper review, collaboration with professionals from other fields, and scientific writings. I would like to thank my dissertation committee of Prof. Weimao Ke, Prof. Erjia Yan, Prof. Bei Yu, and Prof. Yi Fang. Prof. Ke provided many suggestions towards my research projects that relate to information retrieval and my thesis topic. Prof. Yan offered me lots of valuable suggestions for my prospectus and thesis and encouraged me to conduct long-term research. Prof. Yu inspired my interest in natural language processing and text mining, and taught me how to do research from scratch with great patience. Prof. Fang was very supportive to my academic study and career. He gave me detailed suggestions for my research and supported me in many other professional activities including conference talks and job application.

I spend three years at TCL Research America where I learned a lot from industry peers. Dr. Haohong Wang provided me valuable suggestions about research and career development. Dr. Lifan Guo and Dr. Dae Hoon Park spent many hours listening to me talking about my research and helped me flesh out my ideas. I would also like to thank my fellow graduate students for all the help and support they provided: Yuan Ling, Yue Shang, Wanying Ding, and many others. I have also benefited a lot from discussions with Prof. Fang's group, which broadens my vision of research. A special thanks to Travis Ebesu for proofreading my work.

Last but not least, I own my deepest gratitude to my family. Without their love and support, this work would never be possible.

Table of Contents

List of Tables

List of Figures

Abstract

Question-based Text Summarization

Mengwen Liu

In the modern information age, finding the right information at the right time is an art (and a science). However, the abundance of information makes it difficult for people to digest it and make informed choices. In this thesis, we aim to help people who want to quickly capture the main idea of a piece of information before they read the details through text summarization. In contrast with existing works, which mainly utilize *declarative* sentences to summarize a text document, we aim to use a few *questions* as a summary. In this way, people would know what questions a given text document can address and thus they may further read it if they have similar questions in mind.

A question-based summary needs to satisfy three goals, *relevancy*, *answerability*, and *diversity*. Relevancy measures whether a few questions can cover the main points that discussed in a text document; answerability measures whether answers to the questions are included in the text document; and diversity measures whether there is redundant information carried by the questions.

To achieve the three goals, we design a two-stage approach which consists of question selection and question diversification. The question selection component aims to find a set of candidate questions that are relevant to a text document, which in turn can be treated as answers to the questions. Specifically, we explore two lines of approaches that have been developed for traditional text summarization tasks, *extractive* approaches and *abstractive* approaches to achieve the goals of relevancy and answerability, respectively. The question diversification component is designed to re-rank the questions with the goal of rewarding diversity in the final question-based summary.

Evaluation on product review summarization tasks for two product categories shows that the proposed approach is effective for discovering meaningful questions that are representative for individual reviews. This thesis opens up a new direction in the intersection of information retrieval and natural language processing. Despite the evaluation on the product review domain, the thesis provides a general solution for question selection for many interesting applications and discusses the possibility of extending the problem to other domain-specific question-based text summarization tasks.

Chapter 1: Introduction

1.1 Motivation

With the rapid growth of the world wide web, the scale and speed of information have been dramatically increasing. Unfortunately, the large amount of information has caused the problem of "information overload" [102]. It occurs when people have difficulties in finding the right information with limited information processing capacity. To tackle such an issue, various information technologies have been proposed such as search engines [80], recommender systems [2], question and answering systems [89], etc. These solutions aim to quickly provide users with relevant information based on their explicit or implicit needs. Another type of solutions aim to create a simplified representation of information to users, such as information extraction [5], and text summarization [70].

In this thesis, we focus on one type of techniques — automatic text summarization, which aims to help reduce the amount of information that people need to deal with. There is a large body of works that develop various techniques to facilitate automatic text summarization in different domains [70]. The output summaries usually consist of *declarative* sentences, although possibly are not limited to *declarative* sentences. Therefore, we explore the possibility of using another format of sentences — *interrogative* sentences, namely questions, to summarize text documents.

Question-based summaries have two advantages over declarative summaries when a reader reads a piece of text. First of all, people have questions all the time and thus they seek answers by any possible means. Ravi et al. [88] even claimed that when humans start to ask the right questions before they can be answered, they are making progress. Good readers usually have their own questions in mind before they start to search for answers (documents) and to decide which documents are worth reading. Therefore, questions are expected to be more attractive for people to read than declarative sentences are. Second, armed with questions as handy "hints" of a long text document, readers are expected to become interactive readers [22]. In the past 30 years, questions had been

used as motives to help students develop reading comprehension skills. Singer [97] argued that the main strategy for teaching students reading comprehension skills is to ask them questions, before, during, and after reading. Questions asked before reading are helpful to direct and focus students' thinking on the content of the text that will answer the questions. The value of proposed questions for reading comprehension is that they help students maintain a searching attitude during reading. Wood et al. [110] also stated that asking questions during reading is valuable in helping students integrate information, identify main ideas, and summarize information.

Despite the auxiliary power of questions in search and reading activities, few research explores using questions as summaries. In this study, we seek an approach to help readers quickly comprehend a text document through questions. Readers have certain questions in mind and want to search for documents to see if their questions can be answered from documents. Given a concise set of questions for a document, readers can quickly understand its main idea and may further read it only if they have similar questions in their minds. On the other hand, even if a reader does not have any questions in mind, question-based summaries might still be useful as "hints" to stimulate the reader to think about his/her own questions, and thus to be proactive during search and reading.

It is worth noting that we do not aim to replace conventional declarative sentence-based summaries with question-based summaries. Instead, our goal is to explore the possibility of using questions as summaries, which can be complimentary to traditional plain sentence-based summaries. We argue that the question-based summaries are used to reflect the main points discussed in text(s), but not necessarily include the content of them. According to the typology of text summarization systems proposed by Hovy and Lin [38], 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 text(s) without including its contents. After a reader reads an indicative summary, he/she can explain what the input text was about, but not necessarily what was in the input text. An informative summary, on the contrary, aims to reflect (some of) the content of a text document, and thus makes the reader describe (parts of) what was contained in it. In this way, a question-based summary can be regarded as an indicative summary.

1.2 Focus of Thesis

In this thesis[1], we take an initial step to explore the usage of questions as summaries with one type of text resources: product reviews. With the rapid growth of online review sites, more and more customers 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 [109, 56], sentiment polarity classification [45, 61], and aspect-based review summarization [40, 106, 83]. The first two techniques aim to predict numerical ratings and sentiment orientations of reviews. They do not summarize the main points that are discussed in reviews. Review summarization is beneficial for aggregating user opinions towards each aspect of a product (e.g., the autofocus feature of a camera) 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 this thesis, we seek an approach to help customers quickly comprehend a product review through questions. In other words, we aim to find a concise set of questions that are addressed by a given product review as well as cover the main points of it. Many customers 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 sounds impractical at the first sight. A text document usually consists of many declarative sentences instead of questions, thus it is impossible to find useful questions from a text document to represent its content. Thanks to the emergence of Community

[1] This thesis is based on the work described in "Retrieving non-redundant questions to summarize a product review" which appeared in the proceedings of SIGIR 2016 [65] and "Product review summarization through question retrieval and diversification" which appeared in the Journal of Information Retrieval 2017 [64].

Question Answering (CQA), large e-commerce websites now offer CQA services for their products. A notable example is Amazon's Customer Questions & Answers service[2]. In this thesis, we make use of such CQA databases from which we can retrieve real-world user questions to summarize individual reviews.

Take the following segment of a real-world review[3] 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[4] 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.

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

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

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

This task of summarizing a product review through questions is a challenging task. First of all, 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 be able to answer the questions. 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. In summary, the final question-based summary needs to satisfy three goals: *relevancy*, *answerability*, and *diversity*.

To tackle these challenges, we develop a two-stage framework to achieve the goal of finding a set of non-redundant questions to represent a product review. The first stage aims to select candidate questions from existing question databases that are relevant to a review. Meanwhile, the review is expected to be used as answers to questions. We explore two lines of approaches, *extractive* approaches and *abstractive* approaches, which have been extensively used for traditional text summarization tasks. Extractive approaches aim to leverage existing resources (e.g., the original text document) such that the sentences of the summary are coming from those resources. On the contrary, abstractive approaches aim to create a summary without necessarily using the same sentences from the resources. In this way, producing an abstractive summary is similar to how humans create a summary.

Specifically, our extractive approach employs a probabilistic retrieval model to retrieve candidate questions from an existing question and answering database 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, an 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.

The proposed framework and approaches are evaluated in the domain of product review summarization, which is different from existing works as they are focusing on extracting opinions sentences from product reviews. In addition to retrieve a set of questions from an existing question database as a summary for a product review, we also explore the possibility of using a trained RNN Encoder-Decoder to automatically generate a question as an abstractive summary for a text segment.

1.3 Contributions

We introduce a new task of summarizing a text document by questions. To the best of our knowledge, no prior work has been done, as the existing work on text summarization focuses on using declarative sentences as summaries. The question-based summary is expected to satisfy three goals: relevancy, answerability, and diversity. Relevancy measures whether questions cover the main points that are discussed in a text document; answerability measures whether questions could be answered by the text document; and diversity measures whether there are overlap in terms of information among the questions.

We propose a two-stage framework consisting of question selection and question diversification. We explore both extractive and abstractive approaches for question selection, which are common approaches for text summarization tasks. The extractive approach employs a probabilistic retrieval model to retrieve relevant questions from an existing question database. The abstractive approaches employs a sequence-to-sequence learning model to measure the answerability between a question

and a text document. 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 evaluate our approach in the domain of review summarization. We created and annotated 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.

Using questions to summarize text documents has many interesting domain-specific applications as texts could be coming from various resources, such as news, scientific articles, books, forums, etc. There is a large body of works that proposed techniques to tackle the summarization tasks in those domains (see Section 2.1 for related works); but as mentioned before, the summaries are merely restricted to *declarative* sentences. While we experiment with product reviews in this thesis, the proposed framework and its components are general and thus can be applicable to other domains.

In addition to text summarization, the problem of question selection, which corresponds to the first component of our proposed framework, is also an interesting research direction. Our proposed solution for the question selection could be applied to many interesting applications. Now that question is at the heart of reading comprehension, a straightforward application would be question generation in order to facilitate educational material creation such as reading comprehension and vocabulary assessment [34]. Using questions to summarize text documents can be seen as a problem of the transformation from a long text sequence to a short text sequence. Such a problem can be generalized to the problem of finding the mapping between two sequences of text. Examples of applications include machine translation [104, 3], automatic chatbot [95], automatic email responding machine [47], and rap lyrics generation [69]. Moreover, the two sequences are not restricted to text data. A well-known example of mapping multiple modalities is image caption generation [48, 75].

1.4 Outline

This thesis is structured as follows. Chapter 2 summarizes state-of-the-art literature for text summarization and conventional approaches, extractive approaches and abstractive approaches, question

retrieval and generation, and diversified text summaries. Chapter 3 provides the definition of our research questions and specifies the question-based review summarization problem. Chapter 4 presents the overall architecture of our solution and its components: question selection and question diversification. Chapter 5 presents the methods that address the two components. Chapter 6 presents experiments including data annotation, summarization systems, and evaluation metrics. Chapter 7 presents experimental results. Chapter 8 explores the possibility of automatically generating a question to summarize a piece of text. We present our conclusions in Chapter 9 and discuss future directions.

Chapter 2: Related Work

2.1 Text Summarization

The earliest investigation on automatic text summarization dates back to 1958 [67]. The goal of automatic text summarization is similar to the reason that humans need shorter representation of original text. Researchers have provided the definition of text summarization in different ways. For instance, Spärck Jones [46] defines a summary as "reductive transformation of source text to summary text through content reduction by selection and generalization on what is important in the source". Hovy and Lin [38] define a summary as "a text that is produced from one or more texts, that convey important information in the original text(s) and that is no longer than half of the original text(s) and usually significantly less than that".

Hovy and Lin [38] proposed a typology of summaries with respect to the characteristics of source text(s), the characteristics of generated summaries, and the purpose of summarization. Under the "usage" categorization of the typology, the authors proposed two kinds of summaries, indicative and informative summary. An indicative summary aims to provide the gist of the input text(s) without including its contents. An informative summary, on the contrary, aims to reflect (some of) the content of a text document, and thus makes the reader describe (parts of) what was contained in it. Back to our problem, questions can be seen as "indicative" summaries. After a reader reads the questions, he/she is expected to explain what the input text was about, but not necessarily what was contained in it.

Considering text documents in different domains have their own characteristics, various techniques have been developed to tackle domain-specific text summarization problems. Examples include news summarization [54, 7], email summarization [107, 73], blog summarization [41, 52], etc. In this thesis, we are focusing on using questions to summarize a product review. Different from standard text summarization [28], which aims to generate a concise summary for a single [105] or multi-document [29], review summarization aims to integrate users' opinions for a large collection

of reviews with respect to a product [68, 114]. 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 [81, 49, 60]. The difference between our study and existing works on review summarization 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.

Now that we aim to use a few questions to represent a text document, our problem can be treated as a text summarization problem. The difference lies in that the summary consists of questions instead of declarative sentences. To the best of our knowledge, no existing work attempts to use questions for text summarization. Considering the size of text sources, a summary could be generated for a single text document or multiple documents that are thematically related. In our work, we study how to automatically create a question-based summary for a single document and leave multi-document summarization for our future work.

2.2 Extractive Approaches for Text Summarization

There are two lines of approaches that have been developed to solve automatic text summarization problems. One type of approaches is extractive approaches, which aim to select a subset of sentences from original text documents. This process can be seen as identifying the most *central* sentences from original text documents. Centrality of a sentence is often defined in terms of the centrality of the words that it contains. A common way of assessing word centrality is to look at the centroid of the document cluster in a vector space. The centroid of a cluster is a pseudo-document which consists of words that have tf×idf scores above a predefined threshold, where tf is the frequency of a word in the cluster, and idf values are typically computed over a much larger and similar genre dataset. In centroid-based summarization[86, 87], the sentences that contain more words from the centroid of the cluster are considered as central. Later on, Erkan and Radev [23] proposed LexRank, which computes sentence importance based on the concept of eigenvector centrality in a graph representation of sentences. A connectivity matrix based on intra-sentence cosine similarity is used as the adjacency matrix of the graph representation of sentences.

