Coarse-to-Fine Query Focused Multi-Document Summarization

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

1 Introduction
Query Focused Multi-Document Summarization (QFS; Dang 2006) aims to create a short summary from a set of documents that answers a specific query. It has various applications in personalized information retrieval and recommendation engines where search results can be tailored to an information need (e.g., a user might be looking for an overview summary or a more detailed one which would allow them to answer a specific question).

Neural approaches have become increasingly popular in single-document text summarization (Nallapati et al., 2016; Paulus et al., 2018; Li et al., 2017b; See et al., 2017; Narayan et al., 2018; Gehrmann et al., 2018), thanks to the representational power afforded by deeper architectures and the availability of large-scale datasets containing hundreds of thousands of document-summary pairs (Sandhaus, 2008; Hermann et al., 2015; Grusky et al., 2018). Unfortunately, such datasets do not exist in QFS, and one might argue it is unrealistic they will ever be created for millions of queries, across different domains, and languages. In addition to the difficulties in obtaining training data, another obstacle to the application of end-to-end neural models is the size and number of source documents which can be very large. It is practically unfeasible (given memory limitations of current hardware) to train a model which encodes all of them into vectors and subsequently generates a summary from them.

In this paper we propose a coarse-to-fine modeling framework for extractive QFS which incorporates a relevance estimator for retrieving textual segments (e.g., sentences or longer passages) associated with a query, an evidence estimator which further isolates segments likely to contain answers to the query, and a centrality estimator which finally selects which segments to include in the summary. The vast majority of previous work (Wan et al., 2007; Wan, 2008; Wan and Xiao, 2009; Wan and Zhang, 2014) creates summaries by ranking textual segments (usually sentences) according to their relationship (e.g., similarity) to other segments and their relevance to the query. In other words, relevance and evidence estimation are servant to estimating the centrality of a segment (e.g., with a graph-based model). We argue that disentangling these subtasks allows us to better model the query and specialize the summaries to specific questions or topics (Katragadda and Varma, 2009).

A coarse-to-fine approach is also expedient from a computational perspective; at each step the model processes a decreasing number of segments (rather than entire documents), and as a result is insensitive to the original input size and more scalable.

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

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

2 Related Work

Existing research on query-focused multi-document summarization largely lies on extractive approaches, where systems usually take as input a set of documents and select the sentences most relevant to the query for inclusion in the summary.

In Figure 1(a), we provide a sketch of classic centrality-based approaches which have generally shown strong performance in QFS. Under this framework, all sentences within a document cluster, together with their query relevance, are jointly considered in estimating centrality. A variety of approaches have been proposed to enhance the way relevance and centrality are estimated ranging from incorporating topic-sensitive information (Wan, 2008; Badrinath et al., 2011; Xu and Lapata, 2019), predictions about information certainty (Wan and Zhang, 2014), manifold-ranking algorithms (Wan et al., 2007; Wan and Xiao, 2009; Wan, 2009), and Wikipedia-based query expansion (Nastase, 2008). More recently, Li et al. (2015) estimate the salience of text units within a sparse-coding framework by additionally taking into account reader comments (associated with news reports). Li et al. (2017a) use a cascaded neural attention model to find salient sentences, whereas in follow-on work Li et al. (2017b) employ a generative model which maps sentences to a latent semantic space while a reconstruction model estimates sentence salience. There are also feature-based approaches achieving good results by optimizing sentence selection under a summary length constraint (Feigenblat et al., 2017).

In contrast to previous work, our proposal does not simultaneously perform segment selection and query matching. We introduce a coarse-to-fine approach that incorporates progressively more accurate components for selecting segments to include in the summary, making model performance relatively insensitive to the number and size of input documents. Drawing inspiration from recent work on QA, we take advantage of existing datasets in order to reliably estimate the relationship between the query and candidate segments. We focus on two QA subtasks which have attracted considerable attention in the literature, namely answer sentence selection which aims to extract answers from a set of pre-selected sentences (Heilman and Smith, 2010; Yao et al., 2013; Yang et al., 2015) and machine reading comprehension (Rajpurkar et al., 2016; Welbl et al., 2018; Yang et al., 2018), which aims at answering a question after processing a short text passage (Chen, 2018).

