

Text 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

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 summaries from single documents, for *generic* purposes. When using a summarization system, users often have *specific* intents with various language realizations, which, depending on the information need, can range from a single keyword to a long narrative composed of multiple questions. Existing summarization systems, however, often either fail to support or act robustly on this *query focused* summarization task. We introduce LAQSUM, the first unified text summarization system that learns **Latent Queries** from documents for abstractive summarization with any existing query forms. Under a deep generative framework, our system jointly optimizes a *latent query model* and a *conditional language model*, allowing users to plug-and-play queries of any type at test time. Despite learning from only generic summarization data and requiring no further optimization for downstream summarization tasks, our system robustly outperforms strong comparison systems across summarization benchmarks with different query types, document settings, and target domains.

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, *inter alia*) thanks to the availability of large-scale datasets containing hundreds of thousands of document-summary pairs. *Query focused* summarization (QFS; Dang 2005) which aims to create a short summary from one or multiple document(s) that answers a specific query, in comparison, does not have training data of this magnitude; existing QFS corpora (Dang, 2005; Hoa, 2006; Nema et al., 2017; Baumel et al., 2016) are relatively small for training large neural architectures and have been mostly used for evaluation.

Therefore, how to leverage generic summarization data for the benefit of QFS has become a research topic of interest recently (Xu and Lapata, 2020a; Laskar et al., 2020a). It, however, remains a challenging research question due to the absence of queries in generic summarization data.

Early attempts in QFS sidestep this problem by seeking distant supervision from query-relevant NLP tasks (Xu and Lapata, 2020b; Su et al., 2020; Laskar et al., 2020b) and exploiting existing resources, including datasets and pretrained models in question answering (Rajpurkar et al., 2016; Chakraborty et al., 2020) and paraphrase identification (Dolan and Brockett, 2005). Since these resources can also be extremely expensive to acquire (Bajaj et al., 2016), recent work proposes to induce proxy queries from generic summaries (Xu and Lapata, 2020a), enabling an abstractive QFS system to be trained with only query-free resources.

In this work, we note that queries can be realized in various language forms, including but not limited to one or multiple keyword(s) (Baumel et al., 2016; Zhu et al., 2019), a natural question (Nema et al., 2017) and a composition of multiple sub-queries (Dang, 2006). These diversified query languages inevitably lead to a mismatch between the actual queries a system has to take as inputs at test time, and the queries pre-collected or generated from distant supervision for training purposes. Moreover, the scalability of these approaches is constrained in terms of handling different query types. For instance, the performance of an abstractive QFS system trained on proxy queries can be sensitive to the format of target queries at test time which the proxy ones were created to mimic (Xu and Lapata, 2020a). To make a summarization system work well with every new query type, one may have to re-design the proxy algorithm, re-create the proxy queries, and re-train one or more system modules, which is computationally inefficient and sometimes practically infeasible.

In this work, we aim at building an abstractive text summarization system that is robust across *observed* and *latent* query settings. Particularly, we treat generic summarization data as a special case of QFS where the query is unspecified and *empty*, and assume it to be the only accessible resource for *both* model training and development. To build a generalizable summarization system, we propose to model queries as *discrete latent variables* from documents, and build a latent query representation that is compatible with different query language realizations as inputs. Specifically, we formulate an abstractive summarization task as a generative process, and decompose the objective into two components: (1) *latent query modeling* (i.e., generating latent query variables from a document observation) and (2) *conditional language modeling* (i.e., generating an abstractive summary conditioned on the observed document and latent queries). To further enable user-specified query inputs of different formats at test time, we provide a non-parametric calibration to the latent query distribution; it can be plugged into the latent variable model without re-training, and therefore enables *zero-shot QFS*.

Our contributions in this work are threefold: we propose the *first text summarization system that unifies abstractive generic summarization and QFS*. Particularly, *no query-related resource is required for model training or development*; we provide a general, deep generative formulation for text summarization, under which we validate the effectiveness of *representing queries directly from input documents in the latent space*, i.e., without resorting to pipeline-style query extraction or generation; we provide experimental results on a wide spectrum of summarization benchmarks and show that across query types, document settings, and target domains, our system achieves better results than strong comparison systems.

2 Related Work

2.1 Generic Abstractive Summarization

Rush et al. (2015) and Nallapati et al. (2016) are among the first to apply the neural encoder-decoder architecture to abstractive summarization. See et al. (2017) enhance the system with a pointer-generator model where a copy mechanism is proposed to allow words in the source document to be copied directly in summary generation. In a bottom-up fashion, Gehrmann et al. (2018) propose to train a word-level tagging model separately; at test time,

it produces content selection probabilities for each word, which are then used to restrict the copy mechanism by performing hard masking over the input document. Inspired by Gehrmann et al. (2018), we also propose to *train a tagging model for abstractive summarization*. In this work, a tagger serves as a parameterized component of an inference model for latent query variables and is jointly optimized via a generic summarization task and a weakly supervised tagging task.

