THE UNIVERSITY *of* EDINBURGH

Document Summarization with Neural Query Modeling

Yumo Xu

Doctor of Philosophy

Institute for Language, Cognition and Computation

School of Informatics

The University of Edinburgh

2022

Abstract

Document summarization is a natural language processing task that aims to produce a short summary that concisely delivers the most important information of a document or multiple documents. Over the last few decades, the task has drawn much attention from both academia and industry, as it provides effective tools to manage and access text information. For example, through a newswire summarization engine, users can quickly digest a cluster of news articles by reading a short summary of the topic. Such summaries can, meanwhile, be used by news recommendation and question answering engines. Depending on the users' role in the summarization process, document summarization falls into two broad categories: generic summarization and query focused summarization (QFS). The former focuses on information intrinsically salient in the input text, while the latter also caters to requests explicitly specified by users.

Despite the difference between generic summarization and QFS in their task formulations, we argue that all summaries address queries, even if they are not formulated explicitly. In this thesis, we introduce query modeling in the document summarization context as a critical objective for incorporating observed or latent user intent. We investigate different approaches that explore this theme with deep neural networks. We develop novel systems with neural query modeling for both extractive summarization, where summaries are composed of salient segments (e.g., sentences) from the original document(s), and abstractive summarization, where summaries are made up of words or phrases that do not exist in the input.

The recent availability of large-scale datasets has driven the development of neural models that create generic summaries. However, training data in the form of queries, documents, and summaries for QFS is scarce. As most existing research in QFS has employed an extractive approach, we first consider better modeling query-cluster interactions for low-resource extractive QFS. In contrast to previous work with retrieval-style methods for assembling query-relevant summaries, we propose a framework that progressively estimates whether text segments should be included in the summary. Notably, modules of this framework can be independently developed and can leverage training data if available. We present an instantiation of this framework with distant supervision from question answering where various resources exist to identify segments which are likely to answer the query. Experiments on benchmark datasets show that our framework achieves competitive results and is robust across domains.

Ideally, summaries should be abstracts, and the hidden costs incurred by annotating QA pairs should be avoided in query modeling. The second part of this thesis focuses

on the low-resource challenge in abstractive QFS, and builds an abstractive QFS system which is trained query-free. Concretely, we propose to decompose the task into query modeling and conditional language modeling. For query modeling, we first introduce a unified representation for summaries and queries to exploit training resources in generic summarization, on top of which a weakly supervised model is optimized for evidence estimation. The proposed framework achieves state-of-the-art performance in generating query focused abstracts across existing benchmarks.

Finally, the third part of this thesis moves beyond QFS. We provide a unified modeling framework for any kind of summarization, under the assumption that all summaries are a response to a query, which is observed in the case of QFS and latent in the case of generic summarization. We model queries as discrete latent variables over document tokens, and learn representations compatible with observed and unobserved query verbalizations. Requiring no further optimization on downstream summarization tasks, experiments show that our approach outperforms strong comparison systems across benchmarks, query types, document settings, and target domains.

Acknowledgements

Thank you to my supervisor, Mirella Lapata, for your patience, guidance, and support. I have benefited greatly from your wealth of knowledge and meticulous editing. I am extremely grateful that you took me on as a student and continued to have faith in me over the years. It will always be an honor for me to have been your student.

I would like to express my sincere gratitude to Shay Cohen and Laura Perez-Beltrachini, for their helpful feedback on my annual reviews during the course of my PhD degree. My examiners, Ivan Titov and Greg Durrett, also have my deep appreciation for their effort in examining my thesis.

I would like to thank the outstanding researchers in Mirella's group: Hao Zhen, Jiangming Liu, Jianpeng Cheng, Jonathan Mallinson, Laura Perez-Beltrachini, Li Dong, Nelly Papalampidi, Parag Jain, Ratish Puduppully, Reinald Kim Amplayo, Rui Cai, Stefanos Angelidis, Tom Hosking, Tom Sherbone, Yang Liu and Yao Fu, for the helpful discussions at every stage of my PhD research. I would like to extend my sincere thanks to the members of the Edinburgh NLP group: Adam Lopez, Alex Lascarides, Bonnie Webber, Frank Keller, Ivan Titov, Kenneth Heaeld, Mark Steedman, Rico Sennrich, Sharon Goldwater, Shay Cohen, and Walid Magdy, for the valuable comments on my work in our group meetings.

Many thanks to all my colleagues and friends at School of Informatics: Bailin, Biao, Christos, Chunchuan, Elena, He, Nicola, Sabine, Shucong, Xinchi, Xinnuo, Yanpeng, Zhijiang, etc. Also, I would like to offer my special thanks to my office mates, Bowen, Joachim, Kai, Marco, Parag, and Yang. It is delightful to have your company during this journey, and I will very much miss our daily chats and laughs.

Finally, I would like to express my deepest gratitude to my parents and my wife, for their love and unwavering belief in me.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Yumo Xu)*

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview of Document Summarization

Since its inception, the Internet has been growing. A tremendous amount of documents such as web pages, news articles and blogs is circulating the digital space. According to the World Wide Web Size project, it is estimated that Google has indexed over 50 billion webpages in 2021.[1] As a result, users are at a loss to find what they are looking for. For most users, the materials online are either redundant or not relevant, leading to information overload (Feldman et al., 2007). To address this problem, over the years, many Natural Language Processing (NLP) tasks proposed to automatically analyze documents and help users digest information. Document summarization is one core technique that helps users navigate online documents efficiently via reducing their length and condensing them into short summaries.

Maybury (1999) define text summarization as follows:

> Text summarization is a process of distilling the most important information from a source (or sources) to produce an abridged version for a particular user (or users) and task (or tasks).

Based on Maybury's definition, summarization represents different subtasks according to different objectives or factors. Following are three main factors:

- **Generic vs. Query focused**: This factor concerns what summaries focus on. Generic summarization provides a summary of intrinsically important information the input document(s), while query focused summarization caters to user queries which are additionally specified in the input.

[1] https://www.worldwidewebsize.com

- **Extractive vs. Abstractive**: This factor concerns the form of output summaries. Extractive summarization produces summaries by composing text spans selected from the input document(s), while abstractive summarization can generate abstracts with novel words or phrases that do not appear in the input.

- **Single-document vs. Multi-document**: This factor simply concerns the number of documents input into the summarization system. Single-document summarization conditions on one input document, while in multi-document summarization, the input consists of multiple documents (which are usually clustered by similar topics).

In Table 1.1, we show examples of different types of summarization. In this thesis, we will explore both extractive and abstractive approaches to improve document summarization systems, with a special focus on query modeling with neural networks, which we argue plays an important role in not only query focused summarization (QFS), but also generic summarization. Before discussing the role of queries in the next section, we first provide a general overview of research in document summarization and briefly introduce its historical context.

Early Research Early work on document summarization goes back to the extractive methods developed in the 1950s for scientific text processing. To reduce information overload in the scientific literature, Luhn (1958) developed a program that scores sentences based on word frequencies and then extracts high scoring ones as literature abstracts. The assumption is that the importance of sentences can be measured by the frequency of specific content words. Concurrent work by Baxendale (1958) found that sentence position in a document can also indicate sentence importance for inclusion in the summary. Based on this assumption, a simple but effective extractive system for document summarization, LEAD, was proposed, which takes a few lead sentences of an article as the summary. Instead of using a single representation of a document topic, Edmundson (1969) proposed to extract summary sentences based on a combination of factors. Specifically, Edmundson (1969) considered the following four features to measure sentence salience: the frequency of a word used in the article, the position of the sentence, the number of words used in the article title or section heading, and the frequency of cue-words. With a simple linear summation used for sentence scoring, this work set the framework for future machine learning approaches to document summarization.

Generic Document Summarization

Document: Millionaire real estate heir Robert Durst has pleaded not guilty to two weapons charges related to his arrest last month, further delaying his extradition to California to face murder charges. Durst entered his plea during an arraignment in a New Orleans court on weapons charges that accused him of possessing a firearm after a felony conviction and possessing both a firearm and an illegal drug, marijuana. Durst's hands were shackled to his sides, and two defense attorneys lifted him from an armchair to his feet to walk to the podium. Unlikely to face charges in California anytime soon: Robert Durst, 71, pleaded not guilty Thursday to two state gun charges in Louisiana in a case that would delay his extradition to LA to face murder charges. Durst is pictured here last month in New Orleans. Attorney Dick DeGuerin whispered into Durst's ear as he entered the plea. He had to whisper twice before Durst said, 'I am not guilty, your honor.' Judge Franz Zibilich asked if Durst was making that plea to both charges against him. DeGuerin whispered again, and Durst said, 'Yes, your honor.' The weapons arrest has kept Durst in New Orleans even though he waived extradition to California, where he's charged in the December 2000 death of a longtime friend...

Extractive Summary: Unlikely to face charges in California anytime soon: Robert Durst, 71, pleaded not guilty Thursday to two-state gun charges in Louisiana in a case that would delay his extradition to LA to face murder charges. The weapons arrest has kept durst in New Orleans even though he waived extradition to California, where he's charged in the December 2000 death of a longtime friend.

Abstractive Summary: Robert Durst was indicted Wednesday on the two weapons charges that have kept him in New Orleans. Grand jury charged durst with possession of a firearm by a felon, and possession of both a firearm and an illegal drug: 5 ounces of marijuana. On Thursday he appeared in court to plead not guilty. Durst, 71, is wanted in California for the murder of his friend Susan Berman. Berman, an author who formerly acted a media spokeswoman for durst, was shot in the head at her benedict canyon home in 2000.

Query Focused Document Summarization

Query: *Prashant Bhushan, Legal activism, Government accountability*

Document: Feather in cap for graft fighters. New Delhi, March 3: The Supreme Court verdict against P.J. Thomass appointment is not the lone feather in the cap of the petitioner, the Centre for Public Interest Litigation (CPIL), but perhaps the most visible one. The Delhi-based group, a loose collection of activists and lawyers whose aim is to fight corruption, had its previous big hurrah in 2003. That was when it got the apex court to restrain the Centre from divesting majority shares in Hindustan Petroleum and Bharat Petroleum without Parliaments approval. In the 2G allotment case filed by the group, the Supreme Court has already ordered a CBI probe. Another public interest litigation (PIL), filed by group member and senior lawyer Ram Jethmalani, asks that the government be directed to recover Indian black money stashed in foreign banks. Our organisation is devoted to taking up PILs in a systematic, professional and organised manner. We file them on our own or if we are requested to by someone else, said lawyer and group member Prashant Bhushan. The CPIL was founded in the late 1980s by Justice V.M. Tarkunde, who also co-founded the Peoples Union for Civil Liberties...

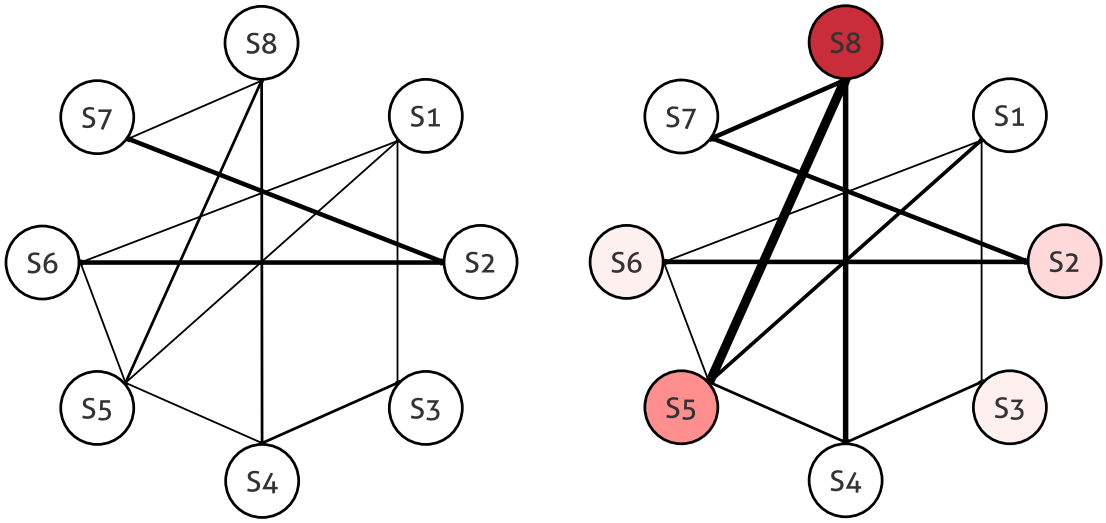
Extractive Summary: New Delhi, March 3: The Supreme Court verdict against P.J. Thomass appointment is not the lone feather in the cap of the petitioner, the Centre for Public Interest Litigation (CPIL), but perhaps the most visible one. That was when it got the apex court to restrain the Centre from divesting majority shares in Hindustan Petroleum and Bharat Petroleum without Parliaments approval. The CPIL was founded in the late 1980s by Justice V.M. Tarkunde, who also co-founded the Peoples Union for Civil Liberties.

Abstractive Summary: The Centre for Public Interest Litigation (CPIL) is a loose collection of activists and lawyers. The group had its big hurrah in 2003 when it got the apex court to restrain the Centre from divesting majority shares in Hindustan Petroleum and Bharat Petroleum.

Table 1.1: Examples of different types of summarization. The top table shows generic summarization, while the bottom table is an example of for query focused summarization (QFS). For simplicity, we show summarization output with a single document as input, however, both generic summarization and QFS can summarize multiple documents. QFS has an additional query as input to which the summary needs to respond. For each task, we show an extractive and an abstractive summary. Extractive summaries are text spans copied from the input document which we highlight in red. In contrast, novel words/phrases that do not appear in the input are used in abstractive summaries.

Statistical Learning In the 1990s, many feature-based learning systems were proposed for document summarization, thanks to advances in statistical machine learning. Based on an extended set of features from Edmundson (1969), Kupiec et al. (1995) formulated extractive summarization as a classification task, and proposed a naive-Bayes classifier to identify summary-worthy sentences. Subsequent work studied better intermediate representations of sentences for extractive summarization and proposed to incorporate richer features such as statistics of noun phrases (Aone et al., 1997). Based on sentence representations, machine learning models such as decision trees (Lin, 1999) and Hidden Markov Models (HMMs; Conroy and O'leary 2001), were implemented to obtain more accurate importance scores. After sentence scoring, sentence selection is usually used to find a subset of sentences in the document as the final summary, considering sentence importance and other factors including information redundancy. Carbonell and Goldstein (1998) first introduced Maximal Marginal Relevance (MMR) as a greedy approach to combine sentence relevance with information novelty. To obtain a globally optimal solution, McDonald (2007) further formulated the sentence selection problem as a constrained optimization problem and solved it with Integer Linear Programming (ILP).

Graph Based Ranking In the early 2000s, the research community witnessed the increasing popularity of graph based ranking models for document summarization. Inspired by the PageRank algorithm (Brin and Page, 1998), LexRank (Erkan and Radev, 2004) and TextRank (Mihalcea and Tarau, 2004) were the two seminal papers that first conceptualized the task of extractive summarization as identifying the most central nodes in a graph that represents the input document(s). In Figure 1.1(a), we show the graph representation of a document which is first segmented into sentences. To construct the graph, each node represents a sentence in the document, and each edge represents the similarity between the node pair it connects. The edge weights are normalized into a Markov chain where each element denotes the probability of transitioning between two states (i.e., nodes). The Markov chain can then be repeatedly run on the graph, which is guaranteed to converge to a stationary distribution that indicates the centrality of nodes in the graph. Finally, sentences can be ranked and selected according to their centrality for inclusion in the summary. LexRank and TextRank, as well as their query focused variants (which will be introduced in Section 1.2.1), are often used as comparison systems in modern extractive summarization research due to their good unsupervised performance.



(a) Graph Representation for Generic Summarization

(b) Graph Representation for Query Focused Summarization

Figure 1.1: Graph representation of a document consisting of 8 sentences $\{s_1, ..., s_8\}$, for (a) generic summarization and (b) query focused summarization. Each node is a sentence, and edge width denotes edge weight. Edges with weights lower than a pre-defined threshold are pruned before graph computation. In generic summarization, edge weights are calculated by the similarity between sentence pairs. In query focused summarization, edge weights are also influenced by the query relevance of sentences. We use red color to show query relevance of sentences and darker color denotes higher query relevance.

Sentence Compression and Fusion Extractive summarization approaches are restricted to sentence-level operations: once a sentence is determined to contain salient information, the whole sentence, which may also contain irrelevant information, will be included in the summary. Sentence compression and fusion are two methods that aim to address this problem (Nenkova and McKeown, 2011). Early work for sentence compression (Jing, 2000; Zajic et al., 2007) employed rules constructed from syntactic and discourse knowledge to determine which phrases in an extracted sentence should be removed. To automatically learn from data compression rules over syntactic constituents, Knight and Marcu (2002) developed two statistical models based on the noisy channel model and decision trees. Clarke and Lapata (2007) presented an ILP-based approach and further incorporated discourse information (encoded as hard constraints) into a statistical learning framework. Primarily applied to multi-document summa-

rization (MDS), sentence fusion generates novel summary sentences with phrases cut and pasted from original document sentences (Jing and McKeown, 2000). In MDS, one common approach is to find similarities across the input documents using sentence clustering and extract a sentence from each cluster. To reduce repetition and improve summary quality, Barzilay and McKeown (2005) introduced a general information fusion approach: phrases across a cluster of similar sentences that convey common information are first identified with pairwise alignment, and then combined into a grammatical sentence using tree traversal for linearization.

Deep Neural Networks Recent representation learning techniques, based on deep neural networks, learn continuous representations for text automatically from data, eschewing the need for human-engineered features which are expensive. After being successfully applied to natural language processing tasks such as sentiment analysis (Socher et al., 2013; Kim, 2014) and machine translation (Bahdanau et al., 2014), neural network models further showed their effectiveness in extractive summarization (Yin and Pei, 2015; Cheng and Lapata, 2016; Nallapati et al., 2017). Based on encoded representations of sentences, neural extractive summarization is formulated as a classification task where a binary label is predicted for each sentence to decide whether it should be included in the summary. Apart from neural sentence extraction, machines can learn to generate fluent abstracts using words that do not appear in the input with an encoder-decoder neural architecture (Sutskever et al., 2014). This has led to a surge of interest in abstractive summarization, which is typically framed as a sequence-to-sequence transduction problem (Rush et al., 2015; See et al., 2017). We provide a more detailed exposition of neural networks and their application to summarization in Chapter 2.

1.2 The Role of Queries in Document Summarization

In document summarization, the concept of a query is usually adopted in the context of query focused summarization (QFS). Apart from QFS, the use of natural questions, which are a specific form of query, has also been investigated for generic summarization. In this section, we will introduce the role of queries in previous document summarization research, including QFS where queries are specified in the input, and generic summarization where queries are manually or automatically created for summary evaluation or refinement.

1.2.1 The Role of Queries in QFS

Document Understanding Conferences (DUC), organized by the National Institute of Standards and Technology (NIST), introduced QFS as a new summarization task for the first time in 2005. This summarization task which was designed to be strongly tied to a user application is defined as follows (Dang, 2005):

> The system task in 2005 will be to synthesize from a set of 25-50 documents a brief, well-organized, fluent answer to a need for information that cannot be met by just stating a name, date, quantity, etc.

From the above task definition, we can deduce that QFS was proposed to address real-world queries, which can be complex and non-factoid. In fact, DUC queries are composite, consisting of a short topic and a long narrative. An example from DUC 2005 is shown below:

> Title: Amnesty International
> Narrative: What is the scope of operations of Amnesty International and what are the international reactions to its activities?

Introducing queries allows users to specify their requests and, as a result, a summarization system can better cater to users' information needs. On the other hand, handling complex queries as the example above requires accurate understanding of query semantics, which unavoidably introduces new research challenges into document summarization. Next, we will discuss how queries are handled in different phases of QFS research, as well as research efforts on query related challenges.

Early Studies on Queries Research related to QFS started prior to its formal proposal in DUC 2005, sometimes with varied terms for the task, including user focused summarization (Mani and Bloedorn, 1998), query oriented summarization (Lin, 1999) and query-relevant summarization (Berger and Mittal, 2000). To involve users in document summarization, early work (Mani and Bloedorn, 1998) defined the overall information need for a user as a set of documents: they asked each subject to pick 10 documents from a corpus that matched the subject's interests, and automatically extracted top content words from the documents. The top content words, in the form of a centroid vector, were then used to score the relevance of each sentence in the input document. As a result, user interest was incorporated in the production of summaries. Following work (Lin, 1999; Berger and Mittal, 2000) treated questions as queries, e.g., *What countries export smoked salmon?* Due to the lack of training data,

Lin (1999) took the evaluation data from a question answer task within the TIPSTER-SUMMAC project (Mani et al., 1999). Berger and Mittal (2000) further proposed to use frequently-asked question (FAQ) documents to train a statistical machine learning model.

Queries in Graph Based Models The wide adoption of graph-based approaches in generic document summarization also influenced QFS research. We show the graph representation for QFS in Figure 1.1(b). Under this query-focused framework, all sentences within the input document(s), together with their query relevance, are jointly considered in estimating centrality. A variety of approaches have been proposed to enhance the way relevance and centrality are estimated ranging from incorporating topic-sensitive information (Wan, 2008; Badrinath et al., 2011), predictions about information certainty (Wan and Zhang, 2014), manifold-ranking algorithms (Wan et al., 2007; Wan and Xiao, 2009), and Wikipedia-based query expansion (Nastase, 2008).

Queries in Neural Networks More recently, neural approaches based on neural networks have been proposed for both extractive QFS (Li et al., 2015, 2017b) and abstractive QFS (Laskar et al., 2020a). Due to the lack of training data, work on neural extractive QFS mostly tries to learn neural networks from a reconstruction objective for unsupervised QFS. On the other hand, research on abstractive QFS has had an even shorter history: it started to emerge after the successful applications of deep neural networks to abstractive systems for generic summarization. Abstractive QFS is also formulated as a sequence-to-sequence problem which relies on a neural architecture. However, QFS takes a user-specified query as input: to generate a responsive summary accordingly, the system needs to accurately understand the query semantics. As a more challenging research question with less training resources, abstractive QFS has received significantly less attention from the research community, compared to its extractive and generic counterparts. Recently, however, the increasing availability of pretrained models has prompted the development of pipeline-style frameworks for QFS which use resources from a wider range of NLP tasks. More details on neural approaches for extractive and abstractive QFS will be given in Section 2.2.2.

1.2.2 The Role of Queries in Generic Summarization

Queries, in the context of question answering (QA), were firstly adopted in generic summarization for system evaluation. In tandem with QA gaining popularity in sum-

mary evaluation, QA-related learning signals have also been studied to improve summary quality. We briefly discuss these approaches below. However, we stress that they are not targeting at query focused summarization; they aim to improve summarization in general by taking QA pairs into account.

Queries for Summary Evaluation In generic summarization, QA methods were first used to evaluate summary quality. Different from the de facto metric ROUGE (Lin and Hovy, 2003) which calculates lexical overlap with the reference summary, QA methods directly compare two summaries based on their common information. Mani et al. (1999) first proposed to use QA as an extrinsic metric to evaluate summaries, based on the assumption that a good summary should answer key questions that a reader may have about a document. Since then, QA has been incorporated into human evaluation for generic summarization, where a few questions about an article are first composed, and participants are asked to answer these questions after reading the summary (Clarke and Lapata, 2010). Recent effects focus on automating this protocol, using rule-based methods (Chen et al., 2018) or fill-in-the-blank questions (Eyal et al., 2019). While Eyal et al. (2019) restrict answers to be named entities, QAEval (Deutsch et al., 2021) take a step further by asking questions about noun phrases, which achieves state-of-the-art performance on summary evaluation. Some recent work (Durmus et al., 2020; Wang et al., 2020a) particularly focuses on automated question generation (QG) from summaries for evaluating summary faithfulness which measures whether the information in the summary is consistent with the input.

Queries for Summary Refinement Another line of research in generic summarization studies how to improve summary quality with QA. Most of this work adopts a Reinforcement Learning (RL) framework, with a QA-related reward model. Arumae and Liu (2019) present a reward function consisting of multiple objectives for extractive summaries: adequate, fluent, length-restricted, and QA-competent. For abstractive summarization, Huang et al. (2020) designed a two-stage system: after maximum likelihood training, the system is further optimized via multi-choice cloze rewards (provided by a pre-trained QA model) to generate more faithful and informative summaries. To improve both the recall and precision of abstractive summaries, Gunasekara et al. (2021) propose an RL framework that considers questions from reference and output summaries: the former promote summary relevance, while the latter refine summaries to be more factually correct.

1.3 Challenges in QFS

Document summarization is a challenging task in that a good summary requires deep document understanding, together with accurate sentence extraction or language generation capabilities. Query focused summarization, as a subtask, shares many of these challenges. Moreover, the additional constraints imposed by queries further introduces a set of novel challenges that need to be handled to produce summaries that can address users' information needs. In this section, we discuss these research challenges, with a special focus on building QFS systems with neural networks.

The Scarcity of QFS Training Data Neural approaches have become increasingly popular in generic text summarization (Nallapati et al., 2016; Paulus et al., 2018; Li et al., 2017b; See et al., 2017; Narayan et al., 2018b; Gehrmann et al., 2018), thanks to the representational power afforded by deeper architectures and the availability of large-scale datasets containing hundreds of thousands of document-summary pairs (Sandhaus, 2008; Hermann et al., 2015; Grusky et al., 2018). Unfortunately, such datasets do not exist in QFS, and one might argue it is unrealistic they will ever be created for millions of queries, across different domains, and languages. The scarcity of training data has previously led to unsupervised extractive formulations of QFS, where graph based (Wan and Zhang, 2014) or autoencoding (Li et al., 2017b) models are adopted. However, this unsupervised formulation prohibits the use of resources distantly related to QFS, and hinders research in abstractive systems that generate high-quality QFS abstracts. With more NLP resources and pretrained models being created in recent years, a machine learning framework that allows the exploitation of weak, indirect summarization signals may offer a more effective solution to the data scarcity challenge described above.

The Cost of Query-Related Resources To alleviate data scarcity in QFS, recent research efforts use query-related resources from a wider range of NLP tasks (Xu and Lapata, 2020; Su et al., 2020; Laskar et al., 2020b), including question answering (Rajpurkar et al., 2016; Chakraborty et al., 2020) and paraphrase identification (Dolan and Brockett, 2005). Despite the effectiveness of these approaches in QFS, relying on query-related resources for distant supervision leads to several other undesirable costs. The first cost is the hidden annotation expense for current QA datasets which can be extremely high (Bajaj et al., 2016; Kwiatkowski et al., 2019). Secondly, there is often a

mismatch between queries in QA datasets and those in QFS scenarios; the two types of queries are not identically distributed and, therefore, lead to distributional divergence between training and testing. Lastly, it is practically infeasible to find appropriate query-related resources for all domains and topics which makes accessibility another issue. How to design a learning scheme without heavy dependency on query-related resources is a research objective in this thesis.

The Diversity of Query Types Building and scaling QFS systems remains challenging due to the many different ways natural language queries express users' information needs. As we can see from the examples in Table 5.1, queries can be one or multiple keyword(s) (Baumel et al., 2016; Zhu et al., 2019), a simple question (Nema et al., 2017), or a longer narrative composed of multiple sub-queries (Dang, 2006) (see the examples in Table 5.1). Although QFS systems can potentially handle queries resembling those seen in training, they are not expected to work well on out-of-distribution queries (Xu and Lapata, 2021). To cover new types of queries, it might be necessary to gather more data, re-design proxy queries, and re-train one or more system components which can be computationally inefficient and in some cases practically infeasible. Therefore, a summarization system that provides a unified framework for all kinds of query verbalizations and handles user requests robustly at test time without retraining is favorable.

Modeling Multiple Documents In addition to the above-mentioned difficulties in resource acquisition and modeling, another obstacle to the application of neural summarization models is the size and number of source documents which can be very large. Given memory limitations of current hardware, it is practically infeasible to train a model which encodes all of them into vectors and subsequently produces a summary from them. Despite being a long-standing problem in generic multi-document summarization (Liu et al., 2018), this research question has not been thoroughly investigated for multi-document QFS. In this thesis, we will explore different approaches to extract and generate query focused summaries from multiple documents without direct supervision.

1.4 Thesis Statement

In this thesis, we aim at developing document summarization systems with effective query modeling while addressing the challenges outlined in the previous section. This is motivated by the assumption that all summaries address queries, even for generic summarization. To begin with, we formally define *query modeling* as a sub-task for document summarization:

Definition 1.4.1 (Query Modeling)**.** Learning a good representation for the semantics of an *observed* or *latent* query that facilitates downstream summarization, including but not limited to query focused summarization.

It is worth mentioning that query modeling has been previously studied in NLP subfields such as information retrieval (Frakes and Baeza-Yates, 1992) and semantic parsing (Srinivasan Iyer and Zettlemoyer, 2017), where only the query is given as input to the system and query modeling, therefore, influences system performance directly. Different from previous uses of the term, this thesis formally introduces query modeling in the context of document summarization, and shows how to instantiate it with neural networks, which we call *neural query modeling*, for the purpose of improving the performance of summarization tasks including QFS and generic summarization.

As most existing research in QFS has employed an extractive approach, we first focus on extractive multi-document QFS in Chapter 3. Our key insight is to treat evidence estimation as a question answering task where a cluster of potentially relevant documents provides support for answering a query (Baumel et al., 2016). Advantageously, we are able to train the evidence estimator on existing large-scale question answering datasets (Rajpurkar et al., 2016; Joshi et al., 2017; Yang et al., 2018), alleviating the data paucity problem in QFS. Existing QFS systems (Wan et al., 2007; Wan, 2008; Wan and Xiao, 2009; Wan and Zhang, 2014) employ classic retrieval techniques (such as TF-IDF) to estimate the affinity between query-sentence pairs. Such techniques can handle short keyword queries, but are less appropriate in QFS settings where query narratives can be long and complex. We argue that a trained evidence estimator might be better at performing semantic matching (Guo et al., 2016) between queries and document segments. To this effect, we experiment with two popular QA settings, namely answer sentence selection (Heilman and Smith, 2010; Yang et al., 2015) and machine reading comprehension (Rajpurkar et al., 2016) which operates over passages than isolated sentences. In both cases, our evidence estimators take advantage of powerful pre-trained encoders such as BERT (Devlin et al., 2019), to better

capture semantic interactions between queries and text units.

In Chapter 4, we avoid the dependency on query-related resources and the hidden costs incurred by annotating QA pairs, by building an abstractive QFS system which is trained query-free. Specifically, we do not assume access to any resources other than those available for generic summarization. To this aim, we decompose abstractive QFS into two subtasks: *query modeling* and *conditional language modeling*. In abstractive QFS, we instantiate the objective of query modeling as finding supportive evidence within a set of documents for a query. As a second stage, conditional language modeling generates an abstractive summary based on found evidence. Under this formulation, we use generic summarization data not only for conditional language modeling, but also for learning an evidence ranker for query modeling. Inspired by the Cloze task and its applications in NLP (Taylor, 1953; Lewis et al., 2019; Lee et al., 2019), we introduce a unified representation for summaries and queries, allowing proxy queries to be constructed from generic summaries to which we have access. Proxy queries are further used as distant supervision to optimize a regression model for evidence estimation and ranking. Based on the selected evidence, we learn a summary generator from generic summarization data to produce query focused abstracts with several controllable factors, including summary length and query influence.

Finally, we focus on the scalability of QFS systems. Most QFS work assumes short queries or compositional queries with an extra-long narrative. However, the actual queries at test time input provided by users can go far beyond one or two specific query types. This is evidenced by the development of different query forms over time in QFS benchmarks. During 2005 and 2007, DUC (Dang, 2005) served as the standard benchmark for QFS requiring participants to handle compositional queries. In 2017, TD-QFS (Baumel et al., 2016) was proposed as a benchmark in the medical domain where queries are just short titles. In the same year, Debatepedia (Nema et al., 2017) was created to cover natural questions in the debate domain for argument retrieval. These questions are not as complex as DUC narratives and they usually do not contain subqueries. More recently, researchers constructed WikiRef (Zhu et al., 2019) from Wikipedia hierarchies, and the section keywords are seen as queries. On top of these, we can also view generic summarization of a single document (Hermann et al., 2015) or multiple documents (Perez-Beltrachini and Lapata, 2021), as a special case of query focused summarization, where the query is unspecified, in other words, *null*.

A natural question then arises: can we build a summarization system that handles all possible query types (including *null*)? The motivation is straightforward: we want

to train a model that works robustly with any type of user input, instead of training a model for every type of query. In Chapter 5, we will answer this research question and present a unified framework under which we can perform both generic summarization and QFS with different query types. The framework is developed under the assumption that all summaries are a response to a query, which is observed in the case of QFS and latent in the case of generic summarization. In this case, we conceptualize query modeling as discrete latent variable modeling over document tokens, and learn representations compatible with observed and unobserved query verbalizations. Our framework formulates summarization as a generative process, and jointly optimizes a *latent query model* and a *conditional language model* using only generic summarization data for model training and development.

To conclude, the main contributions of this thesis are:

1. A coarse-to-fine framework that extracts query-relevant summaries from multiple documents, allowing QA resources to be leveraged in the summarization process for *distant query modeling*.

2. A two-stage abstractive framework that generates query focused summaries whilst removing the dependency on expensive query-related resources with *proxy query modeling*.

3. A unified formulation for generic summarization and QFS, and a deep generative framework that can handle all kinds of query types at test time robustly based on *latent query modeling*.

Experimental results across QFS datasets demonstrate the effectiveness of our proposed approaches in both extractive and abstractive settings. Besides, our system also achieves strong performance on generic summarization benchmarks which we view as special cases of QFS.

1.5 Thesis Outline

The remainder of this thesis is organized as follows:

- Chapter 2 presents background knowledge of the Transformer network and Pretrained Language Models (PLMs). We then discuss related work on document summarization, including both generic and query focused summarization.

- Chapter 3 presents a coarse-to-fine framework for extractive QFS where Question Answering (QA) resources are leveraged for evidence estimation. We first provide background on QA, and then describe our proposed framework which consists of three estimators: a relevance estimator, an evidence estimator, and a centrality estimator. Experimental results across datasets show that the proposed model yields results superior to competitive baselines across domains and query types, contributing to summaries which are more relevant and less redundant.

- Chapter 4 presents an abstractive framework for QFS where no QA training resource is required. We first propose to decouple abstractive QFS into two subtasks: query modeling and conditional language modeling. Then we introduce a unified mask representation for query modeling, which enables generic summaries to serve as proxy queries for model training. Experimental results across datasets show that the proposed system yields state-of-the-art performance despite the weakly supervised setting, and produces more relevant and coherent summaries compared to existing approaches.

- Chapter 5 presents a unified framework for generic summarization and QFS without relying on query-related resources for either model training or development. We first introduce a deep generative formulation for document summarization for any kind of summarization, under the assumption that all summaries are a response to a query, which can be either observed or latent. We then propose to model queries during training as discrete latent variables over document tokens, and learn representations compatible with observed and unobserved query verbalizations. Despite learning from generic summarization data only, our approach outperforms strong comparison systems across benchmarks, query types, document settings, and target domains.

- Chapter 6 concludes the thesis and discusses directions for future work.

Portions of this thesis have been previously published in Xu and Lapata (2020) (Chapter 3), Xu and Lapata (2021) (Chapter 4), and Xu and Lapata (2022) (Chapter 5).

Chapter 2

Background

As introduced in Chapter 1, neural document summarization is typically based on an encoder that represents the input context (in the form of natural language) in the latent semantic space, i.e., as continuous vectors. The input context for summarization is a document or a set of documents, and optionally, a user query that specifies an information request. In addition to the encoder, abstractive document summarization requires a decoder to generate an abstract that summarizes the input. As this encoder-decoder architecture is based on neural networks, in this chapter, we first introduce the basis of neural networks, with a particular focus on the Transformer model (Vaswani et al., 2017), one of the most widely adopted neural models for natural language processing. Then, we show its application to text generation, as well as serving as the backbone of many pretrained models to provide effective network initialization. We will also provide the formulations of generic summarization and query focused summarization, together with discussion of their related work. For each task, we will introduce both extractive and abstractive approaches: the former select sentences from inputs for inclusion in the summary, while the latter adopt the more sophisticated encoder-decoder architecture to produce more human-readable abstracts.