In Chapter 5.1, our work explores the usage of an extractive approach. Specifically, we employ a probabilistic retrieval model, query likelihood language model to retrieve a set of questions from an existing question database that are relevant to a text document.

2.3 Abstractive Approaches for Text Summarization

A large body of work for traditional text summarization has been extractive, which aims to identify salient sentences or passages in the original document and reproduce them as a summary [44, 50, 23, 72, 16, 25, 24]. Humans, on the other hand, tend to paraphrase the original story in their own words. As such, human summaries are abstractive in nature and seldom consist of reproduction of original sentences from the document.

The task of abstractive text summarization was formalized in the DUC-2003 and DUC-2004 competitions [79]. The dataset consists of news stories from different topics with multiple human-generated summaries as references. The best performing system, TOPIARY [115], on the DUC-2004 task used a combination of linguistically motivated compression technique, and an unsupervised topic detection algorithm that appends keywords extracted from the article onto the compressed output. Examples of state-of-the-art abstractive text summarization techniques include traditional phrase-table based machine translation approaches [6], sentence compression using weighted tree-to-tree transformation rules [17], and quasi-synchronous grammar approaches [111].

Abstractive text summarization task is more challenging than extractive text summarization task is, as it requires advanced natural language generation techniques [50, 85]. It requires tailored linguistic patterns and domain knowledge to make sense the generated summaries, which is very time-consuming. Hence, there is still a large gap of linguistic and semantic quality between abstractive summary and human-generated summary.

Recently, the emergence of deep learning techniques and their applications on NLP tasks [18] facilitate the abstractive text summarization tasks. A few recent work utilize an encoder-decoder framework to solve this problem. Such a framework aims to map a pair of input text sequence to another output text sequence through an end-to-end system. The components of the encoder/decoder are deep neural networks. Rush et al. [92] designed a convolutional neural network (CNN) to encode

the source document, and a context-sensitive attentional feed-forward neural network to generate the summary. Similarly, Chopra et al. [14] used a convolutional neural network for the encoder, but a recurrent neural network (RNN) for the decoder. Later on, motivated by the success of applying the RNN Encoder-Decoder framework to machine translation task [13], researchers start to employ the same framework for abstractive text summarization. Lopyrev [66] utilize an attentional RNN Encoder-Decoder with LSTM units for news headline generation task. Hu et al. [39] utilize RNN Encoder-Decoder for Chinese short text summarization. Nallapati et al. [77] revise the attentional RNN Encoder-Decoder framework by 1) enriching the input of encoder with POS, NER, and TF-IDF values; 2) replacing unseen words in the output sequence generated by the decoder with words in the input sequence; and 3) capturing attentional alignment between input and output text sequence through both word and sentence level. The authors apply this framework on DUC corpus and achieve significant results over baselines.

In Chapter 5.2, we propose an abstractive approach, the state-of-the-art Recurrent Neural Network(RNN)-Encoder Decoder to enrich our extractive approach, the probabilistic retrieval models, to find relevant questions given a text document from which the answers are expected to be found. In Chapter 8, our work starts with the RNN-Encoder Decoder framework, and apply it to generate a question for an input text sequence.

2.4 Review Summarization

Automatic review summarization has been a hot research topic in recent decades. Different from standard text summarization [28], which aims to generate a concise summary for a single [105] or multi-document [29], review summarization aims to integrate users' opinions for a large collection of reviews with respect to a product [68, 114]. 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 [81, 49, 60]. 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.

2.5 Question Retrieval

As our goal is to use a text document to find short representative questions as summaries, our problem relates to the problem of information retrieval with verbose queries [31]. 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 [53, 42], query reformulation [19, 113], and query segmentation [8, 82].

Our goal of finding a set of representative questions to summarize a text document 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 [118] who proposed a context-aware model for addressing the lexical gap problem between questions; and [119] 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 text documents 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 documents.

In our study, we first use the entire document as a query to retrieve relevant questions. In order to incorporate the answerability measurement between a question and a document, we split a document into sections, and score each pair of question and document 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 text document.

2.6 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 [91]. Most AQG approaches focus on generating factual 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 [35] proposed rule-based question transformations from declarative sentences. The transformed questions were then ranked and selected by a logistic regression model. Zhao et al. [117] developed a method to automatically generate questions from short user queries in community question and answering (CQA). Chali et al. [12] developed a method to generate all possible questions in regards a topic. Liu et al. [63] 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 [51], speech recognition[36], and natural language processing tasks [9, 74, 100]. Among the various deep learning architectures, Recurrent Neural Network (RNN) Encoder-Decoder [13] is a representative one to learn the mapping between pair of data, one is an input sequence, and the other one is an output sequence. 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. Examples of sequences include but are not limited to text, image, voice; and the input/output sequences are not necessary the same type of modality. Such an architecture has been successfully applied to machine translation [104, 3], image caption generation [75], sentence summarization [92], etc. The advantage of using RNN Encoder-Decoder architecture over template-based AQG approaches is that 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 thesis, in addition to retrieval-based models, we will explore the usage of RNN Encoder-Decoder to measure whether a text document can be used to answer a set of questions.

2.7 Diversified Text Summarization Results

As our goal aims to find a set of non-redundant questions to summarize a text document, our problem relates to search result diversification [33, 99, 98]. The approaches can be categorized into implicit and explicit approaches. Implicit approaches assume similar documents cover similar aspects. [11] 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). [20] 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 [93], 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 design a monotone submodular objective function to determine a set of non-redundant questions. Different from the aforementioned approaches, the submodular objective function can be maximized approximately by efficient greedy algorithm that results in a constant-factor approximation of the optimal solution.

Chapter 3: Problem Statement

3.1 Research Questions

In this thesis, our task is to generate an indicative summary for a text document through questions. The document in turn is supposed to contain the answers to those questions. The main research question of this thesis is:

> *Given a text document, how to automatically find a few questions that can be used to summarize the document?*

We can treat this problem as an inverse Question Answering (QA) problem, as the questions are used as a summary, while the document is used as "answers" to the questions.

The questions need to satisfy three goals in order to be used as a summary to represent a text document.

- **Relevancy**: The questions are expected to reflect the main points that are discussed in a text document.
- **Answerability**: The questions are expected to be able to answered by the text document.
- **Diversity**: The questions are expected to be non-redundant in terms of the information they convey.

To solve this problem with the aforementioned goals, we discuss three sub-questions in the following sections.

3.1.1 Research Question 1: Where to find the questions?

Now that we expect to use questions as a summary to represent a text document, we need to first decide where the questions come from. Can questions be extracted from original text documents? Can they be selected from existing question corpus/databases? Can they be fully automatically generated?

One line of approaches for conventional text summarization is extractive approaches [32], which aim to extract useful sentences from original text(s) to generate a summary. Such an approach might not be useful for question-based summary generation from input text(s), as a text document might contain few interrogative sentences. Thanks to the emergence of Community Question Answering (CQA) services, such as Yahoo! Answers[1], Quora[2], and Stack Overflow[3], their CQA databases can be used as candidate question sets for summary generation. One good thing about CQA databases is that they are contributed by real-world users, so the questions are representative of real-world questions. The task of aligning questions from an existing question database with a text document is a challenging task. A text document is usually long, while questions are much shorter. Directly matching questions to a text document may lead ot unsatisfactory results. Second, the matched questions should have the "answerability" to the text document. Last but not least, the questions should cover as many aspects as possible so that they are not redundant.

Another line of approaches is abstractive approaches [27], which aim to produce a grammatical summary that is close to what a human might generate. Such a summary is not necessarily contain words or phrases from original text(s). Abstractive approaches are harder to develop and extend to broader domains than extractive approaches are, as they usually require advanced natural language generation techniques [21]. Recently, with the rapid growth of deep learning techniques, developing abstractive approaches has become a hot research topic. Recurrent Neural Network (RNN) Encoder-Decoder has been successfully applied to the task of sentence summarization [76], in which the summary is generated in a fully data-driven way. Employing a data-driven approach is challenging as it requires a lot of labeled training examples to obtain a decent model.

3.1.2 Research Question 2: How to measure the quality of a question-based summary?

Once we decide where to obtain questions, we need to design a way to measure the compatibility between a question-based summary and an input text document. Let Q be a finite question set, which could be a CQA database/corpus, or generated questions in a fully automatic way. Similar

[1] `https://answers.yahoo.com/`
[2] `https://www.quora.com/`
[3] `http://stackoverflow.com/`

to conventional text summarization tasks [70], the quality of selected/generated questions can be quantified by a utility function $\mathcal{F} : 2^Q \rightarrow \mathbb{R}$. In addition, the subset S should satisfy certain constraints. Formally, our task is to find the optimal question subset S^* defined as the following combinatorial optimization problem:

$$\begin{aligned} S^* &= \arg\max_{S \subseteq Q} \mathcal{F}(S) \\ s.t. &: \sum_{q \in S} c(q) \leq b, \end{aligned} \tag{3.1}$$

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 utility function $\mathcal{F}$ in Eqn.(3.1) measures the quality of the selected question subset S. As mentioned at the beginning of this Chapter, the question-based summary needs to fulfill three goals: relevancy, answerability, and diversity. The utility function $\mathcal{F}$ is expected to reflect these three measurement. The choice of $\mathcal{F}$ depends on the property of the questions that we desire. In general, Eqn.(3.1) would be an NP-hard problem. Fortunately, if $\mathcal{F}$ satisfies non-decreasing submodular [26], the optimization problem can be solved by efficient greedy algorithms with a close approximation. We introduce the background on submodular functions in Section 5.3.1.

3.1.3 Research Question 3: How to evaluate the effectiveness of a question-based summary?

Since using questions to summarize a text document is a novel task, there is no ground-truth dataset available for evaluation. Therefore, we need to follow the evaluation on conventional text summarization tasks to create gold standard summaries. We could request a human to read a text document and generate a corresponding summary using questions. We assume that a human would be able to effectively identify the most important parts in an article and thus generate a question-based summary. If the set of questions selected/generated by an automatic method has a high overlap with the human generated summary, the automatic method should be regarded as effective

[38].

In summary, it is our goal to: 1) decide whether we make use of existing CQA databases or generate summaries in a fully abstractive way; 2) design a utility function $\mathcal{F}$ to measure the quality of the question-based summary which is expected to be relevant, answerable, and non-redundant; and 3) create gold-standards for the evaluation of automated generated question-based summaries.

3.2 Focus of Thesis

In this thesis, we start with one type of text resources: product reviews, to explore the usage of questions as a summary. We focus on using 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. If we employ an extractive approach, 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. If we employ an abstractive approach, our task is to generate a set of questions S based learning the linguistic patterns from Q.

We choose to employ an existing CQA database from which we select candidate questions to summary a product review. In Chapter 8, we discuss the usage of such a CQA database to train a sequence-to-sequence learning model, which can be used to automatically generate new questions given a text segment.

To measure the quality of the final question set S, namely, relevancy, answerability, and diversity, we need to solve the optimization problem defined in Eqn.(3.1). It is worth noting that we do not solve Eqn.(3.1) 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 5.1. Given these questions, we then apply Eqn.(3.1) to select a few questions (e.g., 5) as the final results by considering both question coverage (including relevancy and answerability) and diversity. Thus, this module can be viewed as re-ranking for achieving diversified results. We present our formulation of Eqn.(3.1) in Section 5.3.

Chapter 4: Overall Framework

4.1 Overview

Our task to use a few questions to summarize a product review. Before a customer reads a lengthy review, he/she could quickly understand the main points that are discussed in the review through the questions. 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?"*. 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 consisting of question selection and question diversification to find a set of questions that can be used to summarize a review. Figure 4.1 shows the overall architecture. The question selection component is used to rank a list of questions from an existing community question answering (CQA) database based on the relevancy and answerability between a review and questions, while the question diversification component is used to promote the diversity of questions. This architecture is flexible that during the question selection stage, we can define any ranking functions that are used to select a subset of questions from a large question pool; and during the diversification stage, we can

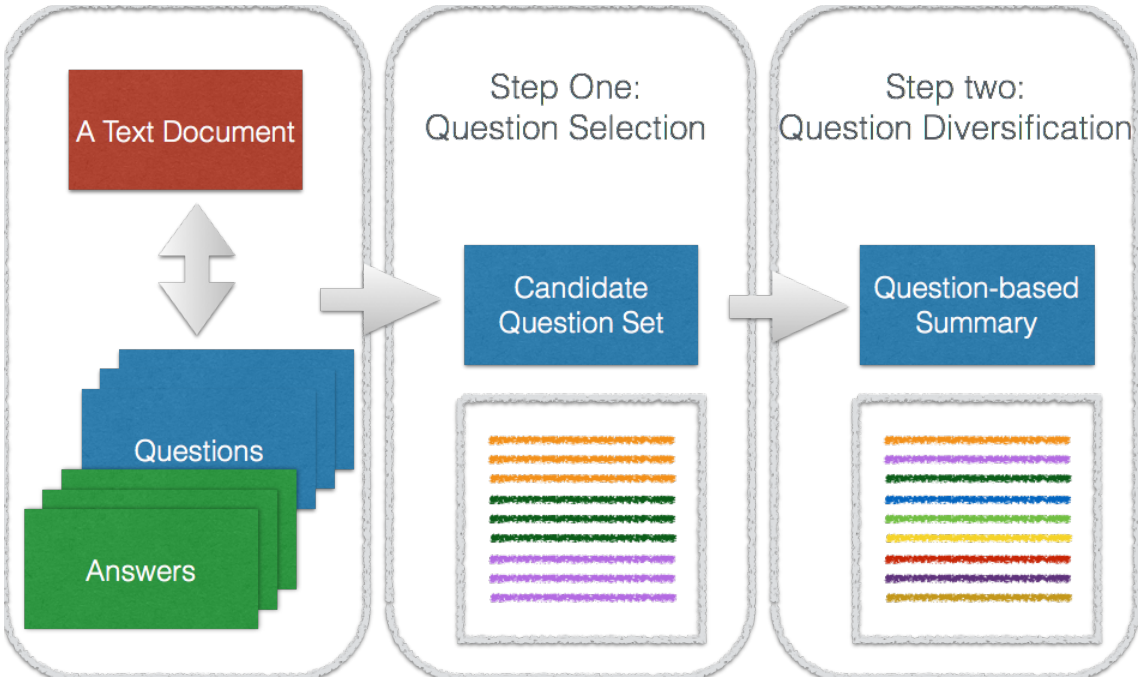

Figure 4.1 The overall architecture of question-based text summarization.

define any ranking functions that consider the trade-off between question coverage and diversity. In Chapter 7, we compare the performance of different summarization systems that are derived based on the variant of our framework. In the following sections, we explain the two components and how to achieve the three goals through the two stages.