Another line of research in abstractive summarization propose to control text generation via topics (Perez-Beltrachini et al., 2019; Wang et al., 2020), retrieved summaries (Cao et al., 2018), or factual relations (Zhu et al., 2021). Notably, Dou et al. (2020) propose a controllable system for abstractive summarization which achieves state-of-the-art performance by taking sentences extracted by another state-of-the-art *extractive* system (Zhong et al., 2020) as guidance at test time. Despite the conceptual similarity between guidance and queries, we note that one fundamental difference between guided summarization and query focused summarization lies in their task objectives: guidance is created for improving generic summarization on aspects such as informativeness (Cao et al., 2018) and faithfulness (Zhu et al., 2021), while QFS handles user intents of various forms in a low-resource setting where the lack of training data for high-quality guidance creation renders a guided system not directly applicable.

2.2 Query Focused Summarization

Extractive QFS, the dominant approach in QFS research, composes summaries by selecting *central* and *query-relevant* sentences in documents, based on different ways of estimating and incorporating centrality and relevance (Wan et al., 2007; Badrinath et al., 2011; Wan and Zhang, 2014; Li et al., 2017b,a). More recently, Xu and Lapata (2020b) propose a coarse-to-fine framework that leverages distant supervision from question answering for summary sentence extraction.

Abstractive QFS, compared to either its generic or extractive counterpart, has received significantly less attention, due to generation models being particularly data-hungry (Lebanoff et al., 2018; Liu and Lapata, 2019a). To alleviate the scarcity of QFS data, resources from a wider range of NLP tasks have been investigated for generating query focused abstracts. Su et al. (2020) rank document

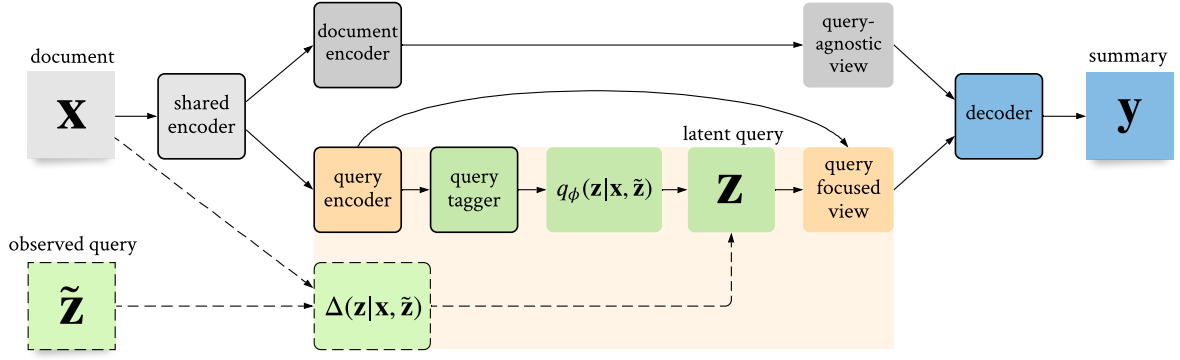


Figure 1: Proposed text summarization framework. Dashed lines denote non-parametric query observation modeling at testing, which is optional. Shadow denotes components for latent query modeling, which infers discrete latent variables as queries. Latent queries create a query focused view of the input document, which, together with a query-agnostic view, is input to a conditional language model for summary generation.

paragraphs against queries with a plethora of QA and machine reading datasets (Su et al., 2019; Rajpurkar et al., 2016), then summarize selected paragraphs iteratively. Similarly, Laskar et al. (2020b) jointly exploit supervision from QFS data (typically reserved for evaluation) and related QA and paraphrase identification tasks.

Since external query-related resources such as QA datasets can also be costly to obtain (Bajaj et al., 2016; Kwiatkowski et al., 2019), Xu and Lapata (2020a) assume access to only *query-free* resources for abstractive QFS training. They discover a type of connection between queries and summaries, and, alternatively, create proxy queries from generic summaries. These proxy queries are created by selectively masking information slots in summaries. To minimize covariance shift, they are created to be as close to target queries as possible, on aspects such as content and length. Despite the promising system performance and relaxed training resource assumption, this approach still assumes prior knowledge of the target query type at testing, and therefore relies on a development set for proxy query generation and model selection (Xu and Lapata, 2020a). Also, the system is particularly tailored for *multi-document* QFS with an evidence selection component.

In this work, we aim at *zero-shot* transfer setting for QFS tasks, i.e., we do not assume the availability of even one small QFS dataset for development purposes, as it is challenging to obtain a development set for every query language form. We present a system that can performing well on both single- and multi-document QFS across both observed and latent query settings.