2.1 Neural Networks

Prior to Transformer models, Recurrent Neural Networks (RNNs) and their variants, such as Long Short-Term Memory Networks (LSTMs; Hochreiter and Schmidhuber 1997), have been successfully applied to various in natural language processing tasks such as machine translation (Sutskever et al., 2014) and language modeling (Jozefowicz et al., 2016). The sequential computation of input words offered by RNNs,

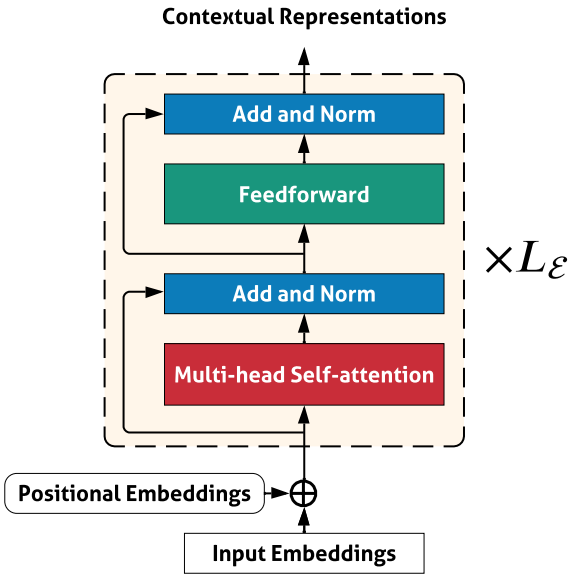


Figure 2.1: Self-attentive encoder in Transformer (Vaswani et al., 2017) stacking $L_{\mathcal{E}}$ identical layers.

however, is constrained by the recurrent nature and follows a strict temporal order. The Transformer model overcomes this constraint, with a self-attention mechanism that allows access to any position in the input sequence. This design allows for more parallelization in computation and higher learning efficiency, which has facilitated the development of large-scale model pretraining in natural language processing.

2.1.1 Transformer Models

We illustrate in Figure 2.1 one layer of the Transformer model, which can be stacked to $L_{\mathcal{E}}$ layers. Information on the relative or absolute position of each token in a sequence is represented by the use of positional encodings which are added to input embeddings (see the bottom of Figure 2.1). One Transformer layer comprises a multi-head self-attention sublayer and a position-wise fully-connected feed-forward network. After each sub-layer, layer normalization is applied to facilitate model learning. Next we will discuss these components one-by-one.

Positional Embeddings Without the recurrence mechanism in RNNs, the Transformer model augments the input elements with positional embeddings to incorporate the order of the input sequence. Specifically, the ith input element is represented with

a vector $\mathbf{h}_i \in \mathbb{R}^{d\times 1}$:

$$\mathbf{h}_i = \mathbf{p}_i + \mathbf{x}_i \tag{2.1}$$

where $\mathbf{p}_i \in \mathbb{R}^{d\times 1}$ and $\mathbf{x}_i \in \mathbb{R}^{d\times 1}$ are the positional embedding and the input embedding, respectively, and d denotes the input embeddings size.

The value of the jth element in the ith positional embedding is defined as follows:

$$\mathbf{p}_{i,j} = \begin{cases} \sin(j/10000^{2k/d}) & \text{if } j = 2k \\ \cos(j/10000^{2k/d}) & \text{if } j = 2k+1. \end{cases} \tag{2.2}$$

where $k \in \mathbb{N}^+$ is a positive integer. This sinusoidal positional encoding has the following property: for any input index i and a constant offset Δi, there always exists a linear transformation $T(\Delta i)$ so that $\mathbf{p}_{i+\Delta i} = T(\Delta i)\mathbf{p}_i$. As a result, it is easy for the Transformer model to learn to attend by relative positions.

Multi-head Attention Multi-head attention is the first sublayer of a Transformer layer. We first introduce single-head attention as its simpler variant. The single-head operation takes three inputs: a query matrix $\mathbf{Q} = \mathbf{H}\mathbf{W}_q$, a key matrix $\mathbf{K} = \mathbf{H}\mathbf{W}_k$, and a value matrix $\mathbf{V} = \mathbf{H}\mathbf{W}_v$, where $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v$ are three learnable weight matrices to project the hidden states $\mathbf{H} \in \mathbb{R}^{n\times d}$ into the query, key, and value spaces, respectively. For the ith input element, the operation of single-head attention is calculated as:

$$\text{head}_i = \text{softmax}(\frac{\mathbf{q}_i\mathbf{K}^\intercal}{\sqrt{d}})\mathbf{V} \tag{2.3}$$

where a scaled dot operation is applied between $\mathbf{q}_i$ and $\mathbf{K}$, followed by a softmax function to form a distribution over the input sequence. The distribution is then used as weights to aggregate the sequence values $\mathbf{V}$ into a contextual representation head_i.

Multi-head attention extends single-head attention by allowing input elements to jointly attend to information from multiple representation subspaces. Specifically, outputs of multiple single-head operations are first concatenated, and then linearly transformed via a learnable weight matrix $\mathbf{W}_c$:

$$\text{multi-head}_i = \text{concat}(\text{head}_i^{(1)}, .., \text{head}_i^{(K)})\mathbf{W}_c \tag{2.4}$$

where K is the number of heads. Note that each $\text{head}_i^{(\cdot)}$ operates on a different set of queries, keys and values: $\mathbf{Q}^{(\cdot)} = \mathbf{H}\mathbf{W}_q^{(\cdot)}, \mathbf{K}^{(\cdot)} = \mathbf{H}\mathbf{W}_k^{(\cdot)}, \mathbf{V}^{(\cdot)} = \mathbf{H}\mathbf{W}_v^{(\cdot)}$ where linear transformations $\mathbf{W}_q^{(\cdot)}, \mathbf{W}_k^{(\cdot)}, \mathbf{W}_v^{(\cdot)}$ project hidden states into multiple head-specific subspaces.

Feed Forward Networks The second sublayer in the Transformer model is a fully-connected feed-forward network (FFN) that is applied to each position separately and identically. FFN has two feed-forward layers and employs ReLU as the activation function. Specifically, for an input hidden vector $\mathbf{h}_i \in \mathbb{R}^{d\times 1}$, FFN is calculated as follows:

$$\text{FFN}(\mathbf{h}_i) = \mathbf{W}_1 \max(0, \mathbf{W}_0\mathbf{h}_i + \mathbf{b}_0) + \mathbf{b}_1. \tag{2.5}$$

where $\mathbf{W}_0, \mathbf{W}_1$ are two weight matrices, $\mathbf{b}_0, \mathbf{b}_1$ are biases, and max is an element-wise maximum operator.

Layer Normalization When neural networks are deep, internal covariance shift affects the stability of gradients negatively and, as a result, delays the convergence of model learning (Ioffe and Szegedy, 2015). Layer normalization (Ba et al., 2016) was proposed to alleviate this problem. For a hidden state $\mathbf{h}_i \in \mathbb{R}^{d\times 1}$, layer normalization first estimates its mean and variance:

$$\mu = \frac{1}{d}\Sigma_j^d \mathbf{h}_{i,j} \tag{2.6}$$

$$\sigma^2 = \frac{1}{d}\Sigma_j (\mathbf{h}_{i,j} - \mu)^2 \tag{2.7}$$

and then normalizes it as:

$$\text{LayerNorm}(\mathbf{h}_i) = \boldsymbol{\gamma}_i * \frac{\mathbf{h}_i - \mu}{\sigma} \tag{2.8}$$

where $\boldsymbol{\gamma}_i$ is a learnable re-scale factor (usually initialized to 1).

2.1.2 Encoder-Decoder Architectures

Many natural language generation (NLG) tasks, such as machine translation and abstractive summarization, are typically formulated as a sequence-to-sequence modeling problem: the input is composed of a sequence of words, and the model maps it to another sequence of words as the output. The neural encoder-decoder architecture has been developed for this modeling objective, and has been proven to be powerful in a wide range of NLG tasks. In this architecture, an encoder first encodes the input sequence into continuous vectors as source representations. Conditioned the source representations, a decoder then autoregressively generates a sequence of words as the output. Formally, we denote (X, Y) as an input-output pair for a language generation task where the input $X = \{x_1, ..., x_M\}$ is a sequence of words, and the output $Y = \{y_1, ..., y_T\}$ is another sequence of words conditioned on the input sequence.

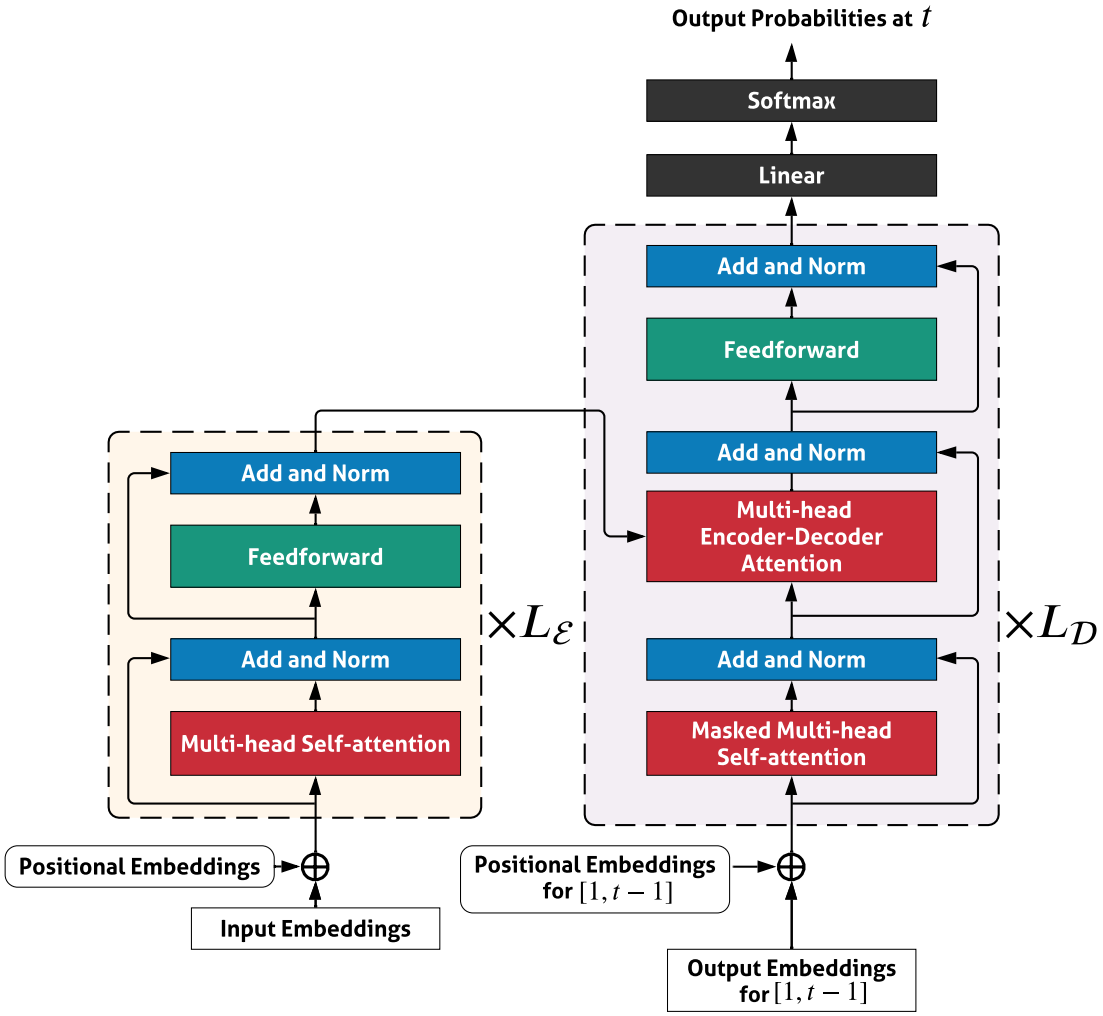


Figure 2.2: Transformer-based encoder-decoder model (Vaswani et al., 2017). The encoder consists of $L_{\mathcal{E}}$ identical encoding layers and the decoder is a stack of $L_{\mathcal{D}}$ identical decoding layers, both operating on inputs augmented with positional embeddings.

Figure 2.2 shows an instantiation of the encoder-decoder architecture based on the previously introduced Transformer model. On the left side, an encoder comprising of $L_{\mathcal{E}}$ vanilla Transformer layers (see Section 2.1.1 for details) first encodes the source X into a sequence of continuous vector representations $[\mathbf{h}_1, ..., \mathbf{h}_M]$. At the target end, as the text generation process is auto-regressive, only words that have already been generated, i.e., the generation history, can be used for the generation of the next word, and the rest should be masked. Therefore, the first sublayer in a Transformer-based decoding layer is a masked multi-head self-attention layer. In addition to the components in the encoding layer, the decoding layer includes a sublayer, called multi-head encoder-decoder attention, to attend to the source representations produced by the encoder.

Therefore, at each generation step t, a Transformer decoder of $L_{\mathcal{D}}$ layers computes the hidden state $\mathbf{h}'_t \in \mathbb{R}^{d\times 1}$ based on on the encodings for the source sequence X and generation history $\hat{y}_{<t}$, and outputs probabilities over the vocabulary V:

$$p(\hat{y}_t|\hat{y}_{<t}, X) = \text{softmax}(\mathbf{W}_y\mathbf{h}'_t + \mathbf{b}_y) \tag{2.9}$$

where $\mathbf{W}_y \in \mathbb{R}^{d\times|V|}$ is a weight matrix and $\mathbf{b}_y \in \mathbb{R}^{|V|\times 1}$ is a bias term.

2.1.3 Pretrained Models

Pretrained language models (Devlin et al., 2019; Yang et al., 2019b; Bao et al., 2020; Lewis et al., 2020) have recently advanced the state-of-the-art of natural language processing, thanks to their ability to learn universal language representations from a vast amount of unlabelled text. Compared to learning a new task from scratch, these rich representations provide a better model initialization to downstream tasks. In this section, we focus on three Transformer-based pretrained models considering their successful applications in document summarization: BERT (Devlin et al., 2019), UniLMv2 (Bao et al., 2020), and BART (Lewis et al., 2020). In this thesis, BERT will be used as the backbone of the query models proposed in Chapter 3 and 4, and different summarization systems based on UniLMv2 and BART will be introduced in Chapter 4 and 5, respectively. In this section, for each pretrained model, we first introduce its learning objective for pretraining, and then detail its neural architecture.

BERT To build contextual representations, BERT (Devlin et al., 2019), standing for Bidirectional Encoder Representations from Transformers, introduces Masked Language Modeling (MLM) as its pretraining objective. MLM randomly masks some tokens with a special token `[MASK]` in an input sequence, and aims to recover these tokens conditioned on their left and right contexts encoded by the bidirectional Transformer model (Vaswani et al., 2017). Particularly, 15% of the token positions are randomly chosen for pretraining prediction. To mitigate the mismatch between pretraining (where the `[MASK]` token exists) and fine-tuning (where `[MASK]` does not appear), each chosen token is:

1. Replaced with the `[MASK]` token, with 80% probability.
2. Replaced with a random token, with 10% probability.
3. Unchanged, with 10% probability.

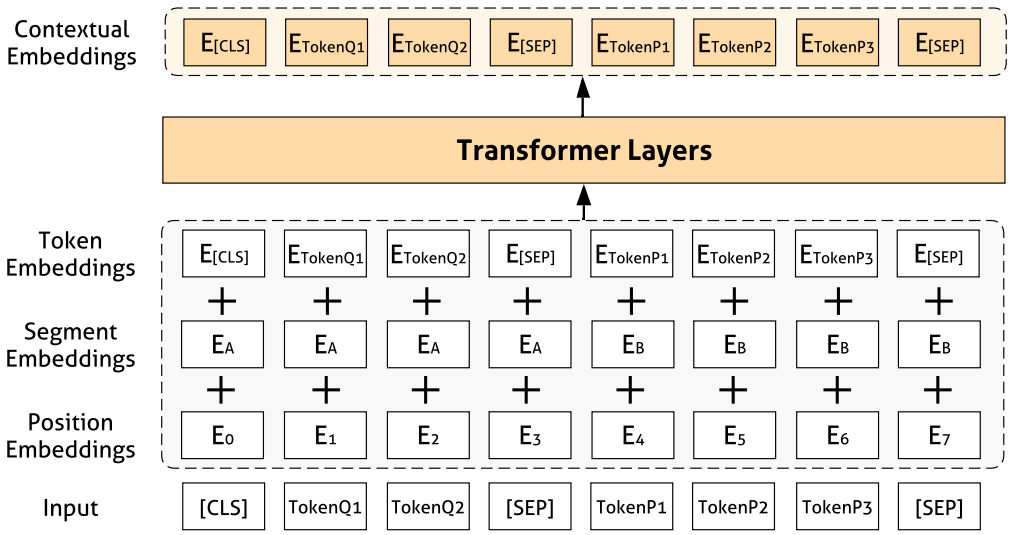


Figure 2.3: Input representation for Machine Reading Comprehension (MRC) with Bidirectional Encoder Representations from Transformers (BERT; Devlin et al. 2019). Question tokens (Q1-2) and passage tokens (P1-3) are separated with a special `[SEP]` token. To form the final input sequence, the input tokens are prepended with a `[CLS]` token and appended with a `[SEP]` token.

The final-layer hidden representations at the chosen positions are used to predict the original token with cross entropy loss which is then backpropagated for pretraining.

Figure 2.3 shows the input representation of BERT. BERT constructs the representation of an input token by summing its corresponding token, position, and segment embeddings. Identical to the Transformer model, the token and position embeddings indicate the meaning of a token and the position of a token in the input sequence, respectively. Additionally, segment embeddings are introduced to discriminate two input segments of different types. We show in Figure 2.3 an example for Machine Reading Comprehension (MRC). A typical MRC input consists of a question and a passage providing the context, and the objective is to find the answer span in the context. The BERT input for this task starts with a special class token `[CLS]`, followed by the concatenation of tokens from a question and a passage. To discriminate question and passage tokens, BERT adopts different segment embeddings for the two subsequences and separates them with another special token `[SEP]`.

UniLMv2 Different from BERT, UniLMv2 (Bao et al., 2020) aims to jointly learn to understand and generate language. To this end, UniLMv2 adopts Pseudo-Masked

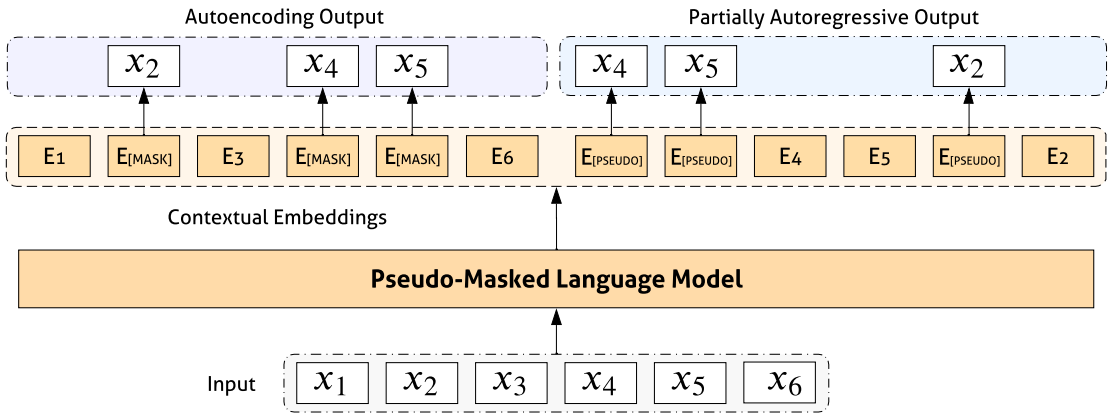


Figure 2.4: The Pseudo-Masked Language Model (PMLM; Bao et al. 2020) jointly optimized by two pretraining objectives: the autoencoding (AE) objective and the partially autoregressive (PAR) objective. Given the input token sequence $\{x_1, x_2, ..., x_6\}$, tokens $\{x_2, x_4, x_5\}$ are randomly masked with two types of special tokens: the conventional mask `[MASK]` and the pseudo mask `[PSEUDO]`. The masked tokens are jointly predicted for AE, while PAR follows a specific factorization order which is uniformly produced: tokens $\{x_4, x_5\}$ are jointly predicted conditioned on $\{x_1, x_3, x_6\}$, and then the prediction for x_2 is made conditioned on all the other tokens.

Language Modeling (PMLM) as its learning objective, which consists of bidirectional language modeling and sequence-to-sequence language modeling: the former employs an autoencoding (AE) objective identical to Devlin et al. (2019), while the latter is partially autoregressive (PAR) and decomposes the probability of masked tokens in input sequence X as:

$$p(x_F \mid x_{\backslash F}) = \prod_{i=1}^{|F|} \prod_{f \in F_i} p(x_f \mid x_{\backslash F_{\geq i}}) \tag{2.10}$$

where F is the uniformly-produced factorization order. The masked position set F_i at the ith factorization step can be either a token or a n-gram block. x_F is a set of x_{F_i}, and similarly, $x_{\backslash F}$ is a set of $x_{\backslash F_i}$. Figure 2.4 shows an example of how an input sequence for pretraining is constructed to compute the PMLM objective.

Similar to BERT, UniLMv2 also takes the Transformer model as the backbone network. In UniLMv2, different self-attention masks are used to control the context access for each token under different language modeling objectives. As a result, model parameters are shared across the two pretraining objectives, allowing UniLMv2 to efficiently perform the two types of language modeling in one forward pass.

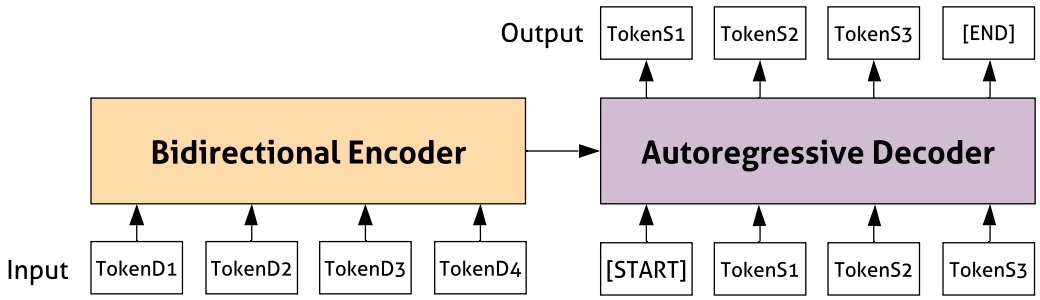


Figure 2.5: BART (Lewis et al., 2020) based on a neural encoder-decoder architecture and an autoencoding objective. During pretraining, the encoder represents a corrupted input sequence with bidirectional contexts, and the decoder aims to generate the original input sequence autoregressively.

BART BART (Lewis et al., 2020) is a denoising autoencoder for pre-training sequence-to-sequence models. Specifically, the pre-training objective is to map a corrupted document input to the original document it was derived from. Unlike existing denoising autoencoders (Yang et al., 2019b; Dong et al., 2019; Joshi et al., 2020), BART allows any type of document corruption, and is found to perform the best by (1) randomly shuffling the order of the original sentences and (2) masking text spans of arbitrary length.

As shown in Figure 2.5, BART adopts a standard Transformer-based encoder-decoder architecture (described in Section 2.1.2), with each decoding layer in BART attends over the hidden states in the final layer of its encoder. Cross-entropy loss is used as the construction loss between the decoder's output and the original document for pretraining.

2.2 Document Summarization

In this section, we describe the background of generic summarization and query focused summarization (QFS). For each, we start with the problem formulation, and then describe existing systems for extractive and abstractive summarization. Extractive summaries consist of salient segments, e.g., sentences, that are taken from the original document set, while abstractive summaries can contain novel words and phrases that are not in the input.

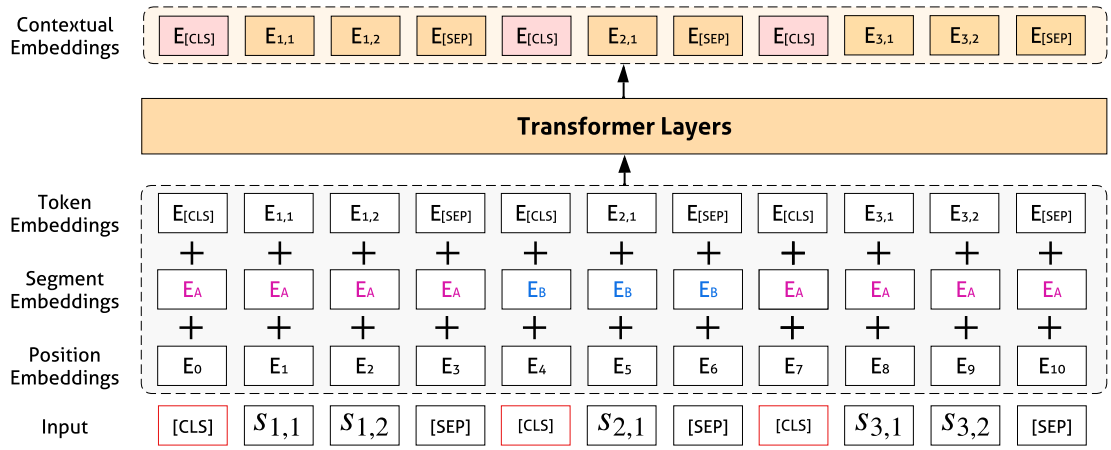


Figure 2.6: Architecture of BERTSUM (Liu and Lapata, 2019b). In the input sequence, $s_{i,j}$ denotes the jth token in the ith document sentence. Compared to the original Bert model (see Figure 2.3), BERTSUM inserts an additional `[CLS]` token (illustrated in red border color) before each input sentence and uses interval segmentation embeddings (illustrated in pink and blue font) to distinguish sentences. The contextual representations of the `[CLS]` tokens are used for predicting which sentences should be included in the summary.

2.2.1 Generic Summarization

Problem Formulation Let $\{(\mathcal{D}, S)\}$ denote a generic summarization dataset where $\mathcal{D} = \{D_1, D_2, \ldots, D_{|\mathcal{D}|}\}$ is a collection of documents with corresponding summaries S. $|\mathcal{D}| = 1$ for single-document summarization (SDS) and $|\mathcal{D}| > 1$ for multi-document summarization (MDS). Without lack of generality, in this section, we will formulate the summarization task for a single document, and use $D = \{s_1, ..., s_N\}$ to denote the input document and its sentences. We will introduce different approaches that extend the applicability of SDS models to multiple documents in Chapter 4 and 5.

Neural Extractive Summarization Extractive summarization is defined as the task of selecting a subset of sentences $[\hat{s}_1, \hat{s}_2, ..., \hat{s}_{N'}]$ in D as summary sentences, where $\hat{s}_j \in D, N' < N$. There are usually two basic components in extractive summarization: one for sentence representation, and the other for sentence selection based on the representations.

Much early work in document summarization has focused on effective neural architectures for sentence representation, such as sentence-level Convolutional Neural Networks (CNNs; Yin and Pei 2015), sentence-level CNNs augmented with document-

level Recurrent Neural Networks (RNNs; Cheng and Lapata 2016), and RNNs for both sentence- and document-level context modeling (Nallapati et al., 2017). More recently, pretrained Tranformers have been employed to construct more effective contextual representations for sentences and BERTSUM (Liu and Lapata, 2019b) was among the first to apply BERT to text summarization. As shown in Figure 2.6, BERTSUM extends the original BERT architecture with interval segment embeddings and makes sentence-level predictions with the special token `[CLS]` inserted before each sentence. BERTSUM achieves strong performance on three news summarization benchmarks without complex mechanisms such as Reinforcement Learning (Narayan et al., 2018b; Dong et al., 2018), showing the importance of sentence representations for document summarization.

Sentence selection, on the other hand, takes sentence representations as input, and decides which ones should be included in the final summary considering evaluation criteria such as redundancy and coverage. The predictions can be made with an autoregressive architecture (Narayan et al., 2018b) that conditions on previously selected sentences $\hat{s}_{<j}$ for the prediction of the jth sentence. Alternatively, the task can be formulated as a sequence labeling problem (Cheng and Lapata, 2016; Nallapati et al., 2017; Liu and Lapata, 2019b). In this case, binary labels are used to denote whether sentences should be included in the summary, and are estimated for all input sentences at once.

Apart from the two basic components, recent work has also explored techniques for post-processing extracted sentences, including sentence compression (Xu and Durrett, 2019) and summary ranking (Zhong et al., 2020), to further improve the quality of extractive summaries.

Neural Abstractive Summarization As mentioned in Section 2.1.2, abstractive summarization is typically seen as a sequence-to-sequence problem, handled with the encoder-decoder neural architecture. In abstractive document summarization, the goal is to generate S, a sequence of summary words, conditioned on D, its corresponding document words via modeling the conditional probability distribution $p(S|D)$. In the encoder-decoder architecture (described in Section 2.1.2), an encoder is employed to encode D into a sequence of continuous vector representations, from which a decoder then generates the summary sequence autoregressively.

Rush et al. (2015) and Nallapati et al. (2016) were among the first to apply the neural encoder-decoder architecture to abstractive summarization. See et al. (2017)

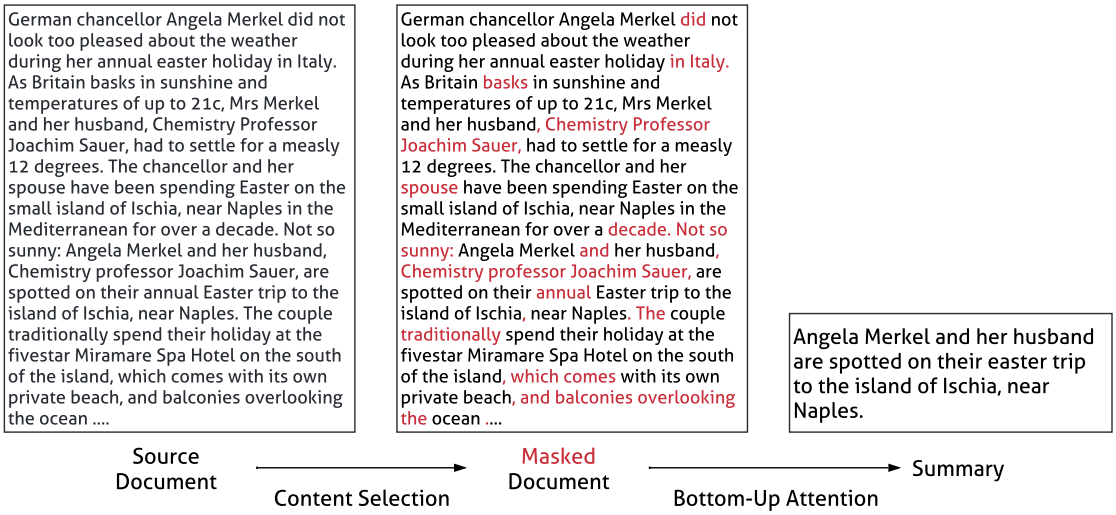


Figure 2.7: Overview of the bottom-up abstractive summarization (Gehrmann et al., 2018). A content selector is trained separately with a word tagging objective, and then applied to the input document at test time to generate a document mask. The document mask restricts the copy mechanism from accessing words that are not selected to be part of the summary during decoding.

enhance their approach with a pointer-generator model, essentially a copy mechanism allowing words from the source document to be copied directly in the summary. Gehrmann et al. (2018) incorporate a content selection model which decides on relevant aspects of the source document. Figure 2.7 shows an overview of this two-step summarization process. They frame the content selection task as a word-level tagging problem, with the objective of separately identifying tokens from a document that should be part of its summary; at test time, their model produces content selection probabilities for each word, which are then used to restrict the copy mechanism by performing hard masking over the input document. Recently, pretrained models have also shown their effectiveness in abstractive summarization. To adapt BERT to summary generation, Liu and Lapata (2019b) combine an encoder which is initialized with a pretrained BERT model (see Figure 2.6 for details), and a randomly initialized decoder which is optimized from scratch. To address the mismatch between learning a pretrained encoder and a random decoder, much recent work has proposed to pretrain a decoder together with an encoder in a sequence-to-sequence framework, which further improves the quality of abstractive summaries (Bao et al., 2020; Lewis et al., 2020; Raffel et al., 2020; Zhang et al., 2020; Zou et al., 2020).

Another line of research controls summary generation via topics (Perez-Beltrachini

et al., 2019; Wang et al., 2020b), retrieve-and-edit methods (Cao et al., 2018), or factual relations (Jin et al., 2020). More recently, Dou et al. (2021) propose various types of external guidance to control the summary content such as keywords, relational triples, or preselected source sentences, and develop GSUM, a general framework for guided summarization. GSUM extends the original BART model (Lewis et al., 2020) with a guidance encoder, and contains an additional cross-attention layer in the decoder to incorporate guidance information. They experimentally demonstrate that extracting a subset of important sentences from the source document provides the best guidance for summary decoding.

2.2.2 Query Focused Summarization

Problem Formulation Query focused summarization aims to create a short summary for a set of documents that answers a specific query. Formally, we denote a QFS dataset as $\{(\mathcal{D}, Q, S)\}$, where $\mathcal{D}$ is a document set, Q is a query that specifies information requests, and S is a short text that summarizes important information in $\mathcal{D}$ while answering Q. In Document Understanding Conferences (DUC; Dang 2005) benchmarks, the query Q consists of a short title (e.g., *Amnesty International*), and a narrative which is longer and more detailed (e.g., *What is the scope of operations of Amnesty International and what are the international reactions to its activities?*). However, the task is not restricted to such queries (Baumel et al., 2016; Nema et al., 2017; Zhu et al., 2019), as introduced in Section 1.3: in reality, queries can be expressed as a question (e.g., *Is euthanasia better than withdrawing life support?*), a phrase (e.g., *Alzheimers Disease*), or even a few keywords (e.g., *Marina Beach, Incidents*).

Extractive QFS Existing research on query-focused summarization largely lies on extractive approaches, where systems select the sentences from $\mathcal{D}$ which are most relevant to the query Q for inclusion in the summary.

In the previous chapter, Figure 1.1(b) illustrated classic centrality-based approaches which have generally shown strong performance in QFS (Wan, 2008). Under this framework, query relevance is first calculated for all sentences within the input document(s), and is then considered in estimating centrality for the input sentences which are viewed as nodes in a graph. A variety of approaches have been proposed to enhance the way relevance and centrality are estimated including adopting manifold-ranking algorithms (Wan et al., 2007; Wan and Xiao, 2009), incorporating topic-sensitive in-

formation (Badrinath et al., 2011), and considering information certainty of candidate sentences (Wan and Zhang, 2014). To mitigate the mismatch between queries and document sentences, Nastase (2008) employs Wikipedia as an external knowledge source and expands queries with related concepts.

More recently, Li et al. (2015) estimate the salience of text units within a sparse-coding framework by additionally taking into account reader comments (associated with news reports). Li et al. (2017a) use a cascaded neural attention model to find salient sentences, whereas in follow-on work Li et al. (2017b) employ a generative model which maps sentences to a latent semantic space while a reconstruction model estimates sentence salience. Specifically, the generative model uses Variational Auto-encoders (VAEs; Kingma and Welling 2014) to represent observed sentences with term representations $\mathbf{X}$ and latent semantic representations $\mathbf{Z}$. The reconstruction model aims to reconstruct these two types of sentence representations $\{\mathbf{X}, \mathbf{Z}\}$ jointly, using several parameterized vectors to represent different latent aspects of a topic. Sentence salience is then estimated from the attention scores in the reconstruction model. Finally, an Integer Linear Programming (ILP) framework is employed to select salient noun phrases (NPs) and verb phrases (VPs) from the constituency trees of salient sentences to produce the summary. Despite the differences in the actual model design, most recent work proposes to learn neural networks from a reconstruction objective for unsupervised extractive QFS.

Abstractive QFS Similar to abstractive systems for generic summarization, abstractive QFS is also formulated as a sequence-to-sequence problem, where the input sequence consists of the semantics of a document set and a query, and the output sequence is decoded from the encoded context representations in an autoregressive manner (Nema et al., 2017).

Abstractive QFS has received significantly less attention. Abstractive summarization models are known to be particularly data-hungry (Lebanoff et al., 2018) due to the challenging nature of the generation task: compared to its extractive counterpart, an abstractive system should learn from data how to perform text abstraction, including paraphrasing, generalization and sentence fusion (Jing and McKeown, 1999). This usually leads to a more complex model design, e.g., an encoder-decoder neural architecture (Sutskever et al., 2014), and therefore, system performance relies heavily on the size of training data available (Liu and Lapata, 2019a). In the case of QFS, the scarcity of training data makes the abstractive task even more challenging, as QFS ad-

ditionally requires the query semantics to be modeled and the generated abstracts to be query focused.

Recently, however, the increasing availability of pretrained models has prompted the development of pipeline-style frameworks for QFS which use resources from a wider range of NLP tasks. For example, Su et al. (2020) fine-tune BART (Lewis et al., 2020) on CNN/DailyMail (Hermann et al., 2015), a generic, single-document summarization dataset, and generate abstracts for multi-document QFS by iteratively summarizing paragraphs to a budget. For paragraph selection, they learn a QA module based on a plethora of QA and machine reading datasets. Specifically, the QA module in their work is an ensemble of two QA models: HLTC-MRQA (Su et al., 2019) and BioBERT (Lee et al., 2020). With XLNet (Yang et al., 2019b) as its backbone, HLTC-MRQA is fine-tuned on six QA datasets via multi-task learning for generalizable language representations across QA tasks. To access rich domain knowledge, Su et al. (2020) further fine-tune BioBERT (Lee et al., 2020), a pretrained language model for biomedical text mining, on SQuAD, a reading comprehension dataset proposed in Rajpurkar et al. (2016). The QA module combines these two models and ranks paragraphs per their answer relevance to the query. Iteratively, each of the top k paragraphs is input to BART fine-tuned for summarization, and the output paragraph-level summaries are concatenated to form the final cluster-level summary.