4.2 Question Selection

For the first stage, we explore both extractive and abstractive approaches, which are common approaches for traditional text summarization tasks, to measure the relevancy and answerability of a question-based summary, respectively. For the extractive approach, we 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. In Section 5.1, we first present a probabilistic retrieval model and then explain how to take into consideration answers to mitigate the vocabulary gap between questions and a review.

The abstractive approach is employed to measure whether the review can be used as answers to questions. Specifically, we employ a sequence-to-sequence learning architecture, an 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. In Section 5.2, we first present how to enrich the relevancy score by the probabilistic retrieval models and then present a sequence-to-sequence model to compute the answerability score.

4.3 Question Diversification

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 relevancy and answerability of the questions and diversifying the questions. To select the final question set for review summarization, we need to design a utility function $\mathcal{F}$ and formulate an optimization problem defined in Eqn.(3.1). In general, Eqn.(3.1) would be an NP-hard problem. Fortunately, if $\mathcal{F}$ satisfies non-decreasing submodular [26], the optimization problem can be solved by efficient greedy algorithms with a close approximation. 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. In Section 5.3, we first present the background on submodular functions, and proposed our set functions that enjoy the properties of submodular functions and a greedy algorithm to optimize the set functions.

Chapter 5: Methods

In this Chapter, we present the approaches in the question selection and diversification components presented in Chapter 4. Recall that a question-based summary needs to satisfy three requirements: relevancy, answerability, and diversity. The question selection component is designed to measure the first two metrics, while the question diversification component is designed to measure the third metric. We present the methods that deal with each of the measurements in the following three sections.

5.1 Relevancy

5.1.1 Query Likelihood Language Model

To retrieve candidate questions that are relevant to a given review, we employ query likelihood language model [10]. 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:

$$\text{score}(r, q) = \log P(q|r) \tag{5.1}$$

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:

$$\begin{aligned} P(q|r) &= \frac{P(r|q)P(q)}{P(r)} \\ &\propto P(r|q)P(q) \end{aligned} \tag{5.2}$$

In Eqn.(5.2), $P(r)$ 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 $P(r|q)$:

$$P(r|q) = \prod_{w \in r} P(w|q) \tag{5.3}$$

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 [116] to avoid zero probabilities of unseen words in q:

$$P(w|q) = (1 - \lambda)P_{ml}(w|q) + \lambda P_{ml}(w|C) \tag{5.4}$$

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

$$P_{ml}(w|q) = \frac{\text{count}(w, q)}{|q|} \tag{5.5}$$

$$P_{ml}(w|C) = \frac{\text{count}(w, C)}{|C|} \tag{5.6}$$

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.(5.2) 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:

$$P(q) \propto \frac{1}{|q|} \tag{5.7}$$

$P(q)$ 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.(5.3) and Eqn.(5.7) into Eqn.(5.2), we can obtain the relevance scores for all questions in the question corpus.

5.1.2 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.(5.4) [112]. After including all the answers a of question q, the relevance score becomes:

$$\text{score}\big(r,(q,a)\big) = \log P\big((q,a)|r\big). \tag{5.8}$$

Based on the Bayes' theorem, we have:

$$\begin{aligned} P((q,a)|r) &= \frac{P\big(r|(q,a)\big)P(q,a)}{P(r)} \\ &\propto P\big(r|(q,a)\big)P(q,a) \\ &= P\big(r|(q,a)\big)P(a|q)P(q) \\ &\propto P\big(r|(q,a)\big)P(q) \end{aligned} \tag{5.9}$$

The above derivation is based on the following reasoning. Similar to Eqn.(5.2), $P(r)$ 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 $P\big(r|(q,a)\big)$:

$$\begin{aligned} P\big(r|(q,a)\big) &= \prod_{w\in r} P\big(w|(q,a)\big) \\ &= \prod_{w\in r} (1-\lambda)P_{mx}\big(w|(q,a)\big) + \lambda P_{ml}\big(w|C'\big) \end{aligned} \tag{5.10}$$

where C' denotes the whole question and answer corpus, and $P_{ml}\big(w|C'\big)$ is the collection language model which is estimated based on Eqn.(5.6). 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:

$$\begin{aligned} P_{mx}(w|(q,a)) &= (1-\alpha)P_{ml}(w|q) + \alpha P_{ml}(w|a) \\ &= (1-\alpha)\frac{\text{count}(w,q)}{|q|} + \alpha\frac{\text{count}(w,a)}{|a|} \end{aligned} \tag{5.11}$$

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

The prior probability $P(q)$ can be calculated in the same way as in Eqn.(5.7). By plugging $P\big(r|(q,a)\big)$ and $P(q)$ in Eqn.(5.9), we can obtain the relevance scores in Eqn.(5.8). The top-k questions are then retrieved as candidates and to be re-ranked by promoting diversity among them.

5.2 Answerability

5.2.1 Enrich Relevance Score with Answerability Score

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 answerable measurement between a review and questions. In tackling the issue, we enrich the ranking score between a review and a question defined in Eqn.(5.1) by integrating the answerability of the review to the question:

$$\text{score}(q,r) = (1-\gamma)\ \text{score}_r(q,r) + \gamma\ \text{score}_a(q,r) \tag{5.12}$$

where $\text{score}_r(q,r)$ denotes the relevance score between the question q and r, $\text{score}_a(q,r)$ 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 Eqn.(5.1); when $\gamma = 1$, the scoring function is fully dominated by the answerable 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 answerable measurement $\text{score}_a(q,r)$ in Eqn.(5.12) could

be calculated based on the summation of the answerability scores of each section of a review and a question:

$$\text{score}_a(q|r) = \sum_s P_a(q, s|r) \tag{5.13}$$

$$= \sum_s P_a(q|s)P(s|r) \tag{5.14}$$

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. We assume q is only dependent on a section of review s instead of the entire review r.

$P_a(q|s)$ in Eqn.(5.13) denotes the most likely question given a review section (answer).

$$P_a(q|s) = \arg\max_{q \in Q} P(q|s) \tag{5.15}$$

where Q denotes the set of all possible questions.

5.2.2 Recurrent Neural Network (RNN) Encoder-Decoder

To find the highest scoring question, we employ a sequence-to-sequence learning model, Recurrent Neural Network (RNN) Encoder-Decoder [104, 13], which can learn semantic relations between a pair of sequences of text data. The RNN Encoder-Decoder consists of two RNNs: one RNN encodes a review section $s = \{s_1, ..., s_{s_m}\}$ of s_m tokens into a fixed-length vector representation c which summarizes the information of the input sequence. Figure 5.1 depicts the architecture of sequence-to-sequence learning with Gated Recurrent Units (GRUs) by treating answer/review as input and question as output. Mathematically,

$$h_t = f(s_t, h_{t-1})$$

$$c = u(h_1, ..., h_{s_m})$$

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

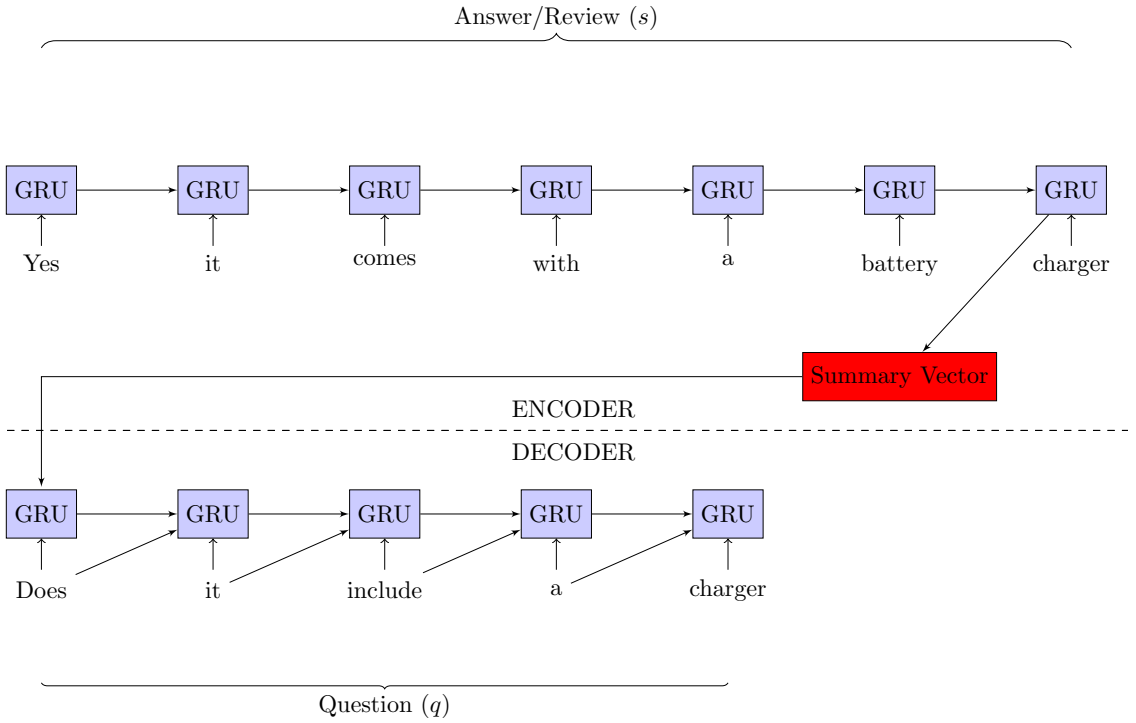

Figure 5.1 The architecture of the sequence-to-sequence learning model with Gated Recurrent Units (GRUs) for learning semantic relations between answer/review (input) and question (output).

as long short-term memory (LSTM) [37], bidirectional neural networks (BRNN) [94], and gated recurrent neural networks (GRU) [15]. u is a nonlinear function. For instance, [104] uses LSTM for f and $c = u(h_1, ..., h_{s_m}) = h_{s_m}$.

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

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

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

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

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 Eqn.(5.15). We will discuss the implementation details of the model in Section 6.2.

In addition to the incorporation of answerable 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 [3]. 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 (Eqn.(5.2)) 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 (Eqn.(5.9)). 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 time-consuming to pair all questions with each of the review sections.

Instead, we only pair each review section with the most relevant questions based on $\text{score}_r(q, r)$ in Eqn.(5.12). 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:

$$\begin{aligned} \text{score}_r(q, r) &= P(q|r) \\ &= \sum_s P(q, s|r) \\ &= \sum_s P(q|s)P(s|r) \\ &= \sum_s \frac{P(s|q)P(q)P(r|s)}{\sum'_q P(s|q')P(q')} \end{aligned} \tag{5.16}$$

where $P(q|s)$ could be calculated from Eqn.(5.2), and $P(q)$ can be calculated from Eqn.(5.7).

5.3 Diversification

To select the final question set for review summarization, we need to design a utility function $\mathcal{F}$ and formulate an optimization problem defined in Eqn.(3.1). In general, Eqn.(3.1) would be an NP-hard problem. Fortunately, if $\mathcal{F}$ satisfies non-decreasing submodular [26], the optimization problem can be solved by efficient greedy algorithms with a close approximation. We first present the background on submodular functions, and proposed our set functions that enjoy the properties of submodular functions.

5.3.1 Submodular Functions

Submodular functions are discrete functions that model laws of diminishing returns [96]. They have been used in a wide range of applications such as sensor networks [55], information diffusion [30], and recommender systems [84]. Recently, it has been explored in multi-document summarization [58, 59]. Following the notations introduced in the previous section, some basic definitions of submodular functions are given as follows.

Definition 1. A set function $\mathcal{F} : 2^Q \to \mathbb{R}$ 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 $\mathcal{F} : 2^Q \to \mathbb{R}$ 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 $\mathcal{F} : 2^Q \to \mathbb{R}$ 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 $\mathcal{F} : 2^Q \to \mathbb{R}$ is a submodular function, $g(S) = \phi(\mathcal{F}(S))$, where $\phi(\cdot)$ is a concave function, is also a submodular function* [96].

In Section 5.3.2, 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 [78] for the optimization problem.