3 Problem Formulation

Let $\{(\mathcal{D}, \mathcal{Q}, \mathcal{S})\}$ denote a summarization dataset, where \mathcal{D} is a document token sequence with corresponding summary \mathcal{S} , and query \mathcal{Q} additionally specifies an information request. We have $\mathcal{Q} = \emptyset$ in generic summarization, while in QFS, depending on the test set, \mathcal{Q} can be in various formats, from keywords to composite questions (see Table 2 for examples of different query types).

In this work, we aim at building a summarization system that learns only from generic summarization data, while robustly generalizing to a range of summarization tasks at test time, including in-domain generic summarization and out-of-domain QFS tasks. One common characteristic of generic summarization and QFS is the *under-specified* user intent: even in QFS, despite $\mathcal{Q} \neq \emptyset$, a query usually presents incomplete guidance signals and does not fully represent a user’s intent, as a query usually *seeks* information, so it is unlikely for a user to specify every aspect of what is needed to compose a good summary (Xu and Lapata, 2020a). Therefore, for both generic summarization and QFS, it is favorable that a system can identify *latent* query signals from \mathcal{D} , to which, \mathcal{Q} can, optionally, serve as an additional observation for belief update.

Generative Model We model the observed input document \mathcal{D} as a sequence of random variables $\mathbf{x} = [\mathbf{x}_1; \mathbf{x}_2; \dots; \mathbf{x}_M]$ where \mathbf{x}_i is a token and M length of document. We define *latent query*, a sequence of discrete latent states over the input document tokens: $\mathbf{z} = [\mathbf{z}_1; \mathbf{z}_2; \dots; \mathbf{z}_M]$. Specifically, from each document token \mathbf{x}_i , we generate a binary query variable \mathbf{z}_i , whose distribution $p(\mathbf{z}_i)$ indicates the belief that \mathbf{x}_i contributes to a potential

query for the document \mathcal{D} . The output variable $\mathbf{y} = [\mathbf{y}_1; \mathbf{y}_2; \dots; \mathbf{y}_T]$ is then generated from $\{\mathbf{x}, \mathbf{z}\}$ using teacher-forcing at training time. Note that at test time, one may have access to additional query knowledge \mathbf{Q} ; we also *ground* this optional query information to the input document as discrete *observed* variables $\tilde{\mathbf{z}} = [\tilde{\mathbf{z}}_1; \tilde{\mathbf{z}}_2; \dots; \tilde{\mathbf{z}}_M]$, and generate \mathbf{y} by additionally conditioning on $\tilde{\mathbf{z}}$ (if it exists) in an autoregressive manner.

We consider modeling the **conditional distribution** $p_\theta(\mathbf{y}|\mathbf{x})$ and write down its factorization according to the generative process described above:

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

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

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

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

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

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

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

Since we cannot solve Equation (4) directly, we apply the Karush–Kuhn–Tucker conditions (KKT; Kuhn et al. 1951) and cast the constrained optimization problem into unconstrained optimization, with the following form of ELBO objective:

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

$$- \beta D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}) || p_\theta(\mathbf{z}|\mathbf{x}))$$

In this work, to minimize our system’s dependence on testing data, we adopt a uniform prior $p_\theta(\mathbf{z}|\mathbf{x})$, based on the hypothesis that given all instances of \mathbf{x} in data (which may come from different domains), the aggregated probability of variable \mathbf{z} , i.e., whether a word is a query word under various contexts, follows a uniform distribution. In this

case, minimizing the KL term is equivalent to maximizing the entropy of the variational posterior.¹ For posterior approximation, we further assume $\mathbf{z} \perp \mathbf{y}$ and therefore $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}) = q_\phi(\mathbf{z}|\mathbf{x})$. While this reduces the risk of exposure bias from \mathbf{y} during learning, it is more challenging to learn meaningful latent variables as they solely condition on \mathbf{x} . We mitigate this issue by introducing a new type of weak supervision $o(\hat{\mathbf{z}}|\mathbf{x}, \mathbf{y})$ that we are able to automatically extract from data. This weak supervision, which is in the form of sequence tagging and will be discussed in Section 4, leads an extra objective term for posterior regularization. Formally, we rewrite the final optimization objective as:

$$\mathcal{L} = \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z})]}_{\text{conditional language modeling}} \quad (6)$$

$$+ \underbrace{\beta \mathcal{H}(q_\phi(\mathbf{z}|\mathbf{x})) - \omega \mathcal{H}(o(\hat{\mathbf{z}}|\mathbf{x}, \mathbf{y}), q_\phi(\mathbf{z}|\mathbf{x}))}_{\text{latent query modeling}}$$

where $\mathcal{H}(\cdot)$ denotes posterior entropy and $\mathcal{H}(\cdot, \cdot)$ denotes cross entropy for posterior regularization from weak supervision. Particularly, we decompose a summarization task into two modeling objectives, *latent query modeling* and *conditional language modeling*. Inside query modeling, hyperparameter ω controls the influence from the weak supervision $\hat{\mathbf{z}}$, while β controls the strength of a form of label smoothing (see Section 4 for details).

