Document Summarization with Latent Queries

Yumo Xu and Mirella Lapata

Institute for Language, Cognition and Computation
School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh EH8 9AB, United Kingdom
yumo.xu@ed.ac.uk mlap@inf.ed.ac.uk

Abstract

The availability of large-scale datasets has driven the development of neural models that create *generic* summaries for single or multiple documents. For *query-focused* summarization (QFS), labeled training data in the form of queries, documents, and summaries is not readily available. 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. Our framework formulates summarization as a generative process, and jointly optimizes a *latent query model* and a *conditional language model*. Despite learning from generic summarization data only, our approach outperforms strong comparison systems across benchmarks, query types, document settings, and target domains.[1]

1 Introduction

Recent years have witnessed substantial progress in *generic* summarization (See et al., 2017; Gehrmann et al., 2018; Liu and Lapata, 2019a, *inter alia*) thanks to neural architectures based on the encoder-decoder paradigm (Sutskever et al., 2014) and the availability of large-scale datasets containing hundreds of thousands of document-summary pairs. Unfortunately, training data of this magnitude is not readily available for the related task of *query-focused* summarization (QFS; Dang 2005) which aims to create a summary from one or multiple document(s) that answers a specific query. Existing QFS benchmarks (Dang, 2005; Hoa, 2006; Nema et al., 2017; Baumel et al., 2016) have been constructively used for evaluation but are relatively small for training large neural models.

To make up for the absence of labeled QFS data, recent work has resorted to distant supervision provided by pretrained models, paraphrase identification, and question-answering datasets (Xu and Lapata, 2020; Su et al., 2020; Laskar et al., 2020b). Other work induces proxy queries (Xu and Lapata, 2021) from generic summarization datasets, without additional question-answering resources that can be also extremely expensive to acquire (Bajaj et al., 2016). Despite this progress, building and scaling QFS systems remains challenging due to the many different ways natural language queries express users' information needs. For instance, queries can have 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 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), namely, queries with different surface forms from those seen in training. In order 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 that can be computationally inefficient and in some cases practically infeasible.

In this work, 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

[1] Our code and models can be found at https://github.com/yumoxu/lqsum.

Transactions of the Association for Computational Linguistics, vol. 10, pp. 623–638, 2022. https://doi.org/10.1162/tacl_a_00480
Action Editor: Wenjie (Maggie) Li. Submission batch: 8/2021; Revision batch: 12/2021; Published 5/2022.

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*
DUC 2007	MDS	Newswire	1,125 (45)	540.3/30.5/250	Composite	*of operations of Amnesty International and what are the international reactions to its activities?*
TD-QFS	MDS	Medical	7,099 (50)	182.9/3.0/250	Title	*Alzheimer's Disease*

Table 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*.

framework formulates abstractive summarization as a generative process, and decomposes the learning objective into: (1) latent query modeling (i.e., generating latent query variables from document observations) and (2) conditional language modeling (i.e., generating summaries conditioned on observed documents and latent queries). To further handle 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 work are threefold: (a) we bring together generic summarization and QFS under a unified modeling framework that 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, that is, 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.

2 Related Work

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) 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 that decides on relevant aspects of the source document. They frame this 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, they produce content selection probabilities for each word, which are then used to restrict the copy mechanism by performing hard masking over the input document. Another line of research controls summary generation via topics (Perez-Beltrachini et al., 2019a; Wang et al., 2020), retrieve-and-edit methods (Cao et al., 2018), factual relations (Jin et al., 2020), keywords, relational triples, or preselected source sentences (Dou et al., 2021).

The majority of previous QFS approaches have been extractive and compose summaries by selecting *central* and *query-relevant* sentences (Wan et al., 2007; Badrinath et al., 2011; Wan and Zhang, 2014; Li et al., 2017b,a). More recently, Xu and Lapata (2020) propose a coarse-to-fine framework that leverages distant supervision from question answering for summary sentence extraction. Abstractive QFS has received significantly less attention in comparison, due to generation models being particularly data-hungry (Lebanoff et al., 2018; Liu and Lapata, 2019a). As a result, resources from a wider range of NLP tasks have been used. Su et al. (2020) rank document paragraphs against queries with the aid of QA and machine reading datasets (Su et al., 2019; Rajpurkar et al., 2016), and then iteratively summarize selected paragraphs. Similarly, Laskar et al. (2020b) jointly exploit supervision from QFS data (typically reserved for evaluation) and related QA and paraphrase identification tasks.