5.3.2 Optimization Problem

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.(3.1) as follows:

$$\begin{aligned} \arg\max_{S \subseteq V} \mathcal{F}(S) &= \mathcal{L}(S) + \eta \mathcal{R}(S) \\ s.t. \sum_{q \in S} \text{length}(q) &\leq b \end{aligned} \tag{5.17}$$

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 $\sum_{q \in S} \text{length}(q) \leq b$ 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 utility 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,

$$\mathcal{L}(S) = \log \big(\sum_{q \in S} \text{score}(q) - c \big) \tag{5.18}$$

where $\text{score}(q)$ is the relevance score of question q. It can be calculated based on the query likelihood language models without (Eqn.(5.1)) or with (Eqn.(5.8)) 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 utility 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,

$$\mathcal{R}(S) = \sum_{i=1}^{T} \log\big(\epsilon + \sum_{q \in P_i \cap S} r_q\big), \tag{5.19}$$

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 [59]. 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 Eqn.(5.17) satisfies monotonicity and submodularity as it is the summation of two monotone submodular functions $\mathcal{L}(S)$ and $\mathcal{R}(S)$. □

5.3.3 A Greedy Algorithm

The submodular optimization problem in Eqn.(5.17) is still NP-hard, but Nemhauser et al. [78] 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.(5.17). 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 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 $q \in VnS$ **do**
    if $l + \text{length}(q) < b$ **then**
      $S_q \leftarrow S \cup \{q\}$
      $\mathcal{L}(S_q) \leftarrow \log\left(\sum_{q \in S_q} \text{score}(q) - c\right)$
      $\mathcal{R}(S_q) \leftarrow \sum_{t=1}^{T} \log\left(\epsilon + \sum_{q \in P_t \cap S_q} \frac{1}{|V|} \sum_{v \in V} w_{qv}\right)$
      $\mathcal{F}(S_q) \leftarrow \mathcal{L}(S_q) + \eta \mathcal{R}(S_q)$
      $A \leftarrow A \cup \{q\}$
    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

Chapter 6: Experiments

6.1 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)[1].

We choose to focus on products from Amazon[2], 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[3] 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 6.1 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 $2,000$ words. After obtaining the 500 reviews for the two product categories, we follow the guidelines for summary generation of NIST DUC[4]. 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

[1] http://duc.nist.gov/duc2004/
[2] http://www.amazon.com/
[3] http://www.nltk.org/
[4] http://duc.nist.gov/duc2004/t1.2.summarization.instructions

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 Section 7.1 demonstrate such an issue. 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.

In Section 5.2, 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 6.2.

6.2 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:

Table 6.1 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

Table 6.2 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

- (1) Query Likelihood Model: The query generation probability is estimated based on question corpus (Eqn.(5.2)).
- (2) Combined Query Likelihood Model: The query generation probability is estimated based on question and answer corpus (Eqn.(5.9)).
- (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.(5.12)).
- (4) Query Likelihood Model with Maximal Marginal Relevance (MMR): re-rank retrieved questions by query likelihood model (system (1)) using MMR [11], 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 [59].
- (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 (Eqn.(5.17)).
- (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.

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 (Eqn.(5.4)). 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 (Eqn.(5.17)) is set to 100, and the number of clusters (Eqn.(5.19)) 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 (Eqn.(5.11)) and diversity regularizer η (Eqn.(5.17)), which will be shown in Section 7.3.

For system (3) and system (8), we employ an attentional Recurrent Neural Network (RNN) Encoder-Decoder [3] 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[5] [71] 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 [1]. We set the

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

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 Eqn.(5.12) between 0.1 and 1.0.

6.3 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-n[6] (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 [57]. In our experiment, we compare unigram and bigram-based ROUGE scores, which are denoted as ROUGE-1 and ROUGE-2. The ROUGE-n scores are defined as follows:

$$\text{ROUGE-n precision} = \frac{\text{number\_of\_overlapping\_words}}{\text{number\_of\_words\_in\_system\_summary}} \tag{6.1}$$

$$\text{ROUGE-n recall} = \frac{\text{number\_of\_overlapping\_words}}{\text{number\_of\_words\_in\_system\_summary}} \tag{6.2}$$

$$\text{ROUGE-n } F_1\text{-score} = \frac{2 * \text{ROUGE-n precision} * \text{ROUGE-n recall}}{\text{ROUGE-n precision} + \text{ROUGE-n recall}} \tag{6.3}$$

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. We leaven other types of evaluations for future work.

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

Chapter 7: Results

7.1 Qualitative Analysis

7.1.1 The Impact of Relevancy

We first show the feasibility of using query likelihood model to retrieve relevant questions and whether incorporating answers would help bring up more meaningful questions. We select a text segment from a product review from Amazon:

> I didn't want anything complicated, and I mostly use this on the auto setting. It takes beautiful photos, particularly in low light. I rarely need the flash.

These sentences discuss low light, flash, image quality, and setting of a camera. Table 7.1 show the questions retrieved by query likelihood language model without incorporating answers to build the langage model. The first, fourth, and seventh questions are relevant to the review segment, as they address the low light, setting features of a camera. After incorporating answers, the ranked list is shown in Table 7.2. The top three questions and the seventh question are relevant to the review segment. The third question does not exist in the previous ranked list that retrieved by a simple query likelihood model. The second and fifth questions are promoted after incorporating answers to the simple query likelihood model.

However, incorporating answers do not always bring good results. Here is another review segment talking about image quality:

> Perfect camera, it was exactly what I wanted, a very good photo, with lots of effects, the convenience to carry everywhere because it is small, has wifi, that I just loved, and to finish its LCD turns 180 degrees in order to take selfies. Even though I have bought it used, the camera seems new.

Table 7.3 shows questions selected by simple query likelihood model. Only the fourth question is relevant to the review segment. The questions are about video quality. The review segments contains

Table 7.1 Case 1: Questions retrieved by Query Likelihood Language Model without incorporating answers

(1) Don't have time to learn photography, but want higher quality photos than compact cameras in low light?
(2) Could someone let me know what came with this purchase? It doesn't say anything in the description about what comes with the camera.
(3) What are the compatible regular Remote Controls for it? (if i dont want to use the app/nfc)
(4) I just got this Camera, and cant figure out whats the best setting to keep it on for everyday photos. any suggestions ?
(5) Can anyone tell me more about how the Wifi connectivity works on the camer? Do you need a phone with NFC to use the wifi option?
(6) Sometimes the image in the screen flips upside down. What is this, a setting? If so I can't find it. Or is this a flaw in the camera? Anyone else?
(7) is there a manual setting?
(8) Does this camera have the ability to auto focus itself while in video?
(9) Can I change language setting?

Table 7.2 Case 1: Questions retrieved by Query Likelihood Language Model with answers

(1) Don't have time to learn photography, but want higher quality photos than compact cameras in low light?
(2) I just got this Camera, and cant figure out whats the best setting to keep it on for everyday photos. any suggestions ?
(3) Does this camera have built-in flash?
(4) The flash rating of A5000 is 4m at ISO100, down from 6m (Nex 3N). Does anyone experience weak fill-in flash in the day? Thanks
(5) is there a manual setting?
(6) Can anyone tell me more about how the Wifi connectivity works on the camera? Do you need a phone with NFC to use the wifi option?
(7) Does this camera have image stabilization?

a word "LCD" and questions that contain the word are retrieved. Table 7.4 shows the ranked results after incorporating answers. The query likelihood model does not retrieve any relevant questions. Incorporating answers might also bring noises. Such a finding is consistent with the quantitative analysis in the following Section 7.2.

7.1.2 The Impact of Answerability

In this section, we explain 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 7.5 to 7.7 show the questions annotated by human, retrieved by simple query likelihood language model, and selected by query likelihood language model incorporating answerability measurement

Table 7.3 Case 2: Questions retrieved by Query Likelihood Language Model without answers

(1) Is this a good camera for taking videos? (2) I know this camera isn't available but I wonder if the LCD screen is touch?? please
(3) I bought this camera six months ago. The image in the viewing screen flips upside down for no
particular reason. What is wrong with it? Very unhappy.
(4) Does this camera have the ability to automatically upload photos to a PC over a WiFi connection as the pictures are being taken?
(5) Does the camera make a noise when you take a picture? (The NEX-3NL/B has a c̈licks̈ound so loud i couldnt use it during choir concerts
(6) Does this camera has bluetooth ?
(7) what is the optical zoom of this camera?
(8) Is there a pop-up flash on this camera? Where?
(9) Does this camera have built-in flash?
(10) does the camera come with the lens?

Table 7.4 Case 2: Questions retrieved by Query Likelihood Language Model with answers

(1) How is the video quality on this camera?
(2) Is this a good camera for taking videos?
(3) I bought this camera six months ago. The image in the viewing screen flips upside down for no particular reason. What is wrong with it? Very unhappy.
(4) this or the canon eos M?
(5) Does this camera has bluetooth ?
(6) The nex-5t and this camera are priced similarly. which is overall the better camera? please help (7) How long does the battery take to charge?
(8) What are the warranty terms for a manufacturer refurbished camera?
(9) I know this camera isn't available but I wonder if the LCD screen is touch?? please
(10) This camera doesn't allow panoramic shooting?

for a camera review[1]. 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 7.7) 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

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

Table 7.5 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 7.6 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 7.7 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 (ie; 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?

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

Table 7.8 Questions retrieved by Query Likelihood Model incorporating Answerability and their Answers from Review (Nikon D3300)

Question: Dose it have selective color mode like D5200? **Answer**: 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).
Question: What are the essential differences between the D5200 and D3200? Anything actually affecting image quality? **Answer**: 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 & color rendition.
Question: 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? **Answer**: 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.
Question: What is the highest iso setting? **Answer**: 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-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 7.7). The answers of the 1st, 2nd, 5th, and 7th questions are addressed in multiple sections of the review (shown in Table 7.8). 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.

7.1.3 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[2] 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:

[2] `http://www.amazon.com/gp/customer-reviews/R360W96STAOKUI?ASIN=B0158SRJVQ`

> ...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 7.9 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 7.10 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 7.11 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.

Table 7.9 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 7.10 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 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 7.11 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??

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:

> ...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.

7.2 Quantitative Analysis

The results on the two datasets (introduced in Section 6.1) achieved by different summarization systems (introduced in Section 6.2) are shown in Table 7.12 to 7.15. 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 at the 0.05 level, and ‡ indicates the corresponding method outperforms all the other methods significantly at the 0.05 level. As the length of the summary increases, the ROUGE precision scores decrease but the recall scores decrease for both datasets. Such a finding demonstrates the trade-off between precision and recall as increasing the length of the summary would include both meaningful words as well as noises.

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

Table 7.12 Summarization Results (Unigram-ROUGE Scores) on TV Dataset

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

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

Table 7.13 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**†

vocabulary size of answer collections for the camera category is larger than that of the TV category according to Table 6.1. 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.

Table 7.14 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	**0.367**‡	**0.229**‡	**0.271**‡

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 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.

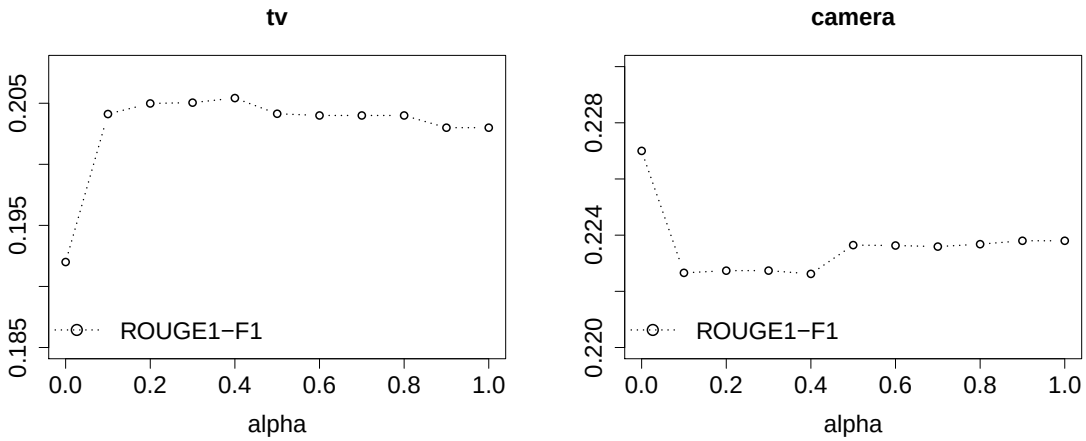

Figure 7.1 ROUGE-1 F_1 Scores on TV and Camera Datasets by Combined Query Likelihood Language Model with different Weights of Answer Collection

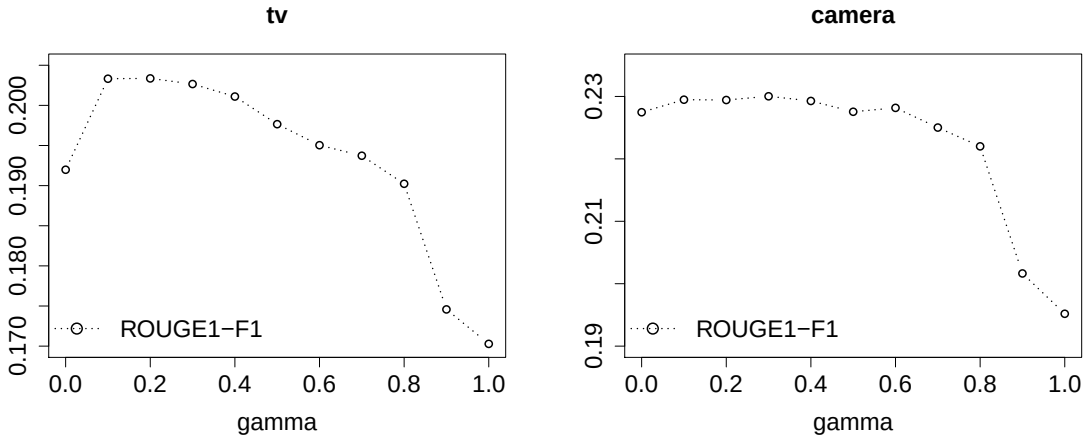

Figure 7.2 ROUGE-1 F_1 Scores on TV and Camera Datasets by Query Likelihood Language Model with different Weights of Answerability Measurement