Similarly, Laskar et al. (2020b) fine-tune BERT (Devlin et al., 2019) on CNN/Daily Mail. To reduce labeling efforts, they propose a three-stage system which creates additional weak supervision using supervision from QFS data (typically reserved for evaluation) and related QA and paraphrase identification tasks. In its supervised framework, two years' DUC datasets are used for training and one for testing. As the first step, a pseudo reference summary for each DUC document in the training set is created. Specifically, they first extract query-relevant sentences with a QA model optimized on the Microsoft Machine Reading Comprehension Dataset (MS-MARCO; Bajaj et al. 2016), a large-scale QA dataset consisting of questions generated from real anonymized Bing user logs. Then they replace some of these extracted sentences with reference summary sentences via a paraphrase model optimized on the Microsoft Research Paraphrase Corpus (MRPC; Dolan and Brockett 2005) which contains 5801 pairs of sentences, each manually labeled with a binary judgment indicating whether the sentence pair constitutes a paraphrase. The next step is to use these document-level pseudo summaries to further fine-tune BERTSUM (Liu and Lapata, 2019b). During fine-tuning, the query is prepended to the source document to form the input sequence.

The fine-tuned BERTSUM can therefore generate query-focused single-document summaries for documents in each test cluster. The last step aims to merge the multiple single-document summaries generated for each cluster into one multi-document summary: all sentences in the generated summaries are re-ranked by the paraphrase model, and the top ranked sentences (which are expected to be more query-relevant) are selected to form the final multi-document summary.

2.3 Summary

In this chapter, we introduced the basics of the Transformer model, one of the most widely adopted neural networks in natural language processing. Specifically, we introduced its application in the encoder-decoder architecture and pretrained models. We also described the tasks of generic and query focused document summarization. For each of these tasks, we provided the problem formulation, and introduced previous work on extractive and abstractive summarization. In the next chapter, we will discuss how to leverage distant training resources to learn a neural query model, and effectively incorporate it into an extractive system for query focused summarization.

Chapter 3

Coarse-to-Fine Query Focused Summarization

As most existing research in QFS has employed an extractive approach, in this chapter, we focus on extractive multi-document QFS. To facilitate query focused extraction from a cluster consisting of multiple documents, we consider the problem of how to improve the modeling of query-cluster interactions. Due to the lack of training data, existing work relies heavily on retrieval-style methods for assembling query relevant summaries. In this chapter, we propose a *coarse-to-fine* modeling framework which employs progressively more accurate modules for estimating whether text segments are relevant, likely to contain an answer, and central. The modules can be independently developed and leverage training data if available. We present an instantiation of this framework with a trained *evidence* estimator which relies on distant supervision from question answering where various resources exist to identify segments which are likely to answer the query and should be included in the summary. Our framework is robust across domains and query types (i.e., long vs short) and outperforms strong comparison systems on benchmark datasets.

3.1 Introduction

As introduced in Chapter 1, multi-document QFS (Dang, 2005) aims to create a short summary from a set of documents that answers a specific query. It has various applications in personalized information retrieval and recommendation engines where search results can be tailored to an information need. For instance, a user might be looking for an overview summary or a more detailed one which would allow them to answer a

specific question. As existing QFS research is dominated by extractive systems (Wan et al., 2007; Nastase, 2008; Baumel et al., 2016; Li et al., 2017b), we aim at building a more effective extractive system for multi-document QFS in this chapter.

Deep neural network models have made significant progress in single-document generic summarization (Nallapati et al., 2016; Paulus et al., 2018; Li et al., 2017b; See et al., 2017; Narayan et al., 2018b; Gehrmann et al., 2018), while multi-document QFS has been relatively neglected, partially due to the paucity of large-scale training data for the application of learning methods. The CNN/DailyMail dataset (Hermann et al., 2015) and the NYT dataset (Sandhaus, 2008) are two widely-used single-document generic summarization, which contain 312,085 and 110,540 samples, respectively. On the other hand, high-quality multi-document QFS datasets, i.e., document clusters paired with multiple human-written summaries, have been produced for the Document Understanding Conferences (DUC), but are relatively small, i.e., around 50 samples, for optimizing deep neural networks. Besides, the size and number of source documents which can be very large also makes it challenging to apply end-to-end neural models to the multi-document setting. As a result, the two basic assumptions which underlie in single-document generic summarization may not be realistic for multi-document QFS: (a) human-annotated training data for millions of samples across different domains and languages is accessible, or can be potentially created with relatively low cost, and (b) a neural network model can be trained to encode the whole input into vectors, without being constrained by the memory of current hardware.

In this chapter, we attempt to address these research challenges and propose a coarse-to-fine modeling framework for extractive QFS. Specifically, our proposed framework incorporates:

1. A *relevance* estimator for retrieving textual segments, e.g., sentences or longer passages, associated with a query.

2. An *evidence* estimator which further isolates segments likely to contain answers to the query.

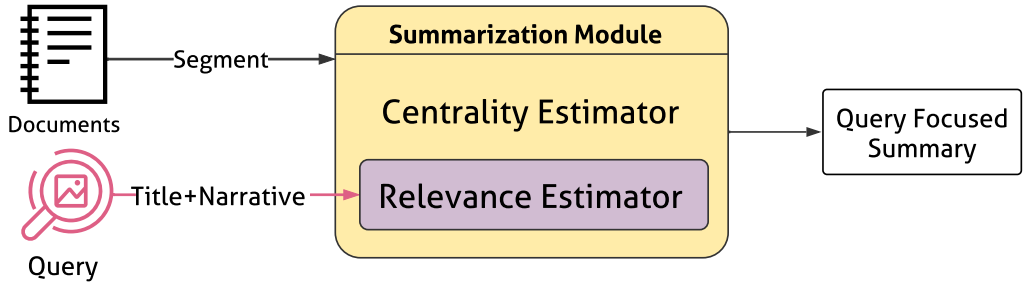
3. A *centrality* estimator which finally selects which segments to include in the summary.

The vast majority of previous work (Wan et al., 2007; Wan, 2008; Wan and Xiao, 2009; Wan and Zhang, 2014) creates summaries by ranking textual segments (usually

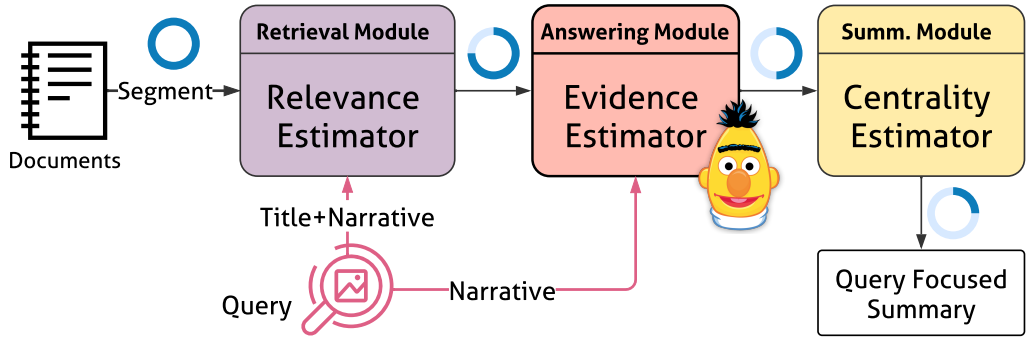
sentences) according to their relationship (e.g., similarity) to other segments *and* their relevance to the query. In other words, relevance and evidence estimation are subservient to estimating the centrality of a segment (e.g., with a graph-based model). We argue that disentangling these subtasks allows us to better model the query and specialize the summaries to specific questions or topics (Katragadda and Varma, 2009). A coarse-to-fine approach is also expedient from a computational perspective; at each step the model processes a decreasing number of segments (rather than entire documents), and as a result is insensitive to the original input size and more scalable.

Our key insight is to treat evidence estimation as a question answering task where a cluster of potentially relevant documents provides support for answering a query (Baumel et al., 2016). Advantageously, we are able to train the evidence estimator on existing large-scale question answering datasets (Rajpurkar et al., 2016; Joshi et al., 2017; Yang et al., 2018), alleviating the data paucity problem in QFS. Existing QFS systems (Wan et al., 2007; Wan, 2008; Wan and Xiao, 2009; Wan and Zhang, 2014) employ classic retrieval techniques (such as TF-IDF) to estimate the affinity between query-sentence pairs. Such techniques can handle short keyword queries, but are less appropriate in QFS settings where query narratives can be long and complex. We argue that a trained evidence estimator might be better at performing *semantic matching* (Guo et al., 2016) between queries and document segments. To this effect, we experiment with two popular QA settings, namely answer sentence selection (Heilman and Smith, 2010; Yang et al., 2015) and machine reading comprehension Rajpurkar et al. (2016) which operates over passages than isolated sentences. In both cases, our evidence estimators take advantage of powerful pre-trained encoders such as BERT (Devlin et al., 2019), to better capture semantic interactions between queries and text units.

Our contributions in this work are threefold: (a) we propose a coarse-to-fine model for QFS which we argue allows to introduce trainable components taking advantage of existing datasets and pre-trained models; (b) we capitalize on the connections of QFS with question answering and propose different ways to effectively estimate the query-segment relationship; and (c) we provide experimental results on several benchmarks which show that our model consistently outperforms strong comparison systems across domains (news articles vs. medical text) and query types (long narratives vs. keywords).



(a) Classic Framework for Extractive QFS



(b) Coarse-to-Fine Framework for Extractive QFS

Figure 3.1: Classic (a) and proposed framework (b) for query focused summarization. The classic approach involves a relevance estimator nested within a summarization module while our framework takes document clusters as input, and *sequentially* processes them with three individual modules (relevance, evidence, and centrality estimators). The blue circles indicate a coarse-to-fine estimation process from original articles to final summaries where modules gradually discard segments (i.e., sentences or passages). With regard to evidence estimation, we adopt pretrained BERT (Devlin et al., 2019) which is further fine-tuned with distant signals from question answering.

3.2 Related Work

Existing research on query-focused multi-document summarization largely lies on extractive approaches, where systems usually take as input a set of documents and select the sentences most relevant to the query for inclusion in the summary. In the previous chapters, we introduced existing approaches for extractive QFS, and presented a detailed description of the classic graph-based QFS framework in Figure 1.1(b). We further provide a sketch of these centrality-based approaches which involve a relevance estimator subservient to a centrality estimator in Figure 3.1(a).

In contrast to previous work, our proposal does not simultaneously perform seg-

ment selection and query matching. We introduce a coarse-to-fine approach that incorporates progressively more accurate components for selecting segments to include in the summary, making model performance relatively insensitive to the number and size of input documents.

Drawing inspiration from recent work on QA, we take advantage of existing datasets in order to reliably estimate the relationship between the query and candidate segments. We focus on the following two QA subtasks which have attracted considerable attention in the literature:

- **Answer sentence selection**: The goal of answer sentence selection is to extract answers from a set of pre-selected sentences. As one of the initial efforts in this research direction, Wang et al. (2007) first collected TrecQA from TREC QA 8-13. TrecQA includes editor-generated questions and candidate answer sentences selected by matching content words in the question. Since its inception, TrecQA has sparked follow-on work (Heilman and Smith, 2010; Yao et al., 2013) and has become a commonly-used QA benchmark. Yang et al. (2015) further constructed WikiQA from Bing query logs which is more than an order of magnitude larger than the previous dataset. Compared to TrecQA, WikiQA includes questions for which there are no correct sentences, and also relaxes the assumption that the answer sentence has to share some content with the question.

- **Machine reading comprehension**: Reading comprehension is a QA task that aims at answering a question after processing a short text passage. The Stanford Question Answering Dataset (SQuAD; Rajpurkar et al. 2016) is one of the most widely-cited reading comprehension benchmarks, created to promote research in reading comprehension. SQuAD 1.0 consists of questions composed by crowdworkers on Wikipedia articles, and their answers in the format of a segment of text, i.e., span, from the corresponding reading passage. In contrast to prior datasets for answer sentence selection, SQuAD does not provide a list of candidate answer sentences, and systems need to cope with all possible spans in the context as candidates. SQuAD 2.0 (Rajpurkar et al., 2018) further extends the dataset with over 50,000 unanswerable questions adversarially created by crowdworkers. Apart from its standard formulation set by SQuAD, research tasks in reading comprehension are many and varied, such as multi-hop (Welbl et al., 2018; Yang et al., 2018), open-domain (Wang et al., 2019; Qi et al., 2019) and conversational (Saeidi et al., 2018; Reddy et al., 2019) reading comprehen-

sion. As an initial effort on leveraging QA resources for QFS, in this chapter, we will describe how to adapt a standard reading comprehension model (optimized on SQuAD 2.0) to extract query focused summaries from multiple documents.

We note that QA and QFS are related but ultimately different tasks. QA aims at finding the *best* answer in a span or sentence, while QFS extracts a *set* of sentences based on user preferences and the content of the input documents under a length budget (Wan, 2008; Wan and Zhang, 2014). QA questions are often short and fact-based while QFS narratives can be longer and more complex (see the example in Section 3.3) and as a result simply localizing an answer within a cluster is not optimal.

3.3 Problem Formulation

Let Q denote an information request and $\mathcal{D} = \{D_1, D_2, \ldots, D_{|\mathcal{D}|}\}$ a set of topic-related documents. It is often assumed (e.g., in DUC competitions) that Q consists of a short title (e.g., *Amnesty International*) highlighting the topic of interest, and a query narrative which is considerably longer and detailed (e.g., *What is the scope of operations of Amnesty International and what are the international reactions to its activities?*).

We illustrate our proposed framework in Figure 3.1(b). We first decompose documents into segments, i.e., passages or sentences, and retrieve those which are most relevant to query Q (Relevance Estimator). Then, a trained estimator quantifies the semantic match between selected segments and the query (Evidence Estimator) to further isolate segments for consideration in the output summary (Centrality Estimator). We propose two variants of our evidence estimator; a context agnostic variant infers evidence scores over individual sentences, while a context aware one infers evidence scores for tokens within a passage which are further aggregated into sentence-level evidence. Passages might allow for semantic relations to be estimated more reliably since neighboring context is also taken into account.

3.3.1 Relevance Estimator

Our QFS system operates over documents within a cluster which we segment into sentences. The latter serve as input to the context agnostic evidence estimator. For the context aware variant, we obtain passages with a sliding window over continuous sentences in the same document.

During inference, we first retrieve the top k^{IR} answer candidates (i.e., sentences or passages) which are subsequently processed by our evidence estimator. We do this following an *adaptive* method that allows for a variable number of segments to be selected for each query. Specifically, for the ith query-cluster pair, we first rank all segments in the cluster based on term frequency with respect to the query, and determine k_i^{IR} such that it reaches a fixed threshold $\theta \in [0,1]$. Formally, k_i^{IR}, the number of retrieved segments, is given by:

$$k_i^{\text{IR}} = \max_k \sum_{j=1}^{k} r_{i,j} < \theta \tag{3.1}$$

where $r_{i,j}$ is the relevance score for segment j (normalized over segments in the ith cluster). Although we adopt term frequency as our relevance estimator, there is nothing in our framework which precludes the use of more sophisticated retrieval methods (Dai and Callan, 2019; Akkalyoncu Yilmaz et al., 2019). We investigated approaches based on term frequency-inverse sentence frequency (Allan et al., 2003) and BM25 (Robertson et al., 2009), however, we empirically found that they are inferior, having a bias towards shorter segments which are potentially less informative for summarization.

3.3.2 Evidence Estimator

We argue that relevance matching is not sufficient to capture the semantics expressed in the query narrative and its relationship to the documents in the cluster. We therefore leverage distant supervision signals from existing QA datasets to train our evidence estimator and use the trained estimator to rerank answer candidates selected from the retrieval module. For the ith cluster, we select the top $\min\{k^{\text{QA}}, k_i^{\text{IR}}\}$ candidates as answer evidence (where k^{QA} is tuned on the development set).

Sentence Selection Let Q denote a sequence of query tokens and $\{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_N\}$ the set of candidate answers (also token sequences) obtained from the retrieval module. Our learning objective is to find the correct answer(s) within this set. We concatenate query Q and candidate sentence $\mathcal{S}$ into a sequence `[CLS]`, Q, `[SEP]`, $\mathcal{S}$, `[SEP]` to serve as input to a BERT encoder (we pad each sequence in a minibatch of L tokens). The `[CLS]` vector $\mathbf{t}_i$ serves as input to a single layer neural network to obtain the distribution over positive and negative classes:

$$\mathbf{p}_0^{(i)} = \frac{1}{Z}\exp\left(\mathbf{t}_i^{\intercal}\mathbf{W}_{:,0}\right), \mathbf{p}_1^{(i)} = \frac{1}{Z}\exp\left(\mathbf{t}_i^{\intercal}\mathbf{W}_{:,1}\right) \tag{3.2}$$

where $Z = \sum_c \exp\left(\mathbf{t}_i^T \mathbf{W}_{:,c}\right)$ and matrix $\mathbf{W} \in \mathbb{R}^{d\times 2}$ is a learnable parameter. We use a cross entropy loss where 1 denotes that a sentence contains the answer (and 0 otherwise):

$$\mathcal{L} = -\sum_{i=1}^{N} (y \log \mathbf{p}_1^{(i)} + (1-y) \log \mathbf{p}_0^{(i)}). \tag{3.3}$$

We treat the probability of the positive class as evidence score $q = \mathbf{p}_1^{(i)} \in (0,1)$ and use it to rank all retrieved segments for each query.

Span Selection A span selection model allows us to capture more faithfully the answer, its local context and their interactions. Again, let $\mathcal{Q}$ denote a query token sequence and $\mathcal{P}$ a passage token sequence. Our training objective is to find the correct answer span in $\mathcal{P}$. Similar to sentence selection, we concatenate the query $\mathcal{Q}$ and the passage $\mathcal{P}$ into a sequence `[CLS]`, $\mathcal{Q}$, `[SEP]`, $\mathcal{P}$, `[SEP]` and pad it to serve as input to a BERT encoder. Let $\mathbf{T} = [\mathbf{t}_i]_{i=1}^N$ denote the contextualized vector representation of the entire sequence obtained from BERT. We feed $\mathbf{T}$ into two separate dense layers to predict probabilities $p_{\mathcal{S}}$ and $p_{\mathcal{E}}$:

$$\mathbf{p}_{\mathcal{S}}^{(i)} = \frac{\exp\left(\mathbf{t}_i^\intercal \mathbf{w}_{\mathcal{S}}\right)}{\sum_j \exp\left(\mathbf{t}_j^\intercal \mathbf{w}_{\mathcal{S}}\right)} \tag{3.4}$$

$$\mathbf{p}_{\mathcal{E}}^{(i)} = \frac{\exp\left(\mathbf{t}_i^\intercal \mathbf{w}_{\mathcal{E}}\right)}{\sum_j \exp\left(\mathbf{t}_j^\intercal \mathbf{w}_{\mathcal{E}}\right)} \tag{3.5}$$

where $\mathbf{w}_{\mathcal{S}}$ and $\mathbf{w}_{\mathcal{E}}$ are two learnable vectors denoting the beginning and end of the (answer) span, respectively. During training we optimize the log-likelihood of the correct start and end positions. For passages without any correct answers, we set these to 0 and default to the `[CLS]` position.

At inference time, to allow comparison of results across passages, we remove the final softmax layer over different answer spans. Specifically, we first calculate the (unnormalized) start and end scores for all tokens in a sequence:

$$\mathbf{u} = \exp\left(\mathbf{T}\mathbf{w}_{\mathcal{S}}\right), \mathbf{v} = \exp\left(\mathbf{T}\mathbf{w}_{\mathcal{E}}\right). \tag{3.6}$$

And collect sentence scores from token scores as follows. For each sentence starting at token i and ending at token j, we obtain score matrix $\mathbf{Q}$ via:

$$\tilde{\mathbf{Q}} = \left(\mathbf{u}_{[i:j]} \mathbf{v}_{[i:j]}^\intercal \mathbf{A}\right)^{\frac{1}{2}} \tag{3.7}$$

$$\mathbf{Q} = \tanh(\tilde{\mathbf{Q}}) \tag{3.8}$$

where we collect all possible span scores within a sentence in matrix $\mathbf{S}$ where $\mathbf{S}_{i',j'}$ denotes the span score from token i' to token j' ($i \leq i' < j' \leq j$). Matrix $\mathbf{A}$ is an upper triangular matrix masking all illegitimate spans whose end comes before the start. The tanh function scales the magnitude of extreme scores (e.g., scores over 100 or under 0.01), as a means of reducing the variance of $\tilde{\mathbf{Q}}$. And finally, we use max pooling to obtain a scalar score q:

$$q = \text{max-pool}(\mathbf{Q}) \in (0,1). \tag{3.9}$$

It is possible to produce multiple evidence scores for the same sentence since we use overlapping passages; we select the score with the highest value in this case.

Ensemble Selection We can also build an ensemble by linearly interpolating evidence scores from the two estimators based on sentence selection and span extraction. Let $(\mathcal{E}^{\mathcal{S}}, q^{\mathcal{S}})$ and $(\mathcal{E}^{\mathcal{P}}, q^{\mathcal{P}})$ denote the selected sentence sets and their evidence scores produced by the sentence selection estimator and span extraction estimator, respectively. We obtain the ensemble score for sentence e via:

$$q_e = \begin{cases} \mu * q_e^{\mathcal{S}} + (1-\mu) * q_e^{\mathcal{P}} & e \in \mathcal{E}^{\mathcal{S}} \cap \mathcal{E}^{\mathcal{P}} \\ \mu * q_e^{\mathcal{S}} & e \in \mathcal{E}^{\mathcal{S}} \wedge e \notin \mathcal{E}^{\mathcal{P}} \\ -\infty & e \notin \mathcal{E}^{\mathcal{S}} \end{cases} \tag{3.10}$$

where the coefficient was set to $\mu = 0.9$.

3.3.3 Centrality Estimator

Graph Construction Inspired by Wan (2008), we introduce as our centrality estimator an extension of the well-known LexRank algorithm (Erkan and Radev, 2004), which we modify to incorporate the evidence estimator introduced in the previous section.

For each document cluster, LexRank builds a graph $G = (\mathcal{V}, \mathcal{E})$ with nodes $\mathcal{V}$ corresponding to sentences and (undirected) edges $\mathcal{E}$ whose weights are computed based on similarity. Specifically, matrix $\mathbf{E}$ represents edge weights where each element $\mathbf{E}_{i,j}$ corresponds to the transition probability from vertex i to vertex j. The original LexRank algorithm uses TF-IDF (Term Frequency Inverse Document Frequency) to measure similarity; since our framework operates over sentences rather than "documents", we use TF-ISF (Term Frequency Inverse Sentence Frequency), with ISF

defined as:

$$\text{ISF}(w) = 1 + \log(|C|/\text{SF}(w)) \tag{3.11}$$

where $|C|$ is the total number of sentences in the cluster, and $\text{SF}(w)$ is the number of sentences in which w occurs.

We integrate our evidence estimator into the original transition matrix as:

$$\tilde{\mathbf{E}} = \phi * [\tilde{\mathbf{q}}; ...; \tilde{\mathbf{q}}] + (1 - \phi) * \mathbf{E} \tag{3.12}$$

where $\phi \in (0, 1)$ controls the extent to which query-specific information influences sentence selection for the summarization task; and $\tilde{\mathbf{q}}$ is a distributional evidence vector which we obtain after normalizing the evidence scores $\mathbf{q} \in \mathbb{R}^{1 \times |V|}$ obtained from the previous module ($\tilde{\mathbf{q}} = \mathbf{q} / \sum_{v}^{|V|} \mathbf{q}_v$).

Summary Generation In order to decide which sentences to include in the summary, a node's centrality is measured using a graph-based ranking algorithm (Erkan and Radev, 2004; Xu and Lapata, 2019). Specifically, we run a Markov chain with $\tilde{\mathbf{E}}$ on G until it converges to stationary distribution $\mathbf{e}^*$ where each element denotes the salience of a sentence. In the proposed algorithm, $\mathbf{e}^*$ jointly expresses the importance of a sentence in the document *and* its semantic relation to the query as modulated by the evidence estimator and controlled by ϕ. We rank sentences according to $\mathbf{e}^*$ and select the top k^{Sum} ones, subject to a budget (e.g., 250 words).

To reduce redundancy, we apply the diversity algorithm proposed in Wan (2008) which iteratively penalizes the salience of sentences according to their similarities with those already selected to appear in the summary. We also remove the sentences which have high cosine similarities (i.e., ≥ 0.6) with any sentence already included in the summary (Cao et al., 2015; Angelidis and Lapata, 2018).

3.4 Experimental Setup

3.4.1 Summarization Datasets

We performed QFS experiments on the DUC 2005-2007 benchmarks and the **T**opically **D**iverse QFS dataset (TD-QFS; Baumel et al. 2016). DUC benchmarks contain long query narratives over 50 clusters with 32–25 documents each, and cover multiple domains. TD-QFS focuses on medical texts, contains short keyword queries over 4 clusters with 185 documents each. As a result, TD-QFS clusters are less topically concentrated, with larger amounts of query-irrelevant information (Baumel et al., 2016).

	DUC			
Dataset	2005	2006	2007	TD-QFS
Domain	Cross	Cross	Cross	Medical
Query Narrative	Long	Long	Long	Short
#Clusters	50	50	45	4
#Queries/Cluster	1	1	1	10
#Documents/Cluster	32	25	25	185
#Summaries/Query	4-9	4	4	3
#Words/Summary	250	250	250	250

Table 3.1: Multi-document QFS dataset statistics. DUC benchmarks span over three DUC years: 2005, 2006 and 2007. DUC benchmarks contain long query narratives and cross-domain news articles, while TD-QFS focus on short queries and medical texts.

Although our approach is motivated by the desire to better model long and complex queries, experiments on TD-QFS examine whether it generalizes to out-of-domain queries and clusters. We used DUC 2005 as a development set to optimize hyper-parameters and evaluated performance on DUC 2006-2007 and TD-QFS. A summary of the characteristics of these datasets is provided in Table 3.1, and examples are shown in Table 3.2.

We used three datasets for training our evidence estimator, including WikiQA (Yang et al., 2015), TrecQA (Yao et al., 2013), and SQuAD 2.0 (Rajpurkar et al., 2018). WikiQA and TrecQA are benchmarks for answer sentence selection while SQuAD 2.0 is a popular machine reading comprehension dataset (which we used for span selection). Compared to SQuAD, WikiQA and TrecQA are smaller and we therefore integrate them for model training (Yang et al., 2019a). We show statistics for QA datasets in Table 3.3 and examples in Table 3.4.

3.4.2 Implementation Details

We used the publicly released BERT model[1] and fine-tuned it on our QA tasks with 4 GTX 1080TI GPUs with 11GB memory. Considering the maximum input length BERT allows (512 tokens) and the query narrative (which in DUC is fairly long), we set the maximum passage size to 8 sentences (with maximum sentence length of 50 tokens). To ensure all sentences are properly contextualized, we used a stride size of

[1] https://github.com/huggingface/pytorch-transformers

DUC

Query: INTERNATIONAL ORGANIZED CRIME – *Identify and describe types of organized crime that crosses borders or involves more than one country. Name the countries involved. Also identify the perpetrators involved with each type of crime, including both individuals and organizations if possible.*

Summary: The main types of international organized crime are drug trafficking and drug money laundering. Major players in these activities are the Colombian Medellin and Cali cartels, which dominate the world cocaine trade. Also involved are the Mexican Sinoloa cartels and, in the US, Los Angeles street gangs allied with Colombian cartels. Former Panamanian leader, General Manuel Noriega, was convicted of drug trafficking, money laundering, and conspiring with the Medellin cartel. Cuban military officers have been involved in smuggling drugs, and Fidel Castro has been accused of mediating on behalf of the Medellin cartel. Other Central and South American countries involved in drug trafficking include Belize, Costa Rica, Guatemala, Honduras, Peru, and Bolivia. Drugs also are smuggled into the US through the Bahamas. In Western Europe, Italy's Sicilian Mafia, Cosa Nostra, and Camorra engage in drug trafficking and money laundering, in association with Colombian cartels. Italian organized crime deals in arms trafficking, as well. Russian crime syndicates in Eastern Europe work with the Italian Mafia and Colombian cartels to funnel drugs into the US. In Africa, crime syndicates deal in ivory, rhino horn, diamonds, arms, and drugs. Nigerian drug rings smuggle heroin and cannabis throughout the world. Chinese Triads and Japanese Yakuza work with crime syndicates in other countries. Other international organized crimes include cigarette smuggling between the US and Canada, illicit arms sales between Israel and Colombian cartels, heroin smuggling from Turkey and along the Afghan/Pakistan border, human smuggling of prostitutes in Italy and illegal Chinese immigrants in the US.

TD-QFS

Query: *Asthma Causes*

Summary: Asthma is a chronic disease that affects your airways. Your airways are tubes that carry air in and out of your lungs. If you have asthma, the inside walls of your airways become sore and swollen. That makes them very sensitive, and they may react strongly to things that you are allergic to or find irritating. When your airways react, they get narrower and your lungs get less air. This can cause wheezing, coughing, chest tightness and trouble breathing, especially early in the morning or at night. When your asthma symptoms become worse than usual, it's called an asthma attack. The exact cause of asthma isn't known, Different factors may be more likely to cause asthma in some people than in others.Researchers think some genetic and environmental factors interact to cause asthma, most often early in life. These factors include: an inherited tendency to develop allergies, called atopy (AT-o-pe), parents who have asthma, certain respiratory infections during childhood, contact with some airborne allergens or exposure to some viral infections in infancy or in early childhood when the immune system is developing. If you have asthma, you may react to just one trigger or you may find that several things act as triggers. Triggers are things that can cause asthma symptoms. If you have asthma, you may react to just one trigger or you may find that several things act as triggers. Some common triggers are: getting a cold or flu, pollen, dust and animals (especially cats), cold weather, smoking, exercise.

Table 3.2: Examples for DUC (Dang, 2005) and TD-QFS (Baumel et al., 2016). DUC queries consist of a TITLE and a *narrative* while TD-QFS has short queries in the medical domain. Only one reference summary is shown for each example, however, both datasets have multiple human-written reference summaries.

Dataset	Sentences			Spans
	WikiQA	TrecQA	Total	SQuAD
#Train	8,672	53,417	62,089	130,318
#Dev	1,130	1,148	2,278	11,872

Table 3.3: Question answering dataset statistics. We use the union of WikiQA and TrecQA for answer sentence selection and SQuAD for span selection.

4 sentences to create overlapping passages. For the answer sentence selection model, BERT was fine-tuned with a learning rate of 3×10^{-6} and a batch size of 16 for 3 epochs. For span selection, we adopted a learning rate of 3×10^{-5} and a batch size of 64 for 5 epochs.

During inference, the confidence threshold for the relevance estimator was set to $\theta = 0.75$ (Kratzwald and Feuerriegel, 2018) for both sentence and passage retrieval. For the evidence estimator, k^{QA} was tuned on the development set. We obtained 90 and 110 evidence sentences from the sentence selection and span selection models, respectively. For the centrality estimator, the influence of the query was set to $\phi = 0.15$ (Wan, 2008; Wan and Zhang, 2014).

The TD-QFS dataset used in this work is publicly available at `https://www.cs.bgu.ac.il/~talbau/TD-QFS/dataset.html`. DUC 2005-2007 datasets can be requested from NIST: `https://www-nlpir.nist.gov/projects/duc/data.html`. Our code is available at: `https://github.com/yumoxu/querysum`.

3.4.3 Evaluation Metrics

Following standard practice in DUC evaluations, we used ROUGE as our automatic evaluation metric[2] (Lin and Hovy, 2003). We report F1 for ROUGE-1 (unigram-based), ROUGE-2 (bigram-based), and ROUGE-SU4 (based on skip bigram with a maximum skip distance of 4).

We also evaluated model summaries in a judgment elicitation study via Amazon Mechanical Turk. Native English speakers (self-reported) were asked to rate query-summary pairs on: *Succinctness* (does the summary avoid unnecessary detail and redundant information?) and *Coherence* (does the summary make logical sense?). The ratings were obtained using a five point Likert scale. In addition, participants were

[2]We used `pyrouge` with the following parameter settings: ROUGE-1.5.5.pl -a -c 95 -m -n 2 -2 4 -u -p 0.5 -l 250.

Sentence Selection

Question: *What bird family is the owl?*

Candidate Sentences:

- Owls are a group of birds that belong to the order strigiformes, constituting 200 extant bird of prey species.
- Most are solitary and nocturnal, with some exceptions (e.g., the northern hawk owl).
- Owls hunt mostly small mammals, insects, and other birds, although a few species specialize in hunting fish.
- They are found in all regions of the earth except antarctica, most of greenland and some remote islands.
- Owls are characterized by their small beaks and wide faces, and are divided into two families: the typical owls, strigidae; and the barn-owls, tytonidae.

Span Selection (**answerable**)

Question: *By what main attribute are computational problems classified utilizing computational complexity theory?*

Context: Computational complexity theory is a branch of the theory of computation in theoretical computer science that focuses on classifying computational problems according to their inherent difficulty, and relating those classes to each other. A computational problem is understood to be a task that is in principle amenable to being solved by a computer, which is equivalent to stating that the problem may be solved by mechanical application of mathematical steps, such as an algorithm.

Answer: inherent difficulty

Span Selection (**unanswerable**)

Question: *What was the name of the 1937 treaty?*

Context: Other legislation followed, including the Migratory Bird Conservation Act of 1929, a 1937 treaty prohibiting the hunting of right and gray whales, and the Bald Eagle Protection Act of 1940. These later laws had a low cost to society: the species were relatively rare and little opposition was raised.

Plausible Answer: Bald Eagle Protection Act

Table 3.4: Examples for two types of question answering datasets for evidence estimation: answer sentence selection and span selection. Blue denotes answers while red denotes a plausible answer to the question that cannot be answered from the given context. We use the union of WikiQA (Yang et al., 2015) and TrecQA (Yao et al., 2013) for answer sentence selection and SQuAD 2.0 (Rajpurkar et al., 2018) for span selection. SQuAD 2.0 contains both answerable and unanswerable questions and we show one example for each of them.

	DUC 2006			DUC 2007		
Upper Bound and Baselines	R-1	R-2	R-SU4	R-1	R-2	R-SU4
GOLD	45.4	11.2	16.8	47.5	14.0	18.9
ORACLE	40.6	9.1	14.8	41.8	10.4	16.0
LEAD	32.1	5.3	10.4	33.4	6.5	11.3
Graph-based	R-1	R-2	R-SU4	R-1	R-2	R-SU4
LEXRANK	34.2	6.4	11.4	35.8	7.7	12.7
GRSUM	38.4*	7.0*	12.8*	42.0	10.3	15.6
CTSUM	—	—	—	42.6	10.8	16.2
Autoencoder-based	R-1	R-2	R-SU4	R-1	R-2	R-SU4
C-ATTENTION	39.3	8.7	14.1	42.3	10.7	16.1
VAESUM	39.6	8.9	14.3	42.1	11.0	16.4
Coarse-to-Fine	R-1	R-2	R-SU4	R-1	R-2	R-SU4
QUERYSUM$_{\mathcal{S}}$	41.1	**9.6**	15.1	42.9	11.6	16.7
QUERYSUM$_{\mathcal{P}}$	41.3	9.1	15.0	**43.4**	11.2	16.5
QUERYSUM$_{\mathcal{S}+\mathcal{P}}$	**41.6**	9.5	**15.3**	43.3	**11.6**	**16.8**

Table 3.5: System performance on DUC 2006 and 2007. R-1, R-2 and R-SU4 stand for the F1 score of ROUGE 1, 2, and SU4, respectively. Results with $*$ were obtained based on our own implementation.

asked to assess the *Relevance* of the summary sentences to the query and sentence scores were averaged to obtain a relevance score. Detailed instructions of human evaluation can be found in Appendix A.

3.5 Results

3.5.1 Automatic Evaluation

Our results on DUC are summarized in Table 3.5. The first block reports upper bound performance (GOLD) which we estimated by treating a (randomly selected) reference summary as the output of a hypothetical system and comparing it against the remaining (three) ground truth summaries. ORACLE uses reference summaries as queries to retrieve summary sentences, and LEAD returns all lead sentences (up to 250 words) of the most recent document.

The second block in Table 3.5 compares our model to various *graph-based* approaches which include: LEXRANK (Erkan and Radev, 2004), a widely used unsupervised method based on Markov random walks. LEXRANK is query-free, it measures relations between all sentence pairs in a cluster and sentences recommend other similar sentences for inclusion in the summary. GRSUM (Wan, 2008), a Markov random walk model that integrates query-relevance into a **G**raph **R**anking algorithm; and CTSUM (Wan and Zhang, 2014) which is based on GRSUM but additionally considers the factor of information **C**er**T**ainty in sentence ranking. Wan and Zhang (2014) manually annotated 1,000 sentences from the FactBank corpus (Saurí and Pustejovsky, 2009) with certainty labels (using a five point Likert scale), and trained a SVM regression model for information certainty estimation. The SVM regression model estimates certainty scores for sentences in news articles, and these scores are incorporated into the graph-based ranking algorithm for extractive summarization.