Because query-related resources can be also costly to obtain (Bajaj et al., 2016; Kwiatkowski et al., 2019), Xu and Lapata (2021) use none whatsoever. Instead, they create *proxy queries* by selectively masking information slots in generic summaries. Despite promising system performance,

their approach assumes prior knowledge of target queries (proxies are created to match their length, and content), and a development set is used (Xu and Lapata, 2021). Also, their system is particularly tailored to multi-document QFS and includes a sophisticated evidence selection component. Our work is closely related to theirs in that we also do not take advantage of query-related resources. We go a step further and do not require a development set either, allowing our model to be independent of specific query verbalizations and produce QFS summaries in *zero-shot* settings.

Our approach is generally applicable to single- and multi-document QFS. For any summarization task we assume 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 that rely on 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).

3 Problem Formulation

Let $\{(\mathcal{D}, \mathcal{Q}, \mathcal{S})\}$ denote a summarization dataset, where document $\mathcal{D}$ is a sequence of tokens, and $\mathcal{S}$ its corresponding summary; query $\mathcal{Q}$ additionally specifies an information request. In generic summarization, $\mathcal{Q} = \emptyset$, whereas in QFS $\mathcal{Q}$ can assume various formats, ranging from keywords to composite questions (see Table 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., $\mathcal{Q} \neq \emptyset$), they are incomplete expressions of intent as it is unlikely to specify queries to the level of detail necessary to compose a good summary (Xu and Lapata, 2021). We thus identify *latent* query signals from $\mathcal{D}$, and optionally take advantage of $\mathcal{Q}$ as additional observation for belief update.

Generative Model We model an observed input document $\mathcal{D}$ 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 $\mathcal{D}$. Modeling latent queries at the token-level allows us to regularize the model—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 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 $\mathcal{Q}$; 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 1) as:

$$p_\theta(\mathbf{y}|\mathbf{x}) = \sum_{\mathbf{z}} p_\theta(\mathbf{y}|\mathbf{z}, \mathbf{x}) p_\theta(\mathbf{z}|\mathbf{x}) \quad (1)$$
$$= \sum_{\mathbf{z}} p_\theta(\mathbf{y}|\mathbf{z}, \mathbf{x}) \prod_i p_\theta(\mathbf{z}_i|\mathbf{x}_i)$$

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

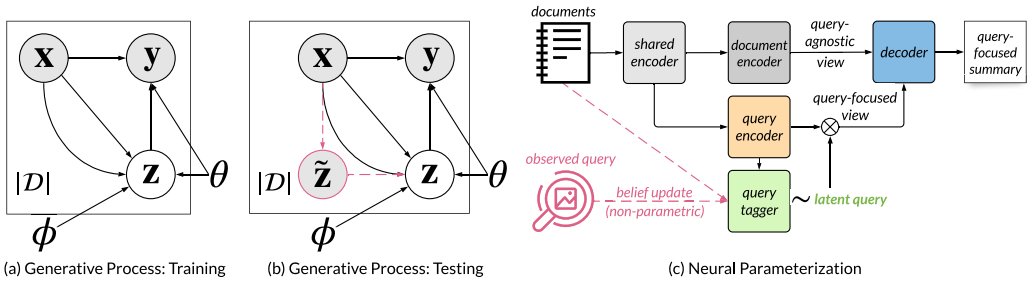
$$p_\theta(\mathbf{z}|\mathbf{x}, \mathbf{y}) = \frac{p_\theta(\mathbf{x}, \mathbf{y}, \mathbf{z})}{p_\theta(\mathbf{x}, \mathbf{y})} = \frac{p_\theta(\mathbf{x}, \mathbf{y}, \mathbf{z})}{\sum_{\mathbf{z}} p_\theta(\mathbf{x}, \mathbf{y}, \mathbf{z})} \quad (2)$$