QA and QFS are related but ultimately different tasks. QA aims at finding the best answer in a span or sentence, while QFS extracts a set of sentences based on user preferences and the content of the input documents under a length budget (Wan, 2008; Wan and Zhang, 2014). QA questions are often short and fact-based while QFS narratives can be longer and more complex (see the example in Section 3) and as a result simply localizing an answer within a cluster is not optimal.
Figure 1: Classic (a) and proposed framework (b) for query-focused summarization. The classic approach involves a relevance estimator nested within a summarization module while our framework takes document clusters as input, and sequentially processes them with three individual modules (relevance, evidence, and centrality estimators). The blue circles indicate a coarse-to-fine estimation process from original articles to final summaries where modules gradually discard segments (i.e., sentences or passages). With regard to evidence estimation, we adopt pretrained BERT (Devlin et al., 2019) which is further fine-tuned with distant signals from question answering.

3 Problem Formulation

Let $Q$ denote an information request and $D = \{d_1, d_2, \ldots, d_M\}$ a set of topic-related documents. It is often assumed (e.g., in DUC competitions) that $Q$ consists of a short title (e.g., Amnesty International) highlighting the topic of interest, and a query narrative which is considerably longer and detailed (e.g., What is the scope of operations of Amnesty International and what are the international reactions to its activities?).

We illustrate our proposed framework in Figure 1(b). We first decompose documents into segments, i.e., passages or sentences, and retrieve those which are most relevant to query $Q$ (Relevance Estimator). Then, a trained estimator quantifies the semantic match between selected segments and the query (Evidence Estimator) to further isolate segments for consideration in the output summary (Centrality Estimator). We propose two variants of our evidence estimator; a context agnostic variant infers evidence scores over individual sentences, while a context aware one infers evidence scores for tokens within a passage which are further aggregated into sentence-level evidence. Passages might allow for semantic relations to be estimated more reliably since neighboring context is also taken into account.

3.1 Relevance Estimator

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

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

$$k_{IR}^i = \max_k \sum_{j=1}^{k} r_{i,j} < \theta$$

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

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

Sentence Selection Let \(Q\) denote a query (in practice a sequence of tokens) and \(\{S_1, S_2, \ldots, S_N\}\) the set of candidate answers (also token sequences) obtained from the retrieval module. Our learning objective is to find the correct answer(s) within this set. We concatenate query \(Q\) and candidate sentence \(S\) into a sequence \([CLS], Q, [SEP], S, [SEP]\) to serve as input to a BERT encoder (we pad each sequence in a minibatch of \(L\) tokens). The \([CLS]\) vector serves as input to a single layer neural network to obtain the distribution over positive and negative classes:

\[
p_0(i) = \frac{1}{Z} \exp(t_i^TW_{:,0}), p_1(i) = \frac{1}{Z} \exp(t_i^TW_{:,1}) \tag{2}
\]

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

\[
\mathcal{L} = - \sum_{i=1}^N (y \log p_1(i) + (1 - y) \log p_0(i)). \tag{3}
\]

We treat the probability of the positive class as evidence score \(q = p_1(i) \in (0, 1)\) and use it to rank all retrieved segments for each query.

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

\[
p_S(i) = \frac{\exp(t_i^Tw_S)}{\sum_j \exp(t_j^Tw_S)} \tag{4}
\]

\[
p_E(i) = \frac{\exp(t_i^Tw_E)}{\sum_j \exp(t_j^Tw_E)} \tag{5}
\]

where \(w_S\) and \(w_E\) are two learnable vectors denoting the beginning and end of the (answer) span, respectively. During training we optimize the log-likelihood of the correct start and end positions. For passages without any correct answers, we set these to 0 and default to the \([CLS]\) position.

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

\[
u = \exp(Tw_S), v = \exp(Tw_E). \tag{6}
\]

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

\[
\tilde{Q} = \left(u_{[i,j]}v_{[i,j]}^T A \right)^{\frac{1}{2}} \tag{7}
\]

\[
Q = \tanh(\tilde{Q}) \tag{8}
\]

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

\[
q = \text{max-pool}(Q) \in (0, 1). \tag{9}
\]

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

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

\[
q_e = \begin{cases} 
\mu * q^S_e + (1 - \mu) * q^P_e & e \in \mathcal{E}^S \cap \mathcal{E}^P \\
\mu * q^S_e & e \in \mathcal{E}^S \land e \notin \mathcal{E}^P \\
-\infty & e \notin \mathcal{E}^S
\end{cases}
\]

(10)

where the coefficient was set to \( \mu = 0.9 \).