The third group in the table shows the performance of *autoencoder-based* neural approaches. C-ATTENTION (Li et al., 2017a) is based on a **C**ascaded attention model that learns the salience information of sentences and words for compressive multi-document summarization: the model captures sentence-level salience with attention weights which are optimized by an unsupervised reconstruction objective, and it also incorporates word salience to generate condensed information by adding sparsity constraints on the number of output vectors. VAESUM (Li et al., 2017b) employs a generative model based on **VA**riational auto**E**ncoders (Kingma and Welling, 2014; Rezende et al., 2014) and a data reconstruction model for sentence salience estimation. VAESUM represents the state-of-the-art amongst neural systems on DUC. The salience estimation module is further integrated in an integer linear program which selects VPs and NPs to create the final summary (see Section 2.2.2 for details). Similar to our experimental setting, its hyperparameters are optimized on a development set.

The last block in Table 3.5 presents different variants of our query-focused summarizer which we call QUERYSUM. We show automatic results with distant supervision based on isolated $\mathcal{S}$entences (QUERYSUM$_{\mathcal{S}}$), $\mathcal{P}$assages (QUERYSUM$_{\mathcal{P}}$), and an ensemble model (QUERYSUM$_{\mathcal{S}+\mathcal{P}}$) which combines both. As can be seen, our models outperform strong comparison systems on both DUC test sets: QUERYSUM$_{\mathcal{S}}$ achieves the best R-1 while QUERYSUM$_{\mathcal{P}}$ achieves the best R-2 and R-SU4. Perhaps unsurprisingly, both models fall behind the human upper bound but close to the oracle.

Our results on the TD-QFS dataset are summarized in Table 3.6. In addition to LEAD and LEXRANK, we compared to KLSUM, the best performing system on this

Upper Bound & Baselines	R-1	R-2	R-SU4
GOLD	52.2	27.0	30.2
ORACLE	44.9	18.9	23.0
LEAD	33.5	5.2	10.4
LEXRANK	35.3	7.6	12.2
KLSUM	41.5	11.3	16.6
Coarse-to-Fine	R-1	R-2	R-SU4
QUERYSUM$_{\mathcal{S}}$	**44.4**	**16.2**	**20.8**
QUERYSUM$_{\mathcal{P}}$	43.5	14.8	19.7
QUERYSUM$_{\mathcal{S}+\mathcal{P}}$	44.3	16.1	20.7

Table 3.6: System performance on TD-QFS. R-1, R-2 and R-SU4 stand for the F1 score of ROUGE 1, 2, and SU4, respectively.

dataset (Baumel et al., 2016). KLSUM selects a subset of sentences from retrieved candidates by minimizing the Kullback-Leibler Divergence between the unigram distribution in the selected sentences and the source cluster. QUERYSUM$_{\mathcal{S}}$ and our ensemble model achieve superior results across all ROUGE metrics.

3.5.2 Human Evaluation

For the DUC benchmarks, participants assessed summaries created by VAESUM a neural state-of-the-art system, QUERYSUM$_{\mathcal{S}+\mathcal{P}}$, and the LEAD baseline. For TD-QFS, we evaluated summaries created by KLSUM, QUERYSUM$_{\mathcal{S}+\mathcal{P}}$, and LEAD. We also included a randomly selected GOLD standard summary as an upper bound. We sampled 20 query-cluster pairs from DUC (2006, 2007; 10 from each set), and 20 pairs from TD-QFS (5 from each cluster). We collected three responses per query-summary pair.

Table 3.7 shows the ratings for each system. As can be seen, participants find QUERYSUM summaries on DUC more relevant and with less redundant information compared to LEAD and VAESUM. Our multi-step estimation process also produces more coherent summaries (as coherent as LEAD) even though coherence is not explicitly modeled. Overall, participants perceive QUERYSUM summaries as significantly better ($p < 0.05$) compared to LEAD and VAESUM. QUERYSUM is also considered as the best performing system across metrics on TD-QFS. This further demonstrates the robustness of our system on unseen domains and query types.

DUC	Rel	Suc	Coh	All
LEAD	$3.75^{\triangleright\dagger\circ}$	$3.60^{\dagger\circ}$	$4.27^{\triangleright}$	$3.96^{\dagger\circ}$
VAESUM	4.28	$3.62^{\dagger\circ}$	$4.05^{\dagger\circ}$	$4.03^{\dagger\circ}$
QUERYSUM	**4.32**	**3.93**$^{\triangleright}$	**4.27**$^{\triangleright}$	**4.22**$^{\triangleright}$
GOLD	4.36	$3.93^{\triangleright}$	$4.35^{\triangleright}$	$4.26^{\triangleright}$

TD-QFS	Rel	Suc	Coh	All
LEAD	$3.97^{\triangleright\dagger\circ}$	3.93°	4.04°	$3.98^{\dagger\circ}$
KLSUM	4.24°	4.13°	4.00°	4.12°
QUERYSUM	**4.47**	**4.13**$^{\circ}$	4.02°	**4.21**$^{\circ}$
GOLD	$4.60^{\triangleright}$	$4.41^{\triangleright\dagger}$	$4.33^{\triangleright\dagger}$	$4.45^{\triangleright\dagger}$

Table 3.7: Human evaluation results on DUC (above) and TD-QFS (below): average **Rel**evance, **Suc**cinctness, **Coh**erence ratings; **All** is the average across ratings; $\triangleright$: sig different from VAESUM or KLSUM; †: sig different from QUERYSUM; $^{\circ}$: sig different from Gold (at $p < 0.05$, using a pairwise t-test).

3.5.3 Examples of System Output

We provide system outputs in Table 3.9, 3.10, and 3.11 for one cluster from DUC 2006, 2007 and TD-QFS, respectively. As we can see, our system produces summaries that cover diverse aspects of the input query and contain less query-irrelevant information.

3.5.4 Ablation Studies

We also conducted ablation experiments to verify the effectiveness of the proposed coarse-to-fine framework. We present results in Table 3.8 when individual modules are removed. In the $-$Relevance setting, all text segments (i.e., sentences or passages) in a cluster are given as input to the evidence estimator module. The $-$Evidence setting treats all retrieved segments as evidence for summarization. Note that since our summarizer operates on sentences, we can only assess this configuration with the QUERYSUM$_S$ model; we take the top k^{QA} sentences from the retrieval module as evidence. The $-$Centrality setting treats the (ranked) output of the evidence estimator as the final summary. For the sake of brevity, we report results on DUC 2007 and TD-QFS (DUC 2006 follows a very similar pattern).

As can be seen, removing the retrieval module leads to a large drop in the per-

	DUC 2007			TD-QFS		
Systems	R-1	R-2	R-SU4	R-1	R-2	R-SU4
QUERYSUM$_{\mathcal{S}}$	42.9	11.6	16.7	44.4	16.2	20.8
−Relevance	↓1.5	↓1.4	↓1.2	↓2.7	↓3.9	↓3.0
−Evidence	↓0.3	↓0.4	↓0.4	↓0.7	↓0.4	↓0.2
−Centrality	↓2.3	↓1.3	↓1.3	↓0.9	↓1.1	↓0.9
QUERYSUM$_{\mathcal{P}}$	43.4	11.2	16.5	43.5	14.8	19.7
−Relevance	↓0.2	↑0.2	↑0.1	↓4.2	↓5.4	↓4.8
−Centrality	↓3.2	↓2.1	↓2.0	↓3.3	↓3.5	↓3.3

Table 3.8: Ablation results (absolute performance decrease/increase denoted by ↓/↑).

formance of QUERYSUM$_{\mathcal{S}}$. This indicates that the (deep) semantic matching model trained for sentence selection can get distracted by noise which a (shallow) relevance matching model can help pre-filter. Interestingly, on DUC, when the matching model is trained on passages, the retrieval module seems more or less redundant, there is in fact a slight improvement in R-2 and R-SU4 (see row QUERYSUM$_{\mathcal{P}}$, −Relevance in Table 3.8). This suggests that the evidence estimator trained on passages is more robust and captures the semantics of the query more faithfully. Moreover, since it takes contextual signals into account, it is able to recognize irrelevant information and unanswerability is explicitly modeled. We show in Figure 3.2 how ROUGE-2 varies over k^{IR} best retrieved segments. We compare three different types of query settings, the short *title*, the *narrative*, and the *full* query with both the title and the narrative. As expected, recall increases with k^{IR} (i.e., when more evidence is selected) and then finally converges. For both sentence and passage retrieval settings, the full query achieves best performance over k^{IR}, with the narrative being most informative when it comes to relevance estimation.

Performance also drops in Table 3.8 when the evidence estimator is removed (see QUERYSUM$_{\mathcal{S}}$, −Evidence in Table 3.8). In Figure 3.3, we plot how ROUGE-2 varies with increasing k^{QA} when the evidence component is estimated on passages and sentences for the full model. As can be seen, the model trained on passages surpasses the model trained on sentences roughly when $k^{\text{QA}} = 80$. For comparison, we also show the performance of the retrieval module by treating the top sentences as evidence. The retrieval curve is consistently under the passage curve, and under the sentence curve when $k^{\text{QA}} < 140$. Since the quality of top sentences directly affects the quality of the

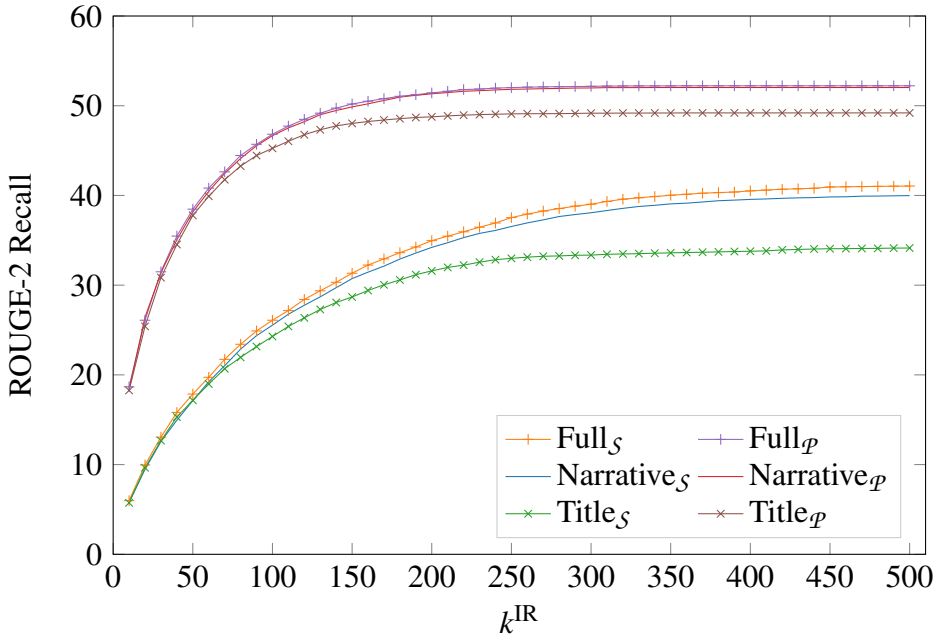


Figure 3.2: Performance (ROUGE-2 Recall) over k^{IR} best retrieved segments (DUC 2005; development set). $\mathcal{S}$ and $\mathcal{P}$ refer to sentence and passage retrieval, respectively. *Full* is the concatenation of the query title and narrative.

summarization module, this further demonstrates the effectiveness of evidence estimation in terms of reranking retrieved segments.

Finally, Table 3.8 shows that the removal of the centrality estimator decreases performance even when the query and appropriate evidence are taken into account. This suggests that the centrality estimator further learns to select important summary worthy sentences from the available evidence. Interestingly, the gain on the DUC datasets is slight but considerable on TD-QFS, suggesting that in less topically concentrated clusters where multiple high-quality answers can be available, the soft discrimination between answer candidates based on their answerability can be useful during the final summary sentence selection.

3.6 Summary

In this chapter, we proposed a coarse-to-fine estimation framework for query focused multi-document summarization. We explored the potential of leveraging distant supervision signals from Question Answering to better capture the semantic relations between queries and document segments. Experimental results across datasets show that the proposed model yields results superior to competitive baselines contributing

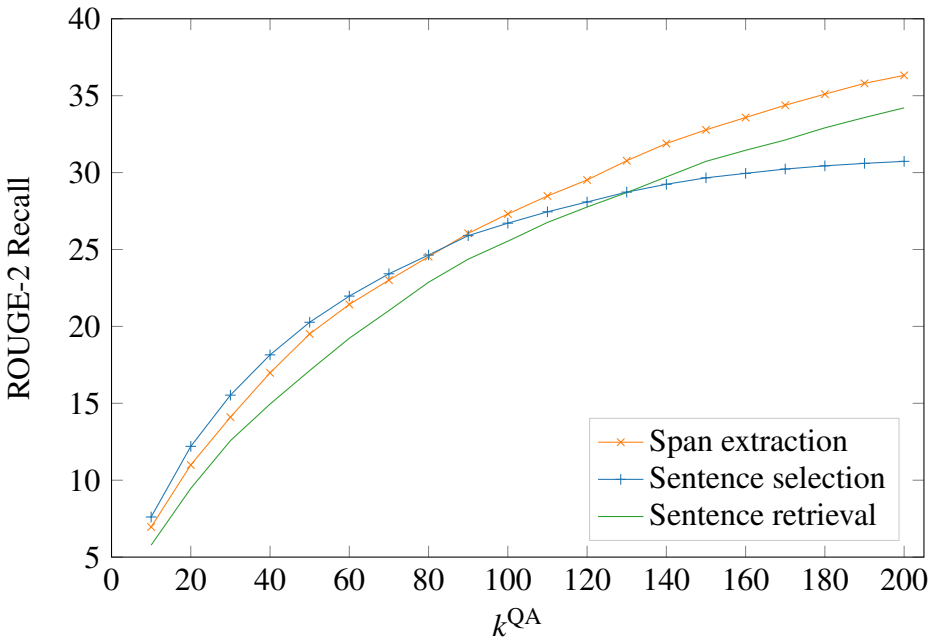


Figure 3.3: Performance (ROUGE-2 Recall) over k^{QA} best evidence sentences selected by estimators trained on sentences and passages (DUC 2005; development set).

to summaries which are more relevant and less redundant. We have also shown that disentangling the tasks of relevance, evidence, and centrality estimation is beneficial allowing us to progressively specialize the summaries to the semantics of the query.

Distant query modeling with QA resources can be usefully employed in extractive QFS. However, QA resources can be expensive, and humans usually prefer abstracts over extracts due to higher readability. In the next chapter, we will move on to abstractive QFS and describe how to generate query focused summaries without using QA training data.

Query: CRIME AND LAW ENFORCEMENT IN CHINA – *Give examples of criminal activity in China. Name those involved, if possible. What is China doing to fight crime?*
GOLD: In 1996, China began cracking down on crime. Extensive investigations and citizen tips led to hundreds of arrests for such crimes as drug trafficking; firearms, ammunition and explosives manufacturing, sales, smuggling and possession; burglary and robbery; murder; hooliganism; kidnapping; racketeering; gambling; and blackmail. The perpetrators are often gangs of thieves and criminals, and members of international criminal gangs operating between China and Hong Kong or China and Macau. In 1998, 60% of criminal suspects arrested were minors. Chinese authorities broke up a Hong Kong-based gang operating between Hong Kong and the mainland. Its leader was tried, convicted, and sentenced to death in China. Chinese authorities apprehended members of a Macau gang in its Guangdong Province. As part of its "Strike Hard national crime-fighting campaign, China agreed to participate in the UN Commission on Crime Prevention and Criminal Justice. China revised its criminal and procedural laws and enacted new laws. Its Criminal Law was amended to include terrorist crime, organized crime, money-laundering, illegal immigrant trafficking, and environment-related crimes. China signed legal assistance agreements with 28 countries and extradition agreements with ten. China pledged increased cross-border anti-crime cooperation and urged Portugal to take tougher measures against gang-related crime in preparation for the 1999 handover of the Portuguese colony. After the handover, China will station troops in Macau to better fight organized criminal activity there. The Chinese government pledges to increase efforts to crack down on corruption, smuggling, and other economic crimes as well as criminal acts in 2000.
LEAD: Members of a criminal gang in Foshan city of south China's Guangdong province, which was controlled by a larger and more notorious gang in neighboring Macao, have been apprehended by local police. Police arrested 28 people who have been involved in more than 30 cases of blackmail, gambling, illegal use of guns and other crimes. The gambling cases involved more than 50 million yuan (about six million U.S. dollars) of illicit money. Police also seized a number of guns and ammunition, including eight military pistols. The gang was established by Zeng Qiqiang in 1996, as a branch of the Shuifang Bang, a large criminal gang in Macao. The gang in Foshan, with more than 100 members, used to help the Shuifang Bang" run its gambling operations and collect money from people by force. To date, the provincial public security department of Guangdong and the local police in Foshan have completely uprooted the gang which seriously threatened the security of Foshan and Macao.

Table 3.9: System outputs for cluster D0621C in DUC 2006. The gold summary answers the query covering four main aspects (denoted with different colors): (1) general facts and vision; (2) criminal activities in southeastern China, including HongKong and Macau; (3) international corporations; (4) law revision and enforcement. Our system produces more diverse content that represents these aspects compared to other systems.

VAESUM: Police working with Hong Kong authorities had arrested 18 members of the gang in southern Guangdong province, which is adjacent to Hong Kong. As a reputed local crime boss fights his death sentence in China, reports Thursday said Hong Kong officials had previously asked mainland counterparts to consider sparing the lives of territory residents convicted of capital offenses in China. A police chief of a southern Chinese city where a reputed Hong Kong crime boss is on trial has stepped up security following assassination threats by gang members. Gang members loyal to "Big Spender" Cheung Tze-keung have put a 4 million Hong Kong dollar (U.S. dlrs 512,000 million) price tag for the death of Guangzhou police chief Zheng Guoqiang. The arrests are the latest in a series by Chinese and Hong Kong police to crack down on criminal activities related to 43-year-old Hong Kong gang boss Cheung Tze-keung, alias "Big Spender. Charges against the reputed gangsters center around the killing of a mainland Chinese businessman and a Hong Kong resident, armed robberies, smuggling explosives into Hong Kong, and the kidnapping of the two Hong Kong businessmen for more than 1.6 billion Hong Kong dollars (U.S. dlrs 205 million) in ransom. Hong Kong officials would appeal on grounds that the mainland had no jurisdiction over Cheung's case since many of Cheung's alleged crimes, including kidnappings of two Hong Kong tycoons, were committed in Hong Kong. 18 were Hong Kong residents and 14 were from mainland China.
QUERYSUM: Zhang Fusen, head of the Chinese delegation, told the fifth session of the UN commission on Crime Prevention and Criminal Justice (CCPCJ) that China will participate in united nations activities in crime prevention and criminal justice. China has revised the criminal law and criminal procedure law, promulgated and enforced new laws such as the lawyers' law and the law on administrative punishment to strengthen the judicial guarantee for human rights during that period of time, the paper says. As a reputed local crime boss fights his death sentence in china, reports Thursday said Hong Kong officials had previously asked mainland counterparts to consider sparing the lives of territory residents convicted of capital offenses in China. China is ready to strengthen cooperation with other countries and international organizations in combating and preventing organized transnational crime, a senior Chinese official said here today. Zhang said that in the past few years, China's law enforcement authorities cracked numerous cases in southeast china involving killing, kidnapping and racketeering by members of criminal gangs which entered china from overseas. Statistics show that in 1996, courts throughout the country sentenced 322,382 criminal offenders who had seriously endangered public security by committing crimes of violence, crimes involving the use of guns, and gang-related crimes. Speaking at the opening ceremony of the seventh world conference of Asia Crime Prevention Foundation (ACPF), deputy procurator-general of the supreme people's procuratorate of China Liang Guoqing called for enhancing cooperation among asian countries to fight crimes and set up a crime prevention regime.

Table 3.9: Continued.

Query: SOUTHERN POVERTY LAW CENTER – *Describe the activities of Morris Dees and the Southern Poverty Law Center.*
GOLD: Morris Dees is a co-founder and leader of the Southern Poverty Law Center, located in Montgomery, Alabama. It was founded to battle racial bias and has expanded its efforts by tracking hate crimes and the increasing spread of racist organizations across the US. "Teaching Tolerance" is a major program of the Center. Under that program, a magazine promoting interracial and intercultural understanding goes to more than 400,000 teachers. Other publications of the Center include the magazine "Intelligence Report" and pamphlets "Ten Ways to Fight Hate" and "Fighting Hate at School". Dees has determined that the civil courts are an effective forum in which to attack and destroy hate groups. He has used the civil lawsuit like a "Buck Knife, carving financial assets out of hate group leaders". Some skeptics thought that Dees sought out victims of hate groups to profit from their tragedy. However, Dees does not charge the groups and the Center estimates that it collects only 2% on successful judgments. Dees has a perfect record in the major lawsuits he has prosecuted. Successful judgments include one for $21.5M against a South Carolina branch of the Ku Klux Klan for burning the Macedonia Baptist Church. Others include $6.3M against Aryan Nation's leader Richard Butler and $7M against a Klan group that killed a black man in Mobile, Alabama. The Center operates mostly on contributions that in the late 1990s have increased to around $100 Million annually.
LEAD: Spokane, Wash. (AP) – facing eviction from its compound in northern Idaho, the aryan nations may move its annual white supremacist gathering to Pennsylvania next year. The news was posted on the Neo-Nazi group's web site Friday, a week after the group was slapped with a $6.3 million judgment in a civil lawsuit. The compound is scheduled to be seized on sept. 29 and the assets sold to satisfy a portion of the judgment due to two people who sued the group after they were assaulted by aryan nations' guards. The notice was the first indication that the lawsuit, brought by the southern poverty law center, may drive the group out of Idaho. "I have been asked if I would continue to host the yearly national congress and my answer was, of course, an astounding yes!" wrote august B. Kreis III, web master for the Aryan nations and a posse comitatus leader in Pennsylvania. Kreis wrote that if the compound is lost, the Aryan nations "National Congress 2001" would be planned for a site near ulysses, pa. Aryan nations leader Richard Butler declined to talk with reporters Friday. He is appealing the judgment to the Idaho supreme court, but that appeal is not expected to halt the seizure of the group's 20-acre compound north of Hayden lake. Morris Dees, the civil rights lawyer who led the plaintiffs' legal team, has said he expected the judgment to bring a quick end to the aryan nations and its racist, anti-semitic message.

Table 3.10: System outputs for cluster D0701A in DUC 2007. The gold summary answers the query covering three main aspects (denoted with different colors): (1) Southern Poverty Law Center and its activities; (2) Morris Dees and his activities; (3) representative successful lawsuits. For this document cluster, summarization systems are prone to extract unnecessary lawsuit details, which indirectly relate to the given query but are not the query focus. Our system contains more summary-worthy facts that succinctly respond to the given query compared to other systems.

VAESUM: A state jury in northern Idaho Thursday ordered leaders of the Aryan nations to pay more than $6 million to the victims of an attack two years ago by men who were serving as security guards at the group's compound near here. Coeur d'Alene, Idaho – issuing a verdict that civil rights organizations hope will bankrupt one of the nation 's largest white-supremacist groups and limit its ability to preach hate. Aryan nations leader Richard Butler vowed Saturday he will not leave northern Idaho, despite a $6.3 million judgment against his racist organization. Coeur d'Alene, idaho – Morris S. Dees JR. , who has won a series of civil rights suits against the Ku Klux Klan and other racist groups in a campaign to put them out of business, came to court here Monday to try to seize the Aryan nations compound that has nurtured white supremacists for more than 20 years. Her son who were attacked by Aryan nations guards outside the white supremacist group's north Idaho headquarters. One of two men convicted of assaulting a woman and her son outside the headquarters of the Aryan nations denied being a member of the white supremacist group Thursday during testimony in a civil rights case filed against them, the aryan nations and the group's founder, Richard Butler. Morris Dees, co-founder of the southern poverty law center in Montgomery, Ala., has said he intends to take everything the aryan nations owns to pay the judgment, including the sect's name.
QUERYSUM: Morris Dees, the co-founder of the southern poverty law center in Montgomery, Ala., and one of the attorneys for the plaintiffs, said he intended to enforce the judgment, taking everything the Aryan nations owns, including its trademark name. Dees, founder of the southern poverty law center, has won a series of civil right suits against the Ku Klux Klan and other racist organizations in a campaign to drive them out of business. But since co-founding the southern poverty law center in 1971, Dees has wielded the civil lawsuit like a buck knife, carving financial assets out of hate group leaders who inspire followers to beat, burn and kill. In a lawsuit that goes to trial Monday, attorney Morris Dees of the southern poverty law center is representing a mother and son who were attacked by security guards for the white supremacist group. The southern poverty law center tracks hate groups, and intelligence report covers right-wing extremists. Over the last two decades, the southern poverty law center has taken the Ku Klux Klan and other hate groups to court, starting with a successful suit against the invisible empire Klan, which in 1979 attacked a group of peaceful civil rights marchers in Decatur, Ala. He said Gilliam also told the informant someone should kill the FBI sniper who killed the wife of white supremacist randy weaver during an 11-day standoff in 1992 at Ruby Ridge, Idaho, along with civil rights lawyer Morris Dees of the Montgomery-based southern poverty law center.

Table 3.10 Continued.

Query: *Alzheimer Memory*
GOLD: Alzheimer's is the most common form of dementia, a general term for memory loss and other intellectual abilities serious enough to interfere with daily life. The main underlying cause of memory loss and confusion is the progressive damage to brain cells caused by Alzheimer's disease. The brain region called the hippocampus is the center of learning and memory in the brain, and the brain cells in this region are often the first to be damaged. People in the early stages of Alzheimer's disease may experience lapses of memory and have problems finding the right words. As the disease progresses, they may: become confused and frequently forget the names of people, places, appointments and recent events experience mood swings, feel sad or angry, or scared and frustrated by their increasing memory loss, become more withdrawn, due either to a loss of confidence or to communication problems have difficulty carrying out everyday activities. In the early stages of dementia, memory aids such as lists, diaries, clocks and clear, written instructions can help jog the person's memory if they are willing and able to make use of them. As the dementia progresses, the person may become less able to understand what the aids are for. Alzheimer's does not affect all memory capacities equally. Older memories of the person's life (episodic memory), facts learned (semantic memory), and implicit memory (the memory of the body on how to do things, such as using a fork to eat) are affected to a lesser degree than new facts or memories.
LEAD: No medications are currently approved by the U.S Food and Drug Administration (FDA) to treat mild cognitive impairment. Drugs approved to treat symptoms of Alzheimers disease have not shown any lasting benefit in delaying or preventing progression of MCI to dementia. The following coping strategies may be helpful for those with MCI Some studies suggest that these strategies may help slow decline in thinking skills, although more research is needed to confirm their effect. Exercise on a regular basis to benefit your heart and blood vessels, including those that nourish your brain. Control cardiovascular risk factors to protect your heart and blood vessels, including those that support brain function. Participate in mentally stimulating and socially engaging activities, which may help sustain brain function. Find a clinical trial join a clinical study to help improve our understanding of MCI. Find a trial. Experts recommend that a person diagnosed with MCI be re-evaluated every six months to determine if symptoms are staying the same, improving or growing worse. MCI increases the risk of later developing dementia, but some people with MCI never get worse. Others with MCI later have test results that return to normal for their age and education. It's not yet possible to tell for certain what the outcome of MCI will be for a specific person or to determine the underlying cause of MCI from a persons symptoms. Researchers hope to increase the power to predict MCI outcomes by developing new diagnostic tools to identify and measure underlying brain changes linked to specific types of dementia.

Table 3.11: System outputs for cluster 3-0 in TD-QFS. Summary sentences include different aspects of *Alzheimer Memory* with varied degrees of query relevance (denoted with different colors): (1) directly relevant aspects, such as memory loss or dementia; (2) indirectly relevant aspects, such as Mild Cognitive Impairment (MCI) and general symptoms of Alzheimers. Compared to other systems, our system contains more information that directly respond to the given query.

KLSUM: Here are some ways that depression in a person with Alzheimers may be different: may be less severe, may not last as long, and symptoms may come and go. The person with Alzheimers may be less likely to talk about or attempt suicide. As a caregiver, if you see signs of depression, discuss them with the primary doctor of the person with dementia. The main underlying cause of memory loss and confusion is the progressive damage to brain cells caused by Alzheimers disease. As the disease progresses, people with Alzheimers will need more support from those who care for them. Some of the symptoms common to both Alzheimers and depression include: loss of interest in once-enjoyable activities and hobbies, social withdrawal, memory problems, sleeping too much or too little, impaired concentration. With so much overlap in symptoms, it can be hard to distinguish between the two disorders, especially since they so often occur together. Many older adults without dementia also notice changes in their sleep, but these disturbances occur more frequently and tend to be more severe in Alzheimers. Older memories of the persons life (episodic memory), facts learned (semantic memory), and implicit memory (the memory of the body on how to do things, such as using a fork to eat) are affected to a lesser degree than new facts or memories. Thus, the genotype most at risk for Alzheimer disease and at an earlier age is APOE 4,4.
QUERYSUM: These symptoms occur when the brain is damaged by certain diseases and conditions, including Alzheimers disease. They often are present along with changes caused by other types of dementia, including Alzheimers disease and dementia with lewy bodies. These diseases include Alzheimers disease and vascular dementia. Memory loss is a distressing part of dementia, both for the person with dementia and for those around them. The main underlying cause of memory loss and confusion is the progressive damage to brain cells caused by Alzheimers disease. Alzheimers disease is distinguished from other forms of dementia by characteristic changes in the brain. Alzheimers disease, first described by the german neurologist Alois Alzheimer, is a physical disease affecting the brain. Experts believe that many cases but not all result from brain changes occurring in the very early stages of Alzheimers disease or other dementias. For a person with Alzheimers or dementia, poor nutrition may increase behavioral symptoms and cause weight loss. Alzheimers is the most common form of dementia, a general term for memory loss and other intellectual abilities serious enough to interfere with daily life. Alzheimers disease is the most common cause of dementia, affecting around 496,000 people in the UK. Cognitive stimulation therapy involves activities and exercises that are designed to improve memory and communication skills. Stage 2: very mild cognitive decline (may be normal age-related changes or earliest signs of Alzheimers disease) the person may feel as if he or she is having memory lapses forgetting familiar words or the location of everyday objects.

Table 3.11 Continued.

Chapter 4

Generating Query Focused Summaries with Query-Free Resources

We have shown in the previous chapter that question answering data and models can be useful for extractive approaches to QFS. However, extractive summaries are often considered suboptimal due to redundancy and coherence issues. In this chapter, we aim at building an abstractive summarization system that generates better summaries for QFS when training data in the form of queries, documents, and summaries is not readily available. Also, as question answering data can be costly to obtain, we further constrain the available resources for this task and investigate: can we eliminate dependency on question answering and generate better summaries for QFS with only query-free resources?

To answer this question, we first propose to decompose the task of abstractive QFS into two components: query modeling and conditional language modeling. As no query is assumed accessible for training, we introduce a unified representation for summaries and queries, so that summaries in generic data can be converted into *proxy queries* to learn a query model, without relying on distant QA resources as in the previous chapter. We present a Masked ROUGE Regression framework for *proxy query modeling*, where sentences are ranked per their estimated evidence, and query focused summaries can be generated from the selected sentences. Experiments across QFS benchmarks show that our model achieves state-of-the-art performance despite learning from weak supervision, and produces summaries that are more relevant and coherent compared to existing systems.

4.1 Introduction

The neural encoder-decoder framework has become increasingly popular in generic summarization (See et al. 2017; Gehrmann et al. 2018; Liu and Lapata 2019a; Fabbri et al. 2019, *inter alia*) thanks to the availability of large-scale datasets containing hundreds of thousands of document-summary pairs. Training data of this magnitude is not readily available for QFS which aims to create a short summary from a set of documents that answers a specific query. Existing corpora (Nema et al., 2017; Dang, 2005; Hoa, 2006; Baumel et al., 2016) are relatively small for modern data-hungry neural architectures and have been mostly used for evaluation purposes.

A major bottleneck in leveraging generic summarization data for QFS is the absence of queries (Nema et al., 2017); the majority of existing datasets consist of document-summary pairs, while QFS summaries are expected to answer specific queries. Recent work (Xu and Lapata, 2020; Su et al., 2020; Laskar et al., 2020b) sidesteps this problem by resorting to distant supervision from query-relevant NLP resources including question answering (Rajpurkar et al., 2016; Chakraborty et al., 2020) and paraphrase identification (Dolan and Brockett, 2005). Such approaches incorporate query modeling in the summarization process but are even more data hungry compared to generic summarization ones, since they additionally require access to QA datasets which can be extremely costly to create (Bajaj et al., 2016; Kwiatkowski et al., 2019). Moreover, there is often a mismatch between queries in QA datasets and those in QFS scenarios (Xu and Lapata, 2020); the two types of queries are not identically distributed and it is practically infeasible to find appropriate query-related resources for all domains and topics.

In this chapter, we do not assume access to any resources other than those available for generic summarization. We further decompose abstractive QFS into two subtasks:

1. **Query modeling**: Representing the semantics for a given query and finding its supportive evidence within a set of documents.

2. **Conditional language modeling**: Generating an abstractive summary based on found evidence.

Under this formulation, we use generic summarization data not only for conditional language modeling, but also for learning an evidence ranking model. Inspired by the Cloze task and its applications in NLP (Taylor, 1953; Lewis et al., 2019; Lee et al., 2019), we propose MARGE, a **Ma**sked **R**OU**GE** regression framework for evidence

estimation and ranking. MARGE introduces a unified representation for *summaries* and *queries*, so that summaries in generic data can be converted into *proxy queries* for learning a query model. Based on the evidence selected by MARGE, we generate abstractive summaries whilst controlling their length and the extent to which the query influences their content.

Our contributions in this chapter are threefold: (a) we propose a weakly supervised system for abstractive QFS where no query-related resources are required; (b) we discover a new type of connection between generic summaries and QFS queries, and provide a universal representation for them which allows generic summarization data to be exploited for QFS; and (c) we provide experimental results on QFS benchmarks, and show that across query types and domains our system achieves state-of-the-art results on both evidence ranking and abstractive QFS.

4.2 Related Work

The majority of previous QFS approaches have been extractive, operating over queries and document clusters from which they select query-relevant sentences to compose a summary. They mostly differ in the way centrality and relevance are estimated and incorporated, e.g., via manifold ranking (Wan et al., 2007), using a look-ahead strategy (Badrinath et al., 2011), uncertainty prediction (Wan and Zhang, 2014), or attention mechanisms (Li et al., 2017a,b). In the previous chapter, we also showed how to leverage distant supervision from question answering to extract summary-worthy content.

Abstractive QFS has received significantly less attention from the research community, due to generation models being particularly data-hungry (Lebanoff et al., 2018; Liu and Lapata, 2019a) and the scarcity of QFS training data. However, the recent increasing availability of pretrained models has promoted the adoption of resources from a broader range of NLP tasks to generate query focused abstracts. For example, Su et al. (2020) learn a paragraph selector based on query relevance from a plethora of QA and machine reading datasets (Su et al., 2019; Rajpurkar et al., 2016). They then fine-tune BART (Lewis et al., 2020) on CNN/DailyMail (Hermann et al., 2015), a single-document summarization dataset, and generate abstracts for QFS by iteratively summarizing the selected paragraphs to a budget. Similarly, Laskar et al. (2020b) fine-tune BERT (Devlin et al., 2019) on CNN/DailyMail, and employ a three-stage system which uses supervision from QFS data and related QA and paraphrase identification tasks. We reviewed these existing approaches for abstractive QFS in Section 2.2.2.

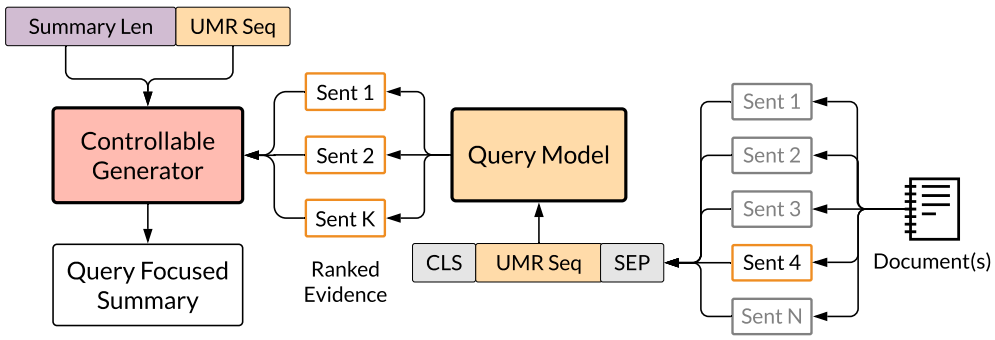


Figure 4.1: Overview of our abstractive QFS approach. Summaries and The summarization framework consists of a query model and a controllable generator. The query model ranks sentences in the input document(s) which provide evidence to answer the query; the generator operates over evidence bearing sentences to generate the final summary.

We also focus on abstractive QFS in this chapter, however, we do not assume access to any additional training resources over and above generic summarization datasets, even for query modeling. Moreover, our system is able to generate long QFS abstracts *all at once*, instead of *iteratively* creating bullet-style summaries which often lack coherence.