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:

$$\max_{\phi,\theta} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}} \left[\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})} \log p_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z}) \right] \quad (3)$$

$$\text{subject to } D_{\mathrm{KL}}\left(q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}) \| p_\theta(\mathbf{z}|\mathbf{x})\right) < \delta \quad (4)$$

Because we cannot solve Equation (4) directly, we invoke the Karush-Kuhn-Tucker conditions



(a) Generative Process: Training (b) Generative Process: Testing (c) Neural Parameterization

Figure 1: Proposed summarization framework: generative process and neural parametrization. Shaded nodes represent observed variables, unshaded nodes indicate latent variables, arrows represent conditional dependencies between variables, and plates refer to repetitions of sampling steps. 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.

(Kuhn et al., 1951) and cast the above constrained optimization problem into unconstrained optimization, with the following ELBO objective:

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log p_\theta(\mathbf{y}|\mathbf{x},\mathbf{z})\right] - \beta D_{\text{KL}}\left(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})||p_\theta(\mathbf{z}|\mathbf{x})\right) \quad (5)$$

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) 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 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 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:

$$\mathcal{L} = \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{y}|\mathbf{x},\mathbf{z})\right]}_{\text{conditional language modeling}} + \underbrace{\beta\mathcal{H}\left(q_\phi(\mathbf{z}|\mathbf{x})\right) - \omega\mathcal{H}\left(o(\hat{\mathbf{z}}|\mathbf{x},\mathbf{y}), q_\phi(\mathbf{z}|\mathbf{x})\right)}_{\text{latent query modeling}} \quad (6)$$

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

As can be seen from Equation (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 (6) with a *latent query model* and a *conditional language model* illustrated in Figure 1. 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 $\mathcal{D}$. 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 previous work (Xu and Lapata, 2021), the latent query model and

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

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.

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 (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 $\mathbf{H}_q \in \mathbb{R}^{M \times d_h}$, we project it to $\mathbb{R}^{M \times 2}$ with a two-layer MLP as a scoring function:

$$\mathbf{H}_s = \text{ReLU}(\mathbf{H}_q \mathbf{W}_h + \mathbf{b}_h^\intercal) \tag{7}$$

$$\boldsymbol{\pi} = \mathbf{H}_s \mathbf{W}_s + \mathbf{b}_s^\intercal \tag{8}$$

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:

$$\begin{aligned} q_\phi(\mathbf{z}_i = \ell|\mathbf{x}) &= \text{softmax}_\ell([\boldsymbol{\pi}_0 + g_0, \boldsymbol{\pi}_1 + g_1]) \\ &= \frac{\exp((\boldsymbol{\pi}_\ell + g_\ell)/\tau)}{\sum_{\ell' \in [0,1]} \exp((\boldsymbol{\pi}_{\ell'} + g_{\ell'})/\tau)} \end{aligned} \tag{9}$$

where τ is the temperature controlling how close $q_\phi(\mathbf{z}|\mathbf{x})$ is to $\arg\max_\ell q_\phi(\mathbf{z}|\mathbf{x})$, 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 $\mathcal{D}$ (which we discuss in Section 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:

$$\mathbf{Q}_i = q_\phi(\mathbf{z}_i = 1|\mathbf{x}) \cdot \mathbf{H}_{q,i} \tag{10}$$

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:

$$\frac{\partial \mathbf{Q}_i}{\partial \mathbf{H}_{q,i}} = \underbrace{\left(1 - q_\phi^{(1)}\right)}_{\text{carry gate}} \cdot \frac{\partial \Delta\pi}{\partial \mathbf{H}_{q,i}} \odot \mathbf{Q}_i + \underbrace{q_\phi^{(1)}}_{\text{transform gate}} \cdot \mathbf{1} \tag{11}$$

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 (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.

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 (6)), thereby mitigating its collapse (i.e., the decoder may learn to ignore the query-focused view and instead solely rely on the query-agnostic view).