3.3 Centrality Estimator

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

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

\[
\text{ISF}(w) = 1 + \log(|C|/\text{SF}(w))
\]

(11)

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

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

\[
\tilde{E} = \phi \ast [\tilde{q}; \ldots; \tilde{q}] + (1 - \phi) \ast E
\]

(12)

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

**Summary Generation** In order to decide which sentences to include in the summary, a node’s centrality is measured using a graph-based ranking algorithm (Erkan and Radev, 2004; Xu and Lapata, 2019). Specifically, we run a Markov chain with \( \tilde{E} \) on \( G \) until it converges to stationary distribution \( e^* \) where each element denotes the salience of a sentence. In the proposed algorithm, \( e^* \) jointly expresses the importance of a sentence in the document and its semantic relation to the query as modulated by the evidence estimator and controlled by \( \phi \). We rank sentences according to \( e^* \) and select the top \( \ell_{\text{Sum}} \) ones, subject to a budget (e.g., 250 words). To reduce redundancy, we apply the diversity algorithm proposed in Wan (2008) which penalizes the salience of sentences according to their overlap with those already selected to appear in the summary. We also remove the sentences which have high cosine similarities (i.e., \( \geq 0.6 \)) with any sentence already included in the summary (Cao et al., 2015; Angelidis and Lapata, 2018).

4 Experimental Setup

**Datasets** We performed QFS experiments on the DUC 2005-2007 benchmarks and the Topically Diverse QFS dataset (TD-QFS; Baumel et al. 2016). DUC benchmarks contain long query narratives over 50 clusters with 32–25 documents each, and cover multiple domains. TD-QFS focuses on medical texts, contains short keyword queries over 4 clusters with 185 documents each. As a result, TD-QFS clusters are less topically concentrated, with larger amounts of query-irrelevant information (Baumel et al., 2016). Although our approach is motivated by the desire to better model long and complex queries, experiments on TD-QFS examine whether it generalizes to out-of-domain queries and clusters. We used DUC 2005 as a development set to optimize hyperparameters and evaluated performance on DUC 2006-2007 and TD-QFS. A summary of the characteristics of these datasets

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

Table 1: QFS dataset statistics.

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

Table 2: Question answering dataset statistics. We use the union of WikiQA and TrecQA for answer sentence selection and SQuAD for span selection.
Table 3: Examples for two types of question answering datasets for evidence estimation: answer sentence selection and span selection. Red denotes answers while blue denotes a plausible answer to the question that cannot be answered from the given context. We use the union of WikiQA (Yang et al., 2015) and TrecQA (Heilman and Smith, 2010) for answer sentence selection and SQuAD 2.0 (Rajpurkar et al., 2016) for span selection. SQuAD 2.0 contains both answerable and unanswerable questions and we show one example for each of them.

| Question | What bird family is the owl? |
|------------------|-----------------------------|
| Candidate Sentences | Owls are a group of birds that belong to the order Strigiformes, constituting 200 extant bird of prey species. Most are solitary and nocturnal, with some exceptions (e.g., the northern hawk owl). Owls hunt mostly small mammals, insects, and other birds, although a few species specialize in hunting fish. They are found in all regions of the earth except Antarctica, most of Greenland and some remote islands. Owls are characterized by their small beaks and wide faces, and are divided into two families: the typical owls, Strigidae; and the barn-owls, Tytonidae. |

**Span Selection (answerable)**

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

**Span Selection (unanswerable)**

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

is provided in Table 1.

We used three datasets for training our evidence estimator, including WikiQA (Yang et al., 2015), TrecQA (Yao et al., 2013), and SQuAD 2.0 (Rajpurkar et al., 2018). WikiQA and TrecQA are benchmarks for answer sentence selection while SQuAD 2.0 is a popular machine reading comprehension dataset (which we used for span selection). Compared to SQuAD, WikiQA and TrecQA are smaller and we therefore integrate them for model training (Yang et al., 2019). We show statistics for QA datasets in Table 2 and examples in Table 3.