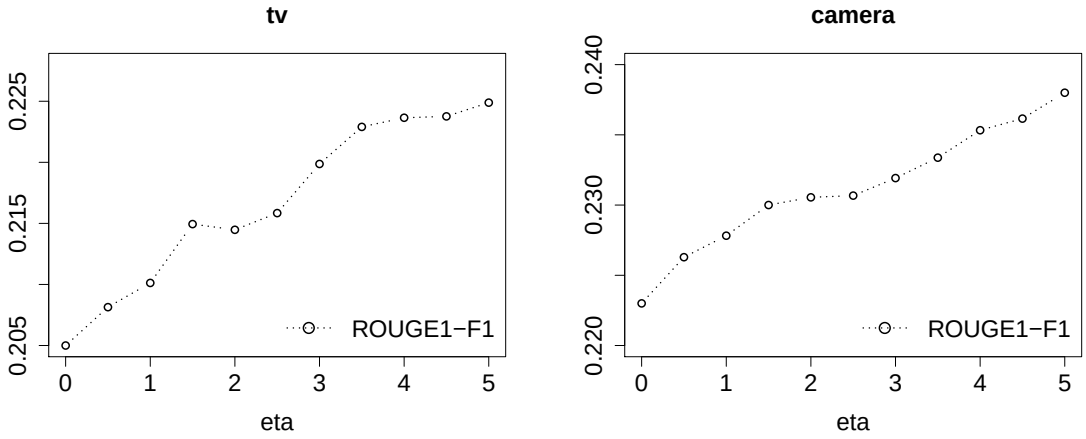

Figure 7.3 ROUGE-1 F_1 Scores on TV and Camera Datasets by Combined Query Likelihood Language Model with different Diversity Regularizer

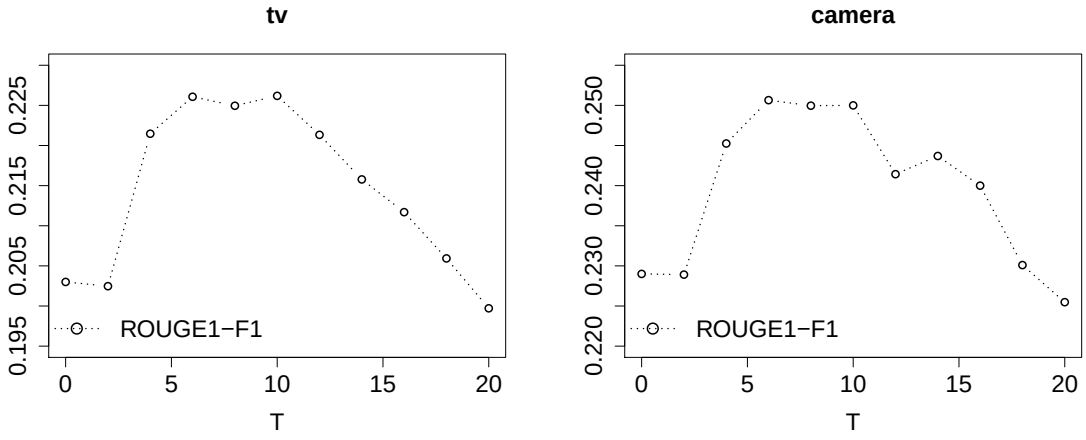

Figure 7.4 ROUGE-1 F_1 Scores on TV and Camera Datasets by Query Likelihood Language Model with Answerability Measurement with different Number of Question Clusters

Table 7.15 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**

7.3 Parameter Analysis

In order to examine the impact of the smoothing parameter α of the answer collection (Eqn.(5.9)), the trade-off between relevancy and answerability (Eqn.(5.12)), diversity regularizer η and number of question clusters T for the sumbodular function (Eqn.(5.17)), we examine the summarization performance (unigram-ROUGE F_1 scores) achieved by system (2), (3), (7), and (8) (introduced in Section 6.2) 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 ROUGE curves achieved with other threshold settings follow similar patterns so we leave them out.

Figure 7.1 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 7.2 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 consistent with the analysis in Section 7.2. 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 7.3 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 7.4 shows the impact of number of question clusters T on the query likelihood language model with answerability measurement and diversity promotion. The JM 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 set to between 6 and 10. The score decrease when T is greater than 10. We set $T = 10$ for both datasets.

Chapter 8: Discussions

8.1 From Extractive Summaries to Abstractive Summaries

The experimental results presented in Chapter 7 show the feasibility of our proposed framework. The query likelihood language model is used to retrieve candidate questions that are relevant to a review. The Recurrent Neural Network(RNN) Encoder-Decoder is used to enrich the relevance score obtained by the query likelihood language model in order to consider the answerability between the questions and a review. The diversity of the final question-based summary is measured by submodular optimization. One limitation of this approach is that the questions are coming from an existing community question and answer database. In this chapter, we explore question-based text summarization from a more general perspective and generate question-based *abstracts* rather than *extracts*. Our goal is to find a question-based summary of the original text document which is grammatical and conveys the most important information without necessarily using the same words in the same order.

It is natural to cast the abstractive question-based text summarization task as mapping an input sequence of words in a source document to an output sequence of words that are used to produce a (few) question(s). Recently, sequence-to-sequence learning models which are used to generate an output sequence given inputs, have achieved great success in the domain of machine translation [13, 104, 3], speech recognition [4], and video captioning [108].

Among the aforementioned tasks, machine translation is the most related one as it aims to transform an input text sequence in one language to an output sequence in another language. This task requires the information conveyed in the original source is kept when it is translated to another language. Thus, it is intuitive to expect almost one-to-one alignment of words between source and target. Abstractive text summarization, on the other hand, aims to produce a condensed representation of original text that captures the main points of the source. It is not necessary to find one-to-one word level alignment between the source and summary.

A notable sequence-to-sequence learning model for machine translation task is attentional Recurrent Neural Network (RNN)-Encoder Decoder [3]. This model is a general end-to-end approach for translating a sequence to another without assuming sequence structure. Specifically, it use a multilayer RNN to map the input sequence to a vector of a fixed dimensionality, and then another deep RNN to decode the target sequence from the vector. To ensure the one-to-one word level alignment between the source and target, the model automatically search for word positions in the source when predicting the target word. This model has achieved state-of-the-art performance in machine translation.

In Section 5.2, we introduce a sequence-to-sequence learning model, the Recurrent Neural Network (RNN) Encoder-Decoder, to measure the answerability between a question and a document. The RNN Encoder-Decoder is trained with a large amount of answer/question pairs. It is worth noting that even though we consider this approach as an abstractive approach for question-based summarization, we did not use this trained model to generate questions. Instead, the resultant model was used to score a pair of unseen document (which can be treated as an approximation of answers) and question. In this chapter, we explore the usage of an RNN Encoder-Decoder to automatically generate questions given a text document.

8.2 Question Generation

Inspired by the recent success of neural machine translation, we apply the sequence-to-sequence learning framework to our abstractive question-based summarization task. Our encoder is modeled off the attention-based encoder of Bahdanau et al. [3] in that it learns a latent soft alignment over the input text document to help information the question-based summary. Crucially both the encoder and the decoder are trained jointly.

Once we move away from extractive summarization, we must find an appropriate training set for our question-based abstractive task. The community question and answering (Q&A) databases introduced in Section 6.2 are natural resources. The answers could be regarded as the original document, while the questions could be seen as the summary. We train an RNN-Encoder Decoder to learn the compression from answers to questions. The resultant model could be used to generate

natural questions given a text.

In what follows, We first present the problem and present the model in Section 8.2.1, and present the experiments in Section 8.2.3. The experimental results on a public product Q&A dataset are presented in Section 8.2.6. We finally conclude this chapter in Section 8.3.

8.2.1 Problem Definition

In Chapter 3, we define our problem as using a few questions to summarize a text document. Abstractive text summarization itself is more complicated than extractive summarization task as it requires advanced natural language generation techniques. To simplify the problem, we use only one question to summarize a text document.

We formalize the task of abstractive question-based text summarization as follows. Denote the input document x as a sequence of words $\boldsymbol{x} = \{x_1, ..., x_M\}$ with M words, in which each word $\boldsymbol{x}_i$ comes from a fixed vocabulary V of size $|V|$. Question-based abstractive text summarization aims to take $\boldsymbol{x}$ as an input, and generates a short question $\boldsymbol{y} = \{y_1, ..., y_N\}$, in which each word y_i comes from another fixed vocabulary U of size $|U|$. Our goal is to maximize the conditional probability of $\boldsymbol{y}$ given $\boldsymbol{x}$, i.e., argmax $P(\boldsymbol{y}|\boldsymbol{x})$. It is worth noting that in traditional abstractive text summarization task, U could be the same as V, but since we aim to use a question to summarize a text document, the input and output sequences are not exact parallel text, so we expect the vocabulary of the output sequence U should be different from V, or we could say $U \subseteq V$.

8.2.2 A Neural Attentional Model for Question Generation

Recurrent Neural Networks

A recurrent neural network (RNN) is a neural network that consists of a hidden state $\boldsymbol{h}$ and an optional output $\boldsymbol{y}$ which operates on a variable-length sequence $\boldsymbol{x} = \{x_1, ..., x_M\}$. At each time t, the hidden state $\boldsymbol{h}_t$ of the RNN is updated by:

$$\boldsymbol{h}_t = f(\boldsymbol{h}_{t-1}, x_t), \tag{8.1}$$

where f is a non-linear activation. f may be as simple as an element-wise logistic sigmoid function and as complex as a long short-term memory (LSTM) [37] or gated recurrent unit (GRU) [15]. In this study, we adopt the GRU and will introduce this function later.

An RNN can learn a probability distribution over a sequence by being trained to predict the next symbol in a sequence. In that case, the output at each timestep t is the conditional distribution $P(x_t|x_1, ..., x_{t-1})$. For example, a multinomial distribution (1-of-K coding) can be output using a softmax activation function:

$$P(x_{t,j} = 1|x_1, ..., x_{t-1}) = \frac{\exp(\boldsymbol{w}_j \boldsymbol{h}_t)}{\sum_{j'=1}^{K} \exp(\boldsymbol{w}_{j'} \boldsymbol{h}_t)}, \tag{8.2}$$

for all possible symbols $j = 1, ..., K$, where $\boldsymbol{w}_j$ are the rows of a weight matrix W. By combining these probabilities, we can compute the probability of the sequence x using

$$P(\boldsymbol{x}) = \prod_{t=1}^{T} P(x_t|x_1, ..., x_{t-1}). \tag{8.3}$$

From this learned distribution, it is straightforward to sample a new sequence by iteratively sampling a symbol at each time step. In the following sections, we present how two RNNs are used to establish a sequence-to-sequence learning framework to map a sequence of words into another sequence.

RNN Encoder-Decoder

We employ a sequence-to-sequence learning architecture that learns to *encode* a variable-length sequence into a fixed-length vector representation and to *decode* a given fixed-length vector representation back into a variable-length vector sequence. From a probabilistic perspective, this new model is a general method to learn the conditional distribution over a variable-length sequence conditioned on yet another variable-length sequence, e.g., $P(y_1, ..., y_N|x_1, ..., x_M)$, where one should note that the input and output sequence lengths M and N may differ.

The encoder is essentially an RNN that reads each symbol of an input sequence $\boldsymbol{x}$ sequentially. As it reads each symbol, the hidden state of the RNN changes according to Eqn. (8.1). After reading

the end of the sequence(marked by an end-of-sequence symbol), the hidden state of the RNN is a summary $\boldsymbol{c}$ of the whole input sequence.

Specifically, for each input word x_i, and then we obtain the vector

$$\boldsymbol{c} = g(\boldsymbol{h}_1, ..., \boldsymbol{h}_M), \tag{8.4}$$

where g is a function that transform hidden states to the summary vector $\boldsymbol{c}$.

The decoder is another RNN which is trained to generate the output question sequence by predicting the next symbol y_t given the hidden state $\boldsymbol{h}_t$. However, unlike the RNN described in Section 8.2.2, both y_t and $\boldsymbol{h}_t$ are also conditioned on y_{t-1} and on the summary $\boldsymbol{c}$ of the input sequence. Hence, the hidden state of the decoder at time t is computed by

$$\boldsymbol{h}_i = f(x_i, \boldsymbol{h}_{i-1}), \tag{8.5}$$

and similarly, the conditional distribution of the next symbol is

$$P(y_t|y_1, ..., y_{t-1}, \boldsymbol{c}) = r(\boldsymbol{h}_t, y_{t-1}, \boldsymbol{c}), \tag{8.6}$$

for given activation functions f, g, and r (the latter must produce valid probabilities, e.g., with a softmax).

The framework is shown in Fig. 8.1.

The two components of the RNN Encoder-Decoder are jointly trained to maximize the conditional log-likelihood

$$\max_{\boldsymbol{\theta}} \frac{1}{N} \sum_{i=1}^{N} \log P(\boldsymbol{y}_i|\boldsymbol{x}_i; \boldsymbol{\theta}), \tag{8.7}$$

where $\boldsymbol{\theta}$ is the set of the model parameters and each $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ is an (input sequence, output sequence) pair from the training set. In our case, as the output of the decoder, starting from the input, is differentiable, we can use a gradient-based algorithm to estimate the model parameters.