Let $t_1, \ldots, t_n$ denote binary tags for each of the source tokens, that is, 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 gold-standard 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. If there exist multiple identical LCSs, only the one appearing at the earliest document position is

[4] We also experimented with drawing hard samples from $\mathbf{z}$ via the straight-through trick (Jang et al., 2016), which is differentiable with biased gradient estimation. However, it did not yield better results than continuous relaxation.

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 our tagging model is therefore able to capture dependencies between tokens, albeit indirectly.

Training To optimize the variational inference model, that is, the MLP defined in Equations (7–9), we use a cross entropy loss for token tagging, with the posterior entropy term from Equation (6). Formally, we write the query modeling loss as follows:

$$\begin{aligned}\mathcal{L}_{\text{query}} &= -\omega\mathcal{L}_{\text{tag}} + \beta\mathcal{L}_{\text{entropy}} \\ &= -\sum_{j=1}^{N}\sum_{i=1}^{M}\big((\omega\hat{\mathbf{z}}_i^j - \beta q_\phi^{(1)})\log q_\phi^{(1)} \\ &+ (\omega(1-\hat{\mathbf{z}}_i^j) - \beta q_\phi^{(0)})\log q_\phi^{(0)}\big)\end{aligned} \tag{12}$$

where $\hat{\mathbf{z}}_i$ is a binary label automatically assigned via BPE-LCS$(\mathcal{D}, \mathcal{S})$, 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 that 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 (12). We then linearly anneal α over optimization steps so that the gradients from the summarization objective (which we introduce in Section 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 $\mathcal{Q} \neq \emptyset$, 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$(\mathcal{D}, \mathcal{Q})$, and all other tokens to zero. We further incorporate query information via a simple calibration as:

$$\begin{aligned}q_\phi(\mathbf{z}_i = 1|\mathbf{x}, \tilde{\mathbf{z}}) = \min\{1, \\ q_\phi(\mathbf{z}_i = 1|\mathbf{x}) + \Delta(\mathbf{z}_i = 1|\mathbf{x}, \tilde{\mathbf{z}})\}\end{aligned} \tag{13}$$

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 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 (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 (10)). As shown in Figure 1(c), our encoder module consists of three encoders: a shared encoder, a document encoder, and a query encoder. Because both views are created from the same document, we use a shared encoder for general document understanding that 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:

$$\begin{aligned}\mathbf{H}^{(\text{enc})} &= \text{LN}\big(\mathbf{H}^{(\text{enc})} + \text{MHA}\big(\mathbf{H}^{(\text{enc})}, \mathbf{H}^{(\text{enc})}, \mathbf{H}^{(\text{enc})}\big)\big) \\ \mathbf{H}^{(\text{enc})} &= \text{LN}\big(\mathbf{H}^{(\text{enc})} + \text{FFN}\big(\mathbf{H}^{(\text{enc})}\big)\big)\end{aligned} \tag{14}$$

where LN denotes layer normalization. As shown in Figure 1(c), 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:

$$\begin{aligned}\mathbf{H}^{(\text{dec})} &= \text{LN}\big(\mathbf{H}^{(\text{dec})} + \text{MHA}\big(\mathbf{H}^{(\text{dec})}, \mathbf{H}^{(\text{dec})}, \mathbf{H}^{(\text{dec})}\big)\big) \\ \mathbf{H}^{(\text{dec})} &= \text{LN}\big(\mathbf{H}^{(\text{dec})} + \text{MHA}\big(\mathbf{H}^{(\text{dec})}, \mathbf{Q}, \mathbf{Q}\big)\big) \\ \mathbf{H}^{(\text{dec})} &= \text{LN}\big(\mathbf{H}^{(\text{dec})} + \text{MHA}\big(\mathbf{H}^{(\text{dec})}, \mathbf{D}, \mathbf{D}\big)\big) \\ \mathbf{H}^{(\text{dec})} &= \text{LN}\big(\mathbf{H}^{(\text{dec})} + \text{FFN}\big(\mathbf{H}^{(\text{dec})}\big)\big)\end{aligned} \tag{15}$$

After taking the context of the previous generation $\mathbf{H}^{(\text{dec})}$ 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:

$$\mathcal{L}_{\text{lm}} = \sum_{j=1}^{N}\sum_{t=1}^{T} \log p_\theta \left(\mathbf{y}_t^j | \mathbf{y}_{<t}^j, \mathbf{D}^j, \mathbf{Q}^j\right) \quad (16)$$

which is jointly trained with the query model (see Equation (12)) as: $\mathcal{L} = \mathcal{L}_{\text{lm}} + \mathcal{L}_{\text{query}}$.