Neural Parameterization We show our framework in Figure 1. We parameterize the two modeling objectives in Equation (6) with a *latent query model* and an *conditional language model*, respectively. The query model is optimized to estimate latent query \mathbf{z} from its input variable \mathbf{x} . At inference time, it, optionally, conditions on prior query knowledge $\hat{\mathbf{z}}$ (when available in QFS data). The conditional language model, on the other hand, augments the standard encoder-decoder structure to take as input two encoding sources: a *query-agnostic view* and a *query focused view*. We call them two *views* as they are separate encodings of the same input \mathcal{D} , with the only difference lies in the direct dependence on \mathbf{z} generated from the query model. In contrast to using only the query-focused view, maintaining two views allows the original document context to be retained as a complementary feature space. Finally, a decoder is used

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

Error	Data	Example
Type I	Summary	Real Madrid slump to defeat against Athletic Bilbao. Solitary goal from Aritz Aduriz enough to give the Basques victory. Bayern Munich continue Bundesliga domination.
	Document	..., with a convincing 4-1 win over Lens at the Parc de Princes. ...
	Annotation	..., with#0 a#0 convincing#0 4-1#0 win#0 over#0 Lens#0 at#0 the#0 Parc#0 de#1 Princes.#0 ...
Type II	Summary	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	... For \$89, self-styled entrepreneur Kyle Waring will ship you 6 pounds of Boston-area snow in an insulated Styrofoam box ...
	Annotation	... For#1 \$89,#1 self-styled#0 entrepreneur#0 Kyle#0 Waring#0 will#1 ship#1 you#0 6#1 pounds#1 of#1 Boston-area#0 snow#1 in#1 an#1 insulated#1 Styrofoam#1 box#1 ...

Table 1: Error analysis for longest common sub-sequence tagging scheme (LCS; Gehrmann et al. 2018) with instances from CNN/DM validation set. Document tokens and binary annotations are separated with #. Bold fonts denote erroneous annotations and their corresponding token in summaries/documents.

to generate summary \mathbf{y} auto-regressively. Different from previous abstractive QFS formulation (Xu and Lapata, 2020a), we jointly train these two models in a fully differentiable summarization system.

4 Latent Query Model

In this section we detail the neural parameterization for latent query modeling in Equation (6), including the architecture for query inference and document representation in a query focused view. We also introduce learning from weak supervision $o(\hat{\mathbf{z}}|\mathbf{x}, \mathbf{y})$ and testing with belief update.

Inference Network for Latent Queries We construct a neural network model to infer query belief for each token in the input document. 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:

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

$$\boldsymbol{\pi} = \mathbf{H}_s \mathbf{W}_s + \mathbf{b}_s^\top \quad (8)$$

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 $g_\ell \sim G(0)$, $\ell \in [0, 1]$ are i.i.d. gumbel noise. We normalize $\boldsymbol{\pi}$ to form a variational distribution as:

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

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 In addition to the query-agnostic document representation, a direct encoding of the input document \mathcal{D} adopted by a plethora of summarization models (which we leave to Section 5), we further introduce a query focused view, an encoding factorized via latent queries \mathbf{z} .

Specifically, for the i th token, we take the continuous relaxation of its discrete latent variable \mathbf{z}_i , and ground the query to the input document in the representation space via:²

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

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

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

where $q_\phi^{(\ell)}$ is the shorthand for the variational probability of $\mathbf{z}_i = \ell|\mathbf{x}$, and $\Delta\pi = \boldsymbol{\pi}_1 - \boldsymbol{\pi}_0$ (see Equation (8)). $\mathbf{1}$ denotes an all-one vector. This can be seen as a special case of highway networks (Srivastava et al., 2015) with a zero-map transform function, where the transform gate $q_\phi^{(1)}$ controls the information compression rate.

²We also experimented with drawing hard samples of \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.

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	\emptyset
WikiRef	SDS	Wiki	12,000	398.7/6.7/36.2	Keywords	<i>Marina Beach, Incidents</i>
Debatepedia	SDS	Debate	1,000	66.4/10.0/11.3	Question	<i>Is euthanasia better than withdrawing life support?</i>
DUC 2006	MDS	Cross	1,250 (50)	699.3/32.8/250	Composite	<i>Amnesty International - What is the scope of operations of Amnesty International and what are the international reactions to its activities?</i>
DUC 2007	MDS	Cross	1,125 (45)	540.3/30.5/250		
TD-QFS	MDS	Medical	7,099 (50)	182.9/3.0/250	Title	<i>Alzheimer's Disease</i>

Table 2: Test data statistics. SDS and MDS stand for single- and multi-document summarization, respectively. Size refers to number of documents for single-document test set; for MDS, we additionally specify number of clusters in brackets. In the composite query example, red and blue fonts denote its title and narrative, respectively.