Once the RNN Encoder-Decoder is trained, the model can be used in two ways. One way is

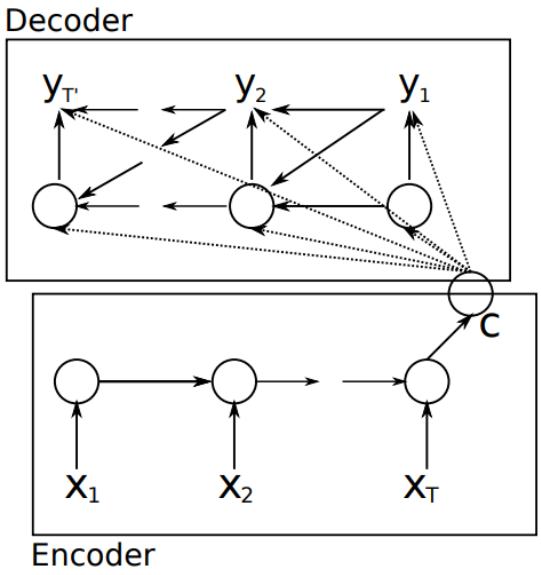

Figure 8.1 Recurrent Neural Network Encoder-Decoder [13]

to use the model to generate a target sequence given an input sequence. On the other hand, the model can be used to score a given pair of input and output sequences, where the score is simply a probability $P(\boldsymbol{y}|\boldsymbol{x})$ from Eqn. (8.3) and (8.7). Recall that in Section 5.2, we enrich the ranking score from the retrieval-based model by incorporating the score produced by an RNN Encoder-Decoder.

Gated Recurrent Unit

A gated recurrent unit (GRU) was proposed by Cho et al. [13] to make each recurrent unit to adaptively capture dependencies of different time scales. The GRU has two gating units, *reset* gate and *update* gate that modulate the flow of information inside the unit. Fig. 8.2 depicts the graphical depiction of GRU. The update gate z selects whether the hidden state is to be updated with a new hidden state $\tilde{h}$. The reset gate r decides whether the previous hidden state is ignored.

We describe how the activation of the j^{th} hidden unit at time t is computed.

First, the reset gate r_j is computed by

$$r_j = \sigma\Big([\boldsymbol{W}_r\boldsymbol{x}]_j + [\boldsymbol{U}_r\boldsymbol{h}_{t-1}]_j\Big), \tag{8.8}$$

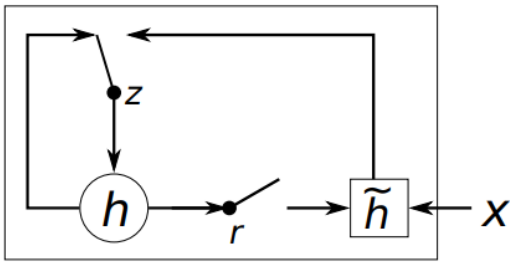

Figure 8.2 Gated Recurrent Unit [13]

where σ is the logistic sigmoid function, and $[.]_j$ denotes the j^{th} element of a vector. $\boldsymbol{x}$ and $\boldsymbol{h}_{t-1}$ are the input and the previous hidden state, respectively. $\boldsymbol{W}_r$ and $\boldsymbol{U}_r$ are weight matrices which are learned.

Similarly, the update gate is computed by

$$z_j = \sigma\Big([\boldsymbol{W}_z\boldsymbol{x}]_j + [\boldsymbol{U}_z\boldsymbol{h}_{t-1}]_j\Big), \tag{8.9}$$

where the difference with reset gate is that the second term changes to a transformation of the hidden state obtained in previous step.

The actual activation of the unit $[\boldsymbol{h}_t]_j$ is then computed by

$$h_j^t = z_j h_j^{t-1} + (1 - z_j)\tilde{h}_j^t, \tag{8.10}$$

where

$$\tilde{h}_j^t = \phi\Big([\boldsymbol{W}\boldsymbol{x}]_j + [\boldsymbol{U}\big(\boldsymbol{r} \odot \boldsymbol{h}_{t-1}\big)]_j\Big) \tag{8.11}$$

In the above formulation, when the reset gate is close to 0, the hidden state is forced to ignore the previous hidden state and reset with the current input only. This effectively allows the hidden state to drop any information that is found to be irrelevant later in the future, thus, allowing a more compact representation.

On the other hand, the update gate controls how much information from the previous hidden

state will carry over to the current hidden state. This helps the RNN to remember long-term information.

As each hidden unit has separate reset and update gates, each hidden unit will learn to capture dependencies over different time scales. Those units that learn to capture short-term dependencies will tend to have reset gates that are frequently active, but those that capture longer-term dependencies will have update gates that are mostly active.

An Attentional RNN Encoder-Decoder

Our task of question-based abstractive text summarization aims to reduce the content of a long text document to a short question. In Eqn.(8.6), the encoder transforms a sequence of input into a fixed-length vector. This may make it difficult for the neural network to cope with long sentences, especially those that are longer than the sentences in the training corpus. Therefore, we employ an attentional mechanism [3] to address this issue.

In an attentional RNN Encoder-Decoder, the conditional probability in Eqn.(8.6) is defined as

$$P(y_i|y_1, ..., y_{i-1}, \boldsymbol{x}) = g(\boldsymbol{s}_i, y_{i-1}, \boldsymbol{c_i}), \tag{8.12}$$

where s_i is an RNN hidden state at time i, computed by

$$s_i = f(s_{i-1}, y_{i-1}, c_i). \tag{8.13}$$

It should be noted that unlike the encoder-decoder defined in Eqn.(8.6), the context vector c is no longer fixed. Here the probability is conditioned on a distinct context vector c_i for each target word y_i.

The context vector c_i depends on a sequence of annotations $(h_1, ..., h_M)$ to which an encoder maps the input sequence. Each annotation h_i contains information about the whole input sequence with a strong focus on the parts surrounding the i^{th} word of the input sequence.

The context vector c_i is then computed as a weighted sum of these annotations h_i

$$c_i = \sum_{j=1}^{M} \alpha_{ij} h_j. \tag{8.14}$$

The weight α_{ij} of each annotation h_j is computed by

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{j'=1}^{M} \exp(e_{ij'})}, \tag{8.15}$$

where

$$e_{ij} = a(s_{i-1}, h_j) \tag{8.16}$$

is an *alignment* model which scores how well the inputs around position j and the output at position i match. The score is based on the RNN hidden state s_{i-1} (just before emitting Eqn.(8.12)) and the j^{th} annotation h_j of the input sequence.

The alignment model is parametrized as a feedforward neural network as follows

$$a(s_{i-1}, h_j) = v_a^\top \tanh(\boldsymbol{W}_a s_{i-1} + \boldsymbol{U}_a h_j), \tag{8.17}$$

where $\boldsymbol{W}_a$ and $\boldsymbol{U}_a$ are weighted matrices.

The alignment model is jointly trained with all the other components of the RNN Encoder-Decoder framework. It is worth noting that the alignment model directly computes a soft alignment, which allows the gradient of the cost function to be back-propagated through. This gradient can be used to train the alignment model as well as the whole model jointly.

We can understand the approach of taking a weighed sum of all the annotations as computing an *expected* annotation, where the expectation is over possible alignments. Let α_{ij} be a probability that the target word y_i is aligned to, or translated from, a source word x_j. Then, the i^{th} context vector c_i is the expected annotation over all the annotations with probabilities α_{ij}.

The probability α_{ij}, or its associated energy e_{ij}, reflects the importance of the annotation h_j with respect to the previous hidden state s_{i-1} in deciding the next state s_i and generating y_i. Intuitively,

this implements a mechanism of attention in the decoder. The decoder decides parts of the source sentence to pay attention to. By letting the decoder have an attention mechanism, we relieve the encoder from the burden of having to encode all information in the source sentence into a fixed-length vector. With this new approach the information can be spread throughout the sequence of annotations, which can be selectively retrieved by the decoder accordingly.

8.2.3 Experiments

A stumbling block to studying question-based abstractive text summarization is the lack of widely available corpora for training and testing. Although there exist some benchmark datasets for abstractive text summarization, such as DUC [79], they have not been created with generating question-based summaries. Since training a deep RNN Encoder-Decoder requires a large amount of dataset, e.g., more than 100K pairs of input/output text sequences, it is not wise to manually create question-based summaries for a large number of text documents. It is worth noting that we rely on human-generated question-based summaries for evaluating retrieval-based methods in Chapter 7, as they do not require any training examples. When we try to incorporate semantic and answerable measurement between review and questions, we leverage a large amount of community question and answer (Q&A) database for training an RNN Encoder-Decoder.

In this study, we also utilize the Q&A database for training and testing the RNN Encoder-Decoder. The answers could be treated as an approximation of text documents and their questions as summaries. We employ the same dataset[1] used in previous chapter. Specifically, we test different summarization systems on six different product categories, Automotive, Electronics, Health and Personal Care, Home and Kitchen, Sports and Outdoors, and Tools and Home Improvement. It is noted that we do not differentiate factoid questions with non-factoid questions, and leave the exploration of generating factoid questions and non-factoid questions for future work.

We apply the following data preprocessing steps on all the above six datasets using NLTK. As mentioned before, the two text collections of answers and questions are not parallell in terms of vocabulary. We apply data preprocessing steps for each of them. For answers, we apply sentence

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

segmentation, word tokenization and lemmatization. We remove punctuations, tokens that indicate urls, unicode characters and tokens inside a parentheses which we assume are less relevant information contained in answers. Observing that the content of answers include a lot of numbers and dimensions, we replace them with two special tokens, _digit and _dim. The answers contain a lot of product names or model names (e.g., Nikon D5500 and 5801XL razor), which usually consist of a mix of digits and capitalized letters. We replace those tokens with a special token _dlmix. We remove tokens whose document frequency is less than 3 and replace tokens whose document frequency is less than 5 with a special token _UNK. Considering the answers need to convey information asking aspects about a product, we remove short answers whose length is less than 5. In addition, since we want to use one question to summarize an answer, we do not expect the answers to be very long, but we still want to keep the length variety of answers to test the robustness of the RNN Encoder-Decoder, so we keep the maximum length of answers as 50. For questions, sometimes people ask multiple questions at the same time, which is out of the scope of this study. Hence, we set the constraint that there is only one question paired with an answer. We apply the same preprocessing steps described for answer preprocessing to question data except sentence segmentation in Chapter 6. We keep the question length between 5 to 15. The statistics for the six datasets are summarized in Table 8.1 and 8.2. The average question length for all datasets is 9 and average answer length is 19 to 20. It is noting that the number of questions and answers in each of the datasets are not necessary the same. Multiple answers could have the same question as a summary. Therefore, the number of questions is smaller than that of answers in all datasets.

For each dataset, we randomly split those datasets into three partitions for training (90%), validation (5%) and test (5%), respectively. We apply such a partition in order to feed a large amount of training examples into the RNN Encoder-Decoder.

The training is run using the TensorFlow library [1]. In the RNN Encoder-Decoder architecture, each Gate Recurrent Unit (GRU) has 2 hidden layers, and each layer has 128 hidden units. 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.

Table 8.1 Statistics of Amazon Q&A Data (1)

Dataset	Avg. Question Length	Avg. Answer Length
Automative	9.486	19.616
Electronics	9.774	20.464
Health	9.502	20.761
Home and Kitchen	9.711	20.392
Sports	9.668	20.547
Tools and Home	9.895	20.648

Table 8.2 Statistics of Amazon Q&A Data (2)

Dataset	#Question	#Answer	Question Vocabulary Size	Answer Vocabulary Size
Automative	40,112	42,306	2,872	4,469
Electronics	116,909	125,887	4,909	7,679
Health	30,356	33,060	2,742	4,911
Home and Kitchen	66,742	75,146	3,660	6,142
Sports	47,057	49,998	3,525	5,496
Tools and Home	37,670	39,524	2,859	4,619

8.2.4 Summarization Systems

Since the task of generating questions for text summarization is new, there is no existing models that can be readily trained on the question-based abstractive text summarization data. Since our problem could be treated as an inverse QA problem, we adopt a retrieval-based method as a baseline. Specifically, we employ a query likelihood language model, which is used as an extractive approach for question retrieval in Section 5.1, to retrieve a candidate list of questions that are relevant to a given answer. The second one is an abstractive method, which is a variant of RNN Encoder-Decoder with and without attentional mechanism.

8.2.5 Evaluation Metrics

We follow the way of question and answering [103] for evaluating the models. For every answer in the test set, we use the query likelihood model to retrieve 29 questions from the question pool, which are used as negative examples. We evaluate the aforementioned summarization system from the ranking perspective in order to see how the RNN Encoder-Decoder re-rank the result list. The evaluation metrics include: Mean Average Precision (MAP), Recall@k, and Normalized Discounted

Cumulative Gain@k (NDCG@k). The definitions of these metrics are defined as follows:

$$\text{MAP} = \frac{1}{|A|} \sum_{a \in A} AP(a), \tag{8.18}$$

where Q is the set of test answers, and $AP(q)$ stands for the precision of the relevant questions retrieved by answer given a ranked list.

$$\text{Recall@}k = \frac{1}{|A|} \sum_{a \in A} \sum_{i=1}^{k} Recall(a), \tag{8.19}$$

where Q is the set of test answers, and $Recall(q)$ stands for the probability that a relevant question is retrieved by an answer at top-k position. We set k to 1, 3, and 5, respectively.

$$\text{NDCG@}k = \frac{1}{|A|} \sum_{a \in A} \frac{1}{Z_a} \sum_{i=1}^{k} \frac{2^{r_i^a} - 1}{\log(1+i)}, \tag{8.20}$$

where A is the set of test answers, k indicates the top k questions obtained in a ranked list, Z_a is a normalization factor, and r_i^a indicates the boolean value of the question in the i-th position in the ranked list in descending order of values for answer a. This metric measures the ranking quality with respect answers at position k [43]. We set k to 5, 10, and 15, respectively.

8.2.6 Results

Quantitative Analysis

Table 8.3 to 8.8 show the ranking results obtained by different methods: query likelihood model (QL), Recurrent Neural Network (RNN) Encoder-Decoder with and without attention mechanism. Boldface stands for best performance per column. We conduct paired t tests for all the comparisons of results achieved by two different methods. † indicates the corresponding method outperforms the query likelihood baseline statistically significantly at the 0.01 level, and ‡ indicates the corresponding method outperforms all the other methods significantly at the 0.01 level.