6 Experimental Setup

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 1 summarizes the characteristics of these datasets and presents test set statistics. Note that in contrast to Xu and Lapata (2021), we do not make use of development data for our QFS tasks.

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.

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

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 DUC (Dang, 2005) and report R-SU4 (based on skip bigram with maximum skip distance of 4) instead of R-L.[5]

7.1 Supervised Setting

Table 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 that 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

[5]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 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.

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., 2019b)	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 4: Multi-document summarization, zero-shot setting, **WikiCatSum** test set. Results averaged over three domains: *Company*, *Film*, *Animal*.

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 2 for an overview). PTGEN (See et al., 2017) and BOTTOMUP (Gehrmann et al., 2018) do not use pretrained 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.

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 3 for details).

7.2 Zero-Shot Setting

Multi-Document Summarization We evaluated our model's ability to summarize multiple documents on WikiCatSum (Perez-Beltrachini et al., 2019b), 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., 2019b) 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 4 we present upper bound and baseline results. The second block contains results for two supervised systems, a sequence-to-sequence model 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. 2019b). 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 and 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 1). The first block in both tables shows results for the ORACLE upper bound, LEAD, and LEXRANK$_Q$, a query-focused version of LEXRANK

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

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
hline 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 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., 2018).

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 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 1).

We applied our summarization model 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).[6] Repeated sentences were skipped to reduce redundancy in the final summary.

Our results are given in Table 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 (Xu and Lapata, 2021), 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 that takes advantage of existing QA datasets and adopts a coarse-to-fine salience estimation procedure. BART-CAQ (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 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

[6] An alternative would be to generate a long summary at once. However, this requires a model to be trained on a MDS dataset, or at least a proxy thereof (Xu and Lapata, 2021).

[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 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.

	CNN/DM			WikiRef			Debatepedia			DUC 2006			DUC 2007			TD-QFS		
Model	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-SU4	R-1	R-2	R-SU4	R-1	R-2	R-SU4
LQSUM	45.1	22.0	41.9	31.1	12.6	27.1	23.5	7.2	20.6	39.1	8.5	13.7	40.4	10.2	15.0	45.7	18.1	22.1
$-\Delta(\hat{\mathbf{z}}\|\mathbf{x},\mathbf{z})$	—	—	—	↓0.1	↓0.2	↓0.2	↓0.5	↓0.3	↓0.6	↓0.6	↓0.2	↓0.6	↑0.1	↓0.1	↓1.3	↑0.1	↓0.6	↓0.4
−Joint training	↓0.4	↓0.3	↓0.4	↓2.9	↓0.9	↓2.8	↓2.8	↓1.1	↓2.8	↓2.9	↓1.7	↓1.6	↓2.4	↓2.0	↓1.7	↓0.7	↓0.6	↓0.4
−Weak supervision	↓0.6	↓0.7	↓0.7	↓0.7	↓0.2	↓0.5	↓1.0	↓0.5	↓1.3	↓0.2	↓0.2	↓0.2	↓0.2	↓0.3	↓0.3	↓0.1	↓0.3	↓0.0
−Dual view	↓2.7	↓3.5	↓2.5	↓12.2	↓9.3	↓10.5	↓7.9	↓3.3	↓6.6	↓6.3	↓1.8	↓1.8	↓6.5	↓3.0	↓2.5	↓2.5	↓3.3	↓2.8
−Posterior dropout	↓0.7	↓0.6	↓0.8	↓0.8	↓0.3	↓0.7	↓1.1	↓0.3	↓1.2	↓0.2	↓0.2	↓0.2	↓0.4	↓0.4	↓0.5	↑0.2	↓0.0	↑0.1

Table 8: LQSUM ablation results; ↑/↓: absolute increase/decrease.

various *zero-shot* systems including BART and GSUM+LEXRANK$_Q$.