Sequence Tagging as Weak Supervision Since our system is fully differentiable, it is possible to optimize latent queries solely based on conditional language modeling. In this work, we additionally propose to exploit sequence tagging as weak supervision. This can be advantageous since it imposes extra regularization via posterior constraints to prevent its collapse, in which case the decoder may learn to ignore the query focused view and instead solely rely on the query agnostic view.

A challenge with applying sequence tagging to summarization is the absence of gold query annotation in training data. Inspired by Gehrmann et al. (2018), we align summary and document by searching for their longest common sub-sequences (LCS). Nevertheless, we note that there exist a few drawbacks hindering its direct application as weak supervision. Primarily, LCS treats a document as a word sequence and annotates it at word-level, while our tagging model (built on top of a pretrained language model) operates on subwords. Apart from this incompatibility of granularity, LCS can lead to false-negative word labels (Type II error), and it is also sensitive to false-positive cases (Type I error) since a summary is seen as a *character* sequence in LCS. We provide examples of these two error types in Table 1. We propose a simple but effective solution to fix the abovementioned issues: we first byte-pair encode (BPE; Sennrich et al. 2016) documents and summaries, and then search LCS over paired document-summary BPE sequences. Compared to the other way around, doing BPE as the first step allows finer-grained unit matching (which reduces Type II error), while still retains basic semantics (which reduces Type I error). We annotate BPEs in the extracted LCS as 1 and the rest as 0. Note that if there exist multiple identical LCS, only the one appearing at the earliest document position is tagged as positive. We refer to this query tagging scheme as BPE-LCS.

Training We use a cross entropy loss for sequence tagging, with a posterior entropy term in Equation (6):

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

where $\hat{\mathbf{z}}_i$ is a binary annotation automatically assigned via BPE-LCS(\mathcal{D}, \mathcal{S}). As we can see, the entropy term smooths the weak annotation $\hat{\mathbf{z}}_i$, with a dynamic smoothing strength dependent on q_{ϕ} . We optimize ω, β on the development set.

We notice that at the initial training phase, the under-optimized tagger produces inaccurate posterior $q_{\phi}(\mathbf{z}_i|\mathbf{x})$, and, consequently, hurts learning an abstractive summarization model which heavily relies on a high-quality query focused view. To tackle this issue, we propose a *posterior dropout* mechanism: with a probability δ , we replace the estimated posterior with the weak supervision $o(\hat{\mathbf{z}}|\mathbf{x})$. We initialize δ to 1.0, i.e., only $o(\hat{\mathbf{z}}|\mathbf{x})$ is used, and the tagger is supervised via Equation (12). We then linearly anneal δ over optimization steps to δ_{end} , so the gradients from the summarization objective (which will be introduced in Section 5) can further optimize the tagger jointly.

Testing To plug-and-play query focus during testing, we model the optional query knowledge one may have access to, Q , with an query belief up-dator $\Delta(\mathbf{z}_i|\mathbf{x}, \tilde{\mathbf{z}})$. Specifically, when no prior query knowledge Q is accessible (i.e., in generic summarization), one may assume *zero* increment for all tokens’ query belief. While in QFS, we have prior knowledge $Q \neq \emptyset$ that some tokens come with high query belief and therefore should be biased over: we set $\Delta(\mathbf{z}_i = 1|\mathbf{x}, \tilde{\mathbf{z}}) = 1.0, \forall w_i \in \text{BPE-LCS}(\mathcal{D}, Q)$, and the rest to zero.

Systems	R-1	R-2	R-L
ORACLE	55.8	33.2	51.8
LEAD	40.4	17.6	36.7
<i>Extractive</i>			
BERTEXT (Liu and Lapata, 2019b)	43.9	20.3	39.9
MATCHSUM (Zhong et al., 2020)	43.9	20.6	39.8
<i>Abstractive</i>			
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., 2020)	45.9	22.3	42.5
GSUM (our implementation)	45.0	21.9	41.8
LAQSUM	45.1	22.0	41.9

Table 3: Supervised performance on **CNN/DM** test set. R-1, R-2 and R-L stand for the F1 score of ROUGE 1, 2, and L, respectively. GSUM (our implementation) uses the same training configurations as our model.