4.3 Problem Formulation

Consistent with previous chapters, we let $\{(S, \mathcal{D})\}$ denote a generic summarization dataset where $\mathcal{D} = \{D_1, D_2, \ldots, D_{|\mathcal{D}|}\}$ is a collection of documents with corresponding summaries S. $|\mathcal{D}| = 1$ for single-document summarization (SDS) and $|\mathcal{D}| > 1$ for multi-document summarization (MDS). In QFS, a query Q additionally specifies an information request, $\{(S, \mathcal{D}, Q)\}$. It is often assumed (e.g., in DUC benchmarks) that Q consists of a short title (e.g., *Amnesty International*), and a query narrative which is longer and more detailed (e.g., *What is the scope of operations of Amnesty International and what are the international reactions to its activities?*).

In this chapter, we propose to decompose QFS into two sub-tasks, namely *query modeling* and *conditional language modeling*. The query model $q_\theta(D|Q;\theta)$ estimates whether textual units (e.g., sentences) within document cluster D are relevant to query Q, while $p_\phi(S|D,Q;\phi)$ generates summary S conditioned on evidence provided by the query model and (optionally) the query itself (see Figure 4.1 for an illustration). When

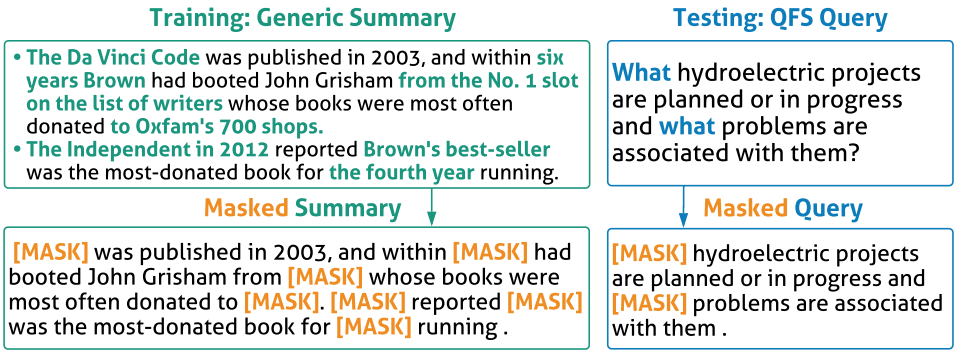


Figure 4.2: Overview of the proposed Unified Masked Representation (UMR). Summaries and queries are rendered with UMR for training and testing, respectively.

$S \perp\!\!\!\perp Q$, we have a *query-agnostic* conditional language model $p_\phi(S|D;\phi)$. Otherwise, the conditional language model is *query-guided*. Our query model is trained with distant supervision derived from *generic* summarization data which is easier to obtain (e.g., from online sources) compared to QA datasets which must be annotated from scratch (e.g., for different types of questions and domains). Although queries are not verbalized in generic summarization, we hypothesize that the summaries themselves constitute a response to *latent* queries.

So, how can we reverse-engineer the queries from the summaries? Inspired by the standard Cloze task (Taylor, 1953) and its recent variants (Lewis et al., 2019; Lee et al., 2019), we render queries and summaries in a *Unified Masked Representation* (UMR) which enables summaries to serve as *proxy* queries for model training, as shown in Figure 4.2. We further assume that the answer to these queries can be found in sentences which form part of the document collection $\mathcal{D}$. Although we do not know for certain what these sentences are we can assume that if they have a high ROUGE score against the reference summary they are likely to contain an answer. We therefore use ROUGE as a distant supervision signal, and train a model that takes a query and document sentence as input and estimates their relevance. At inference time, we also render actual queries in UMR and rank all sentences in the document collection with our trained model. The most relevant sentences serve as input to a conditional language model to generate query focused abstractive summaries.

Algorithm 1 Generate Masked Summary

function MASKSUMMARY($\mathcal{S}$, γ) ▷ Summary sentences and mask ratio
  Parse each $s \in S$ with OpenIE to extract information slots I
  Reveal budget $B = |I| * \gamma$ ▷ Reveal information partially
  Initialize revealed token number $b = 0$
  Initialize masked summary $\mathcal{M}$ to S and fill with `[MASK]`
  Initialize `EOM` = false ▷ End of Masking
  while true **do**
    $\mathcal{S}_a$ =GETAVAILABLE($\mathcal{S}$) ▷ Sentences with masked slots
    for $s \leftarrow \mathcal{S}_a$ **do**
      $b = b +$ REVEAL(s) ▷ Sample and reveal a slot; record its #tokens
      if $b \geq B$ **then** `EOM` = true
    if `EOM` **then** ▷ Start post-processing
      for $m \leftarrow \mathcal{M}$ **do**
        MERGE(m) ▷ Merge adjacent `[MASK]` tokens
      return $\mathcal{M}$
end function

4.4 Query Modeling

As explained earlier, we train a query model $q_\theta(D|Q;\theta)$ on summary-sentence pairs via distant supervision. We use a summary-based proxy query $\texttt{UMR}_S$ during training and an actual query $\texttt{UMR}_Q$ during testing. In the following, we first describe how UMRs are obtained and then discuss how the query model is trained.

Unified Masked Representation The intuition behind UMR is that a summary will encapsulate most salient information a user needs, while a query typically covers only a small fraction. We thus add one or more "placeholders" to the query to represent missing information the user actually seeks. We also identify such information in generic summaries for selective masking, to reduce the distributional shift during training.

The UMR for a summary is the concatenation of its sentential UMRs. To convert a sentence from natural language to UMR, we parse it with Open Information Extraction (Open IE; Stanovsky et al. 2018) to a set of propositions consisting of verbs and their arguments. The latter are considered candidate *information slots* I. We initialize Algorithm 1, by replacing all such slots with a `[MASK]` token. We subsequently sample and reveal a set of slots subject to a budget constraint. We define the budget as $B = \gamma * |I|$

where $\gamma \in [0,1]$ modulates the proportion of tokens to be revealed within I slots (and is optimized on the development set). Finally, in order to keep the representation of $\texttt{UMR}_S$ and $\texttt{UMR}_Q$ consistent (see next paragraph), we merge adjacent `[MASK]` tokens to one `[MASK]` resulting in a partially masked summary.

We mask QFS queries by considering their structure and lexical makeup. Queries in DUC benchmarks often contain *interrogative* words (e.g., *how is A* and *what is B*) and *request* words (e.g., *describe A* and *tell me B*). Following this observation, we manually collect a small set of such query words and replace them with `[MASK]`. For queries with a title and a narrative, we first mask the narrative and then prepend "`[MASK]` $\mathcal{T}$.", where $\mathcal{T}$ is a sequence of title tokens. Figure 4.2 shows examples of a masked query and summary.

Evidence Ranking We represent sentences in a document collection and UMR queries with a pre-trained BERT model (Devlin et al., 2019). Specifically, we concatenate a UMR query and a candidate sentence to sequence "`[CLS]` $\mathcal{U}$ `[SEP]` $\mathcal{C}$ `[SEP]`" where $\mathcal{U}$ is a sequence of tokens within a UMR query and $\mathcal{C}$ a sequence of tokens in a document sentence (we pad each sequence in a minibatch of L tokens). The `[CLS]` vector serves as input to a single layer neural network which estimates whether the sentence contains sufficient evidence to answer the query (see Figure 4.1 right). We use the mean-square error to compute the loss and update the encoding parameters in BERT via standard backpropagation:

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{B}|} \sum_{(S,C)\sim\mathcal{B}} \left[(y - \hat{y}(S,C;\theta))^2 \right]. \tag{4.1}$$

where S,C is a summary-sentence pair sampled from a minibatch $\mathcal{B}$ and y the training signal. Recall the summary is rendered as $\texttt{UMR}_S$.

Previous work (Liu and Lapata, 2019a) has used ROUGE-2 as training signal for paragraph ranking. However, sentences are significantly shorter than paragraphs, and we observe a number of instances with a ROUGE-2 score of 0. We therefore perform label smoothing and define y as the F1 interpolation of ROUGE-2 and ROUGE-1: $y = \mathrm{R}_2(S,C) + \lambda * \mathrm{R}_1(S,C)$ where λ is optimized on the development set. At inference time, we use the trained model to compute the affinity score between $\texttt{UMR}_Q$ and all candidate sentences in $\mathcal{D}$ and rank them accordingly. The highest ranked sentences are deemed query-relevant and passed on to our summary generation model.[1]

[1]The Cloze task has been also employed in recent work in generic summarization (Huang et al., 2020). In comparison, we address a different research question (i.e., query modeling vs. summary evaluation) based on a different formulation (masked ROUGE regression vs. multiple-choice QA).

Query Narrative Expansion In some cases queries may be relatively short and narratives absent. This can be problematic for our setup since query proxies (in the form of summaries) are typically long and detailed. For datasets with short queries we automatically create query narratives in an unsupervised fashion. We employ LexRank (Erkan and Radev, 2004) to select a subset of representative sentences under a word budget and concatenate them to form narratives (which we append to the original queries).

4.5 Query Focused Generation

We also leverage generic summarization datasets to fine-tune a pretrained language model for abstractive QFS. In experiments we employ the publicly released UNILMV2 (Bao et al., 2020) to instantiate the controllable generator shown in Figure 4.1, however any other language model could have been used instead.

With Transformer (Vaswani et al., 2017) as the backbone network, UNILMV2 is jointly pretrained for natural language understanding and generation. Specifically, a bidirectional model employs an autoencoding objective (AE; identical to Devlin et al. 2019), while a partially autoregressive (PAR) sequence-to-sequence model decomposes the probability of masked tokens in the input sequence. The pretraining loss is computed as $\mathcal{L}_{\text{AE}} + \mathcal{L}_{\text{PAR}}$. More details on UNILMV2 were provided in Section 2.1.3.

At inference, UNILMV2 operates over sentences deemed relevant by the query model and decodes summaries autoregressively (see Figure 4.1 left).

Synthetic MDS Data The pre-trained language model can be fine-tuned on MDS datasets (e.g., Multi-News; Fabbri et al. 2019) which are perhaps better aligned with the QFS task since both MDS and QFS operate over document clusters. We additionally propose a way to create synthetic MDS datasets based on SDS data. This is advantageous for two reasons. Firstly, MDS resources are fairly limited compared to SDS data (Zhang et al., 2018; Lebanoff et al., 2018). And secondly, by construction, we can ensure various data characteristics which might be desirable (e.g., the number of topics represented in the document collection).

A challenge with leveraging SDS for QFS is the summary length (Lebanoff et al., 2018). Summaries in SDS datasets such as CNN/DailyMail (Hermann et al., 2015), are on average 30 tokens long. In contrast, query focused summaries can be as long as 250 tokens. We sidestep this problem by adopting a *retrieval*-based solution. Specifically,

Document 1: (CNN) – The only thing crazier than a guy in snowbound Massachusetts boxing up the powdery white stuff and offering it for sale online? People are actually buying it. For $89, self-styled entrepreneur Kyle Waring will ship you 6 pounds of Boston-area snow in an insulated Styrofoam box – enough for 10 to 15 snowballs, he says. But not if you live in New England or surrounding states. "We will not ship snow to any states in the northeast!" says Waring's website, ShipSnowYo.com. "We're in the business of expunging snow!". His website and social media accounts claim to have filled more than 133 orders for snow – more than 30 on Tuesday alone, his busiest day yet. With more than 45 total inches, Boston has set a record this winter for the snowiest month in its history. Most residents see the huge piles of snow choking their yards and sidewalks as a nuisance, but Waring saw an opportunity...	**Summary 1**: A man in suburban Boston is selling snow online to customers in warmer states. For $89, he will ship 6 pounds of snow in an insulated Styrofoam box.
Document 2: It may be the first day of spring, but don't pack away those snow shovels just yet. Up to 36 million people are under some sort of winter weather advisory, while forecasters have warned Winter Storm Ultima will dump six inches of snow on the Northeast and mid-Atlantic on Friday. And a few locations, particularly in the higher elevations, could see even more of the white stuff, meteorologist Bruce Terry of the National Weather Service warned. South central Pennsylvania will be in the bulls-eye of the storm and receive up to 10 inches of snow, he said on Thursday. Western Maryland could get slammed with up to 8 inches. New England will be on the lower end of the snow totals but even Boston, which has seen a record 108.6 inches of snow, could get an inch or two more. The snow in New England - including Boston - will start Friday and possibly stretch through Saturday night...	**Summary 2**: Winter Storm Ultima is expected to dump six inches of snow on the Northeast and mid-Atlantic on Friday. New York City could get four to six inches of wet snow as temperatures plunge into the 30s, while Boston could get one or two inches. But higher elevations in central Pennsylvania could get 10 inches. The winter storm will move north this weekend and warmer temperatures are expected to return next week.
Document 3: Most of America has spent this winter shivering in a colder-than-usual polar plunge that's seen almost every state turned white and the Great Lakes freeze over. But in Alaska, residents are wondering what's become of the blizzards and arctic lows that usually characterize the northernmost state. The biggest city, Anchorage, is so unseasonably warm that a winter festival could only go ahead after trucks drove in snow from a stockpile, and hide it under PVC to stop it from melting away. Bemused residents even took to asking Boston - which has been swamped with more than 100 inches of snow in a record-breaking winter - if they can have their winter back. Speaking to the Boston Globe, Anchorage-dweller Danielle Crelley, 19, said said: 'This is the worst winter ever... We cant even go sledding. I just want to build a snowman.' Another, store owner Nina Walker, proposed a trade between Massachusetts and Alaska. She said: 'You give us your snow, and well give you the Palins.' Cameras from local station KTUU showed the snow-less scenes in the city...	**Summary 3**: Winter has seen snow in almost every state, and frigid lows in the Northeast - but Alaska is balmy by comparison. In largest city, Anchorage, snow for winter festival was driven in from stockpiles after less than an inch fell last month. Dog-sledding forced to move 260 miles north to get enough snow - the first time since the event began in 1946. Residents jokingly asked Boston - been buried by more than 100 inches in recent months - for its snow back. Alaskan warmth and frigid lows further south are both caused by atmospheric movements in the jet stream.

Table 4.1: Example of the synthetic MDS data from the original document-summary pairs in CNN/DM. Summary 1 is used as a query which retrieves topically-related summaries 2-3 (in this example, the topic being snow and winter storm). We view documents 1-3 as a synthetic document cluster, and the summary for this cluster is formed by the concatenation of summaries 1-3, with redundant sentences removed.

we first build a database with all summaries in the original dataset. For each sample (D,S), we query the database with summary S. We retrieve other summaries $\mathcal{S}$ with the bigram hashing and TF-IDF matching method described in Chen et al. (2017). Then, we fetch their corresponding articles $\mathcal{D}$, and form a cluster as:

$$\mathcal{D}^* = \{D\} \bigcup \mathcal{D} \tag{4.2}$$

$$\hat{S}_i^* = \text{concat}\left(S, \mathcal{S}_1, \ldots, \mathcal{S}_{|\mathcal{S}|}\right) \tag{4.3}$$

where $\mathcal{D}^*$ are the source documents, and $\hat{S}^*$ is a potentially redundant summary of them. We set $|\mathcal{S}|$ to minimize the length difference between $\hat{S}^*$ and our summary length requirement (e.g., 250 tokens). To obtain the final summary S^*, we eliminate redundancy by selecting sentences from the start of $\hat{S}^*$, skipping sentences that have high cosine similarity with those which have already been selected. We show an example of the synthetic MDS data in Table 4.1.

Summarization Input In generic MDS, the input to the summarization model is a long sequence, i.e., documents within a cluster are concatenated together and sentences in each document follow their original order (Fabbri et al., 2019). In QFS, information about absolute (document) position is lost after evidence ranking. As a result, there is a discrepancy between training and testing for our generation model. To mitigate this, we collect all sentences across documents for each training sample and rank them in descending order according to their ROUGE-2 score against the reference summary. The pretrained language model is fine-tuned against this evidence-ranked list of sentences. During inference, when *actual* queries are available, we instead use the top sentences ranked by our query model as input to summary generation.

Query Guidance Given that summarization input essentially consists of sentences that are highly relevant to the query, an obvious question concerns the usefulness of explicitly modeling the query during generation. We thus instantiate two conditional language models. For a *query-guided* summarizer $p_\phi(S|D,Q;\phi)$, we prepend $\texttt{UMRS}_S$ to the selected evidence during training and $\texttt{UMR}_Q$ at inference. While for a *query-agnostic* summarizer $p_\phi(S|D;\phi)$, we only consider the selected evidence as input to our summarizer and this setting is identical to generic MDS.

Length Control QFS tasks usually require summaries of a fixed length budget (e.g, 250 words), whereas summary length is bound to be variable in the training data.

Query Modeling	Multi-News	CNN/DM
#Sentence/Doc	20	3
#Train	1,615,508	1,719,210
#Validation	200,824	80,052
#Words/Proxy Query	111.7	26.0
#Masks/Proxy Query	35.6	8.1

Summary Generation	Multi-News	CNN/DM
#Clusters	44,972	287,227
#Documents/Cluster	2.8	4.1
#Words/Summary	257.2	261.3

Table 4.2: Training data for query modeling and summary generation. CNN/DM statistics for summary generation refer to the synthetic MDS dataset proposed in this work (based on CNN/DM).

Inspired by Fan et al. (2018), we quantize summary length into discrete bins. We augment each training instance with this information, i.e., we prepend a length token (e.g., `[230]`) to document sentences. At inference, we inform the model of the summary budget by prepending the expected length token (e.g., `[250]`) to the sentences selected by the evidence ranker (see Figure 4.1).

4.6 Experimental Setup

4.6.1 Summarization Datasets

We performed experiments on the same datasets as in the previous chapter: DUC 2005-2007 benchmarks and TD-QFS (Baumel et al., 2016). Statistics for both datasets are given in Table 3.1. DUC benchmarks contain long query narratives while TD-QFS focuses on medical texts with short keyword queries. We used DUC 2005 as a development set to optimize hyperparameters and select abstractive models, and evaluated performance on the other three datasets.

We used Multi-News (Fabbri et al., 2019) and CNN/DailyMail (Hermann et al., 2015) as our generic summarization datasets to train MARGE (for evidence ranking) and to fine-tune UNILMV2 (for summary generation). Multi-News and CNN/DailyMail

can be downloaded from https://github.com/Alex-Fabbri/Multi-News and https://github.com/abisee/cnn-dailymail, respectively. Data statistics are shown in Table 4.2. To create the training and development sets for optimizing MARGE, we sampled sentences from each dataset. Specifically, we took the first and last 20 sentences from each cluster in Multi-News and the first and last three sentences from each article in CNN/DailyMail. For fine-tuning UNILMV2, we used the original Multi-News and the synthetic multi-document version of CNN/DailyMail described in Section 4.5.

4.6.2 Implementation Details

We used the publicly released BERT model[2] and fine-tuned it for ROUGE regression with a learning rate of 3×10^{-5} and a batch size of 128 for 3 epochs on 8 GPUs (GTX 2080 Ti). We trained two summarization models on CNN/DailyMail and Multi-News, respectively, with the same hardware. For both models, we set the maximum input length to 768, and fine-tuned the publicly released UNILMV2 model[3] with a learning rate of 7×10^{-5} and a batch size of 16 for 40,000 steps with gradient accumulation every 4 steps. During decoding, we used beam search with beam size 5 and Trigram Blocking (Paulus et al., 2018) to reduce redundancy. The cosine similarity threshold for redundancy removal was set to 0.6 and summary length was discretized to 10 bins. The λ parameter for label smoothing was set to 0.15. We set γ, the parameter which modulates the proportion of information slots to reveal during masking, to 0 (see Section 4.7.1 for detailed analysis of γ and its effect on model performance).

4.7 Results

Our experiments evaluate both components of the proposed approach, namely query modeling and summary generation. We assess the evidence ranker and the effectiveness of the unified masking. We also compare our summaries against competitive abstractive and extractive systems using automatic and human-based evaluation.

4.7.1 Query Modeling

Evaluation Metrics We evaluate query modeling with *retrieval* and *summarization* metrics. For the former evaluation, we follow Liu and Lapata (2019a), concatenate the

[2]https://github.com/huggingface/pytorch-transformers
[3]https://github.com/microsoft/unilm

Models	DUC 2006			DUC 2007			TD-QFS		
	R@10	R@30	R@50	R@10	R@30	R@50	R@10	R@30	R@50
ORACLE	6.7	16.2	22.7	8.4	19.1	26.2	17.2	35.6	44.6
TERMFREQ	7.2	15.1	20.8	8.5	18.5	25.2	14.2	25.9	34.0
BERTQA	8.5	16.3	22.1	10.2	20.2	26.1	9.8	21.9	29.1
BERTMRC	8.2	16.6	22.3	9.0	19.2	25.2	8.1	16.4	23.2
MARGE-MN	**11.1**	**20.2**	**25.9**	**13.8**	**25.3**	**31.8**	11.2	21.6	29.4
+EXPAND	—	—	—	—	—	—	**18.1**	**32.9**	**39.1**
MARGE-CD	9.1	17.4	23.3	11.1	22.1	28.8	10.0	18.7	26.2
+EXPAND	—	—	—	—	—	—	17.2	27.7	26.2

Table 4.3: Retrieval performance of evidence rankers. R@k is ROUGE-2 recall against the top k sentences. MARGE models are trained on Multi-News (MN) and CNN/DailyMail (CD) datasets.

top k ranked sentences, and calculate recall against gold summaries. We additionally propose to evaluate model output as if it were an extractive summary, to better assess coverage and informativeness. We thus take the top sentences subject to a budget of 250 tokens, and remove redundancy by selecting sentences from the top and skipping sentences that have high cosine similarity (e.g., ≥ 0.6) with selected ones. We use ROUGE F1 to evaluate the resulting summaries so that *precision* is also taken into account.

Results We compare MARGE against Term Frequency, a simple but effective retrieval method that performs particularly well on DUC datasets (Katragadda and Varma, 2009). We also compare to two semantic matching models used for extractive QFS in Chapter 3: BERTQA which is trained on the joint set of WikiQA (Yang et al., 2015) and TrecQA (Yao et al., 2013) for answer sentence selection, BERTMRC which is fine-tuned on SQuAD 2.0 (Rajpurkar et al., 2018) for answer span extraction. ORACLE uses reference summaries as queries to retrieve summary sentences. For *summarization* evaluation, we report upper bound performance (GOLD) which we estimated by comparing a (randomly selected) reference summary against the remaining three reference summaries. In addition, we compare to LEAD which returns all lead sentences of the most recent document (up to 250 words) and LEXRANK (Erkan and Radev, 2004), a widely-used unsupervised method based on Markov random walks on

Models	DUC 2006			DUC 2007			TD-QFS		
	R-1	R-2	R-SU4	R-1	R-2	R-SU4	R-1	R-2	R-SU4
GOLD	45.4	11.2	16.8	47.5	14.0	18.9	52.2	27.0	30.2
ORACLE	40.6	9.1	14.8	41.8	10.4	16.0	44.9	18.9	23.0
LEAD	32.1	5.3	10.4	33.4	6.5	11.3	33.5	5.2	10.4
TERMFREQ	36.5	7.0	12.6	38.5	9.0	14.2	35.7	6.5	12.0
LEXRANK	34.2	6.4	11.4	35.8	7.7	12.7	35.3	7.6	12.2
BERTQA	38.6	8.4	13.9	39.8	10.0	14.9	39.5	10.5	16.1
BERTMRC	**39.6**	7.8	13.6	39.9	8.9	14.3	36.6	8.4	13.2
MARGE-MN	39.0	**9.3**	**14.5**	**41.6**	**11.6**	**16.6**	38.8	10.5	15.9
+EXPAND	—	—	—	—	—	—	**45.9**	**18.8**	**23.0**
MARGE-CD	38.4	8.6	13.9	40.7	10.8	15.8	40.1	11.6	16.9
+EXPAND	—	—	—	—	—	—	**45.9**	18.3	22.7

Table 4.4: Performance of evidence rankers on extractive QFS. R-1, R-2 and R-SU4 stand for the F1 score of ROUGE 1, 2, and SU4, respectively.

sentence-similarity graphs which does not take query into account.

We summarize ranking and summarization results in Tables 4.3 and 4.4. As we can see, despite learning from weak signals, i.e., proxy queries and proxy answers, MARGE outperforms the strongest baseline, BERTQA, under both evaluation tasks. Without recourse to any question/answer annotations or dataset-specific retrieval methods, our model provides more informative input to the downstream generation task. As anticipated, query expansion (+EXPAND) gives a big boost on TD-QFS (which has short queries) leading to better coverage. A comparison between the outputs of MARGE for retrieval and summarization evaluation is shown in Table 4.5.

Ablation Studies Table 4.6 shows the outcome of various ablation studies which assess the effectiveness of masking and how to best instantiate it. Specifically, −Verb additionally treats verbs as information slots for sampling and masking; −Mask removes masking entirely so that the whole summary is revealed; −Query removes the proxy query (at training time) and the actual query (at inference time); this is to investigate whether our model simply learns to judge sentence salience based on its own features, instead of performing semantic matching with the given query; −OpenIE removes the dependency on Open IE and chooses words to mask at random. Specif-

Retrieval (Top 10 Ranked Sentences)	Summarization (250 Words)
1. In a lawsuit that goes to trial Monday, attorney Morris Dees of the Southern Poverty Law Center is representing a mother and son who were attacked by security guards for the white supremacist group.	1. In a lawsuit that goes to trial Monday, attorney Morris Dees of the Southern Poverty Law Center is representing a mother and son who were attacked by security guards for the white supremacist group.
2. Dees, founder of the Southern Poverty Law Center, has won a series of civil right suits against the Ku Klux Klan and other racist organizations in a campaign to drive them out of business.	2. Dees, founder of the Southern Poverty Law Center, has won a series of civil right suits against the Ku Klux Klan and other racist organizations in a campaign to drive them out of business.
3. Morris Dees, the co-founder of the Southern Poverty Law Center in Montgomery, ALA., and one of the attorneys for the plaintiffs, said he intended to enforce the judgment, taking everything the Aryan Nations owns, including its trademark name.	3. Morris Dees, the co-founder of the Southern Poverty Law Center in Montgomery, ALA., and one of the attorneys for the plaintiffs, said he intended to enforce the judgment, taking everything the Aryan Nations owns, including its trademark name.
4. He said Gilliam also told the informant someone should kill the FBI sniper who killed the wife of white supremacist randy weaver during an 11-day standoff in 1992 at ruby ridge, Idaho, along with civil rights lawyer Morris Dees of the Montgomery-based Southern Poverty Law Center.	4. He said Gilliam also told the informant someone should kill the FBI sniper who killed the wife of white supremacist randy weaver during an 11-day standoff in 1992 at ruby ridge, Idaho, along with civil rights lawyer Morris Dees of the Montgomery-based Southern Poverty Law Center.
5. Morris Dees, co-founder of the Southern Poverty Law Center in Montgomery, ALA., represented the Keenans and has said he intends to take everything the aryan nations owns to pay the judgment, including the sect's name.	5. Washington, March 3 (Xinhua) – the number of organized hate groups in the united states grew last year, mostly through new chapters of established white power organizations, the Southern Poverty Law Center said in a report released Tuesday.
6. Morris Dees, co-founder of the Southern Poverty Law Center and a crusader against intolerance, says the answer is not to censor the Internet.	6. The Southern Poverty Law Center, which was founded in the 1970s to battle racial bias, won major legal fights against the Ku Klux Klan and other white supremacist groups.
7. Triggs called Morris Dees, co-founder of the Southern Poverty Law Center, a non-profit civil rights organization, to ask what East Peoria could do.	7. Carrier said the Southern Poverty Law Center will distribute a million free copies of the booklet and a companion, "responding to hate at school."
8. Lawyer Morris Dees, the co-founder of the Southern Poverty Law Center who is representing Victoria Keenan and her son, Jason, introduced letters, photographs and depositions to contradict the men's testimony.	8. Over the last two decades, the Southern Poverty Law Center has taken the Ku Klux Klan and other hate groups to court, starting with a successful suit against the invisible empire Klan, which in 1979 attacked a group of peaceful civil rights marchers in Decatur, ALA.
9. Washington, March 3 (Xinhua) – the number of organized hate groups in the United States grew last year, mostly through new chapters of established white power organizations, the Southern Poverty Law Center said in a report released Tuesday.	
10. The Southern Poverty Law Center, which was founded in the 1970s to battle racial bias, won major legal fights against the Ku Klux Klan and other white supremacist groups.	

Table 4.5: Query modeling outputs of MARGE for cluster D0701A in DUC 2007. Retrieval evaluation (left) simply takes the top k ranked sentences (in this example $k = 10$), while summarization evaluation (right) further removes redundant sentences and includes sentences that do not appear in the top 10 list.

Models	DUC 2006	DUC 2007	TD-QFS
MARGE-MN	14.5	16.6	23.0
−Verb	↓0.5	↓0.3	↓2.8
−Mask	↓0.8	↓1.2	↓1.5
−Query	↓**2.9**	↓**2.9**	↓**12.6**
−OpenIE	↓0.9	↓1.1	↓2.1

Table 4.6: Ablation results on training data (absolute performance decrease in ROUGE SU4 denoted by ↓).

ically, we randomly mask 15% words in summaries as in BERT (Devlin et al., 2019) and merge adjacent `[MASK]` tokens. Performance drops in all cases, especially when queries are removed, underscoring the effectiveness of the proposed representation and training framework.

The Effect of Reveal Ratio We show how the mask reveal ratio γ affects model performance in Figure 4.3. As we can see, performance on the ROUGE regression task improves as γ increases; this is not surprising, the task becomes easier when fewer tokens are masked; when $\gamma = 1.0$, simply counting lexical overlap can solve the task perfectly. However, model performance on the QFS development set (DUC 2005) shows the opposite trend: actual queries *seek* information, instead of providing all the information needed. Therefore, the model is required to perform *semantic matching* (Guo et al., 2016) to accurately estimate evidence scores. Based on our empirical results, a simple but effective strategy is to mask all information slots (i.e., potential arguments) and reveal the rest of the words (including verbs) in the summary to construct proxy queries for training.

4.7.2 Abstractive Summarization

Automatic Evaluation Table 4.7 compares our model, which we call MARGESUM, against existing QFS systems. These include PQSUM-WSL (Laskar et al., 2020b) a supervised abstractive system which represents the state of the art on DUC benchmarks. It first extracts relevant sentences for each document with a QA model, it then replaces some of these with reference summary sentences via a paraphrase model, and uses them to further fine-tune BERTSUM (Liu and Lapata, 2019b). In its supervised incarnation, two years' DUC datasets are used for training and one for testing. QUERYSUM

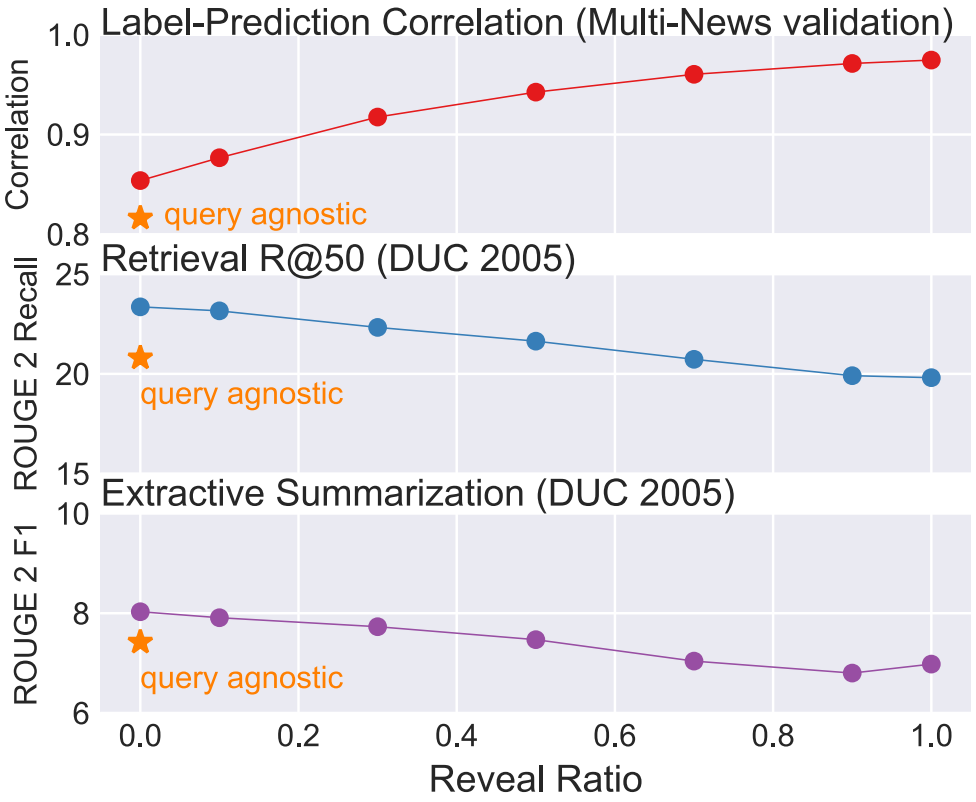


Figure 4.3: Model performance when reveal ratio γ is varied. Correlation refers to the average of Pearson's r correlation between the ground-truth and estimated ROUGE scores. The star marker denotes query-agnostic performance where all query tokens are masked, including information slots.

(Xu and Lapata, 2020) is state-of-the-art extractive system which adopts a coarse-to-fine process for salience estimation.

The second block compares our model with two *distantly supervised* approaches. BART-CAQ (Su et al., 2020) uses an ensembled QA model to extract answer evidence, and fine-tuned BART (Lewis et al., 2020) to iteratively generate summaries from paragraphs. PQSUM (Laskar et al., 2020b), uses fine-tuned BERTSUM to generate summaries for each document in a cluster, and a QA model to rank summary sentences against the query. Table 4.8 compares these models and our own in terms of their training requirements.

The third block presents the performance of UNILM fine-tuned on Multi-News and CNN/DailyMail following the standard setting in Bao et al. (2020). It uses no query guidance or length control. Documents are concatenated as input for training. During testing, sentences are selected with MARGE but ordered according to their original

Models	DUC 2006			DUC 2007			TD-QFS		
	R-1	R-2	R-SU4	R-1	R-2	R-SU4	R-1	R-2	R-SU4
PQSUM-WSL† (Laskar et al., 2020b)	43.5	10.8	16.5	44.7	12.4	17.7	—	—	—
QUERYSUM* (Xu and Lapata, 2020)	41.6	9.5	15.3	43.3	11.6	16.8	44.3	16.1	20.7
BART-CAQ (Su et al., 2020)	38.3	7.7	12.9	40.5	9.2	14.4	—	—	—
PQSUM (Laskar et al., 2020b)	**40.9**	9.4	14.8	42.2	10.8	16.0	—	—	—
UNILM-MN	34.6	6.7	11.8	35.5	7.6	12.3	36.2	8.1	12.9
UNILM-CD	37.6	8.3	13.6	39.6	10.1	14.9	40.1	11.8	16.7
MARGESUM-MN	39.1	9.1	14.3	42.1	11.7	16.5	40.8	11.6	16.5
MARGESUM-CD	40.2	**9.7**	**15.1**	**42.5**	**12.0**	**16.9**	**45.5**	**16.6**	**20.9**

Table 4.7: Performance of abstractive summarization systems. R-1, R-2 and R-SU4 stand for the F1 score of ROUGE 1, 2, and SU4, respectively. ∗/†: extractive/supervised method.

document position. The last block shows two variants of MARGESUM, optimized on Multi-News and a synthetic training set built from CNN/DailyMail. Both take as input sentences selected with MARGE-MN during inference due to its superior performance in query modeling (see Section 4.7.1).

As we can see, without requiring expensive QA data (see Table 4.8), MARGESUM-CD outperforms existing distantly supervised approaches. Its performance on DUC is on par with one of the strongest extractive systems, while on TD-QFS it is superior across metrics. Also note that MARGE trained on synthetic data outperforms MARGESUM-MN. Compared to Multi-News, synthetic summaries cover more topics and are less redundant, which is suited to QFS where there are usually multiple sub-queries to answer.

Examples of System Output We provide examples of summary output from DUC 2005, DUC 2006, and TD-QFS in Table 4.11, 4.12, and 4.13, respectively. In Table 4.11, both outputs from MARGESUM-CD and PQSUM have a good coverage of the main query focuses. Compared to PQSUM, MARGESUM-CD produces a more coherent summary for the given query narrative with a more natural topic flow. In Table 4.12, the output summary from PQSUM fails to respond to plans for future activity, while MARGESUM-CD covers all the aspects specified in the user query. The summary from MARGESUM-CD in Table 4.13 follows a similar general-to-specific pattern

Models	QA	PI	GS	QFS
BART-CAQ (Su et al., 2020)	✓	✗	✓	✗
PQSUM (Laskar et al., 2020b)	✓	✗	✓	✗
PQSUM-WSL (Laskar et al., 2020b)	✓	✓	✓	✓
UNILM (Bao et al., 2020)	✗	✗	✓	✗
MARGESUM	✗	✗	✓	✗

Table 4.8: Training requirements for existing QFS models (QA, PI, GS, and QFS stand for question answering, paraphrase identification, generic summarization and query focused summarization).