The patterns are consistent across different datasets except the "Home and Kitchen" dataset. The RNN Encoder-Decoder performs better than query likelihood model does, and the attentional RNN Encoder-Decoder performs better than the RNN Encoder-Decoder without incorporation of

Table 8.3 Ranking Results of Amazon Q&A Data — Automative

Models	MAP	Recall@5	Recall@10	Recall@15	NDCG@5	NDCG@10	NDCG@15
QL	0.108	0.112	0.148	0.183	0.083	0.094	0.103
RNN	0.302	0.365	0.505	0.624	0.284	0.329	0.360
Attentional RNN	**0.319**‡	**0.390**‡	**0.535**‡	**0.659**‡	**0.303**‡	**0.350**‡	**0.383**‡

Table 8.4 Ranking Results of Amazon Q&A Data — Electronics

Models	MAP	Recall@5	Recall@10	Recall@15	NDCG@5	NDCG@10	NDCG@15
QL	0.099	0.097	0.139	0.175	0.071	0.084	0.094
RNN	0.295	0.348	0.491	0.607	0.274	0.320	0.350
Attentional RNN	**0.306**‡	**0.360**‡	**0.504**‡	**0.623**‡	**0.286**‡	**0.332**‡	**0.363**‡

Table 8.5 Ranking Results of Amazon Q&A Data — Health

Models	MAP	Recall@5	Recall@10	Recall@15	NDCG@5	NDCG@10	NDCG@15
QL	0.136	0.140	0.211	0.252	0.108	0.131	0.141
RNN	0.253	0.317	0.455	0.583	0.235	0.278	0.312
Attentional RNN	**0.271**‡	**0.343**‡	**0.472**‡	**0.612**‡	**0.256**‡	**0.296**‡	**0.333**‡

Table 8.6 Ranking Results of Amazon Q&A Data — Home and Kitchen

Models	MAP	Recall@5	Recall@10	Recall@15	NDCG@5	NDCG@10	NDCG@15
QL	0.126	0.135	0.181	0.216	0.101	0.116	0.125
RNN	**0.294**†	0.359	0.492	0.612	**0.277**†	0.320	0.351
Attentional RNN	0.287	**0.370**†	**0.520**‡	**0.642**‡	0.273	**0.321**†	**0.353**†

Table 8.7 Ranking Results of Amazon Q&A Data — Sports

Models	MAP	Recall@5	Recall@10	Recall@15	NDCG@5	NDCG@10	NDCG@15
QL	0.128	0.138	0.184	0.231	0.103	0.118	0.130
RNN	0.259	0.312	0.470	0.601	0.235	0.286	0.320
Attentional RNN	**0.278**‡	**0.347**‡	**0.504**‡	**0.638**‡	**0.260**‡	**0.310**‡	**0.345**‡

the attention mechanism does. The attentional RNN Encoder-Decoder yields the best results for all ranking metrics on all datasets and perform significantly better than both query likelihood model and RNN Encoder-Decoder with an attention mechanism except for the Recall@5 on the "Tools and Home" dataset. Such a finding is consistent with the results presented in Chapter 7 that incorporating answerability score to the relevancy score is beneficial for re-ranking the candidate questions and thus yield better summarization performance. The results on the "Home and Kitchen" dataset are mixed. The RNN Encoder-Decoder without and with the attention mechanism achieve competitive results and outperform significantly query likelihood model.

Table 8.8 Ranking Results of Amazon Q&A Data — Tools and Home

Models	MAP	Recall@5	Recall@10	Recall@15	NDCG@5	NDCG@10	NDCG@15
QL	0.150	0.174	0.233	0.282	0.128	0.147	0.160
RNN	0.227	0.269	0.429	0.586	0.199	0.250	0.291
Attentional RNN	**0.240**‡	**0.301**‡	**0.459**‡	**0.595**†	**0.218**‡	**0.268**‡	**0.304**‡

Qualitative Analysis

Table 8.9 shows three question generation results by an attentional RNN Encoder-Decoder. The answers are used as the input sequence to feed into a trained RNN Encoder-Decoder. The questions are regarded as the ground truth that are paired with each answer. The list of questions above each question are generated questions using beam search [90]. In the first example, even though there is no word overlap between the answer and question as well as the generation results, the attentional RNN Encoder-Decoder is able to generate meaning questions that can be addressed by the answer. In the second example, the attentional RNN Encoder-Decoder outputs meaning questions that are relevant to different operating systems. In the third example, similar to the first example, there is no identical words that occur in both answer and question, but the attentional RNN is able to generate semantically relevant questions. From the grammar perspective, the top first results for the three examples are correct; but some of the grammar is incorrect in the following results.

Table 8.10 show another three question generation results that are not semantically relevant once the input answer sequence is longer. The generation results in the first and second examples do not match with the semantics conveyed in those answers. The generation results contain special tokens that need to be mapped back to a real token. In this way, domain knowledge is needed to expect more meaningful generation results [77].

Table 8.11 shows the ranking results by different systems for two input answers. Boldface stands for ground truth questions. In the first example, the ground truth is ranked at the 9th position by query likelihood language model. The top-5 questions selected by query likelihood model contain the words from the input answer including “it”, “comes”, “with”, “new”, “cap”; but they are not semantically related with the answer except the first one. In constrast, the RNN Encoder-

Table 8.9 Good Generation Results

Answer: about foot
Question: how long be the entire cord
how long be the cord
how long long cord
how long be the dimension
how long be the cable
how long be this this this cable
how long be this this
how be the dimension
how be the be this cable
how long long dimension
how long be this this this unit
how long dimension
what be the dimension
Answer: definitely i have use it with xp vista and several flavor of linux oss
Question: will this device work on win
will this work with windows
will this work with window
will this work with mac
will this work with a _dlmix
does this work with windows
does this work with window
will this work with iphone
does this work with mac
will this work with a iPhone
Answer: they be _dim
Question: what size be they
what be the dimension
how be the dimension
how be this this this cable
how be this this this unit
what be the dimension or _dlmix
what be be this
how be be the dimension
how be the dimension dimension
how be this this this camera
what be the cord

Decoder promotes the ground truth into the second position; and after incorporating the attention mechanism, the attentional RNN Encoder-Decoder is able to promote the question to the first place. Even though only the question "Does this come with the cap?" is labeled as the ground truth, both the first and second questions selected by RNN Encoder-Decoder without or with attention mechanism could be addressed by the input answer. The fourth question selected by both models is partially semantically aligned with the input answer. Therefore, it could be used as a candidate

summary as well.

In the second example, the query likelihood fails to select the ground truth question in top positions but the RNN Encoder-Decoder and attentional RNN Encoder-Decoder are able to promote the question into second and first places. The second and fourth questions selected by the query likelihood language model contain the word "sir" which is used in the answer as a salute. The "sir" occurred in the second question is a misspelling word which should be "air"; but the second question is partially semantically matched with answer. The "sir" is a car model; but the fourth question is also semantically aligned with the answer. Such a finding reflects the robustness of the query likelihood model. The mismatch of a word does not affect the quality of the retrieved questions.

Similar to the first example, there is more than one question that could be treated as a summary and be addressed by the input answer. The top four questions selected by query likelihood model and top five questions selected by the both RNN models are plausible summaries. However, in our test set, there is only one positive example paired with the input answer. Therefore, the performance of the three models would be under-estimated.

8.3 Summary

In this Chapter, we explore the usage of a sequence-to-sequence learning model, an attentional RNN Encoder-Decoder for question generation. We experiment with a public question and answer database from which we construct six experimental datasets. For each dataset, we use the answers as the input sequences and questions as the output sequences to train an attentional RNN Encoder-Decoder. The encoder is expected to capture the information from the input sequence, and the decoder is expected to learn the grammatical of the output sequence. The attention mechanism is expected to learn the semantic alignment of word that occur in both input and output sequences.

Experimental results show that the attentional RNN Encoder-Decoder performs significantly better than query likelihood model in terms of ranking metrics. We also examine the generation results produced by a trained RNN Encoder-Decoder. We found that when the length of input sequence is short (e.g., less than 10 words), it is more likely to generate semantically and grammatically correct questions. However, if the length of input sequence is longer, the generation results are less plausible

and contain a lot of special tokens which need domain knowledge to recover the special tokens back to meaningful tokens. Therefore, it is important to encode domain knowledge to train an RNN Encoder-Decoder to expect more plausible generation results.

The approach for generating abstractive summary for document could be plugged into the first component in the framework proposed in Section 4. Instead of selecting a set of candidate questions from an existing question database, the sequence-to-sequence learning model can be used to automatically generate questions. If we set the constraint for the diversity optimization problem (Section 5.3) in the second stage to one, we can use the generated question as an abstractive summary. We can also use the sequence-to-sequence learning model to generate multiple questions given different sections of a text document, which is similar to how we measure answerability between a question and document segment in Section 5.2, and then promote the diversity in the question set based on generation scores.

To train the RNN Encoder-Decoder, we experiment with a simple recurrent neural network unit with 2 layer of 128 hidden units. We found that this simple model works well for our dataset. One of the possible reason is that our training data is not very large, compared with over 100 million training examples for training an email responding machine [47], and 9.5 million training examples for training a summarization system for news headline generation [92]. In our experimental dataset, the majority questions are yes/no binary questions. Extension to generating open-ended questions is a future direction.

Last but not least, it is worth noting that the evaluation for abstractive summary might be underestimated. According to the generation results shown in Section 8.2.6, some of the generation results that are not in top-1 position could be used as the questions that can be addressed by the input answer. The ranking results shown in 8.2.6 also demonstrate that there exist false positives. Liu et al. [62] conducted an analysis of different evaluation metrics for dialogue response generation and concluded that system generated results are not strongly correlated with human judgment. Therefore, it is also a future research direction to design appropriate evaluation metrics for automatic text generation tasks.

Table 8.10 Bad Generation Results

Answer: comes with one serial port
Question: how many serial port do it have
does it come with a _dlmix
does it have a _UNK
does it come with a _UNK
does it have a _dlmix
does it come with the _dlmix
does it come with the _UNK
does it come with a camera
does it work with the _dlmix
does it work with a _UNK
does it work with a _dlmix
does this work it
does it have with a _dlmix
does it come with a _UNK _dlmix
does it come with with
does it have with a _UNK
does it have with the _UNK
does this come it
Answer: it adjust the volume of the headphone i use the line out to go to hercules powered speaker the volume be then control through the powered speaker not the _dlmix
Question: do the volume control adjust the line output or just the headphone
what be the warranty
what be the the _dlmix
how be the dimension
what be the dimension
what be the dimension of a _dlmix
will this work with a a _dlmix
what be the _UNK
what be the battery have
Answer: the samsung lens be also _dlmix they will fit and make you very happy ps you should see list on the lens
Question: will these fit the samsung _dlmix lens
will this work with a _dlmix
will this work for a _dlmix
will this work with a _dlmix _dlmix
will this work on a _dlmix
will this work for the _dlmix
will this fit with a _dlmix
will this work with my _dlmix
will this fit on a _dlmix
will this work on the _dlmix
will this work on my _dlmix
will this work for a _dlmix _dlmix
will this work with _dlmix
will this work for my _dlmix
will this work with a canon _dlmix
will this fit with the _dlmix
will this work with a nikon _dlmix
will this fit on the _dlmix
will this fit with a sony _dlmix

Table 8.11 Ranking Results obtained by different Systems

Answer: Yes it comes with new cap.
Question: Does this come with the cap?
QL
(1) Did it come with a new rad cap?
(2) Will it fit my 2013 dodge diesel 2500?
(3) Any chanch of purchasing a new cap from amazon?
(4) Does this scale come with warranty? If yes, what it is?
(5) where do you get new end cap?
.....
(9) **Does this come with the cap?**
RNN
(1) Does this come with a cap?
(2) **Does this come with the cap?**
(3) Does it come with a new blank key?
(4) Does this come with the cap? And is it black?
(5) Do these come with new bolts?
Attentional RNN
(1) **Does this come with the cap?**
(2) Does this come with a cap?
(3) Does this come with the cap? And is it black?
(4) Do these come with black cap?
(5) Does the caps come in sets of 4's?
Answer: as long as your car is 4 lug it will fit sir :)
Question: will it fit my 2004 lancer?
QL
(1) Hi i see they have 5 lugs my car have 4 will they still fit?
(2) Will these fit my 1999 deville and with the sir suspension?
(3) My car is 192 inch long. Will this fit?
(4) Hey guys, can this fit my honda CF4 SIR?
(5) Did anybody have to install long lugs?
.....
(30) **Will it fit my 2004 lancer?**
RNN
(1) Will these fit 8 lug c20?
(2) **Will it fit my 2004 lancer**
(3) Will these fit a 2013 5 lug tacoma?
(4) Will these fit my 6 lug 96 z71 Chevy?
(5) Will they fit a 2000 Chevy 2500 Silverado 8 lug pattern?
Attentional RNN
(1) **Will it fit my 2004 lancer?**
(2) Will these fit 8 lug c20?
(3) Will these fit a 2013 5 lug tacoma?
(4) Will these fit my 6 lug 96 z71 Chevy?
(5) Will they fit a 2000 Chevy 2500 Silverado 8 lug pattern?

Chapter 9: Conclusions and Future Work

9.1 Conclusions

This thesis addresses a novel task: summarizing a text document through questions. Questions are often more attractive for readers to read than declarative sentences are. They can serve as "hints" for readers to decide whether they want to further read a document. To the best of our knowledge, no prior work has studied this task in the literature, as existing work on text summarization mainly focuses on extracting declarative sentences from original text documents or generating declarative summaries.

We propose a two-stage framework consisting of question selection and question diversification. Both extractive and abstractive approaches are explored for the question selection component. Extractive approaches aim to retrieve candidate questions from an existing question database based on relevancy between questions and a text document using a probabilistic retrieval model. Abstractive approaches aim to measure the answerability between questions and a text document using a sequence-to-sequence learning model. For the question diversification component, submodular optimization is used to consider both question coverage and answerability measure of the text document and non-redundancy.