Systems	R-1	R-2	R-L
ORACLE	54.5	37.5	48.5
LEAD	26.3	10.5	21.8
LEXRANK	29.9	12.3	26.1
<i>Supervised (Extractive)</i>			
TRANSFORMER (Zhu et al., 2019)	28.1	12.8	23.8
BERTEXT (Zhu et al., 2019)	36.0	18.8	30.7
<i>Zero-shot Abstractive</i>			
BART (Lewis et al., 2020)	30.0	12.2	26.0
GSUM+QUERY _E	30.2	12.5	26.3
LAQSUM	31.1	12.6	27.1

Table 4: Zero-shot performance on **WikiRef** test set (with keywords as queries). R1, R2 and RL stand for the F1 score of ROUGE 1, 2, and L, respectively.

We further incorporate prior query information via a simple calibration as:

$$q_{\phi}(\mathbf{z}_i = 1|\mathbf{x}, \tilde{\mathbf{z}}) = \max\{1.0, q_{\phi}(\mathbf{z}_i = 1|\mathbf{x}) + \Delta(\mathbf{z}_i = 1|\mathbf{x}, \tilde{\mathbf{z}})\}.$$

Note that our adopt a *non-parametric* query belief calibration, as we do not assume the availability of a development set of each query type for hyperparameter optimization. This enables *zero-shot* transfer to QFS tasks with different settings.

5 Conditional Language Model

In this section we introduce conditional language modeling which models the expectation of the log-likelihood of a summary word sequence over the variational posterior distribution in Equation (6). As shown in Figure 1, we adopt an encoder-decoder architecture tailored for text summarization with latent queries.

Encoder We encode the same input document into two different views, a query-agnostic view,

Systems	R-1	R-2	R-L
LEAD	18.1	5.6	15.9
LEXRANK	17.4	5.3	15.1
<i>Supervised (Abstractive)</i>			
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
<i>Zero-shot Abstractive</i>			
BERTABS [†] (Liu and Lapata, 2019b)	13.3	2.8	2.8
BART (Lewis et al., 2020)	21.4	6.3	18.4
GSUM+QUERY _E	21.2	6.2	18.2
LAQSUM	23.5	7.2	20.6

Table 5: Zero-shot performance on **Debatepedia** test set (with natural questions as queries). R1, R2 and RL stand for the F1 score of ROUGE 1, 2, and L, respectively. † denotes models optimized on XSum (Narayan et al., 2018) and numbers are borrowed from Laskar et al. (2020a).

and a query focused view. Therefore, our encoder module consists of three encoders: a shared encoder, a document encoder, and a query encoder. The intuition is straightforward: since both views are created from the same document, we use a shared encoder for general document understanding which also reduces model parameters. The shared document representation is then input to the other two separate encoders to encode high-level view-specific features. Each encoder contains one or multiple Transformer layers (Vaswani et al., 2017), which is composed of a multi-head attention (MHA) layer and a feed-forward (FFN) layer:

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

where LN denotes layer normalization. As outputs of the encoding module, the query focused view **Q** directly conditions on latent query variables, while the query-agnostic view **D** retains original contexts.

Decoder We adopt a similar decoder structure as in Dou et al. (2020) to handle multiple inputs. Instead of incorporating pre-extracted guidance in Dou et al. (2020), our decoder attends to the two encoded views of the same document sequentially:

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

After taking in the context of the previous generation, the decoder first fuses in query signals from **Q**,

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	45.7	11.2	17.0	47.9	14.1	19.1	—	—	—
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
LEXRANK	34.2	6.4	11.4	35.8	7.7	12.7	35.3	7.6	12.2
<i>Distantly Supervised</i>									
QUERYSUM* (Xu and Lapata, 2020b)	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	—	—	—
<i>Few- or Zero-shot Abstractive</i>									
MARGESUM [†] (Xu and Lapata, 2020a)	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+QUERY _E	38.1	7.9	13.1	39.5	9.5	14.3	45.5	18.0	22.4
LAQSUM	39.1	8.5	13.7	40.4	10.2	15.0	45.7	18.1	22.1

Table 6: Zero-shot performance on multi-document QFS test sets **DUC** (with composed queries) and **TD-QFS** (with titles as queries). */†: extractive/few-shot system. R1, R2 and R-SU4 stand for the F1 score of ROUGE 1, 2, and SU4, respectively.

which then drives the incorporation of original document context **D**. The final summary generation objective is calculated auto-regressively as:

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

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 We used CNN/DM (Hermann et al., 2015), a generic single-document summarization dataset containing news articles and associated highlights, for model training and development (with 287,227/13,368 instances). For model evaluation, we evaluated our system on CNN/DM test set (11,490 instances) under a supervised setting. We also performed experiments on QFS under a *zero-shot* transfer setting, on five test sets with various formats of queries, domains, and document settings, including WikiRef (Zhu et al., 2019), Debatepedia (Nema et al., 2017), DUC 2006-07, and TD-QFS (Baumel et al., 2016). Statistics for all test sets are given in Table 2. Note that we do not assume any development data in QFS, which departs from Xu and Lapata (2020a).