Models	DUC 2006	DUC 2007	TD-QFS
MARGE-CD	15.1	16.9	20.9
BERTQA	↓1.0	↓2.2	↓**6.1**
−Rank	↓**1.7**	↓**3.1**	↓1.3
−Length	↓0.1	↓0.5	↓0.2
−Query	↓0.5	↓0.3	↓0.4

Table 4.9: Ablations for MARGESUM trained on CNN/Daily Mail (performance decrease in ROUGE SU4 denoted by ↓).

as the gold summary. In comparison, the output from QUERYSUM provides details of a dementia category different from the given query and is less coherent.

Ablation Studies Table 4.9 presents the results of ablation studies on MARGESUM-CD. Replacing the input to the summarization component with sentences selected by BERTQA, the answer sentence selection model in Chapter 3, significantly decreases performance, demonstrating that sentences selected by MARGE are useful to downstream abstractive summarization. Removing evidence ranking altogether (−Rank) leads to a large performance drop; this is expected since sentence position information from the original documents does not transfer well to QFS settings. Removing length control (−Length) also hurts performance as does the removal of query guidance (−Query) at inference time.

Human Evaluation We also evaluated model summaries in a judgment elicitation study via Amazon Mechanical Turk. Native English speakers (self-reported) were

DUC	Rel	Suc	Coh
PQSUM-WSL	**2.95**	**3.27**	$2.93^{\dagger\circ}$
QUERYSUM	2.79	3.13	$2.94^{\dagger\circ}$
UNILM-CD	$2.43^{\dagger\circ}$	3.09	3.27
MARGESUM-CD	2.91	3.25	**3.30**
GOLD	3.05	3.29	3.35

TD-QFS	Rel	Suc	Coh
QUERYSUM	4.32	3.90°	$3.80^{\dagger\circ}$
UNILM-CD	$3.63^{\dagger\circ}$	**4.12**	4.28
MARGESUM-CD	**4.55**	4.02	**4.37**
GOLD	4.70	4.23	4.60

Table 4.10: Human evaluation results on DUC (above) and TD-QFS (below): average **Rel**evance, **Suc**cinctness, **Coh**erence ratings; †: sig different from MARGESUM-CD; $^{\circ}$: sig different from Gold (at $p < 0.05$, using a pairwise t-test).

asked to rate query-summary pairs on two dimensions: *Succinctness* and *Coherence*. The ratings were obtained using a fivepoint Likert scale. In addition, participants were asked to assess the *Relevance* of the summary to the query at sentence-level. Sentence scores were averaged to obtain a relevance score for the whole summary. Detailed instructions of human evaluation can be found in Appendix A.

Participants assessed summaries created by PQSUM-WSL, the state-of-the-art abstractive system, QUERYSUM, a state-of-the-art extractive system, UNILM-CD, and MARGESUM-CD.[4] We also randomly selected GOLD standard summaries to include as an upper bound. We sampled 20 query-cluster pairs from DUC (2006, 2007; 10 from each set), and 20 pairs from TD-QFS (5 from each cluster) and collected three responses per pair.

Table 4.10 shows the human ratings for each system. On both DUC and TD-QFS, participants perceive MARGESUM-CD on par with PQSUM-WSL in terms of query relevance and summary succinctness, while significantly better than PQSUM-WSL and QUERYSUM in terms of coherence. In fact, participants find summaries PQSUM-WSL summaries as incoherent as those created by the extractive QUERYSUM; this is proba-

[4]We include PQSUM-WSL only for human evaluation on DUC since it was not evaluated on TD-QFS (Laskar et al., 2020b) and system output is not available.

bly due to the fact that PQSUM-WSL first generates an abstractive summary for each document and then re-ranks the generated sentences. Therefore, final summary sentences are less related to each other. Summaries from our system are also considered significantly more relevant than UNILM-CD. Compared to PQSUM-WSL, although UNILM-CD is not good at producing relevant content, it maintains relatively higher coherence, demonstrating the effectiveness of training abstractive systems with synthetic data from SDS and generating long summaries at once.

4.8 Summary

In this chapter we proposed an abstractive framework for query focused summarization. We provided a unified mask representation for summaries and queries, which enables summaries to serve as proxy queries for model training. As a result, a query model can be trained with generic summarization data without relying on additional question-answering resources. Experimental results across datasets show that the proposed system yields state-of-the-art performance despite the weakly supervised setting, and produces more relevant and coherent summaries compared to existing approaches.

Both proxy query modeling (this chapter) and distant query modeling (Chapter 3) assume prior knowledge of the query form at test time. Under this assumption, systems are trained for a specific query type and how to scale them to handle out-of-distribution queries becomes a nontrivial problem. In the next chapter, we will focus on the scalability of QFS systems, and handle various query types with a unified framework under which generic summarization can be viewed as a special case.

Query: STEROID USE AMONG FEMALE ATHLETES – *Discuss the prevalence of steroid use among female athletes over the years. Include information regarding trends, side effects and consequences of such use.*
GOLD: Increasing numbers of female athletes use anabolic steroids. Among female college athletes, an estimated 5% in swimming, 6% in basketball and 10% in track and field have used steroids. The fastest growing group of users is adolescent females. New steroids are often marketed as dietary supplements and girls are unaware of what they are taking. A young tennis player tested positive for steroids after taking an "all natural" supplement later discovered to contain a variation on the banned steroid nandrolone. A former Olympic track coach believes at least 40% of the U.S. women's team in Seoul had used steroids at some time. Because of their naturally low testosterone levels, steroids have a more dramatic effect on women, boosting their levels up to 10 times. The health effects can be dramatic, including liver damage and tumors, elevated cholesterol, heart attacks, strokes, stunted growth in adolescents, infertility, uncontrollably violent behavior, chronic depression, deeper voices, excessive facial hair, and acne. Many East German female athletes were given steroids, often unbeknownst to them or their parents and suffered serious side-effects. Athletes who have tested positive for steroids include a Bulgarian triple jumper, a Romanian hammer thrower, a Russian runner, a Dominican high, a Jamaican sprinter, a Spanish pole vaulter, a German marathon runner, two Moroccan athletes, and two Chinese Taipei weightlifters. A number of female Chinese athletes have tested positive, including several swimmers and rowers. International organizations impose bans of between two months and life on athletes found guilty of using illegal steroids.
UNILM-CD: At least 23 Chinese athletes, mostly women, have tested positive for steroid use since 1990. Such side-effects of anabolic steroids have been generally known since 1977. Some of the athletes were as young as 12 when they started taking the steroids. EPO is one of at least 20 steroids prohibited by the International Olympic Committee. The drug is legal and sold over-the-counter, but is banned by the National Olympic Committee, the National Football League and the National Collegiate Athletic Association. The National College Athletic Association (ACA) has banned EPO for life. EPO has been linked to a number of health problems, including liver damage and liver cancer. The ACA says EPO's use in the U.S.is safe, effective and safe to use. The National Institute of Drug Abuse (NIDA) is investigating the use of EPO in the United States. The NIDA says it has no evidence to support the claim that EPO can cause liver damage. The drug was approved for use by the U.S. Olympic committee in 1996. The National Association of College Athletes (NAA) is calling for the drug to be banned from all U.S. sports. The NAA says it's time for the U.S. Olympic Committee (IOC) to ban EPO. The drug has been used by more than 200,000 athletes worldwide since it was approved in 1996.

Table 4.11: System outputs for cluster D0602B in DUC 2006. The gold summary answers the query covering four main aspects (denoted with different colors): (1) trend; (2) side-effects; (3) consequences of such use; (4) historical cases.

PQSUM: About 175,000 teen age girls in the United States have reported using anabolic steroids at least once within a year of the time surveyed a rise of 100 percent since 1991. In 1996 Samantha Reeves became the first female tennis player to have tested positive for steroids. The rise of steroid use in female athletes has led to a rise in women's steroid use. Over the years the number of female steroid related women in the U.S has increased There is no mistaking the dangers of steroids for women with documentation provided by trainers and coaches from the former east german sports empire. In 1996 women were banned from the olympics for a positive test of methandienone. The women involved suffered serious side effects from the drugs and that they knew of the potential danger. She was found to have steroid stanozolol in her system. In the last month Cuban sprinter Merlene Ottey withdrew from the world.championships after testing positive for nandrolone. Some were as young as 12 when they started taking the anabolic steroid pills. The drug use has been generally known since 1977. Pospelova would be the seventh athlete to test positive at the games. Such side effects of anabolic steroids are generally known as with all probability linked to doping. An over the counter supplement called andro raises testosterone and estrogen above normal levels and could be dangerous according to a harvard study by major league baseball and its players union. In 1996.
MARGESUM-CD: Penn State professor Charles Yesalis estimates the use of steroids among female athletes at 5 percent in swimming, 6 percent in basketball and 10 percent in track and field, a rise of 100 percent since 1991. The national institutes of drug abuse says 175,000 teenage girls in the united states have reported taking anabolic steroids at least once within a year of the time surveyed. The national institute on drug abuse provides information regarding trends, side effects and consequences of such use. Two Moroccan female athletes have been stripped of gold and bronze medals for using a muscle-building steroid in the first reported cases of doping at the Arab games for using the steroid nandrolone, a steroid that has been linked to liver cancer, heart disease and uncontrollable aggressiveness. Two medical experts testifying in the doping trial of a former east german sports doctor say the female swimmers they examined showed health damage linked to performance-enhancing drugs, including liver damage and excessive facial hair. The study, published in Wednesday's Journal of the American Medical Association, is the first to conclude that high doses of the steroids can elevate testosterone levels and that the hormone can be used as a performance-enhancing steroid, such as epitestosterone, as a marker the testosterone is 6 to 1 in the male sex hormone and 5 to 1 for the female steroid hormone epitestoterone - a metabolite that is used as an indicator of testosterone use - the female sex hormone.

Table 4.11: Continued.

Query: DEEP WATER EXPLORATION – *What is being learned from the study of deep water, seabeds, and deep water life? What equipment and techniques are used? What are plans for future related activity?*
GOLD: Deep-sea explorations worldwide have led to historic discoveries. US teams discovered five Roman shipwrecks during a Mediterranean exploration and two deep-water sea wrecks off the coast of Israel. Divers discovered pharaonic statues, sphinxes, and Byzantine coins off Egypt's Mediterranean coast. One exploration off the New York coast searched for signs of human settlements predating land ones. The Chinese built a robot to survey ocean depths of 6,000 meters. They have also made advances in the exploration of seabed polymetallic nodules. China collected natural benthal electric data using an electromagnetic survey machine at 200 meters under the sea. The country planted seabed forests to restore damage to its coastline from marine development, offshore petroleum exploration and industrial development. Through deep-sea exploration scientists have learned about the ocean environment. Scientists recovered the first rock sample from under the Antarctic Ocean bed. They found that animals living on the ocean floor seem as susceptible to climate as surface dwellers. They found oscillations involving deep ocean currents that appear similar to El Nino and La Nina. They discovered cracks in the seabed that might set off a tsunami, threatening Mid-Atlantic States. Dutch scientists found toxic chemicals in whales that feed in the Atlantic Ocean. US scientists have improved craft and submersibles used for sea explorations in response to international competition. In 2000 President Clinton announced federal participation in new undersea exploration projects on the East, West, and Gulf coasts and called for recommendations for new ocean exploration from a Commerce Department advisory panel.
UNILM-CD: Shell will also carry out deep water explorations in the waters of Angola, Congo and Cote d'Ivoire. Herakleum is the second city which was found deep in the Mediterranean water after Menouthis was discovered in 1934. The oil reserve in the deep waters of the west Africa is as high as 15 billion barrels. Shell has already started its deep-water exploration in Nigeria through the shell Nigeria exploration company limited. The first five licenses to drill for oil in the Israeli deep waters are granted. The "Bonga" oil field is located 120 kilometers off the coast of the oil-producing Niger Delta region in water more than 1,000 meters deep. The company hopes to find evidence of a trade route directly across the sea through deep water. New: shell says it's working with the Nigerian government to develop the oil field. New: the company says it will start drilling in the middle of the year. The oil reserves in the Gulf of Mexico are estimated to be up to 10 billion barrels. The company has been drilling for the oil for more than a decade. New: U.S. state department says it is working with Nigeria to develop its oil reserves. New: Nigeria's president says he is committed to developing the country's oil reserves. New: "we're going to work with the united states on this," the president says.

Table 4.12: System outputs for cluster D0737I in DUC 2007. The gold summary answers the query covering three main aspects (denoted with different colors): (1) discoveries; (2) equipment and techniques; (3) plans for future related activity.

PQSUM: The research of deep water seabeds and deep water life is being used to study the ocean floor at 6000-meter depth. China has developed a complete set of deep-sea mining techniques and plans for commercial mining in the next century. In the Black Sea, a team from Woods Hole Oceanographic Institution and MIT hopes to find evidence of harbors dating from an era when the waters were hundreds of feet shallower and perhaps shipwrecks that could reveal ancient deep water trading routes possibly the old silk road. The robot completed its deep water video record measurements of the ocean bottom investigation of water conditions and search for missing objects while automatically recording data. The world's oldest known deep water sea wrecks were found in Tel Aviv Israel by an underwater robot deep water tracking equipment and a global positioning satellite. The deep ocean currents in effect create a global conveyor belt that transport great amounts of heat around the world. Scientists say they have found cracks in the seabed that could set off a tsunami sending waves as high as 20 feet speeding toward the mid-Atlantic states. Scientists have found toxic industrial chemicals in the tissue of whales that normally feed in deep waters of the Atlantic Ocean. The New Zealand mission is part of a global initiative to determine how the oceans tick and influence climate. In 2000 Shell's deep water oil exploration in its Bonga field came up with exciting results.
MARGESUM-CD: The discovery of natural gas-hydrate resources in the seabed of the Xisha Isles of the South China Sea is being learned from the study of deep water, seabeds, and deep water life. American explorer Robert Ballard has located their two ships – the world's oldest known deep-water sea wrecks – using the same techniques he used to find the titanic. Underwater deep water drilling equipment and techniques are used. There are plans for future related activity in New Zealand waters in August and September with the launch of an advanced French research ship to carry out deep sea drilling in the middle of the Atlantic ocean in the next few months. In the East Pacific, China has recorded new advances in the exploration of seabed polymetallic nodules in the Eastern Pacific and planning for commercial mining in next century. The China Ocean Mineral Resources Research and Development Association (COMRRDA) says the discovery is the first of its kind in China and plans for commercial exploration in the coming century, according to the group's president, David Mindell, who will lead the black sea expedition to find evidence of a trade route across the sea through deep water in the Antarctic Ocean bed 30-100 million years ago. The team's first rock sample from deep beneath the Antarctic sea bed is expected to reveal what happened in the frozen continent 30-50m above the earth's surface in the mid-20th century and the next 50m below the surface of the sea floor in the deep sea.

Table 4.12: Continued.

Query: *Semantic Dementia*
GOLD: Dementia is not a specific disease. It is used to describe a set of symptoms that occur when the brain is damaged by specific diseases. One of these diseases is semantic dementia. Semantic dementia was first described by Arnold Pick in 1904. Semantic dementia (SD) is a progressive neurodegenerative disorder characterized by loss of semantic memory in both the verbal and non-verbal domains. It is characterized as a primary progressive aphasia. However, the most common presenting symptoms are in the verbal domain (with loss of word meaning). Sometimes, SD patients show symptoms of surface dyslexia. They often present with the complaint of word-finding difficulties. Clinical signs include fluent aphasia, anomia, impaired comprehension of word meaning, and associative visual agnosia (inability to match semantically related pictures or objects). As the disease progresses, behavioural and personality changes are often seen similar to those seen in frontotemporal dementia although cases have been described of 'pure' semantic dementia with few late behavioural symptoms. When Structural MRI imaging is performed on SD patients it shows a characteristic pattern of atrophy in the temporal lobes (predominantly on the left), with inferior greater than superior involvement and anterior temporal lobe atrophy greater than posterior. Based on these imaging methods, semantic dementia can be regionally dissociated from the other subtypes of frontotemporal lobar degeneration, frontotemporal dementia and progressive nonfluent aphasia. Semantic dementia is a form of dementia that is different from other more common forms such as Alzheimer's.
UNILM-CD: Alzheimer's disease is a progressive neurodegenerative disorder characterized by loss of semantic memory in both the verbal and non-verbal domains. The most common presenting symptoms are in the verbal domain however it is characterized as a primary progressive aphasia. SD is one of the three canonical clinical syndromes associated with Frontotemporal Lobar Degeneration (FTLD). Researchers at the Linda Crnic Institute for Down Syndrome and Niemann Pick-C Disease found that cholesterol wreaks havoc on the orderly process of cell division, leading to defective daughter cells throughout the body. Researchers believe in the value of live training and the impact that training has on the staff. The NCCDP certified trainers believe that live training has the potential to improve the quality of life for people with Alzheimer's and other neurological disorders. A new study by the University of Wisconsin-Madison has found that the brain of Alzheimer's sufferers is damaged. The brain is damaged when the disease progresses and can lead to memory loss, confusion and even death. The findings could lead to the development of new treatments for the disease, which is estimated to cost the U.S. at least $1 billion a year. The study is the first to quantify the damage caused by Alzheimer, AOS disease, and could one day be used to develop new drugs to treat the disease. The researchers hope that by the end of the century there will be at least 100,000 new Alzheimer's patients in the U.S.

Table 4.13: System outputs for cluster 3-3 in TD-QFS. The given query, *Semantic Dementia*, is a type of dementia. The gold summary starts with general information of dementia, and then progresses to details of semantic dementia, including its proposal, characteristics and symptoms. Details of vascular dementia which is a different category of dementia is also highlighted.

QUERYSUM: These diseases include Alzheimer's disease and vascular dementia. Some symptoms may be similar to those of other types of dementia, such as Alzheimer's disease. Memory loss is a distressing part of dementia, both for the person with dementia and for those around them. The term dementia is used to describe a set of symptoms that occur when the brain is damaged by specific diseases. Vascular dementia, which occurs after a stroke, is the second most common dementia type. For a person with Alzheimer's or dementia, poor nutrition may increase behavioral symptoms and cause weight loss. When vascular dementia develops after an obvious stroke, it is sometimes called post-stroke dementia (or single-infarct dementia). There are different types of vascular dementia. Dementia is caused by damage to brain cells. Alzheimer's disease is distinguished from other forms of dementia by characteristic changes in the brain. Some people may experience sub-cortical vascular dementia as well as stroke. Although the brain damage that causes vascular dementia cannot be reversed, it may be possible to slow the progression of the disease in a number of ways. Sub-cortical vascular dementia is sometimes referred to as small vessel disease. Alzheimer's disease is the most common cause of dementia, affecting around 496,000 people in the UK. The speed that vascular dementia progresses varies from person to person. Sub-cortical vascular dementia is caused by damage to the tiny blood vessels that lie deep within the brain. There is also a specific form of sub-cortical vascular dementia called Binswanger's disease.
MARGESUM-CD: The American Psychiatric Association's diagnostic and statistical manual of mental disorders characterizes Alzheimer's disease as a chronically progressive, global, cognitive loss. Older memories of the person's life (episodic memory), facts learned (semantic memory) and implicit memory (the memory of the body on how to do things) are affected to a lesser degree than new facts or memories. it was first described by Arnold Pick in 1904 and in modern times was characterized by professor Elizabeth Warrington in 1975 as a primary progressive aphasia, frontotemporal dementia and progressive nonfluent aphasia. Semantic dementia is a progressive neurodegenerative disorder characterized by loss of semantic memory in both the verbal and non-verbal domains. Behavioral and personality changes are often seen similar to those seen in frontotemporal lobar degeneration, frontotemporal and progressive nonfluent aphasias, as well as associative visual agnosia (inability to match semantically related pictures or objects). The most common presenting symptoms are in the verbal domain, with loss of word meaning, impaired comprehension of word meanings, and impairments in semantic memory (memory of meanings, concepts, and abstract thinking). Semantic dementia can be regionally dissociated from in the other subtypes of frontotemporal Alzheimer and progressive Alzheimermant dementia although cases have been described of pure semantic dementia with few late behavioral symptoms. Clinical signs include fluent aphasia, anomia, impaired understanding of word meaning, impaired ability to retrieve memories and changes in emotion and personality meta-analyses on MRI and FDG - pet studies confirm these findings by identifying alterations in the inferior temporal poles and amygdalae as the hotspots of disease - brain regions that have been discussed in the context of conceptual knowledge, semantic information processing, and social cognition.

Table 4.13 Continued.

Chapter 5

Document Summarization with Latent Queries

In the previous chapters, we have described several query modeling methods for QFS, including leveraging distant supervision from QA, and generating proxy queries from generic summaries to address the data paucity problem. These methods have shown effectiveness on standard DUC benchmarks, which is not surprising since they were developed to handle DUC-standard queries (i.e., a short title following by a long narrative), based on knowledge of the target query form, e.g., from official DUC examples or a development set. Nevertheless, user queries in real-world scenarios can be verbalized in various ways, from simple keywords to natural questions, and assuming prior knowledge of a specific language realization is neither realistic nor computationally scalable.

In addition, existing research views QFS and generic summarization as two distinct summarization tasks: architecture designs and training strategies specifically developed for QFS cannot be easily applied to generic summarization tasks. As a result, two separate summarization systems need to be trained and deployed to produce both generic and query-focused summaries which we argue is an inefficient solution.

In this chapter, we propose a unified modeling framework for any kind of summarization, including QFS with various query types and generic summarization. To this aim, we assume that all summaries are a response to a query, which is observed in the case of QFS and latent in the case of generic summarization. We model queries as discrete latent variables over document tokens, and learn representations compatible with observed and unobserved query verbalizations. Our framework formulates summarization as a generative process, and jointly optimizes a latent query model and a con-

ditional language model. Despite learning from generic summarization data only and requiring no further optimization for downstream summarization tasks, our approach outperforms strong comparison systems across benchmarks, query types, document settings, and target domains.

5.1 Introduction

In the previous chapters, we have discussed the scarcity of training data in QFS, a fundamental research challenge in QFS: unlike generic summarization for which many large-scale datasets have been proposed recently to enable the training of end-to-end neural summarization systems (Hermann et al., 2015; Narayan et al., 2018a; Fabbri et al., 2019), existing QFS benchmarks (Dang, 2005; Baumel et al., 2016) which are relatively small in size have been primarily used for system evaluation. To make up for the absence of labeled QFS data, since the proposal of the coarse-to-fine framework in Chapter 3, a new line of work has resorted to distant supervision provided by pretrained models, paraphrase identification, and question-answering datasets (Su et al., 2020; Laskar et al., 2020b). As query-related resources can also be expensive to acquire (Bajaj et al., 2016), we presented in the last chapter an alternative approach which eliminates this dependency via the induction of proxy queries from generic summarization data, achieving state-of-the-art performance in the few-shot setting where a small QFS development set is used.

Despite this progress, the diversity of query types, another research challenge introduced in Section 1.3, remains understudied. Table 5.1 shows examples of various query types in existing QFS benchmarks. The experimental results from the previous chapters have shown that QFS systems can potentially handle queries resembling those seen in training, however, they are not expected to work well on out-of-distribution queries, i.e., queries with different surface forms from those seen in training. For instance, due to the distribution divergence between questions and queries, the answering module optimized with QA resources can only perform well on QFS when paired with the retrieval and summarization modules, as shown in Table 3.8. In the last chapter, the reveal ratio, which determines the distribution of proxy queries for training, also largely affects query modeling performance (see Table 4.3). This makes it challenging to scale existing QFS frameworks well over a variety of query expressions. For a trained proxy query model from the last chapter, it might be necessary to execute the following steps to cover new types of queries:

1. Gather more data to obtain knowledge about the new queries, including statistical information such as query length and content.

2. Re-design proxy queries to imitate the target queries, and generate training data accordingly.

3. Re-train one or more system components that rely on the query distribution which has shifted.

As we can see, steps 2–3 are not computationally scalable since data construction and model training have to be repeated for every new incoming query type. Besides, the exact expressions that users input to the summarization system may be unpredictable at test time, so step 1 in some cases may be infeasible in practice. In fact, users may prefer to leave the query *empty* if they do not have a specific query in mind: in this case, the system should simply *recommend* information which might answer a latent query (e.g., *What is important in this document?*), which is, by definition, a *generic* summary. However, building and maintaining a separate generic summarization system for this purpose is also computationally inefficient.

To address these scalability issues, in this chapter, we provide a unified modeling framework for generic summarization *and* QFS, under the assumption that only data for the former is available. Specifically, we treat generic summarization as a special case of QFS where the query is *latent*. We model queries as *discrete latent variables* over document tokens, and learn representations compatible with observed and unobserved query verbalizations. Our framework formulates abstractive summarization as a generative process, and decomposes the learning objective into:

1. **Latent query modeling**: Generating latent query variables from document observations.

2. **Conditional language modeling**: Generating summaries conditioned on observed documents and latent queries.

To further handle optional user queries at test time, we propose a non-parametric calibration of the latent query distribution which allows us to perform *zero-shot* QFS without model re-training.

Our contributions in this chapter are threefold: (a) we bring together generic summarization and QFS under a unified modeling framework which does not require query-related resources for training or development; (b) we provide a deep generative

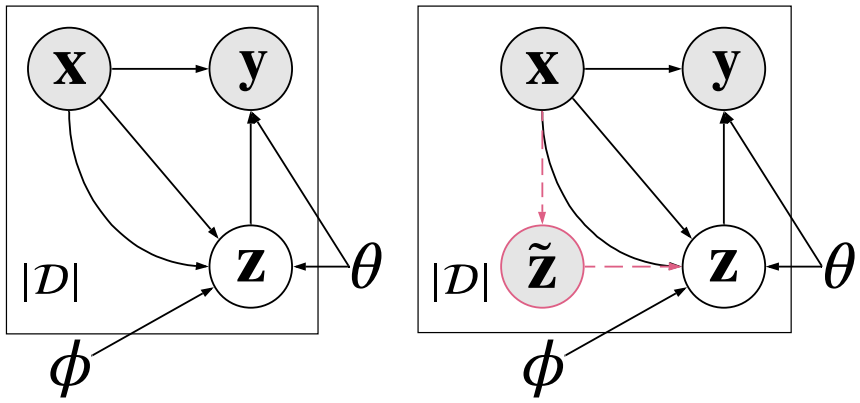
formulation for document summarization, where queries are represented *directly* from input documents in latent space, i.e., without resorting to pipeline-style query extraction or generation; and (c) experiments on a range of summarization benchmarks show that across query types, document settings, and target domains, our model achieves better results than strong comparison systems.

5.2 Related Work

A simple neural encoder-decoder architecture was originally applied to generic abstractive summarization (Rush et al., 2015; Nallapati et al., 2016), and was later enhanced with a copy mechanism (See et al., 2017), content selection (Gehrmann et al., 2018), pretrained models (Liu and Lapata, 2019b; Lewis et al., 2020), and features which control the length or content of the summary (Cao et al., 2018; Dou et al., 2021). We reviewed these existing approaches for generic summarization in Section 2.2.1.

In comparison to its generic summarization, abstractive QFS has received significantly less attention due to data paucity, as discussed in the previous chapter. In Section 4.2, we introduced a line of research adopting query-related resources to generate query focused abstracts (Su et al., 2020; Laskar et al., 2020b). Since query-related resources can be also costly to obtain (Bajaj et al., 2016; Kwiatkowski et al., 2019), the abstractive system we proposed in the last chapter, MARGESUM, employs none whatsoever. Instead, we create proxy queries by selectively masking information slots in generic summaries. Despite promising system performance, MARGESUM assumes prior knowledge of target queries (proxies are created to match their length, and content), and a development set is used. Also, it is particularly tailored to multi-document QFS and incorporates a sophisticated evidence selection component. The methodology in this chapter is closely related to MARGESUM in that we also do not take advantage of query-related training resources. We take a step further in this chapter and do not require a development set either, allowing our model to produce QFS summaries in *zero-shot* settings.

Our approach is generally applicable to single- and multi-document QFS. We assume for both generic summarization and QFS that queries are latent and estimate these jointly via a summarization and (weakly supervised) tagging task. The latter draws inspiration from Gehrmann et al. (2018) under the assumption that document tokens found in the summary also provide evidence for the (latent) query that gave rise to it. Finally, our model is fundamentally different from approaches which rely on



(a) Generative Process: Training (b) Generative Process: Testing

Figure 5.1: Generative processes of the proposed summarization framework. Dashed lines denote optional queries at test time. Shaded nodes represent observed variables, unshaded nodes indicate latent variables, arrows represent conditional dependencies between variables, whereas plates refer to repetitions of sampling steps.

document-based guidance to improve the informativeness (Cao et al., 2018) or faithfulness (Chen et al., 2021) of summaries. While these models exploit guidance from supervision signals in training data, we are faced with the problem of estimating queries when there are none available (at least during training).

5.3 Problem Formulation

Let $\{(D,Q,S)\}$ denote a summarization dataset, where document D is a sequence of tokens, and S its corresponding summary; query Q additionally specifies an information request. In generic summarization, $Q = \emptyset$, whereas in QFS Q can assume various formats, ranging from keywords to composite questions (see Table 5.1 for examples).

Our model learns from generic summarization data alone, while robustly generalizing to a range of tasks at test time, including out-of-domain QFS. A shared characteristic between generic summarization and QFS is the fact that user intent is *underspecified*. Even when queries are available (i.e., $Q \neq \emptyset$), as shown in the last chapter, they are incomplete expressions of intent as it is unlikely to specify queries to the level of detail necessary to compose a good summary We thus identify *latent* query signals from D, and optionally take advantage of Q as additional observation for belief update.

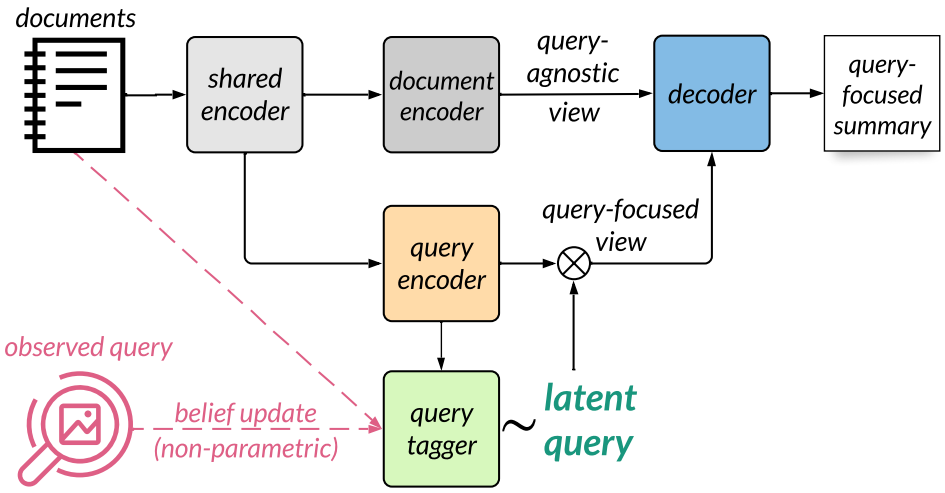


Figure 5.2: Neural parametrization of the proposed summarization framework. Dashed lines denote optional queries at test time. Latent queries create a query-focused view of the input document, which together with a query-agnostic view serve as input to a decoder for summary generation.

Generative Model We model an observed input document D as a sequence of random variables $\mathbf{x} = [\mathbf{x}_1; \mathbf{x}_2; \ldots; \mathbf{x}_M]$ where $\mathbf{x}_i$ is a token and M the length of the document. We define the *latent query* as a sequence of discrete latent states over input document tokens: $\mathbf{z} = [\mathbf{z}_1; \mathbf{z}_2; \ldots; \mathbf{z}_M]$. Specifically, from each document token $\mathbf{x}_i$, we generate a binary query variable $\mathbf{z}_i$, whose distribution $p(\mathbf{z}_i)$ represents the belief that $\mathbf{x}_i$ contributes to a potential query for document D. Modeling latent queries at the token-level allows us to regularize the model, i.e., by taking into account weak supervision in the form of token-level tagging (Gehrmann et al., 2018). It also renders the model independent of the query form, thereby enabling zero-shot inference (see Section 5.4).

The output summary $\mathbf{y} = [\mathbf{y}_1; \mathbf{y}_2; \ldots; \mathbf{y}_T]$ is then generated from $\{\mathbf{x}, \mathbf{z}\}$ using teacher-forcing at training time. At test time, we may additionally be presented with a query Q; we *ground* this optional information to the input document via discrete *observed* variables $\tilde{\mathbf{z}} = [\tilde{\mathbf{z}}_1; \tilde{\mathbf{z}}_2; \ldots; \tilde{\mathbf{z}}_M]$, and generate $\mathbf{y}$ by additionally conditioning on $\tilde{\mathbf{z}}$ (if it exists) in an autoregressive manner.

Our model estimates the conditional distribution $p_\theta(\mathbf{y}|\mathbf{x})$ according to the generative process just described (and illustrated in Figure 5.1) as:

$$p_\theta(\mathbf{y}|\mathbf{x}) = \sum_{\mathbf{z}} p_\theta(\mathbf{y}|\mathbf{z},\mathbf{x}) p_\theta(\mathbf{z}|\mathbf{x}) \tag{5.1}$$
$$= \sum_{\mathbf{z}} p_\theta(\mathbf{y}|\mathbf{z},\mathbf{x}) \prod_i p_\theta(\mathbf{z}_i|\mathbf{x}_i)$$

Inference Model The posterior distribution of latent variable $\mathbf{z}$ is calculated as:

$$p_\theta(\mathbf{z}|\mathbf{x},\mathbf{y}) = \frac{p_\theta(\mathbf{x},\mathbf{y},\mathbf{z})}{p_\theta(\mathbf{x},\mathbf{y})} = \frac{p_\theta(\mathbf{x},\mathbf{y},\mathbf{z})}{\sum_\mathbf{z} p_\theta(\mathbf{x},\mathbf{y},\mathbf{z})}. \tag{5.2}$$

Unfortunately, exact inference of this posterior is computationally intractable due to the joint probability $p_\theta(\mathbf{x},\mathbf{y})$. We therefore approximate it with a variational posterior $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})$. Inspired by β-VAE (Higgins et al., 2017), we maximize the probability of generating summary $\mathbf{y}$, provided the distance between the prior and variational posterior distributions is below a small constant δ:

$$\max_{\phi,\theta} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p(\mathbf{x},\mathbf{y})} \left[\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})} \log p_\theta(\mathbf{y}|\mathbf{x},\mathbf{z})\right] \tag{5.3}$$

$$\text{subject to } D_{\mathrm{KL}}\left(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})||p_\theta(\mathbf{z}|\mathbf{x})\right) < \delta \tag{5.4}$$

Since we cannot solve Equation (5.4) directly, we invoke the Karush-Kuhn-Tucker conditions (KKT; Karush 1939; Kuhn et al. 1951) and cast the above constrained optimization problem into unconstrained optimization, with the following ELBO objective:[1]

$$\mathcal{L}_{\mathrm{ELBO}} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log p_\theta(\mathbf{y}|\mathbf{x},\mathbf{z})\right] - \beta D_{\mathrm{KL}}\left(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})||p_\theta(\mathbf{z}|\mathbf{x})\right) \tag{5.5}$$

where the Lagrangian multiplier β is a hyperparameter. To minimize our model's dependence on queries (which we assume are unavailable for both training and development), we adopt a uniform prior $p_\theta(\mathbf{z}|\mathbf{x})$. In other words, the probability of variable $\mathbf{z}$ being a query word (given all instances of $\mathbf{x}$) follows a uniform distribution. In this case, minimizing the KL term in Equation (5.5) is equivalent to maximizing the entropy of the variational posterior.[2] We further assume that the tokens observed in a document are a superset of potential query tokens, and therefore $\mathbf{z} \perp\!\!\!\perp \mathbf{y}$ and $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y}) = q_\phi(\mathbf{z}|\mathbf{x})$.[3]

While the simplification reduces the risk of exposure to bias from training on $\mathbf{y}$, it makes learning meaningful latent variables more challenging as they depend solely on $\mathbf{x}$. We alleviate this by introducing a new type of weak supervision $o(\hat{\mathbf{z}}|\mathbf{x},\mathbf{y})$ which we automatically extract from data (i.e., document-summary pairs). Essentially, we

[1] For a constrained optimization problem $\max f(\mathbf{x}),\ s.t.\ g(\mathbf{x}) \le 0,\ \mathbf{x} \in \mathbb{R}^n$, if $\mathbf{x}^\star$ is a local optimum and the optimization problem satisfies regularity conditions, then there exists a constant μ such that the following four groups of KKT conditions hold: (1) primal feasibility: $g(\mathbf{x}^\star) \le 0$, (2) dual feasibility: $\mu \ge 0$, (3) complementary slackness: $\mu g(\mathbf{x}^\star) = 0$ and (4) stationarity: $-\nabla f(\mathbf{x}^\star) + \mu \nabla g(\mathbf{x}^\star) = 0$.

[2] When $p_\theta(\mathbf{z}|\mathbf{x}) \sim \mathcal{U}(a,b)$, $D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})||p_\theta(\mathbf{z}|\mathbf{x})) = -\mathcal{H}\left(q_\phi(\mathbf{z}|\mathbf{x})\right) + \log(b-a+1)$ always holds ($\mathbf{z} \in [a,b]$).