**Implementation Details** We used the publicly released BERT model\(^2\) and fine-tuned it on our QA tasks. Considering the maximum input length BERT allows (512 tokens) and the query narrative (which in DUC is fairly long), we set the maximum passage size to 8 sentences (with maximum sentence length of 50 tokens). To ensure all sentences are properly contextualized, we used a stride size of 4 sentences to create overlapping passages. Details on model training and optimization are provided in Appendix A.

**Evaluation** Following standard practice in DUC evaluations, we used ROUGE as our automatic evaluation metric\(^3\) (Lin and Hovy, 2003) We report F1 for ROUGE-1 (unigram-based), ROUGE-2 (bigram-based), and ROUGE-SU4 (based on skip bigram with a maximum skip distance of 4).

We also evaluated model summaries in a judgment elicitation study via Amazon Mechanical Turk. Native English speakers (self-reported) were asked to rate query-summary pairs on two dimensions: **Succinctness** (does the summary avoid unnecessary detail and redundant information?) and **Coherence** (does the summary make logical sense?). The ratings were obtained using a five point Likert scale. In addition, participants were 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., whether it provides an answer to the query), irrelevant (i.e., it does not answer the query), and partially relevant (i.e., it is not clear it directly answers the query). Relevant sentences were awarded 3

\(^2\)https://github.com/huggingface/pytorch-transformers

\(^3\)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.
Table 4: System performance on DUC 2006 and 2007. R-1, R-2 and R-SU4 stand for the F1 score of ROUGE 1, 2, and SU4, respectively. Results with \* were obtained based on our own implementation.

Table 5: System performance on TD-QFS. R-1, R-2 and R-SU4 stand for the F1 score of ROUGE 1, 2, and SU4, respectively.

\*Similar to our experimental setting, its hyperparameters are optimized on a development set. For fair comparison, we leave aside a few symbolic approaches that take advantage of query expansion techniques, and task-specific predictors such as position bias.

For the DUC benchmarks, participants assessed summaries created by superior results across all ROUGE metrics.

The last block in Table 4 presents different variants of our query-focused summarizer which we call QUERYSUM. We show automatic results with distant supervision based on isolated sentences (QUERYSUM$_S$), passages (QUERYSUM$_P$), and an ensemble model (QUERYSUM$_{S+P}$) which combines both. As can be seen, our models outperform strong comparison systems on both DUC test sets: QUERYSUM$_S$ achieves the best R-1 while QUERYSUM$_P$ achieves the best R-2 and R-SU4. Perhaps unsurprisingly, both models fall behind the human upper bound.

Our results on the TD-QFS dataset are summarized in Table 5. In addition to LEAD and LEXRANK, we compared to KLSUM, the best performing system on this dataset (Baumel et al., 2016). KLSUM selects a subset of sentences from retrieved candidates by minimizing the Kullback-Leibler Divergence between the unigram distribution in the selected sentences and the source cluster. QUERYSUM$_S$ and our ensemble model achieve superior results across all ROUGE metrics.

**Human Evaluation** For the DUC benchmarks, participants assessed summaries created by
distant supervision based on \textit{VA}riational auto\textit{E}ncoders (Kingma and Welling, 2013; Rezende et al., 2014) and a data reconstruction model for sentence salience estimation. VAESUM represents the state-of-the-art amongst neural systems on DUC.\* The salience estimation module is further integrated in an integer linear program which selects VPs and NPs to create the final summary.
VAES\textsuperscript{5}, a neural state-of-the-art system, QUERYSUM, and the LEAD baseline. For TD-QFS, we evaluated summaries created by KLSUM, QUERYSUM, and LEAD. We also included a randomly selected GOLD standard summary as an upper bound. We sampled 20 query-cluster pairs from DUC (2006, 2007; 10 from each set), and 20 pairs from TD-QFS (5 from each cluster). We collected three responses per query-summary pair.

Table 6 shows the ratings for each system. As can be seen, participants find QUERYSUM summaries on DUC more relevant and with less redundant information compared to LEAD and VAESUM. Our multi-step estimation process also produces more coherent summaries (as coherent as LEAD) even though coherence is not explicitly modeled. Overall, participants perceive QUERYSUM summaries as significantly better ($p < 0.01$) compared to LEAD and VAESUM (see Appendix B for examples of system output). QUERYSUM is also considered as the best performing system across metrics on TD-QFS. This further demonstrates the robustness of our system on unseen domains and query types.