We verify our proposed framework and methods in the domain of product review summarization. We create and annotate a dataset by manually locating and editting questions for reviews in two product categories. The experimental results demonstrate the proposed methods can effectively find relevant questions from an existing community question and answer database for summarizing a product review, and significantly outperform the baseline methods.

9.2 Future Work

This thesis is an initial step towards a promising research direction. The new problem of question-based text summarization can be further explored from multiple perspectives. First, we could

extend the problem of single document summarization through questions to that of multi-document summarization. Indeed, our proposed framework and its components can be extended to fit this problem. One possible solution could be first retrieving or generating candidate questions for each individual text documents, then synthesizing the questions and re-ranking them by our question diversity component. In Section 5.2, our way of partially matching a question's answerability with a section of a text document can be applied to the multi-document summarization task, since a document can be seen as a collection of multi-text sections. Our proposed framework is flexible that we can plug in other ranking functions to measure the quality of a question-based summary. If the ranking function is used to measure the coverage of a document, it could be included in the question selection stage; if another ranking function is designed to promote the diversity, it could be included in the question diversification stage.

Second, evaluation results show that while our proposed approach can retrieve reasonable questions, there is still a wide gap with human performance which motivates further work on enriching our proposed components with domain knowledge. Examples of domain-specific information in our product review summarization task include product specifications, review sentiments, or question ratings. We do not separate factoid questions with non-factoid questions in our study, but designing a more tailored component for different types of questions would be a future direction [101].

Third, we would like to apply the proposed framework and methods to other question retrieval/generation tasks. In fact, the problem tackled in this thesis can be formulated as an inverse question-answering task by using given "answers" to find/generate relevant questions, which may have numerous applications in the real world. Despite we use product reviews as the experimental dataset, our approach can be applied to other text summarization tasks, such as news, scientific articles, social media, knowledge bases. Examples of applications include question generation for educational material creation [35], automatic email responding machine [47], etc. The framework can even be extended for feeding in more than just text data, such as multimedia data. Example of applications include generating questions for images [75] and videos [108].

Last but not least, we would also like to deploy the proposed method to a real-world review

system and measure the satisfaction of real users. Current evaluation is based on the matching between human generated summaries and system generated summaries. We expect to find whether the question-based review summarization will help improve user's online shopping experience. Comparing the effectiveness of declarative sentence-based summary and question-based summary is also a future direction. As mentioned before, we do not aim to replace question-based summary with declarative sentence-based summary; but we would like to compare the effectiveness of using these two types of summaries and see how different they affect a user's search behavior.

Bibliography

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous systems, 2015. volume 1, 2015.

[2] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749, 2005.

[3] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[4] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio. End-to-end attention-based large vocabulary speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 4945–4949. IEEE, 2016.

[5] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *IJCAI*, volume 7, pages 2670–2676, 2007.

[6] M. Banko, V. O. Mittal, and M. J. Witbrock. Headline generation based on statistical translation. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 318–325. Association for Computational Linguistics, 2000.

[7] R. Barzilay and K. R. McKeown. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328, 2005.

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

[9] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *journal of machine learning research*, 3(Feb):1137–1155, 2003.

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

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

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

[13] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[14] S. Chopra, M. Auli, A. M. Rush, and S. Harvard. Abstractive sentence summarization with attentive recurrent neural networks. *Proceedings of NAACL-HLT16*, pages 93–98, 2016.

[15] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[16] J. Clarke and M. Lapata. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research*, 31:399–429, 2008.

[17] T. Cohn and M. Lapata. Sentence compression beyond word deletion. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 137–144. Association for Computational Linguistics, 2008.

[18] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

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

[20] V. Dang and W. B. Croft. 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*, pages 65–74. ACM, 2012.

[21] D. Das and A. F. Martins. A survey on automatic text summarization. *Literature Survey for the Language and Statistics II course at CMU*, 4:192–195, 2007.

[22] R. Day and P. Jeong-suk. Developing reading comprehension questions. *Reading in a foreign language*, 17(1):60, 2005.

[23] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479, 2004.

[24] K. Filippova, E. Alfonseca, C. A. Colmenares, L. Kaiser, and O. Vinyals. Sentence compression by deletion with lstms. In *EMNLP*, pages 360–368, 2015.

[25] K. Filippova and Y. Altun. Overcoming the lack of parallel data in sentence compression. In *EMNLP*, pages 1481–1491. Citeseer, 2013.

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

[27] P.-E. Genest and G. Lapalme. Fully abstractive approach to guided summarization. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 354–358. Association for Computational Linguistics, 2012.

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

[29] 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.

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

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

[32] V. Gupta and G. S. Lehal. A survey of text summarization extractive techniques. *Journal of emerging technologies in web intelligence*, 2(3):258–268, 2010.

[33] D. Harman. Overview of the trec 2002 novelty track. In *TREC*, 2002.

[34] M. Heilman. *Automatic factual question generation from text*. PhD thesis, Carnegie Mellon University, 2011.

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

[36] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[37] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

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

[39] B. Hu, Q. Chen, and F. Zhu. Lcsts: A large scale chinese short text summarization dataset. pages 1967—-1972, 2015.

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

[41] M. Hu, A. Sun, and E.-P. Lim. Comments-oriented blog summarization by sentence extraction. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 901–904. ACM, 2007.

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

[43] K. Järvelin and J. Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 41–48. ACM, 2000.

[44] H. Jing. Sentence reduction for automatic text summarization. In *Proceedings of the sixth conference on Applied natural language processing*, pages 310–315. Association for Computational Linguistics, 2000.

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

[46] K. S. Jones et al. Automatic summarizing: factors and directions. *Advances in automatic text summarization*, pages 1–12, 1999.

[47] A. Kannan, K. Kurach, S. Ravi, T. Kaufmann, A. Tomkins, B. Miklos, G. Corrado, L. Lukács, M. Ganea, P. Young, et al. Smart reply: Automated response suggestion for email. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, volume 36, pages 495–503, 2016.

[48] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.

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

[50] K. Knight and D. Marcu. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1):91–107, 2002.

[51] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[52] L.-W. Ku, Y.-T. Liang, H.-H. Chen, et al. Opinion extraction, summarization and tracking in news and blog corpora. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, volume 100107, 2006.

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

[54] C.-S. Lee, Z.-W. Jian, and L.-K. Huang. A fuzzy ontology and its application to news summarization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(5):859–880, 2005.

[55] 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.

[56] 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.

[57] 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.

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

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

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

[61] 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.

[62] C.-W. Liu, R. Lowe, I. V. Serban, M. Noseworthy, L. Charlin, and J. Pineau. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*, 2016.

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

[64] M. Liu, Y. Fang, A. G. Choulos, D. H. Park, and X. Hu. Product review summarization through question retrieval and diversification. *Information Retrieval Journal*, pages 1–31, 2017.

[65] M. Liu, Y. Fang, D. H. Park, X. Hu, and Z. Yu. Retrieving non-redundant questions to summarize a product review. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 385–394. ACM, 2016.

[66] K. Lopyrev. Generating news headlines with recurrent neural networks. *arXiv preprint arXiv:1512.01712*, 2015.

[67] H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165, 1958.

[68] 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.

[69] E. Malmi, P. Takala, H. Toivonen, T. Raiko, and A. Gionis. Dopelearning: A computational approach to rap lyrics generation. *arXiv preprint arXiv:1505.04771*, 2016.

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

[71] J. McAuley and A. Yang. Addressing complex and subjective product-related queries with customer reviews. In *Proceedings of the 25th International Conference on World Wide Web*, pages 625–635. International World Wide Web Conferences Steering Committee, 2016.

[72] R. T. McDonald. Discriminative sentence compression with soft syntactic evidence. In *Eacl*, 2006.

[73] K. McKeown, L. Shrestha, and O. Rambow. Using question-answer pairs in extractive summarization of email conversations. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 542–550. Springer, 2007.

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

[75] N. Mostafazadeh, I. Misra, J. Devlin, M. Mitchell, X. He, and L. Vanderwende. Generating natural questions about an image. *arXiv preprint arXiv:1603.06059*, 2016.

[76] R. Nallapati, B. Zhou, Ç. glar Gulçehre, and B. Xiang. Abstractive text summarization using sequence-to-sequence rnns and beyond. pages 280–290, 2016.

[77] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*, 2016.

[78] 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.

[79] P. Over, H. Dang, and D. Harman. Duc in context. *Information Processing & Management*, 43(6):1506–1520, 2007.

[80] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

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

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

[83] 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.

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

[85] D. R. Radev, E. Hovy, and K. McKeown. Introduction to the special issue on summarization. *Computational linguistics*, 28(4):399–408, 2002.

[86] D. R. Radev, H. Jing, and M. Budzikowska. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *Proceedings of the 2000 NAACL-ANLP Workshop on Automatic summarization*, pages 21–30. Association for Computational Linguistics, 2000.

[87] D. R. Radev, H. Jing, M. Styś, and D. Tam. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938, 2004.

[88] S. Ravi, B. Pang, V. Rastogi, and R. Kumar. Great question! question quality in community q&a. *ICWSM*, 14:426–435, 2014.

[89] D. Ravichandran and E. Hovy. Learning surface text patterns for a question answering system. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 41–47. Association for Computational Linguistics, 2002.

[90] D. R. Reddy et al. Speech understanding systems: A summary of results of the five-year research effort. *Department of Computer Science. Camegie-Mell University, Pittsburgh, PA*, 1977.

[91] V. Rus and C. G. Arthur. The question generation shared task and evaluation challenge. In *The University of Memphis. National Science Foundation*. Citeseer, 2009.

[92] A. M. Rush, S. Chopra, and J. Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389. Association for Computational Linguistics, 2015.

[93] R. L. Santos, C. Macdonald, and I. Ounis. Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th international conference on World wide web*, pages 881–890. ACM, 2010.

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

[95] L. Shang, Z. Lu, and H. Li. Neural responding machine for short-text conversation. *arXiv preprint arXiv:1503.02364*, 2015.

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

[97] H. Singer. Active comprehension: From answering to asking questions. *The Reading Teacher*, 31(8):901–908, 1978.

[98] I. Soboroff. Overview of the trec 2004 novelty track. In *TREC*, 2004.

[99] I. Soboroff and D. Harman. Overview of the trec 2003 novelty track. In *TREC*, 2003.

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

[101] R. Soricut and E. Brill. Automatic question answering: Beyond the factoid. In *HLT-NAACL*, pages 57–64, 2004.

[102] C. Speier, J. S. Valacich, and I. Vessey. The influence of task interruption on individual decision making: An information overload perspective. *Decision Sciences*, 30(2):337–360, 1999.

[103] M. Surdeanu, M. Ciaramita, and H. Zaragoza. Learning to rank answers on large online qa collections. In *ACL*, volume 8, pages 719–727, 2008.

[104] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[105] 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.

[106] 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.

[107] J. Ulrich, G. Murray, and G. Carenini. A publicly available annotated corpus for supervised email summarization. In *Proc. of aaai email-2008 workshop, chicago, usa*, 2008.

[108] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to sequence-video to text. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4534–4542, 2015.

[109] 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.

[110] E. Wood, V. Woloshyn, and T. Willoughby. *Cognitive strategy instruction for middle and high schools*. Brookline Books, 1995.

[111] K. Woodsend, Y. Feng, and M. Lapata. Generation with quasi-synchronous grammar. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 513–523. Association for Computational Linguistics, 2010.

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

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

[114] 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.

[115] D. Zajic, B. Dorr, and R. Schwartz. Bbn/umd at duc-2004: Topiary. In *Proceedings of the HLT-NAACL 2004 Document Understanding Workshop, Boston*, pages 112–119, 2004.

[116] 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.

[117] 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.

[118] 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.

[119] 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.

Vita

Mengwen Liu

Education

- Drexel University, Philadelphia, Pennsylvania USA
  - Ph.D., Information Science, December 2017
- Syracuse University, Syracuse, New York USA
  - M.S., Information Management, May 2012
- Capital Normal University, Beijing, China
  - B.S., Information Management and Information Systems (E-Commerce), July 2010

Publications and Patents

- **M. Liu**, Y. Fang, A. G. Choulos, D. H. Park, and X. Hu. Review summarization through question retrieval and diversification. Information Retrieval Journal, 2017
- **M. Liu**, W. Ding, D. H. Park, Y. Fang, R. Yan, and X. Hu. Which used product is more sellable? a time-aware approach. Information Retrieval Journal, 2017
- **M. Liu***, Y. Fang*(equal contribution), D. H. Park, X. Hu, and Z. Yu. Retrieving non-redundant questions to summarize a product review. SIGIR, 2016
- **M. Liu**, L. Guo, and H. Wang. Time-value estimation method and system for sharing environment, Dec. 30 2015. US Patent App. 14/984,812
- **M. Liu**, Y. Shang, L. Guo, and H. Wang. Method and system for multimodal clue based personalized app function recommendation, July 22 2015. US Patent App. 14/805,830
- **M. Liu**, Y. Ling, Y. An, and X. Hu. Relation extraction from biomedical literature with minimal supervision and grouping strategy. BIBM, 2014
- **M. Liu**, Y. An, X. Hu, D. Langer, C. Newschaffer, and L. Shea. An evaluation of identification of suspected autism spectrum disorder (ASD) cases in early intervention (EI) records. BIBM, 2013

Awards

- Student Travel Award, ACM SIGIR, 2016, IEEE BIBM, 2014
- Inaugural CCI Day PhD Poster Award, Drexel University, 2014
- Best Student Paper Award, IEEE BIBM, 2013

Professional Activities

- Program Committee Member, ACM CIKM 2016 Workshop on DDTA
- Reviewer, IEEE J-BHI 2017
- Reviewer, China Institute of Communications 2016
- Reviewer, IEEE ICNC 2016