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.

7.3 Ablation Studies

We further performed a series of ablation studies, reported in Table 8, 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 that 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 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 performance as it leads to more severe error propagation for the downstream generation model.

8 Human Evaluation

Following previous work (Xu and Lapata, 2021, 2020), 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 two dimensions: Succinctness (does the summary avoid unnecessary detail and redundant

[8] −Joint training replaces the $\mathrm{softmax}$ in Equation (9) with $\arg\max$, to stop the gradients from the generation loss in backpropagation. −Weak supervision sets $\omega = 0$.

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†	2.93$^{\dagger\circ}$	2.59†
GSUM+LEXRANK$_Q$	2.88†	3.60	3.49†
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 9: Human evaluation on QFS benchmarks: average **Rel**evance, **Suc**cinctness, **Coh**erence ratings; $\dagger/\circ$: sig different from LQSUM/Gold (at $p < 0.05$, using a pairwise t-test); best system shown in bold.

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. Thomas's 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 Parliament's approval. *The CPIL was founded in the late 1980s by Justice V.M. Tarkunde, who also co-founded the People's 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.
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 10: System output on WikiRef (above; document 3918) and and Debetepedia (below; document 260). Information *irrelevant to the query* or **incoherent in the summary** is highlighted.

information?) and Coherence (does the summary make logical sense?). The ratings were obtained using a five-point Likert scale.

In addition, participants were asked to assess the Relevance of the summary to the query. Crowdworkers read a summary and for each *sentence* decided whether it is relevant (i.e., provides an answer to the query), irrelevant (i.e., does not answer the query), or partially relevant (i.e., unclear it directly answers the query). Relevant sentences were awarded a score of 5, partially relevant ones a score of 2.5, and 0 otherwise. Sentence scores were 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 obtained per-sentence ratings which we then aggregated to an overall summary score. To make this task manageable, raters were asked to provide more coarse-grained ratings.

Participants assessed summaries created by LQSUM (our model), 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

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

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.[10]

We show our results in Table 9 and examples of system output in Table 10. 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 MargeSum on DUC. On Wikiref, BertExt is slightly more relevant but less coherent.

9 Conclusion

We propose 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.

Directions for future work are many and varied. One research challenge is to push this low-resource approach even further and generate abstractive summaries without access to any summaries or queries. We would also like to extend the proposed framework to cross-lingual settings, and satisfy the information needs of users with different language backgrounds through effective query understanding and summary generation.

Acknowledgments

The authors would like to thank the action editor, Wenjie Li, and the anonymous reviewers for their valuable feedback. We acknowledge the financial support of the European Research Council (Lapata; award number 681760). This research was supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via contract FA8650-17-C-9118. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

[10]We are grateful to Md Tahmid Rahman Laskar and Haichao Zhu for providing us with system output.

References

Deen Mohammad Abdullah and Yllias Chali. 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.

Rama Badrinath, Suresh Venkatasubramaniyan, and CE Veni Madhavan. 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. https://doi.org/10.1007/978-3-642-20161-5_64

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2016. MS MARCO: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

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

Tal Baumel, Matan Eyal, and Michael Elhadad. 2018. Query focused abstractive summarization: Incorporating query relevance, multi-document coverage, and summary length constraints into

seq2seq models. *arXiv preprint arXiv:1801.07704*.

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 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.

Sihao Chen, Fan Zhang, Kazoo Sone, and Dan Roth. 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. https://doi.org/10.18653/v1/2021.naacl-main.475

Hoa Trang Dang. 2005. Overview of duc 2005. In *Proceedings of the 2005 Document Understanding Conference*, pages 1–12. Vancouver, Canada.

Hoa Trang Dang. 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. https://doi.org/10.3115/1654679.1654689

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 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. https://doi.org/10.18653/v1/2021.naacl-main.384

Günes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479. https://doi.org/10.1613/jair.1523

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109. Brussels, Belgium. https://doi.org/10.18653/v1/D18-1443

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 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.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-vae: Learning basic visual concepts with a constrained variational framework. In *Proceedings of the 5th International Conference on Learning Representations*. Toulon, France.