[3] We experimentally verified this assumption in several QFS datasets. In WikRef (Zhu et al., 2019) and Debatepedia (Nema et al., 2017), 1.57% and 4.27% of query tokens are not attested in the input document, respectively. In DUC (Dang, 2005) and TD-QFS (Baumel et al., 2016) where the input contains multiple documents, all query tokens are attested. Across all datasets, only 1.69% of query tokens are not attested in the input document/cluster.

tag tokens in the document as likely to be in the summary and by extension in the query. We discuss how this tagger is learned in Section 5.4. For now, suffice it to say that weak supervision is a form of posterior regularization adding an extra term in the objective which we rewrite as:

$$\mathcal{L} = \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{y}|\mathbf{x},\mathbf{z})\right]}_{\text{conditional language modeling}} + \underbrace{\beta\mathcal{H}\left(q_\phi(\mathbf{z}|\mathbf{x})\right) - \omega\mathcal{H}\left(o(\hat{\mathbf{z}}|\mathbf{x},\mathbf{y}), q_\phi(\mathbf{z}|\mathbf{x})\right)}_{\text{latent query modeling}} \tag{5.6}$$

where $\mathcal{H}(\cdot)$ denotes posterior entropy and $\mathcal{H}(\cdot,\cdot)$ denotes cross entropy.

As can be seen from Equation (5.6), we decompose summarization into two modeling objectives, namely *latent query modeling* and *conditional language modeling*. Inside the query modeling term, hyperparameter ω controls the influence of weak supervision $\hat{\mathbf{z}}$, while β controls the strength of label smoothing on the weak annotations.

Neural Parametrization We parametrize the two objectives in Equation (5.6) with a *latent query model* and a *conditional language model* illustrated in Figure 5.2. The query model estimates latent query $\mathbf{z}$ from input variable $\mathbf{x}$. At inference time, it, optionally, conditions on query knowledge $\hat{\mathbf{z}}$ (when this is available). The conditional language model, is based on the vanilla encoder-decoder architecture, the main difference being that it encodes two *views* of input document D. One encoding is query-focused, and depends directly on $\mathbf{z}$ as generated from the query model. The second encoding is query-agnostic, allowing for the original document to provide complementary context. A decoder conditioned on both encodings autoregressively generates the summary $\mathbf{y}$. In contrast to MARGESUM presented in the last chapter, the latent query model and conditional language model are trained jointly in a fully differentiable end-to-end manner. In the following sections we explain in detail how these two models are parametrized.

5.4 Latent Query Model

In this section we discuss how the inference network for latent queries is constructed. We also explain how query-focused document representations are obtained, our attempts to mitigate posterior collapse via weak supervision $o(\hat{\mathbf{z}}|\mathbf{x},\mathbf{y})$ (see Equation (5.6)), and how query belief is updated when queries are available at test time.

Inference Network for Latent Queries We construct a neural network model to infer for each token in the input document whether it constitutes a query term. Given a

contextual token representation matrix $\mathbf{H}_q \in \mathbb{R}^{M \times d_h}$ where d_h denotes the hidden state dimension, we project it to $\mathbb{R}^{M \times 2}$ with a two-layer MLP as a scoring function:

$$\mathbf{H}_s = \text{ReLU}(\mathbf{H}_q \mathbf{W}_h + \mathbf{b}_h^\intercal) \tag{5.7}$$

$$\boldsymbol{\pi} = \mathbf{H}_s \mathbf{W}_s + \mathbf{b}_s^\intercal \tag{5.8}$$

where $\mathbf{W}_h \in \mathbb{R}^{d_h \times d_h}$, $\mathbf{b}_h \in \mathbb{R}^{d_h \times 1}$, $\mathbf{W}_s \in \mathbb{R}^{d_h \times 2}$, and $\mathbf{b}_s \in \mathbb{R}^{2 \times 1}$ are learnable model parameters.

Let $G(0)$ denote the standard Gumbel distribution, and $g_\ell \sim G(0)$, $\ell \in [0,1]$ is i.i.d. drawn Gumbel noise. We normalize $\boldsymbol{\pi}$ to form a variational distribution as:

$$\begin{aligned} q_\phi(\mathbf{z}_i = \ell | \mathbf{x}) &= \text{softmax}_\ell([\boldsymbol{\pi}_0 + g_0, \boldsymbol{\pi}_1 + g_1]) \\ &= \frac{\exp((\boldsymbol{\pi}_\ell + g_\ell)/\tau)}{\sum_{\ell' \in [0,1]} \exp((\boldsymbol{\pi}_{\ell'} + g_{\ell'})/\tau)} \end{aligned} \tag{5.9}$$

where τ is the temperature controlling how close $q_\phi(\mathbf{z}|\mathbf{x})$ is to $\text{argmax}_\ell\, q_\phi(\mathbf{z}|\mathbf{x})$, and is optimized on the development set. Note that Gumbel noise is only applied during learning and is set to its mode, i.e., 0, for inference.

Query-focused View As explained earlier, in addition to a canonical, query-agnostic encoding of the input document D (which we discuss in Section 5.5), we further introduce a query-focused encoding factorized via latent queries $\mathbf{z}$.

Specifically, for the ith token, we take the continuous relaxation of its discrete latent variable $\mathbf{z}_i$, and ground[4] it to the input document via:

$$\mathbf{Q}_i = q_\phi(\mathbf{z}_i = 1 | \mathbf{x}) \cdot \mathbf{H}_{q,i}. \tag{5.10}$$

As we can see, the query-focused view explicitly models the dependency on latent queries. From a learning perspective, this factorization leads to the following partial derivatives of the query encoder states with respect to the query-focused view:

$$\frac{\partial \mathbf{Q}_i}{\partial \mathbf{H}_{q,i}} = \underbrace{\left(1 - q_\phi^{(1)}\right)}_{\text{carry gate}} \cdot \frac{\partial \Delta\pi}{\partial \mathbf{H}_{q,i}} \odot \mathbf{Q}_i + \underbrace{q_\phi^{(1)}}_{\text{transform gate}} \cdot \mathbf{1} \tag{5.11}$$

where $q_\phi^{(\ell)}$ is a shorthand for the variational probability of $\mathbf{z}_i = \ell | \mathbf{x}$, and $\Delta\pi = \boldsymbol{\pi}_1 - \boldsymbol{\pi}_0$ (see Equation (5.8)) and $\mathbf{1}$ denotes an all-one vector. This can be viewed as a special case of highway networks (Srivastava et al., 2015) where transform gate $q_\phi^{(1)}$ compresses the information captured by a token based on its likelihood of being a query term.

[4] We also experimented with drawing hard samples from $\mathbf{z}$ via the straight-through trick (Jang et al., 2017) which is differentiable with biased gradient estimation. However, it did not yield better results than continuous relaxation.

Token Tagging as Weak Supervision Although it is possible to optimize latent queries solely based on conditional language modeling (our approach is fully differentiable), we additionally exploit weak supervision to label tokens in the document as query-specific or not. Weak supervision is advantageous as it imposes extra regularization on the posterior (see Equation (5.6)) thereby mitigating its collapse (i.e., the decoder may learn to ignore the query-focused view and instead rely solely on the query-agnostic view).

Let $t_1, \ldots, t_M$ denote binary tags for each of the source tokens, i.e., 1 if a token is query-specific and 0 otherwise. We could learn such a tagger from training data generated by aligning query tokens to the document. In default of such goldstandard data, we approximate queries by summaries and obtain silver standard token labels by aligning summaries to their corresponding documents. Specifically, inspired by Gehrmann et al. (2018), we assume a token in the document is query-specific if it is part of the longest common sub-sequence (LCS) of tokens in the summary. Our tagging model is built on top of a pretrained language model, and thus operates on subwords. We first byte-pair encode (BPE; Sennrich et al. 2016) documents and summaries, and then search for the LCS over BPE sequences.[5] If there exist multiple identical LCSs, only the one appearing at the earliest document position is tagged as positive. We refer to this tagging scheme as BPE-LCS.

Note that although we model query variables at the token level, we take phrases indirectly into account through LCS which identifies subsequences of tokens (or phrases) as query annotations. Our tagging model is therefore able to capture dependencies between tokens, albeit indirectly.

Training To optimize the variational inference model, i.e., the MLP defined in Equations (5.7–5.9), we use a cross entropy loss for token tagging, with the posterior entropy term from Equation (5.6). Formally, we write the query modeling loss as follows:

$$\begin{aligned}\mathcal{L}_{\text{query}} &= -\omega\mathcal{L}_{\text{tag}} + \beta\mathcal{L}_{\text{entropy}} \\ &= -\sum_{j=1}^{|\mathcal{B}|}\sum_{i=1}^{M}\Big(\big(\omega\hat{\mathbf{z}}_i^j - \beta q_\phi^{(1)}\big)\log q_\phi^{(1)} + \big(\omega\big(1-\hat{\mathbf{z}}_i^j\big) - \beta q_\phi^{(0)}\big)\log q_\phi^{(0)}\Big)\end{aligned} \tag{5.12}$$

where $|\mathcal{B}|$ is the minibatch size and $\hat{\mathbf{z}}_i$ is a binary label automatically assigned via

[5] BPE was initially introduced as a compression algorithm (Gage, 1994), and then adapted to the task of word segmentation (Sennrich et al., 2016). By encoding rare and unknown words as sequences of subword units, BPE represents an open vocabulary with a fixed-size vocabulary of variable-length character sequences.

BPE-LCS(D,S), the alignment procedure described above. As we can see, the entropy term dynamically smooths the weak annotations $\hat{\mathbf{z}}_i$ (the degree of smoothing is modulated by q_ϕ). We optimize ω and β on a development set.

In the initial stages of training, the tagger might lead to inaccurate posterior probability assignments $q_\phi(\mathbf{z}_i|\mathbf{x})$, and, consequently, hurt the summarization model which relies heavily on a high-quality query-focused view. To address this issue, we introduce a *posterior dropout* mechanism which replaces the estimated posterior with weak supervision $o(\hat{\mathbf{z}}|\mathbf{x})$ according to probability α. We initialize α to 1, so that only $o(\hat{\mathbf{z}}|\mathbf{x})$ is used in the beginning of training, and the tagger is supervised via Equation (5.12). We then linearly anneal α over optimization steps so that the gradients from the summarization objective (which we introduce in Section 5.5) can jointly optimize the tagger.

Zero-shot Transfer We now explain how queries are taken into account at test time by performing query belief updates $\Delta(\mathbf{z}_i|\mathbf{x},\tilde{\mathbf{z}})$. In the case of generic summarization where no queries are available, we simply perform no update. When $Q \neq \emptyset$, some tokens in the document become more relevant and we consequently set $\Delta(\mathbf{z}_i = 1|\mathbf{x},\tilde{\mathbf{z}}) = 1$, $\forall w_i \in$ BPE-LCS(D,Q), and all other tokens to zero. We further incorporate query information via a simple calibration as:

$$q_\phi(\mathbf{z}_i = 1|\mathbf{x},\tilde{\mathbf{z}}) = \min\{1, q_\phi(\mathbf{z}_i = 1|\mathbf{x}) + \Delta(\mathbf{z}_i = 1|\mathbf{x},\tilde{\mathbf{z}})\}. \tag{5.13}$$

Note that our calibration is *non-parametric*, since it is not realistic to assume access to a development set for each query type (e.g., in order to perform hyper-parameter tuning). This enables zero-shot transfer to QFS tasks with varying characteristics.

5.5 Conditional Language Model

In this section we describe our conditional language model which estimates the log-likelihood expectation of a summary sequence over the variational posterior (see Equation (5.6)). As mentioned earlier we adopt an encoder-decoder architecture tailored to document summarization with latent queries.

Encoder We encode two views of the input document, a generic query-agnostic view $\mathbf{D}$, and a query-focused one $\mathbf{Q}$ (see Equation (5.10)). As shown in Figure 5.2, our encoder module consists of three encoders: a shared encoder, a document encoder, and a query encoder. Since both views are created from the same document, we use a

Dataset	Task	Domain	Size	D/Q/S Tokens	Query Type	Query Example
CNN/DM	SDS	News	11,490	760.5/0.0/45.7	Empty	∅
WikiCatSum	MDS	Wiki	8,494	800.0/0.0/105.6	Empty	∅
WikiRef	SDS	Wiki	12,000	398.7/6.7/36.2	Keywords	*Marina Beach, Incidents*
Debatepedia	SDS	Debates	1,000	66.4/10.0/11.3	Question	*Is euthanasia better than withdrawing life support?*
DUC 2006	MDS	Newswire	1,250 (50)	699.3/32.8/250	Composite	AMNESTY INTERNATIONAL – *What is the scope of operations of Amnesty International and what are the international reactions to its activities?*
DUC 2007	MDS	Newswire	1,125 (45)	540.3/30.5/250		
TD-QFS	MDS	Medical	7,099 (50)	182.9/3.0/250	Title	*Alzheimers Disease*

Table 5.1: Test data statistics. SDS/MDS stand for single-/multi-document summarization. Size refers to number of test documents; for multi-document QFS, we specify the number of clusters in brackets. D/Q/S are Document/Query/Summary tokens. Composite queries consist of a TOPIC and a *narrative*.

shared encoder for general document understanding which also reduces model parameters. The shared document representation serves as input to more specialized encoders. Each encoder contains one or multiple Transformer layers (Vaswani et al., 2017), each composed of a multi-head attention (MHA) layer and a feed-forward (FFN) layer:

$$\begin{aligned}\mathbf{H}_{\mathcal{E}} &= \mathrm{LN}\big(\mathbf{H}_{\mathcal{E}} + \mathrm{MHA}\big(\mathbf{H}_{\mathcal{E}}, \mathbf{H}_{\mathcal{E}}, \mathbf{H}_{\mathcal{E}}\big)\big) \\ \mathbf{H}_{\mathcal{E}} &= \mathrm{LN}\big(\mathbf{H}_{\mathcal{E}} + \mathrm{FFN}\big(\mathbf{H}_{\mathcal{E}}\big)\big)\end{aligned} \tag{5.14}$$

where LN denotes layer normalization. As shown in Figure 5.2, the query-focused view $\mathbf{Q}$ directly conditions on sampled latent queries, while $\mathbf{D}$ is based on the original document and its content.

Decoder We adopt a decoder structure similar to Dou et al. (2021) to handle multiple inputs. Our decoder sequentially attends to the two encoded views of the same document:

$$\begin{aligned}\mathbf{H}_{\mathcal{D}} &= \mathrm{LN}\big(\mathbf{H}_{\mathcal{D}} + \mathrm{MHA}\big(\mathbf{H}_{\mathcal{D}}, \mathbf{H}_{\mathcal{D}}, \mathbf{H}_{\mathcal{D}}\big)\big) \\ \mathbf{H}_{\mathcal{D}} &= \mathrm{LN}\big(\mathbf{H}_{\mathcal{D}} + \mathrm{MHA}\big(\mathbf{H}_{\mathcal{D}}, \mathbf{Q}, \mathbf{Q}\big)\big) \\ \mathbf{H}_{\mathcal{D}} &= \mathrm{LN}\big(\mathbf{H}_{\mathcal{D}} + \mathrm{MHA}\big(\mathbf{H}_{\mathcal{D}}, \mathbf{D}, \mathbf{D}\big)\big) \\ \mathbf{H}_{\mathcal{D}} &= \mathrm{LN}\big(\mathbf{H}_{\mathcal{D}} + \mathrm{FFN}\big(\mathbf{H}_{\mathcal{D}}\big)\big).\end{aligned} \tag{5.15}$$

After taking the context of the previous generation $\mathbf{H}_{\mathcal{D}}$ into account, the decoder will first attend to signals coming from query $\mathbf{Q}$, then to original document $\mathbf{D}$ (based on guidance provided by the query). The final summary generation objective is calculated autoregressively as:

$$\mathcal{L}_{\mathrm{lm}} = \sum_{j=1}^{|\mathcal{B}|} \sum_{t=1}^{T} \log p_{\theta}\left(\mathbf{y}_t | \mathbf{y}_{<t}, \mathbf{D}, \mathbf{Q}\right) \tag{5.16}$$

which is jointly trained with the query model (see Equation (5.12)) as: $\mathcal{L} = \mathcal{L}_{\mathrm{lm}} + \mathcal{L}_{\mathrm{query}}$.

5.6 Experimental Setup

5.6.1 Summarization Datasets

For model training and development, we used the CNN/Daily Mail dataset (Hermann et al., 2015), a generic single-document summarization benchmark containing news articles and associated highlights (287,227/13,368 instances). We evaluated our model

on the CNN/Daily Mail test set following a generic summarization, supervised setting. We also performed several *zero-shot* experiments, on five benchmarks representing various query formats, domains, and summarization scenarios (e.g., single- vs multiple-documents). Specifically, we report results on WikiCatSum (Perez-Beltrachini and Lapata, 2021) as an example of multi-document generic summarization, and WikiRef (Zhu et al., 2019), Debatepedia (Nema et al., 2017), DUC 2006-07, and TD-QFS (Baumel et al., 2016) as examples of QFS. Table 5.1 summarizes the characteristics of these datasets and presents test set statistics. Note that in contrast to the few-shot approach in the last chapter, we do not make use of development data for our QFS tasks.

5.6.2 Implementation Details

The shared encoder consists of 11 Transformer layers. The document and query encoders have a separate Transformer layer each. All encoders and decoder are initialized with a pretrained BART model (Lewis et al., 2020), while the query encoder is initialized randomly. We used four GeForce RTX 2080 GPUs for training; we set the batch size to 8 (i.e., one sample for each GPU), and accumulate gradients every 32 steps. We fine-tuned BART on CNN/Daily Mail with a learning rate of 3×10^{-5} for 20,000 optimization steps, and a warmup-step of 500. We used half float precision for efficient training and set the maximum length of an input document to 640 tokens, with the excess clipped. We set $\beta = 0.1$ and $\omega = 10$ in the learning objective, and $\tau = 0.9$ for latent query modeling. We annealed the dropout rate α from 1.0 to 0.5 over the whole training session.

5.7 Automatic Evaluation

Before analyzing our model under various zero-shot settings, we first confirm it can indeed produce good quality generic summaries in a supervised setting. There is no point in contemplating zero-shot scenarios if our approach underperforms when full supervision is available. Following standard practice, we use F1 ROUGE as our automatic evaluation metric (Lin and Hovy, 2003). Unigram and bigram ROUGE (R-1 and R-2) are a proxy for assessing informativeness and the longest common subsequence (R-L) represents fluency. For multi-document QFS, we follow the official metrics in DUC (Dang, 2005) and report R-SU4 (based on skip bigram with a maximum skip

Upper Bound & Baselines	R-1	R-2	R-L
ORACLE	55.8	33.2	51.8
LEAD	40.4	17.6	36.7
LEXRANK	33.2	11.8	29.6
Supervised (Extractive)	R-1	R-2	R-L
BERTEXT (Liu and Lapata, 2019b)	43.9	20.3	39.9
MATCHSUM (Zhong et al., 2020)	43.9	20.6	39.8
Supervised (Abstractive)	R-1	R-2	R-L
PTGEN (See et al., 2017)	39.5	17.3	36.4
BOTTOMUP (Gehrmann et al., 2018)	41.2	18.7	38.4
BERTABS (Liu and Lapata, 2019b)	41.7	19.4	38.8
BART (Lewis et al., 2020)	44.2	21.3	40.9
GSUM (Dou et al., 2021)	45.9	22.3	42.5
GSUM (our implementation)	45.0	21.9	41.8
LQSUM	**45.1**	**22.0**	**41.9**

Table 5.2: Generic summarization, supervised setting, **CNN/Daily Mail** test set.

distance of 4) instead of R-L.[6]

5.7.1 Supervised Setting

Table 5.2 summarizes our results on the CNN/Daily Mail test set. As an upper bound (first block) we report the performance of an extractive ORACLE which performs greedy search to find a set of sentences in the source document that maximize ROUGE scores against the reference (Liu and Lapata, 2019b). The LEAD baseline considers the first 3 sentences in a document as the summary. LEXRANK (Erkan and Radev, 2004) estimates sentence-level centrality via a Markov Random Walk on graphs. The second block includes two additional extractive systems. BERTEXT (Liu and Lapata, 2019b) is the first rendition of a summarization system with a pretrained encoder (Devlin et al., 2019). MATCHSUM (Zhong et al., 2020) extracts an optimal set of sentences via semantically matching documents to candidate summaries.

The third block includes various abstractive systems (see Section 5.2 for an overview). PTGEN (See et al., 2017) and BOTTOMUP (Gehrmann et al., 2018) do not use pre-

[6]We used `pyrouge` with the following parameter settings: ROUGE-1.5.5.pl -a -c 95 -m -n 2 -2 4 -u -p 0.5 -l 250.

Model	Size	Components
BART	400M	ENC=12, DEC=12
GSUM	625M	ENC=13, DEC=12, BERT=2 (220M; guidance)
LQSUM	406M	ENC=13, DEC=12, TAG=1 (1M; latent query)

Table 5.3: System comparison. ENC, DEC and TAG denote number of layers for encoding, decoding and tagging, respectively. GSUM (Dou et al., 2021) and LQSUM add a (randomly initialized) encoding layer on top of BART (Lewis et al., 2020) for guidance/query representation. LQSUM replaces guidance extraction in GSUM (i.e., two BERT models) with latent query modeling (i.e., a lightweight tagging layer) which is more parameter efficient.

trained LMs, while BERTABS (Liu and Lapata, 2019b) is built on top of a pretrained BERT encoder. BART (Lewis et al., 2020) is fine-tuned on CNN/DM, while GSUM (Dou et al., 2021) is initialized with BART parameters. We introduced these summarization systems in Section 2.1.3 and 2.2.1.

Our **L**atent **Q**uery **Sum**marization model (LQSUM) outperforms BART by a large margin, which demonstrates the effectiveness of latent queries even for generic summarization. It also performs on par with GSUM, under identical training resources and configurations. GSUM is a state-of-the-art abstractive model, which relies on MATCHSUM (Zhong et al., 2020), a high-performance extractive model to provide guidance to the decoder. Compared to GSUM, LQSUM can be trained end-to-end and requires significantly less parameters (406 M for LQSUM versus 625 M for GSUM); see Table 5.3 for details).

5.7.2 Zero-Shot Setting

Multi-Document Summarization We evaluated our model's ability to summarize multiple documents on WikiCatSum (Perez-Beltrachini et al., 2019), a collection of articles on a specific topic (e.g., Tokyo Olympics) and their corresponding Wikipedia summary. In order to handle multi-document input with a model trained on single-document data, we follow previous work (Perez-Beltrachini et al., 2019) and first select a subset of salient passages which are then concatenated into a sequence and given to our model to summarize.

In the first block of Table 5.4 we present upper bound and baseline results. The second block contains results for two supervised systems, a sequence-to-sequence model

Upper Bound & Baselines	R-1	R-2	R-L
ORACLE	47.2	23.3	42.9
LEAD	22.3	6.9	19.9
LEXRANK	23.3	6.5	20.3
Supervised (Abstractive)	R-1	R-2	R-L
TRANSFORMER (Liu et al., 2018)	35.5	19.0	30.5
CV-S2D+T (Perez-Beltrachini et al., 2019)	36.1	19.9	30.5
Zero-shot Abstractive	R-1	R-2	R-L
BART (Lewis et al., 2020)	27.8	9.8	25.1
GSUM+LEXRANK	27.4	8.2	25.0
LQSUM	**28.7**	**9.9**	**26.1**

Table 5.4: Multi-document summarization, zero-shot setting, **WikiCatSum** test set. Results are averaged over three domains: *Company*, *Film*, and *Animal*.

based on Transformer (Liu et al., 2018), and a state-of-the-art system enhanced with a convolutional encoder, a structured decoder, and a topic prediction module (CV-S2D+T; Perez-Beltrachini et al. 2019). The third block contains zero-shot models, including BART, GSUM and LQSUM. GSUM requires another extractive system's output as guidance during inference, for which we default to LEXRANK. As can be seen, LQSUM performs best among zero-shot models, but lags behind fully-supervised ones which is not surprising (zero-shot models operate over pre-ranked, incoherent passages).

Single-Document QFS Tables 5.5 and 5.6 show results for single-document QFS on two datasets, namely WikiRef (Zhu et al., 2019) and Debatepedia (Nema et al., 2017) which differ in terms of document/summary size and query type (see Table 5.1). The first block in both tables shows results for the ORACLE upper bound, LEAD, and LEXRANK$_Q$, a query-focused version of LEXRANK described in Xu and Lapata (2020). The second block presents various *supervised* systems on WikiRef and Debatepedia, both extractive and abstractive. Note that abstractive QFS systems have not been previously evaluated on WikiRef, while Debatepedia contains short documents and accordingly short summaries and has mainly served as a testbed for abstractive summarization. The third block reports system performance in the zero-shot setting. We compare LQSUM against BART and GSUM which, however, requires guidance

Upper Bound & Baselines	R-1	R-2	R-L
ORACLE	54.5	37.5	48.5
LEAD	26.3	10.5	21.8
LEXRANK$_Q$	29.9	12.3	26.1
Supervised (Extractive)	R-1	R-2	R-L
TRANSFORMER (Zhu et al., 2019)	28.1	12.8	23.8
BERTEXT (Zhu et al., 2019)	35.1	18.2	30.0
Zero-shot Abstractive	R-1	R-2	R-L
BART (Lewis et al., 2020)	30.0	12.2	26.0
GSUM+LEXRANK$_Q$	30.2	12.5	26.3
LQSUM	**31.1**	**12.6**	**27.1**

Table 5.5: Single-document QFS, zero-shot setting, **WikiRef** test set (queries are keywords).

from automatically extracted sentences. Note that MATCHSUM (Zhong et al., 2020), the original extractive system used by GSUM for guidance, is not directly applicable to QFS, as it is trained for generic summarization which does not take queries as input. We made a best effort attempt to adapt GSUM to our QFS setting by using query-focused LEXRANK$_Q$ to extract the top K sentences for each test document as guidance.

Across both datasets LQSUM achieves the highest ROUGE scores in the zero-shot setting, in some cases surpassing the performance of supervised models. Compared to our results on generic summarization, LQSUM also shows a clearer advantage over systems without latent query modeling.

Multi-Document QFS We performed experiments on the DUC 2005-2007 benchmarks and TD-QFS (Baumel et al., 2016). The former contains long query narratives while TD-QFS focuses on short keyword queries (see Table 5.1).

We applied our summarization model which was trained on *single* documents to document *clusters* following a simple iterative approach (Baumel et al., 2018): we first rank documents in a cluster via their query term frequency, and then generate a summary for each document. The summary for the entire cluster is the concatenation of the individual document summaries subject to a budget (i.e., 250 tokens). An alternative is to generate a long summary at once. However, as shown in the last chapter, this

Upper Bound & Baselines	R-1	R-2	R-L
ORACLE	28.9	11.0	24.9
LEAD	18.1	5.6	15.9
LEXRANK$_Q$	17.4	5.3	15.1
Supervised (Abstractive)	R-1	R-2	R-L
DDA (Laskar et al., 2020a)	7.4	2.8	7.2
BERTABS+RANK (Abdullah and Chali, 2020)	19.2	10.6	17.9
BERTABS+CONCAT (Laskar et al., 2020a)	26.4	11.9	25.1
Zero-shot Abstractive	R-1	R-2	R-L
BERTABS† (Liu and Lapata, 2019b)	13.3	2.8	2.8
BART (Lewis et al., 2020)	21.4	6.3	18.4
GSUM+LEXRANK$_Q$	21.2	6.2	18.2
LQSUM	**23.5**	**7.2**	**20.6**

Table 5.6: Single-document QFS, zero-shot setting, **Debatepedia** test set (queries are natural questions). BERTABS† (Laskar et al., 2020a) is optimized on XSum (Narayan et al., 2018a).

requires a model to be trained on a MDS dataset, or at least a proxy thereof. Since we trained our model on single-document summarization data, we opted for the former approach (i.e., to first generate and then compose the cluster summary. Repeated sentences were skipped to reduce redundancy in the final summary.

Our results are given in Table 5.7. The first block reports performance for the ORACLE upper bound and GOLD which was estimated by comparing a (randomly selected) reference summary against the remaining two or three reference summaries.[7] We also include LEXRANK$_Q$, and LEAD which returns all lead sentences (up to 250 words) of the most recent document. The second block contains *distantly supervised* approaches. QUERYSUM (Xu and Lapata, 2020) is an extractive system which takes advantage of existing QA datasets and adopts a coarse-to-fine salience estimation procedure. BARTCAQ (Su et al., 2020) uses an ensembled QA model for answer evidence extraction, and a fine-tuned BART model (Lewis et al., 2020) to iteratively generate summaries from paragraphs. PQSUM (Laskar et al., 2020b) uses fine-tuned BERTSUM to generate summaries for each document in a cluster, and a QA model for summary sentence

[7]We compute this upper bound only for DUC and TD-QFS benchmarks as they include multiple reference summaries.

	DUC 2006			DUC 2007			TD-QFS		
Upper Bound & Baselines	R-1	R-2	R-SU4	R-1	R-2	R-SU4	R-1	R-2	R-SU4
GOLD	45.4	11.2	16.8	47.5	14.0	18.9	52.2	27.0	30.2
ORACLE	47.5	15.8	20.2	47.6	17.1	20.9	64.9	48.3	49.4
LEAD	32.1	5.3	10.4	33.4	6.5	11.3	33.5	5.2	10.4
LEXRANK$_Q$	34.2	6.4	11.4	35.8	7.7	12.7	35.3	7.6	12.2
Distantly Supervised	R-1	R-2	R-SU4	R-1	R-2	R-SU4	R-1	R-2	R-SU4
QUERYSUM* (Xu and Lapata, 2020)	41.6	9.5	15.3	43.3	11.6	16.8	44.3	16.1	20.7
BART-CAQ (Su et al., 2020)	38.3	7.7	12.9	40.5	9.2	14.4	—	—	—
PQSUM (Laskar et al., 2020b)	40.9	9.4	14.8	42.2	10.8	16.0	—	—	—
Few- or Zero-shot Abstractive	R-1	R-2	R-SU4	R-1	R-2	R-SU4	R-1	R-2	R-SU4
MARGESUM† (Xu and Lapata, 2021)	40.2	9.7	15.1	42.5	12.0	16.9	45.5	16.6	20.9
BART (Lewis et al., 2020)	38.3	7.8	13.1	40.2	9.9	14.6	45.1	16.9	21.4
GSUM+LEXRANK$_Q$	38.1	7.9	13.1	39.5	9.5	14.3	45.5	18.0	**22.4**
LQSUM	**39.1**	**8.5**	**13.7**	**40.4**	**10.2**	**15.0**	**45.7**	**18.1**	22.1

Table 5.7: Multi-document QFS, zero-shot setting, **DUC** (queries are narratives) and **TD-QFS** (queries are keywords) test sets. $*/\dagger$ denotes extractive/few-shot systems.

re-ranking.

The third block compares our model against MARGESUM (Xu and Lapata, 2021), a state-of-the-art *few-shot* approach, which uses data for proxy query generation and model development, and various *zero-shot* systems including GSUM+LEXRANK$_Q$ and BART. Across datasets, LQSUM outperforms comparison zero-shot approaches. It also has a clear advantage over MARGESUM on TD-QFS but is slightly worse on DUC. We also see that LQSUM is superior to BART-CAQ which relies on distant supervision from QA data.

5.8 Ablation Studies

We further performed a series of ablation studies in Tables 5.8 and 5.9 to assess the contribution of individual model components. Perhaps unsurprisingly, we observe that not updating the query belief at test time hurts performance ($-\Delta(\hat{\mathbf{z}}|\mathbf{x},\mathbf{z})$). Recall that we adopt a simple method which calibrates the variational posterior distribution. When it comes to learning meaningful latent queries that benefit summarization tasks, relying solely on tagging ($-$Joint training) or generation ($-$Weak supervision) substantially

	CNN/DM			WikiRef			Debatepedia		
Model	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
LQSUM	45.1	22.0	41.9	31.1	12.6	27.1	23.5	7.2	20.6
$-\Delta(\hat{\mathbf{z}}\|\mathbf{x},\mathbf{z})$	—	—	—	↓0.1	↓0.2	↓0.2	↓0.5	↓0.3	↓0.6
−Joint training	↓0.4	↓0.3	↓0.4	↓2.9	↓0.9	↓2.8	↓2.8	↓1.1	↓2.8
−Weak supervision	↓0.6	↓0.7	↓0.7	↓0.7	↓0.2	↓0.5	↓1.0	↓0.5	↓1.3
−Dual view	**↓2.7**	**↓3.5**	**↓2.5**	**↓12.2**	**↓9.3**	**↓10.5**	**↓7.9**	**↓3.3**	**↓6.6**
−Posterior dropout	↓0.7	↓0.6	↓0.8	↓0.8	↓0.3	↓0.7	↓1.1	↓0.3	↓1.2

Table 5.8: LQSUM ablation results on single-document summarization benchmarks CNN/DM, WikiRef, and Debatepedia; ↑/↓: absolute increase/decrease.

	DUC 2006			DUC 2007			TD-QFS		
Model	R-1	R-2	R-SU4	R-1	R-2	R-SU4	R-1	R-2	R-SU4
LQSUM	39.1	8.5	13.7	40.4	10.2	15.0	45.7	18.1	22.1
$-\Delta(\hat{\mathbf{z}}\|\mathbf{x},\mathbf{z})$	↓0.6	↓0.2	↓0.6	↑0.1	↓0.1	↓1.3	↑0.1	↓0.6	↓0.4
−Joint training	↓2.9	↓1.7	↓1.6	↓2.4	↓2.0	↓1.7	↓0.7	↓0.6	↓0.4
−Weak supervision	↓0.2	↓0.2	↓0.2	↓0.2	↓0.3	↓0.3	↓0.1	↓0.3	↓0.0
−Dual view	**↓6.3**	**↓1.8**	**↓1.8**	**↓6.5**	**↓3.0**	**↓2.5**	**↓2.5**	**↓3.3**	**↓2.8**
−Posterior dropout	↓0.2	↓0.2	↓0.2	↓0.4	↓0.4	↓0.5	↑0.2	↓0.0	↑0.1

Table 5.9: LQSUM ablation results on multi-document summarization benchmarks DUC 2006-07 and TD-QFS; ↑/↓: absolute increase/decrease.

decreases performance.[8] Latent query learning balances a trade-off between *direct but weak* supervision from the tagging objective (based on silver standard token labels) and *natural but indirect* supervision from the generation objective (based on human-written summaries). As silver tagging labels provide less accurate supervision than human-written summaries, we observe that −Joint training hurts performance more than −Weak supervision.

Removing the query agnostic view (−Dual view) causes a significant performance drop as the decoder can no longer leverage the original document context which is useful especially when the query model is not accurate. Relying solely on the *estimated* posterior to create the query-focused view for training (−Posterior dropout), also hurts

[8] −Joint training replaces the softmax in Equation (5.9) with argmax, to stop the gradients from the generation loss in backpropagation. −Weak supervision sets $\omega = 0$.

WikiRef	$n=2$	$n=3$	$n=4$
GSUM+LEXRANK$_Q$	16.67	27.20	35.41
LQSUM	22.60	36.15	46.03
GOLD	58.22	72.49	79.81

Debatepedia	$n=2$	$n=3$	$n=4$
GSUM+LEXRANK$_Q$	12.47	16.78	20.88
LQSUM	27.51	35.79	42.61
GOLD	82.93	91.78	94.68

DUC	$n=2$	$n=3$	$n=4$
GSUM+LEXRANK$_Q$	17.62	31.45	41.90
LQSUM	19.43	35.01	46.38
GOLD	56.62	78.48	87.55

TD-QFS	$n=2$	$n=3$	$n=4$
GSUM+LEXRANK$_Q$	8.55	16.89	25.11
LQSUM	8.55	16.80	24.69
GOLD	13.28	23.00	30.30

Table 5.10: Proportion of novel n-grams (%) in model generated summaries and gold summaries on QFS benchmarks.

performance as it leads to more severe error propagation for the downstream generation model.

5.9 Novel N-grams

We further analyzed model generated summaries by calculating the proportion of novel n-grams that appear in the summaries but not in the source documents. We show the results in Table 5.10. As we can see, gold summaries in Debatepedia, which usually consist of one or two sentences answering a given question, are the most abstractive. In contrast, TD-QFS, a multi-document QFS dataset in the medical domain, contains the least proportion of novel n-grams and is therefore the most extractive benchmark. We observe that on all QFS datasets except TD-QFS, our system LQSUM produces more novel n-grams than GSUM+LEXRANK$_Q$. Compared to GSUM+LEXRANK$_Q$ which takes pre-extracted sentences as generation guidance, LQSUM leverages token-level query information which is less redundant (Dou et al., 2021), encouraging the decoder to perform summary abstraction rather than select sentences from the input.