**Ablation Studies** We also conducted ablation experiments to verify the effectiveness of the proposed coarse-to-fine framework. We present results in Table 7 when individual modules are removed. In the −Relevance setting, all text segments (i.e., sentences or passages) in a cluster are given as input to the evidence estimator module. The −Evidence setting treats all retrieved segments as evidence for summarization. Note that since our summarizer operates on sentences, we can only assess this configuration with the QUERYSUM\textsubscript{S} model; we take the top $k_{QA}$ sentences from the retrieval module as evidence. The −Centrality setting treats the (ranked) output of the evidence estimator as the final summary. For the sake of brevity, we report results on DUC-2007 and TD-QFS (DUC-2006 follows a very similar pattern).

As can be seen, removing the retrieval module leads to a large drop in the performance of QUERYSUM\textsubscript{S}. This indicates that the (deep) semantic matching model trained for sentence selection can get distracted by noise which a (shallow) relevance matching model can help pre-filter. Interestingly, on DUC, when the matching model is trained on passages, the retrieval module seems more or less redundant, there is in fact a slight improvement in R-2 and R-SU4 (see row QUERYSUM\textsubscript{F}, −Relevance in Table 7). This suggests that the evidence estimator trained on passages is more robust and captures the semantics of the query more faithfully. Moreover, since it takes contextual signals into account, it is able to recognize irrelevant information and unanswerability is explicitly modeled. We show in Figure 2 how ROUGE-2 varies over $k_{IR}$ best retrieved segments. We compare three different types of query settings, the short *title*, the *narrative*, and the full query with both the title and the narrative. As expected, recall increases with $k_{IR}$ (i.e., when more evidence is selected) and then finally converges. For both sentence and passage retrieval settings, the full query achieves best performance over $k_{IR}$, with the narrative being most informative when it comes to relevance estimation.

Performance also drops in Table 7 when the evidence estimator is removed (see QUERYSUM\textsubscript{S}, −Evidence in Table 7). In Figure 3, we plot how ROUGE-2 varies with increasing $k_{QA}$ when the evidence component is estimated on passages and sentences for the full model. As can be seen, the model trained on passages surpasses the model summary performance in Table 7). This suggests that the evidence estimator is removed (see Q...
trained on sentences roughly when $k^{QA} = 80$. For comparison, we also show the performance of the retrieval module by treating the top sentences as evidence. The retrieval curve is consistently under the passage curve, and under the sentence curve when $k^{QA} < 140$. Since the quality of top sentences directly affects the quality of the summarization module, this further demonstrates the effectiveness of evidence estimation in terms of reranking retrieved segments.

Finally, Table 7 shows that the removal of the centrality estimator decreases performance even when the query and appropriate evidence are taken into account. This suggests that the centrality estimator further learns to select important summary worthy sentences from the available evidence. Interestingly, the gain on the DUC datasets is slight but considerable on TD-QFS, suggesting that in less topically concentrated clusters where multiple high-quality answers can be available, the soft discrimination between answer candidates based on their answerability can be useful during the final summary sentence selection.

### 6 Conclusions

In this work, we proposed a coarse-to-fine estimation framework for query focused multi-document summarization. We explored the potential of leveraging distant supervision signals from Question Answering to better capture the semantic relations between queries and document segments. Experimental results across datasets show that the proposed model yields results superior to competitive baselines contributing to summaries which are more relevant and less redundant. We have also shown that disentangling the tasks of relevance, evidence, and centrality estimation is beneficial allowing us to progressively specialize the summaries to the semantics of the query. In the future, we would like to generate abstractive summaries following an unsupervised approach (Baziotis et al., 2019; Chu and Liu, 2019) and investigate how recent advances in open domain QA (Wang et al., 2019; Qi et al., 2019) can be adapted for query focused summarization.

### Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable feedback. We acknowledge the financial support of the European Research Council (Lapata; award number 681760). This research is based upon work 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.

### References

Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Cross-domain modeling of sentence-level evidence for document retrieval. In Proceedings of the 2019 Conference on
Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming T. Baumel, Raphael Cohen, and Michael Elhadad. 2006. DUC 2005: Evaluation of retrieval and novelty detection at the sentence level. In Proceedings of the 26th Annual ACM SIGIR International Conference on Research and Development in Information Retrieval, pages 314–321, Toronto, Canada.