T. D. Hoa. 2006. Overview of DUC 2006. In *Proceedings of the 2006 Document Understanding Conference*. New York, USA.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Hanqi Jin, Tianming Wang, and Xiaojun Wan. 2020. Semsum: Semantic dependency guided neural abstractive summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8026–8033. New York, USA. https://doi.org/10.1609/aaai.v34i05.6312

H. W. Kuhn, A. W. Tucker. 1951. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492. California, USA.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research.

Transactions of the Association for Computational Linguistics, 7:453–466. https://doi.org/10.1162/tacl_a_00276

Md Tahmid Rahman Laskar, Enamul Hoque, and Jimmy Huang. 2020a. Query focused abstractive summarization via incorporating query relevance and transfer learning with transformer models. In *Canadian Conference on Artificial Intelligence*, pages 342–348. Springer. https://doi.org/10.1007/978-3-030-47358-7_35

Md Tahmid Rahman Laskar, Enamul Hoque, and Jimmy Xiangji Huang. 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.

Logan Lebanoff, Kaiqiang Song, and Fei Liu. 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. https://doi.org/10.18653/v1/D18-1446

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880. Online. https://doi.org/10.18653/v1/2020.acl-main.703

Piji Li, Wai Lam, Lidong Bing, Weiwei Guo, and Hang Li. 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. Brussels, Belgium.

Piji Li, Zihao Wang, Wai Lam, Zhaochun Ren, and Lidong Bing. 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.

Chin-Yew Lin and Eduard Hovy. 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.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating Wikipedia by summarizing long sequences. In *Proceedings of the 6th International Conference on Learning Representations*. Vancouver, Canada.

Yang Liu and Mirella Lapata. 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. https://doi.org/10.18653/v1/P19-1500

Yang Liu and Mirella Lapata. 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. https://doi.org/10.18653/v1/D19-1387

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 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. https://doi.org/10.18653/v1/K16-1028

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. 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. https://doi.org/10.18653/v1/D18-1206

Preksha Nema, Mitesh M. Khapra, Anirban Laha, and Balaraman Ravindran. 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. https://doi.org/10.18653/v1/P17-1098

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

Laura Perez-Beltrachini, Yang Liu, and Mirella Lapata. 2019a. 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. https://doi.org/10.1613/jair.1.12522

Laura Perez-Beltrachini, Yang Liu, and Mirella Lapata. 2019b. Generating Summaries with Topic Templates and Structured Convolutional Decoders. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy. https://doi.org/10.18653/v1/P19-1504

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 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. https://doi.org/10.18653/v1/D16-1264

Alexander M. Rush, Sumit Chopra, and Jason Weston. 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.

Abigail See, Peter J. Liu, and Christopher D. Manning. 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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 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. https://doi.org/10.18653/v1/P16-1162

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 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.

Dan Su, Yan Xu, Genta Indra Winata, Peng Xu, Hyeondey Kim, Zihan Liu, and Pascale Fung. 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. https://doi.org/10.18653/v1/D19-5827

Dan Su, Yan Xu, Tiezheng Yu, Farhad Bin Siddique, Elham Barezi, and Pascale Fung. 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. https://doi.org/10.18653/v1/2020.nlpcovid19-2.14

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. 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.

Xiaojun Wan and Jianmin Zhang. 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.

Zhengjue Wang, Zhibin Duan, Hao Zhang, Chaojie Wang, Long Tian, Bo Chen, and Mingyuan Zhou. 2020. 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. https://doi.org/10.18653/v1/2020.emnlp-main.35

Yumo Xu and Mirella Lapata. 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.

Yumo Xu and Mirella Lapata. 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.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208. Online. https://doi.org/10.18653/v1/2020.acl-main.552

Haichao Zhu, Li Dong, Furu Wei, Bing Qin, and Ting Liu. 2019. Transforming Wikipedia into augmented data for query-focused summarization. *arXiv preprint arXiv:1911.03324*.