5.10 Human Evaluation

We also evaluated query-focused summaries in a judgment elicitation study via Amazon Mechanical Turk. Native English speakers (self-reported) were asked to rate query-summary pairs on *Succinctness* and *Coherence* using a five point Likert scale. Participants were also asked to assess the *Relevance* of the summary sentences to the

WikiRef	Rel	Suc	Coh
BERTEXT	**3.57**	**3.63**	3.72
GSUM+LEXRANK$_Q$	$2.92^{\dagger\circ}$	3.48°	3.72
LEXRANK$_Q$	3.23	3.40	3.68
LQSUM	3.41	3.58	**3.78**
GOLD	3.62	3.73	3.59

Debatepedia	Rel	Suc	Coh
BERTABS	$2.42^{\dagger}$	$2.93^{\dagger\circ}$	$2.59^{\dagger}$
GSUM+LEXRANK$_Q$	$2.88^{\dagger}$	3.60	$3.49^{\dagger}$
LEXRANK$_Q$	3.33	3.47°	3.52
LQSUM	**3.39**	**3.74**	**3.78**
GOLD	3.29	3.76	3.57

DUC	Rel	Suc	Coh
MARGESUM	**4.00**	3.75	$3.65^{\dagger\circ}$
GSUM+LEXRANK$_Q$	3.90	$3.44^{\dagger\circ}$	3.84
LEXRANK$_Q$	$3.59^{\dagger\circ}$	$3.38^{\dagger\circ}$	$3.54^{\dagger\circ}$
LQSUM	3.97	**3.88**	**3.95**
GOLD	4.01	3.94	4.04

TD-QFS	Rel	Suc	Coh
MARGESUM	3.28	3.57	3.62
GSUM+LEXRANK$_Q$	3.26	3.65	3.76
LEXRANK$_Q$	$2.78^{\dagger\circ}$	$3.36^{\dagger\circ}$	$3.33^{\dagger\circ}$
LQSUM	**3.35**	**3.70**	**3.77**
GOLD	3.50	3.88	3.68

Table 5.11: Human evaluation on QFS benchmarks: average **Rel**evance, **Suc**cinctness, **Coh**erence ratings; †/°: sig different from LQSUM/Gold (at $p < 0.05$, using a pairwise t-test); best system shown in bold.

query, and sentence scores were averaged to obtain a relevance score for the whole summary. Detailed instructions of human evaluation can be found in Appendix A.

Participants assessed summaries created by LQSUM (our zero-shot abstractive system), GSUM+LEXRANK$_Q$ (a competitive abstractive system), LEXRANK$_Q$ (an extractive baseline), and GOLD (the ground-truth upper bound). We also compared against BERTEXT on WikiRef, BERTABS on Debatepedia, and MARGESUM on DUC and TD-QFS.[9] We sampled 40 query-document pairs from WikiRef and Debatepedia, 40 query-cluster pairs from DUC (2006, 2007; 20 from each set), and 40 pairs from TD-QFS and collected three responses per pair.

We show our results in Table 5.11 and examples of system output in Table 5.12. On WikiRef, LQSUM outperforms GSUM+LEXRANK$_Q$ significantly in terms of relevance. On Debatepedia it surpasses BERTABS, a supervised model, across all three metrics. On DUC, it outperforms comparison systems in terms of succinctness and coherence. LQSUM avoids repetition by yielding dynamic (latent) query representations for each document in the a cluster. On TD-QFS, all comparison systems perform similarly, except LEXRANK$_Q$ which is significantly worse in terms of relevance and succinctness. As far as Relevance is concerned we observe that LQSUM outperforms comparison systems on Debatepedia and TD-QFS, while being very similar to

[9] BERTEXT and BERTABS are supervised systems, while MARGESUM is a few-shot system.

WikiRef

Query: *Prashant Bhushan, Legal activism, Government accountability*
GOLD: CPIL won a major victory in 2003 when the Supreme Court restrained the Union government from privatising Hindustan Petroleum and Bharat Petroleum without the approval of Parliament.
BERTEXT: New Delhi, March 3: The Supreme Court verdict against P.J. Thomass appointment is not the lone feather in the cap of the petitioner, the Centre for Public Interest Litigation (CPIL), but perhaps the most visible one. That was when it got the apex court to restrain the Centre from divesting majority shares in Hindustan Petroleum and Bharat Petroleum without Parliaments approval. The CPIL was founded in the late 1980s by Justice V.M. Tarkunde, who also co-founded the Peoples Union for Civil Liberties.
GSUM+LEXRANK$_Q$: The Centre for Public Interest Litigation (CPIL) is a loose collection of activists and lawyers whose aim is to fight corruption. Among its members are lawyers Shanti Bhushan, Prashant BhUSHan, Kamini Jaiswal, Ram Jethmalani, Anand Divan and Anil Divan. Another PIL asks that the government be directed to recover Indian black money stashed in foreign banks.
LQSUM: The Centre for Public Interest Litigation (CPIL) is a loose collection of activists and lawyers. The group had its big hurrah in 2003 when it got the apex court to restrain the Centre from divesting majority shares in Hindustan Petroleum and Bharat Petroleum.

Debatepedia

Query: *Effectiveness: Do earmarks allocate spending effectively?*
GOLD: Earmarks are often unrelated to legislation; holds up bill.
BERTABS: Earmarks can be fully examined.
GSUM+LEXRANK$_Q$: Sometimes a good piece of legislation that receives the support of a majority of congressman will be held up and voted down.
LQSUM: Congressmen are using earmarks to hold up bills they don't like, says Rep. Ruben Gallego.

Table 5.12: System outputs on WikiRef (above; document 3918) and Debetepedia (below; document 260). Information irrelevant to the query or incoherent in the summary is highlighted.

MARGESUM on DUC. On Wikiref, BERTEXT is slightly more relevant but less coherent.

5.11 Summary

In this chapter, we moved beyond QFS and proposed a deep generative formulation for document summarization that supports generic and query-focused applications. We represent queries as discrete latent variables, whose approximated posterior distribution can be calibrated with query observations at test time without further adaptation. Our approach does not rely on any query-related resource and can be applied in zero-

shot settings. Experimental results across summarization datasets show that the proposed model yields state-of-the-art QFS performance in zero-shot settings.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this thesis, we focused on document summarization and proposed neural query modeling as an intermediate objective to improve the current state of the art on both query focused and generic summarization. We proposed three different instantiations, namely *progressive*, *proxy*, and *latent* query modeling, and examined their effectiveness on a diverse set of summarization benchmarks.

We first investigated how to improve extractive QFS with *progressive* query modeling. In Chapter 3, we proposed a coarse-to-fine estimation framework for multi-document QFS, and then explored the potential of leveraging distant supervision signals from Question Answering. We experimented with two popular QA settings, namely answer sentence selection and machine reading comprehension which operates over passages than isolated sentences. Experimental results across datasets show that the proposed model yields results superior to existing systems that employ classic retrieval techniques for query-sentence relevance estimation. We showed that large-scale QA datasets can provide supporting evidence for answering queries and help to alleviate the data paucity problem in QFS. We also found that disentangling the tasks of relevance, evidence, and centrality estimation allows us to progressively specialize the summaries to the query semantics, contributing to summaries which are more relevant and less redundant.

In Chapter 4, we moved on to abstractive QFS, and studied the research question of how to avoid the dependency on query-related resources which can be costly to obtain. To this end, we proposed a framework that performs abstractive QFS with *proxy* query modeling and uses only query-free training data. We first decomposed abstractive QFS

into proxy query modeling and conditional language modeling. Inspired by the Cloze task, we provided a unified mask representation for summaries and queries, which enables generic summaries to be transformed into proxy queries for training an evidence ranker. Based on the selected evidence, conditional language modeling generates an abstractive summary in autoregressively. Under this formulation, we used generic summarization data not only for conditional language modeling, but also for learning an evidence ranker for query modeling. We also studied the incorporation of various control attributes into the optimization of an abstractive summarizer. Despite learning from weak supervision, our model outperforms strong question answering models in evidence ranking. Experimental results across datasets also show that the proposed abstractive system yields state-of-the-art summarization performance and generates more relevant and coherent summaries compared to existing approaches.

Can we build a summarization system that robustly handles all possible query types? In Chapter 5, we answered this research question by presenting a unified framework for generic summarization and QFS with *latent* query modeling. We first assumed that all summaries are a response to a query, which is observed in the case of QFS and latent in the case of generic summarization. Based on the assumption, we proposed a deep generative formulation for document summarization that supports generating both generic and query-focused abstracts. We represented queries as discrete latent variables, whose variational posterior can be adapted with additional query observations at test time in a non-parametric manner. Jointly optimized with a conditional language model, the latent query model learns representations compatible with observed and unobserved query verbalizations from both summarization objectives and weak tagging supervisions. As our framework does not rely on any query-related resources for either training or development, it is naturally applicable to zero-shot scenarios. Experimental results across summarization datasets with varied query types, from keywords to natural questions, show that the proposed model yields state-of-the-art QFS performance in zero-shot settings.

6.2 Future work

Potential directions for future research in document summarization with query modeling are many and varied. We discuss four promising topics as follows.

Mixture-of-Experts Summarization Systems Recent work suggests that a large model size may be necessary for better generalization and higher robustness of neural networks (Bubeck and Sellke, 2021). Nevertheless, it is computationally expensive to train such large models from scratch for every new problem. One promising approach to train large models efficiently with limited resources is a mixture-of-experts (MoE; Shazeer et al. 2017) architecture. Different from typical neural networks which are usually dense, a MoE model is sparsely activated and the model learns how to dynamically route tasks through the most relevant neurons conditioned on the input. For a document summarization system, a sparse mixture of independent neural modules can be constructed as experts, each expert specializing for one specific query type (such as a natural question or a keyword) or summarization task (such as extractive or abstractive). In this case, the system can expect to have a larger capacity to generalize over a variety of summarization tasks, while being more computationally efficient due to the network sparsity.

Cross-Lingual Query Modeling To enable users with different linguistic backgrounds to interact with summarization systems, one possible future direction is to extend query modeling to cross-lingual settings. One potential cross-lingual task formulation is to have the input document or document cluster is in one language (i.e., source language), while the query and the summary in another language (i.e., target language). For instance, a user can query a cluster of English documents with French, and receive a query focused summary in French. To achieve this goal, following cross-lingual summarization for generic purposes, machine translation models can either perform separately (Wan et al., 2010) to translate the documents to the target language, or be jointly optimized with the summarization objective (Cao et al., 2020; Dou et al., 2020).

Multi-Modal Query Modeling The new era of technology has enabled users to express themselves in rich media, usually in multiple forms including text, images, audio, and video. To cope with the consequent information overload and improve user experience, the task of multi-modal summarization has attracted lots of research attention in recent years (Yan et al., 2012; Zhu et al., 2018; Li et al., 2018; Zhu et al., 2020). However, how to build a user-centric summarization system and incorporate query understanding into multi-modal models remains an under-studied research question.

Conversational Query Modeling The current formulation of QFS allows users to query summarization systems once, based on the assumption that users' information needs can be satisfied in one turn. However, this strong assumption can fail in real-world scenarios due to various reasons. For instance, as user queries can be ambiguous, the system may need to ask clarification questions before responding, i.e., generating the summary (Rao and Daumé III, 2018). Also, users may sometimes prefer to perform exploratory searches where the information-seeking process can be opportunistic, iterative, and multitactical (White and Roth, 2009). Therefore, to allow document summaries to be produced in a more progressive and interactive manner, the support for multi-turn user-system interaction, i.e., conversational query modeling, is highly favorable for a summarization system.

Appendix A

Instructions for Human Evaluation

We conducted judgment elicitation studies via the Amazon Mechanical Turk platform in Chapters 3, 4 and 5 to evaluate query-focused summaries. Figure A.1 shows the instructions we give to the Amazon Mechanical Turk participants.

Native English speakers (self-reported) are asked to rate query-summary pairs on two dimensions:

- **Succinctness**: Does the summary avoid unnecessary detail and redundant information?
- **Coherence**: Does the summary make logical sense?

The ratings were obtained using a five point Likert scale where 5 denotes very succinct/coherent and 1 denotes the opposite.

In addition, participants are asked to assess the Relevance of the summary to the query. Crowdworkers read a summary and for each *sentence* decided whether it is:

- **Relevant**: The sentence provides an answer to the query.
- **Irrelevant**: The sentence does not answer the query.
- **Partially relevant**: It is unclear the sentence directly answers the query.

Relevant sentences are awarded a score of 5, partially relevant ones a score of 2.5, and 0 otherwise. Sentence scores are averaged to obtain a relevance score for the whole summary. We view Relevance as as more critical for QFS than Coherence or Succinctness. This is why we obtain per-sentence ratings which we then aggregate to an overall summary score. To make this task manageable, raters are asked to provide more coarse grained ratings.

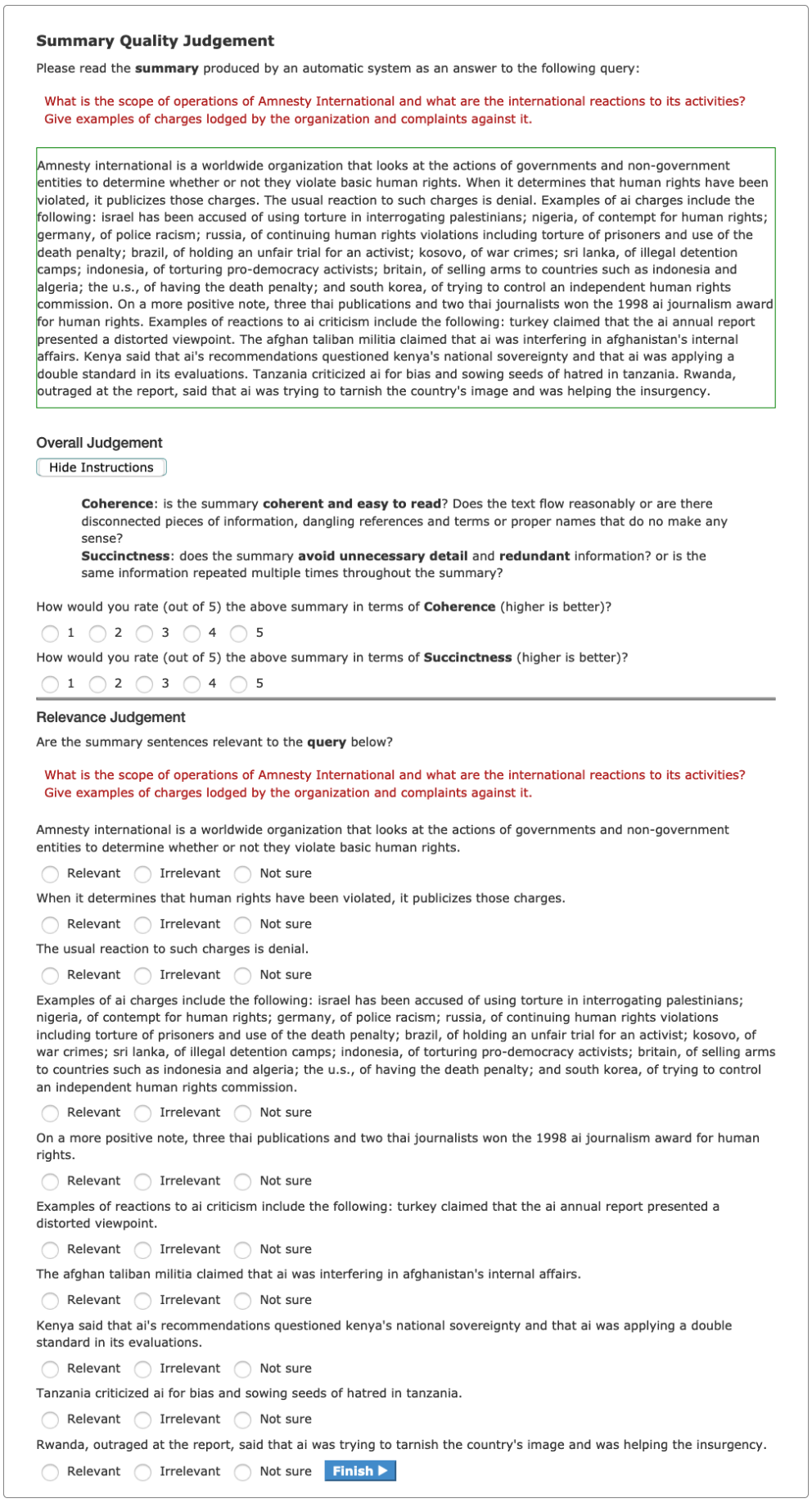


Figure A.1: Instructions for human evaluation of summarization systems on the web-page of Amazon Mechanical Turk platform.

Bibliography

Abdullah, D. M. and Chali, Y. (2020). Towards generating query to perform query focused abstractive summarization using pre-trained model. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 80–85, Dublin, Ireland.

Akkalyoncu Yilmaz, Z., Yang, W., Zhang, H., and Lin, J. (2019). Cross-domain modeling of sentence-level evidence for document retrieval. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3490–3496, Hong Kong, China.

Allan, J., Wade, C., and Bolivar, A. (2003). Retrieval and novelty detection at the sentence level. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 314–321, Toronto, Canada.

Angelidis, S. and Lapata, M. (2018). Multiple instance learning networks for fine-grained sentiment analysis. *Transactions of the Association for Computational Linguistics*, 6:17–31.

Aone, C., Okurowski, M. E., Gorlinsky, J., and Larsen, B. (1997). A scalable summarization system using robust nlp. In *Intelligent Scalable Text Summarization*.

Arumae, K. and Liu, F. (2019). Guiding extractive summarization with question-answering rewards. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2566–2577, Minneapolis, Minnesota.

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

Badrinath, R., Venkatasubramaniyan, S., and Veni Madhavan, C. (2011). Improving query focused summarization using look-ahead strategy. In *Proceedings of the 33rd European Conference on Advances in Information Retrieval*, pages 641–652, Dublin, Ireland.

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

Bajaj, P., Campos, D., Craswell, N., Deng, L., Gao, J., Liu, X., Majumder, R., McNamara, A., Mitra, B., Nguyen, T., et al. (2016). MS MARCO: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Bao, H., Dong, L., Wei, F., Wang, W., Yang, N., Liu, X., Wang, Y., Gao, J., Piao, S., Zhou, M., et al. (2020). UniLMv2: Pseudo-masked language models for unified language model pre-training. In *Proceedings of the 37rd International Conference on Machine Learning*, pages 642–652, Online.

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

Baumel, T., Cohen, R., and Elhadad, M. (2016). Topic concentration in query focused summarization datasets. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 2573–2579, Phoenix, Arizona.

Baumel, T., Eyal, M., and Elhadad, M. (2018). Query focused abstractive summarization: Incorporating query relevance, multi-document coverage, and summary length constraints into seq2seq models. *arXiv preprint arXiv:1801.07704*.

Baxendale, P. B. (1958). Machine-made index for technical literaturean experiment. *IBM Journal of research and development*, 2(4):354–361.

Berger, A. and Mittal, V. O. (2000). Query-relevant summarization using faqs. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 294–301, Hong Kong, China.

Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117.

Bubeck, S. and Sellke, M. (2021). A universal law of robustness via isoperimetry. *Advances in Neural Information Processing Systems*, 34.

Cao, Y., Liu, H., and Wan, X. (2020). Jointly learning to align and summarize for neural cross-lingual summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6220–6231, Online.

Cao, Z., Li, W., Li, S., and Wei, F. (2018). Retrieve, rerank and rewrite: Soft template based neural summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 152–161, Melbourne, Australia.

Cao, Z., Wei, F., Dong, L., Li, S., and Zhou, M. (2015). Ranking with recursive neural networks and its application to multi-document summarization. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 2153–2159, Austin, Texas, USA.

Carbonell, J. and Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 335–336, Melbourne Australia.

Chakraborty, S., Bisong, E., Bhatt, S., Wagner, T., Elliott, R., and Mosconi, F. (2020). BioMedBERT: A pre-trained biomedical language model for QA and IR. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 669–679, Online.

Chen, D., Fisch, A., Weston, J., and Bordes, A. (2017). Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1870–1879, Vancouver, Canada.

Chen, P., Wu, F., Wang, T., and Ding, W. (2018). A semantic qa-based approach for text summarization evaluation. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, volume 32, New Orleans, Louisiana, USA.

Chen, S., Zhang, F., Sone, K., and Roth, D. (2021). Improving faithfulness in abstractive summarization with contrast candidate generation and selection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5935–5941, Online.

Cheng, J. and Lapata, M. (2016). Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 484–494, Berlin, Germany.

Clarke, J. and Lapata, M. (2007). Modelling compression with discourse constraints. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1–11, Prague, Czech Republic.

Clarke, J. and Lapata, M. (2010). Discourse constraints for document compression. *Computational Linguistics*, 36(3):411–441.

Conroy, J. M. and O'leary, D. P. (2001). Text summarization via hidden markov models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 406–407, New Orleans, Louisiana, USA.

Dai, Z. and Callan, J. (2019). Deeper text understanding for IR with contextual neural language modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 985–988, Paris, France.

Dang, H. T. (2005). Overview of duc 2005. In *Proceedings of the 2005 Document Understanding Conference*, pages 1–12, Vancouver, Canada.

Dang, H. T. (2006). DUC 2005: Evaluation of question-focused summarization systems. In *Proceedings of the Workshop on Task-Focused Summarization and Question Answering*, pages 48–55, Stroudsburg, PA, USA.

Deutsch, D., Bedrax-Weiss, T., and Roth, D. (2021). Towards question-answering as an automatic metric for evaluating the content quality of a summary. *Transactions of the Association for Computational Linguistics*, 9:774–789.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, Minneapolis, Minnesota.

Dolan, W. B. and Brockett, C. (2005). Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the 3rd International Workshop on Paraphrasing*, pages 9–16, Jeju Island, Korea.

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M., and Hon, H.-W. (2019). Unified language model pre-training for natural language understanding and generation. *Advances in Neural Information Processing Systems*, 32.

Dong, Y., Shen, Y., Crawford, E., van Hoof, H., and Cheung, J. C. K. (2018). BanditSum: Extractive summarization as a contextual bandit. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3739–3748, Brussels, Belgium.

Dou, Z.-Y., Kumar, S., and Tsvetkov, Y. (2020). A deep reinforced model for zero-shot cross-lingual summarization with bilingual semantic similarity rewards. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 60–68, Online.

Dou, Z.-Y., Liu, P., Hayashi, H., Jiang, Z., and Neubig, G. (2021). GSum: A general framework for guided neural abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4830–4842, Online.

Durmus, E., He, H., and Diab, M. (2020). FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online.

Edmundson, H. P. (1969). New methods in automatic extracting. *Journal of the ACM*, 16(2):264–285.

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

Eyal, M., Baumel, T., and Elhadad, M. (2019). Question answering as an automatic evaluation metric for news article summarization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3938–3948, Minneapolis, Minnesota.

Fabbri, A. R., Li, I., She, T., Li, S., and Radev, D. (2019). Multi-News: A large-scale multi-document summarization dataset and abstractive hierarchical model. In

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1074–1084, Florence, Italy.

Fan, A., Grangier, D., and Auli, M. (2018). Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54, Melbourne, Australia.

Feldman, R., Sanger, J., et al. (2007). *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge university press.

Frakes, W. B. and Baeza-Yates, R. (1992). *Information retrieval: data structures and algorithms*. Prentice-Hall, Inc.

Gage, P. (1994). A new algorithm for data compression. *C Users Journal*, 12(2):23–38.

Gehrmann, S., Deng, Y., and Rush, A. (2018). Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium.

Grusky, M., Naaman, M., and Artzi, Y. (2018). NEWSROOM: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 708–719, New Orleans, USA.

Gunasekara, C., Feigenblat, G., Sznajder, B., Aharonov, R., and Joshi, S. (2021). Using question answering rewards to improve abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 518–526, Punta Cana, Dominican Republic.

Guo, J., Fan, Y., Ai, Q., and Croft, W. B. (2016). A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, pages 55–64, Indianapolis, Indiana.

Heilman, M. and Smith, N. A. (2010). Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1011–1019, Los Angeles, California.

Hermann, K. M., Kočiský, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, pages 1693–1701, Cambridge, MA, USA.

Higgins, I., Matthey, L., Pal, A., Burgess, C. P., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017). beta-VAE: Learning basic visual concepts with a constrained variational framework. In *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France.

Hoa, T. (2006). Overview of duc 2006. In *Proceedings of the 2006 Document Understanding Conference*, pages 1–10, New York, USA.

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

Huang, L., Wu, L., and Wang, L. (2020). Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5094–5107, Online.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 448–456, Lille, France.

Jang, E., Gu, S., and Poole, B. (2017). Categorical reparameterization with Gumbel-softmax. In *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France.

Jin, H., Wang, T., and Wan, X. (2020). Semsum: Semantic dependency guided neural abstractive summarization. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, volume 34, pages 8026–8033, New York, USA.

Jing, H. (2000). Sentence reduction for automatic text summarization. In *Proceedings of the Sixth Conference on Applied Natural language Processing*, page 310315, Seattle, Washington, USA.

Jing, H. and McKeown, K. (2000). Cut and paste based text summarization. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, page 178185, Seattle, Washington, USA.

Jing, H. and McKeown, K. R. (1999). The decomposition of human-written summary sentences. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 129–136, Berkeley, California, USA.

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., and Levy, O. (2020). SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Joshi, M., Choi, E., Weld, D., and Zettlemoyer, L. (2017). TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada.

Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., and Wu, Y. (2016). Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

Karush, W. (1939). Minima of functions of several variables with inequalities as side constraints. *M. Sc. Dissertation. Dept. of Mathematics, Univ. of Chicago*.

Katragadda, R. and Varma, V. (2009). Query-focused summaries or query-biased summaries? In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing*, pages 105–108, Suntec, Singapore.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751, Doha, Qatar.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational Bayes. In *Proceedings of the 2nd International Conference on Learning Representations*, Banff, Canada.

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

Kratzwald, B. and Feuerriegel, S. (2018). Adaptive document retrieval for deep question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 576–581, Brussels, Belgium.

Kuhn, H., Tucker, A., et al. (1951). Nonlinear programming. In *Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, California, USA.

Kupiec, J., Pedersen, J., and Chen, F. (1995). A trainable document summarizer. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 68–73, Seattle, Washington, USA.

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., et al. (2019). Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Laskar, M. T. R., Hoque, E., and Huang, J. (2020a). Query focused abstractive summarization via incorporating query relevance and transfer learning with transformer models. In *Proceedings of the 33rd Canadian Conference on Artificial Intelligence*, pages 342–348, online.

Laskar, M. T. R., Hoque, E., and Huang, J. X. (2020b). WSL-DS: Weakly supervised learning with distant supervision for query focused multi-document abstractive summarization. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5647–5654, Online.

Lebanoff, L., Song, K., and Liu, F. (2018). Adapting the neural encoder-decoder framework from single to multi-document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4131–4141, Brussels, Belgium.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., and Kang, J. (2020). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Lee, K., Chang, M.-W., and Toutanova, K. (2019). Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Pro-*

ceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7871–7880, Online.

Lewis, P., Denoyer, L., and Riedel, S. (2019). Unsupervised question answering by cloze translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4896–4910, Florence, Italy.

Li, H., Zhu, J., Liu, T., Zhang, J., Zong, C., et al. (2018). Multi-modal sentence summarization with modality attention and image filtering. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4152–4158, Stockholm, Sweden.

Li, P., Bing, L., Lam, W., Li, H., and Liao, Y. (2015). Reader-aware multi-document summarization via sparse coding. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 1270–1276, Buenos Aires, Argentina.

Li, P., Lam, W., Bing, L., Guo, W., and Li, H. (2017a). Cascaded attention based unsupervised information distillation for compressive summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2081–2090, Brussells, Belgium.

Li, P., Wang, Z., Lam, W., Ren, Z., and Bing, L. (2017b). Salience estimation via variational auto-encoders for multi-document summarization. In *Proceedings of the 31th AAAI Conference on Artificial Intelligence*, pages 3497–3503, San Francisco, California, USA.

Lin, C.-Y. (1999). Training a selection function for extraction. In *Proceedings of the 8th International Conference on Information and Knowledge Management*, pages 55–62, Kansas City, Missouri, USA.

Lin, C.-Y. and Hovy, E. (2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 71–78, Edmonton, Canada.

Liu, P. J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., and Shazeer, N. (2018). Generating Wikipedia by summarizing long sequences. In *Proceedings of the 6th International Conference on Learning Representations*, Vancouver, Canada.

Liu, Y. and Lapata, M. (2019a). Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081, Florence, Italy.

Liu, Y. and Lapata, M. (2019b). Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3730–3740, Hong Kong, China.

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

Mani, I. and Bloedorn, E. (1998). Machine learning of generic and user-focused summarization. In *Proceedings of the 15th AAAI Conference on Artificial Intelligence*, pages 821–826, Madison, Wisconsin, USA.

Mani, I., House, D., Klein, G., Hirschman, L., Firmin, T., and Sundheim, B. M. (1999). The tipster summac text summarization evaluation. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics*, pages 77–85.

Maybury, M. (1999). *Advances in automatic text summarization*. MIT press.

McDonald, R. (2007). A study of global inference algorithms in multi-document summarization. In *Proceedings of the 29th European Conference on Information Retrieval*, pages 557–564, Rome, Italy.

Mihalcea, R. and Tarau, P. (2004). TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain.

Nallapati, R., Zhai, F., and Zhou, B. (2017). Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pages 3075–3081, San Francisco, California, USA.

Nallapati, R., Zhou, B., dos Santos, C., Gulcehre, C., and Xiang, B. (2016). Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany.

Narayan, S., Cohen, S. B., and Lapata, M. (2018a). Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium.

Narayan, S., Cohen, S. B., and Lapata, M. (2018b). Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1747–1759, New Orleans, Louisiana.

Nastase, V. (2008). Topic-driven multi-document summarization with encyclopedic knowledge and spreading activation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 763–772, Honolulu, Hawaii.

Nema, P., Khapra, M. M., Laha, A., and Ravindran, B. (2017). Diversity driven attention model for query-based abstractive summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1063–1072, Vancouver, Canada.

Nenkova, A. and McKeown, K. (2011). *Automatic summarization*. Now Publishers Inc.

Paulus, R., Xiong, C., and Socher, R. (2018). A deep reinforced model for abstractive summarization. In *Proceedings of the 6th International Conference on Learning Representations*, Vancouver, Canada.

Perez-Beltrachini, L. and Lapata, M. (2021). Multi-document summarization with determinantal point process attention. *Journal of Artificial Intelligence Research*, 71:371–399.

Perez-Beltrachini, L., Liu, Y., and Lapata, M. (2019). Generating summaries with topic templates and structured convolutional decoders. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5107–5116, Florence, Italy.

Qi, P., Lin, X., Mehr, L., Wang, Z., and Manning, C. D. (2019). Answering complex open-domain questions through iterative query generation. In *Proceedings of the*

2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 2590–2602, Hong Kong, China.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Rajpurkar, P., Jia, R., and Liang, P. (2018). Know what you dont know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 784–789, Austin, Texas.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Sydney, Australia.

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

Reddy, S., Chen, D., and Manning, C. D. (2019). CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on International Conference on Machine Learning*, pages 1278–1286, Beijing, China.

Robertson, S., Zaragoza, H., et al. (2009). The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Rush, A. M., Chopra, S., and Weston, J. (2015). A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal.

Saeidi, M., Bartolo, M., Lewis, P., Singh, S., Rocktäschel, T., Sheldon, M., Bouchard, G., and Riedel, S. (2018). Interpretation of natural language rules in conversational machine reading. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2087–2097, Brussels, Belgium.

Sandhaus, E. (2008). The New York Times Annotated Corpus. *Linguistic Data Consortium, Philadelphia*, 6(12).

Saurí, R. and Pustejovsky, J. (2009). FactBank: a corpus annotated with event factuality. *Language resources and evaluation*, 43(3):227–268.

See, A., Liu, P. J., and Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1073–1083, Vancouver, Canada.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. (2017). Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, USA.

Srinivasan Iyer, Ioannis Konstas, A. C. J. K. and Zettlemoyer, L. (2017). Learning a neural semantic parser from user feedback. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 963–973, Vancouver, Canada.

Srivastava, R. K., Greff, K., and Schmidhuber, J. (2015). Training very deep networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2*, pages 2377–2385, Montreal, Quebec, Canada.

Stanovsky, G., Michael, J., Zettlemoyer, L., and Dagan, I. (2018). Supervised open information extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 885–895, New Orleans, Louisiana.

Su, D., Xu, Y., Winata, G. I., Xu, P., Kim, H., Liu, Z., and Fung, P. (2019). Generalizing question answering system with pre-trained language model fine-tuning. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 203–211, Hong Kong, China.

Su, D., Xu, Y., Yu, T., Siddique, F. B., Barezi, E., and Fung, P. (2020). CAiRE-COVID: A question answering and query-focused multi-document summarization system for COVID-19 scholarly information management. In *Proceedings of the 1st Workshop on NLP for COVID-19 at EMNLP 2020*, Online.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, 27.

Taylor, W. L. (1953). Cloze Procedure: A new tool for measuring readability. *Journalism Quarterly*, 30(4):415–433.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Wan, X. (2008). Using only cross-document relationships for both generic and topic-focused multi-document summarizations. *Information Retrieval*, 11(1):25–49.

Wan, X., Li, H., and Xiao, J. (2010). Cross-language document summarization based on machine translation quality prediction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 917–926, Uppsala, Sweden.

Wan, X. and Xiao, J. (2009). Graph-based multi-modality learning for topic-focused multi-document summarization. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 1586–1591, Pasadena, California, USA.

Wan, X., Yang, J., and Xiao, J. (2007). Manifold-ranking based topic-focused multi-document summarization. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2903–2908, Hyderabad, India.

Wan, X. and Zhang, J. (2014). CTSUM: extracting more certain summaries for news articles. In *Proceedings of the 37th international ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 787–796, New York, United States.

Wang, A., Cho, K., and Lewis, M. (2020a). Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online.

Wang, M., Smith, N. A., and Mitamura, T. (2007). What is the Jeopardy model? a quasi-synchronous grammar for QA. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 22–32, Prague, Czech Republic.

Wang, Z., Duan, Z., Zhang, H., Wang, C., Tian, L., Chen, B., and Zhou, M. (2020b). Friendly topic assistant for transformer based abstractive summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 485–497, Online.

Wang, Z., Ng, P., Ma, X., Nallapati, R., and Xiang, B. (2019). Multi-passage BERT: A globally normalized BERT model for open-domain question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 5878–5882, Hong Kong, China.

Welbl, J., Stenetorp, P., and Riedel, S. (2018). Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

White, R. W. and Roth, R. A. (2009). Exploratory search: Beyond the query-response paradigm. *Synthesis lectures on information concepts, retrieval, and services*, 1(1):1–98.

Xu, J. and Durrett, G. (2019). Neural extractive text summarization with syntactic compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3292–3303, Hong Kong, China.

Xu, Y. and Lapata, M. (2019). Weakly supervised domain detection. *Transactions of the Association for Computational Linguistics*, 7:581–596.

Xu, Y. and Lapata, M. (2020). Coarse-to-fine query focused multi-document summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 3632–3645, Online.

Xu, Y. and Lapata, M. (2021). Generating query focused summaries from query-free resources. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6096–6109, Online.

Xu, Y. and Lapata, M. (2022). Document summarization with latent queries. *Transactions of the Association for Computational Linguistics*, 10:1–16.

Yan, R., Wan, X., Lapata, M., Zhao, W. X., Cheng, P.-J., and Li, X. (2012). Visualizing timelines: Evolutionary summarization via iterative reinforcement between text and image streams. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 275–284, Maui, Hawaii, USA.

Yang, W., Zhang, H., and Lin, J. (2019a). Simple applications of BERT for ad hoc document retrieval. *arXiv preprint arXiv:1903.10972*.

Yang, Y., Yih, W.-t., and Meek, C. (2015). WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019b). XLNet: Generalized autoregressive pretraining for language understanding. 32.

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W., Salakhutdinov, R., and Manning, C. D. (2018). HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium.

Yao, X., Van Durme, B., Callison-Burch, C., and Clark, P. (2013). Answer extraction as sequence tagging with tree edit distance. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 858–867, Atlanta, Georgia.

Yin, W. and Pei, Y. (2015). Optimizing sentence modeling and selection for document summarization. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 1383–1389, Buenos Aires, Argentina.

Zajic, D., Dorr, B. J., Lin, J., and Schwartz, R. (2007). Multi-candidate reduction: Sentence compression as a tool for document summarization tasks. *Information Processing and Management*, 43(6):15491570.

Zhang, J., Tan, J., and Wan, X. (2018). Adapting neural single-document summarization model for abstractive multi-document summarization: A pilot study. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 381–390, Tilburg University, The Netherlands.

Zhang, J., Zhao, Y., Saleh, M., and Liu, P. (2020). Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, pages 11328–11339, Online.

Zhong, M., Liu, P., Chen, Y., Wang, D., Qiu, X., and Huang, X. (2020). Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online.

Zhu, H., Dong, L., Wei, F., Qin, B., and Liu, T. (2019). Transforming Wikipedia into augmented data for query-focused summarization. *arXiv preprint arXiv:1911.03324*.

Zhu, J., Li, H., Liu, T., Zhou, Y., Zhang, J., and Zong, C. (2018). MSMO: Multimodal summarization with multimodal output. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4154–4164, Brussels, Belgium.

Zhu, J., Zhou, Y., Zhang, J., Li, H., Zong, C., and Li, C. (2020). Multimodal summarization with guidance of multimodal reference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9749–9756.

Zou, Y., Zhang, X., Lu, W., Wei, F., and Zhou, M. (2020). Pre-training for abstractive document summarization by reinstating source text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 3646–3660, Online.