Stefanos Angelidis and Mirella Lapata. 2018. Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3675–3686, Brussels, Belgium.

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.

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.

Christos Baziotis, Ion Androutsopoulos, Ioannis Konotas, and Alexandros Potamianos. 2019. SEQ3: Differentiable sequence-to-sequence autoencoder for unsupervised abstractive sentence compression. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 673–681, Minneapolis, Minnesota.

Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming Zhou. 2015. Ranking with recursive neural networks and its application to multi-document summarization. In Proceedings of the 29th AAAI Conference on Artificial Intelligence, pages 2153–2159, Austin, Texas, USA.

Danqi Chen. 2018. Neural Reading Comprehension and Beyond. Ph.D. thesis, Stanford University.

Eric Chu and Peter Liu. 2019. MeanSum: A neural model for unsupervised multi-document abstractive summarization. In Proceedings of the 36th International Conference on Machine Learning, pages 1223–1232, Long Beach, California.

Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for IR with contextual neural language modeling. arXiv preprint arXiv:1905.09217.

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.

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.

Guy Feigenblat, Hagai Roitman, Odelia Boni, and David Konopnicki. 2017. Unsupervised query-focused multi-document summarization using the cross entropy method. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 961–964, Tokyo, Japan.

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.

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

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pages 55–64, Indianapolis, Indiana.

Michael Heilman and Noah A Smith. 2010. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In Proceedings of the 2010 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1011–1019, Los Angeles, California.

Karl Moritz Hermann, Tomáš Kočíský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In Advances in Neural Information Processing Systems 28, pages 1693–1701. Morgan, Kaufmann.

Mandal Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pages 1601–1611, Vancouver, Canada.
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, volume 7, 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 and Development in Information Retrieval, pages 787–796. ACM.

Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Multi-passage bert: A globally normalized BERT model for open-domain question answering. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 5881–5885, Hong Kong, China.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. Transactions of the Association for Computational Linguistics, 6:287–302.

Yumo Xu and Mirella Lapata. 2019. Weakly supervised domain detection. Transactions of the Association for Computational Linguistics, 7:581–596.

Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Simple applications of BERT for ad hoc document retrieval. arXiv preprint arXiv:1903.10972.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 2013–2018, Lisbon, Portugal.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2369–2380, Brussels, Belgium.

Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013. Answer extraction as sequence tagging with tree edit distance. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 858–867, Atlanta, Georgia.

A Implementation Details

We used the publicly released BERT model and fine-tuned it on our QA tasks with 4 GTX 1080TI GPUs with 11GB memory. For the answer sentence selection model, BERT was fine-tuned with a learning rate of $3 \times 10^{-6}$ and a batch size of 16 for 3 epochs (Yang et al., 2019). For span selection, we adopted a learning rate of $3 \times 10^{-5}$ and a batch size of 64 for 5 epochs. During inference, the confidence threshold for the relevance estimator was set to $\theta = 0.75$ (Kratzwald and Feuerriegel, 2018) for both sentence and passage retrieval. For the evidence estimator, $k^{QA}$ was tuned on the development set. We obtained 90 and 110 evidence sentences from the sentence selection and span selection models, respectively. For the centrality estimator, the influence of the query was set to $\phi = 0.15$ (Wan, 2008; Wan and Zhang, 2014).

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

B Summary Outputs

We show in Table 8 and Table 9 system outputs for one cluster in DUC 2006 and 2007, respectively.
Table 8: System outputs for cluster D0621C in DUC 2006. The gold summary answers the query covering four main aspects (denoted with different colors): (1) general facts and vision; (2) criminal activities in southeastern China, including Hong Kong and Macau; (3) international corporations; (4) law revision and enforcement. Our system produces more diverse content that represents these aspects compared to other systems.
Table 9: System outputs for cluster D0701A in DUC 2007. The gold summary answers the query covering three main aspects (denoted with different colors): (1) Southern Poverty Law Center and its activities; (2) Morris Dees and his activities; (3) representative successful lawsuits. For this document cluster, summarization systems are prone to extract unnecessary lawsuit details, which indirectly relate to the given query but are not the query focus. Our system contains more summary-worthy facts that succinctly respond to the given query compared to other systems.