Implementation Details The shared encoder consists of 11 Transformer layers. The document and query encoder has one separate Transformer layer each. The shared encoder, document encoder, and decoder are initialized with a pretrained BART model (Lewis et al., 2020), while the query encoder is randomly initialized. We used 4 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. Following the standard configurations for BART finetuning on CNN/DM, we used a learning rate of 3×10^{-5} for 20,000 optimization steps, with a warmup-step of 500. Due to memory constraints, we used half float precision for efficient training and also 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. We used $\tau = 0.9$ for latent query modeling. For the proposed posterior dropout, we annealed the dropout rate δ from 1.0 to 0.5 over the whole training session.

7 Results

Generic Summarization Table 3 summarizes our results on CNN/DM. The first block in the table includes an ORACLE extractive system as an upper bound. LEAD baseline take the first 3 sentences in a document as a summary.

The second block presents two extractive systems. BERTEXT (Liu and Lapata, 2019b) is the first system using as a pretrained encoder (Devlin et al., 2019) for text summarization. MATCHSUM is a state-of-the-art extractive system extracting an optimal set of sentences via text matching.

The third block includes various abstractive systems (see Section 2 for an overview). Particularly, PTGEN (See et al., 2017) and BOTTOMUP (Gehrmann et al., 2018) do not use pretrained LMs, while BERTABS is built on a pretrained BERT encoder, and GSUM (Dou et al., 2020), similar to our work, is initialized with pretrained BART parameters (Lewis et al., 2020). Our system outperforms standard the BART finetuned on CNN/DM by a

Models	CNN/DM	WikiRef	Debatepedia	DUC 2006	DUC 2007	TD-QFS
LAQSUM	41.9	27.1	20.6	13.7	15.0	22.1
- $\Delta(\hat{\mathbf{z}} \mathbf{x}, \mathbf{z})$	—	↓0.2	↓0.6	↓0.6	↓1.3	↓0.4
-Joint training	↓0.4	↓2.8	↓2.8	↓1.6	↓1.7	↓0.4
-Weak supervision	↓0.7	↓0.5	↓1.3	↓0.2	↓0.3	↓0.0
-Dual view	↓2.5	↓10.5	↓6.6	↓1.8	↓2.5	↓2.8
-Posterior dropout	↓0.8	↓0.7	↓1.2	↓0.2	↓0.5	↑0.1

Table 7: Ablation results for our abstractive summarization system. \uparrow / \downarrow : absolute performance increase/decrease in ROUGE-L (on CNN/DM, WikiRef and Debatepedia) or ROUGE-SU4 (on DUC 2006-07 and TD-QFS).

fairly large margin, which demonstrates the effectiveness of modeling the query focused view with latent queries even for generic summarization. Under the same training resources and configurations, it also performs on par with GSUM, a state-of-the-art abstractive model, despite being significantly more computationally efficient, as no access to high-quality guidance sentences (which are produced by another well-trained extractive system, e.g., MATCHSUM) is required.

Single-Document QFS We show results on WikiRef and Debatepedia in Table 4 and Table 5, respectively, for single-document QFS evaluation.

In the first block of these two tables, we show two *unsupervised* extractive baselines: LEAD and LEXRANK which estimates sentence-level centrality via Markov Random Walk on graphs. The second block presents various *supervised* systems on WikiRef and Debatepedia. Note that no abstractive QFS system has been evaluated on WikiRef, while Debatepedia is a short-document, short-summary dataset mainly for abstractive summarization.

The third block of the two tables highlights system performance in the *zero-shot* transfer setting, including BART and GSUM. Particularly, GSUM requires guidance from a *generic* extractive summarization system, which is hard to obtain due to the data scarcity in QFS. Also, it is not straightforward how a given query can be incorporated to generate query-specific summaries from GSUM. To adapt it to QFS test settings, we build GSUM+QUERY_E, where we employ an unsupervised query focused extractive system to pre-extract the top K ranked sentences for each testing document as its guidance. Specifically, we choose a query focused version of LEXRANK described in Xu and Lapata (2020b), which is well-performed on extractive QFS tasks by jointly estimating sentence centrality and query relevance (Wan, 2008).

In an end-to-end fashion, our system achieves the highest ROUGE scores on both datasets in the zero-shot transfer setting. Compared to the results

on generic data, our system shows a clearer edge over systems without latent query modeling.

Multi-Document QFS To apply a summarization system trained on *single*-document data to a *multi*-document setting, we adopt a simple iterative generation approach (Baumel et al., 2018): we first rank documents in a cluster via query term frequency, and then generate summaries iteratively for each document. The final summary for the whole cluster is composed by concatenating document-level summaries.³ Repetitive generated sentences are skipped to remove redundancy.

Table 6 presents results on multi-document QFS datasets. The first block reports performance of two upper-bound systems, GOLD and ORACLE, and two unsupervised systems taken from Xu and Lapata (2020a). The second block contains previous *distantly supervised* approaches. QUERYSUM (Xu and Lapata, 2020b) is state-of-the-art extractive system which adopts QA datasets for a coarse-to-fine salience estimation process. On the abstractive side, BART-CAQ (Su et al., 2020) uses an ensembled QA model for answer evidence extraction, and then use finetuned BART (Lewis et al., 2020) to iteratively generate summaries from paragraphs. PQSUM (Laskar et al., 2020b) uses finetuned 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 with a start-of-the-art *few-shot* approach, MARGESUM (Xu and Lapata, 2020a) which requires a small QFS development set, and *zero-shot* systems including BART and GSUM+QUERY_E. As we can see, without recourse to expensive QA/QFS annotations, our system, achieves significantly better results than BART-CAQ which exploits QA data as ex-

³An alternative is to generate a long summary at once. However, this requires a model to be trained on a MDS dataset, or at least a proxy one (Xu and Lapata, 2020a). Since we build our system also for single-document summarization, we choose to generate and then compose.

ternal training resources, on DUC test sets (except in ROUGE-1 on DUC 2007); on TD-QFS, it surpasses MARGESUM which uses QFS data for proxy query generation and model development, across all metrics. Also, our system outperforms strong zero-shot abstractive systems including BART and GSUM+QUERY_E on all three datasets.

Ablation Studies We provide the results of ablation studies on LAQSUM in Table 7. Removing query belief update at test time ($-\Delta(\hat{\mathbf{z}}|\mathbf{x}, \mathbf{z})$) hurts model performance on QFS test sets, demonstrating the usefulness of incorporating query information via simple calibration on the variational posterior distribution. When it comes to learning meaningful latent queries that benefit summarization tasks, relying on only tagging (-Joint training, where we adopt argmax to stop gradients from the generation loss), or generation (-Weak supervision, where we set $\omega = 0$) significantly decreases performance. We conclude that latent query learning performs a trade-off between exploiting *direct but weak* supervision from the tagging objective (i.e., based on synthetic token annotation), and exploring the *natural but indirect* supervision from the generation objective (i.e., based on human-written summaries). Removing the query agnostic view (-Dual view) causes significant performance drop as it keeps the original document context that the decoder can possibly leverage, especially when the query model is not well-performed. This is also supported by solely using the *estimated* posterior to create query focused view for training (-Posterior dropout), which also hurts model performance as it leads to more severe error propagation to the downstream generation model.

8 Conclusion

In this work we provide a deep generative formulation for text summarization, and present a general text summarization system that supports generating both generic and query focused abstracts. Under this formulation, queries are represented as discrete latent variables, whose approximated posterior distribution can be, optionally, calibrated with additional query observations during testing without further adaptation. As a result, our system does not rely on any query-related resource. Experimental results across datasets of various characteristics show that the proposed system yields strong performance on generic summarization, and state-of-the-art performance on zero-shot abstractive QFS.

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 Venkatasubramanian, 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.
- Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 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.
- Souradip Chakraborty, Ekaba Bisong, Shweta Bhatt, Thomas Wagner, Riley Elliott, and Francesco Mosconi. 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.
- 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.
- 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.

- William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing*, pages 9–16, Jeju Island, Korea.
- Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2020. GSum: A general framework for guided neural abstractive summarization. *arXiv preprint arXiv:2010.08014*.
- 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.
- 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*, page 1693–1701, Cambridge, MA, USA.
- I. Higgins, Loïc Matthey, A. Pal, Christopher P. Burgess, Xavier Glorot, M. Botvinick, S. Mohamed, and Alexander Lerchner. 2017. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*.
- TD 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*.
- HW Kuhn, AW Tucker, et al. 1951. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*.
- Tom Kwiakowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.
- 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.
- 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.
- 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.
- 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. Saliency 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.
- 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.
- 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*, Hong Kong, China.
- 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.
- 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.
- 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.

- Laura Perez-Beltrachini, Yang Liu, and Mirella Lapata. 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.
- 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.
- 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.
- 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, Hyeondev 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.
- 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.
- 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. 2008. Using only cross-document relationships for both generic and topic-focused multi-document summarizations. *Information Retrieval*, 11(1):25–49.
- 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.
- Yumo Xu and Mirella Lapata. 2020a. Abstractive query focused summarization with query-free resources. *arXiv preprint arXiv:2012.14774*.
- Yumo Xu and Mirella Lapata. 2020b. 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.
- 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.
- Chenguang Zhu, William Hinthorn, Ruochen Xu, Qingkai Zeng, Michael Zeng, Xuedong Huang, and Meng Jiang. 2021. Enhancing factual consistency of abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 718–733, Online.
- 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*.