Summarizing News Articles using Question-and-Answer Pairs via Learning

Xuezhi Wang, Cong Yu

Google Research, New York
{xuezhiw,congyu}@google.com

Abstract. The launch of the new Google News in 2018[1] introduced the *Frequently asked questions* feature to *structurally summarize* the news story in its full coverage page. While news summarization has been a research topic for decades, this new feature is poised to usher in a new line of news summarization techniques. There are two fundamental approaches: *mining the questions* from data associated with the news story and *learning the questions* from the content of the story directly. This paper provides the first study, to the best of our knowledge, of a *learning* based approach to generate a structured summary of news articles with question and answer pairs to capture salient and interesting aspects of the news story. Specifically, this learning-based approach reads a news article, predicts its attention map (i.e., important snippets in the article), and generates multiple natural language questions corresponding to each snippet. Furthermore, we describe a mining-based approach as the mechanism to generate weak supervision data for training the learning based approach. We evaluate our approach on the existing SQuAD dataset[2] and a large dataset with 91K news articles we constructed. We show that our proposed system can achieve an AUC of 0.734 for document attention map prediction, a BLEU-4 score of 12.46 for natural question generation and a BLEU-4 score of 24.4 for question summarization, beating state-of-art baselines.

Keywords: Structured summarization · Question answering

1 Introduction

News summarization has been an important topic of natural language research for decades [20]. While there are many approaches, the end result has always been natural sentences that summarize the articles. The launch of the new Google News in 2018 [28] with its *Frequently asked questions* feature showed that *structured summaries* such as question-and-answer (Q/A) pairs can be beneficial to the news consumption experience[3]. Compared with natural language summaries,

[1] https://www.blog.google/products/news/new-google-news-ai-meets-human-intelligence/

[2] https://rajpurkar.github.io/SQuAD-explorer/

[3] Private communication with Google's news team: FAQ is shown to improve users' understanding of the news stories in user studies, which is an important launch criteria.

Q/A pairs offer low cognitive overload because, being very short, questions are easy to digest and users can easily skip those they do not care and read the answer snippets for only those they are interested in. Furthermore, structured summary often does not try to capture an overview of the story, but rather highlights salient aspects that the users would like to know about, making them complementary to the conventional news summaries.

Question answering has been an important research topic for semantic web [11] and information retrieval [15], with the goal of answering users' questions based on the knowledge base or the documents in the corpus. Lately, major search engines have begun to leverage Q/A in more proactive ways. For example, Google search has been using an Q/A feature, *People also ask*, to proactively highlight the most salient aspects of the search results for the users. The success of Q/A features in search no doubt has played a role in the introduction of Q/A features into the various news consumption platforms such as Google News.

For a news article that has been published for a little while and queried by lots of users, many questions would have been asked about it. Thus, the intuitive first approach for generating Q/A pairs for a news article is to mine the query log for questions, cluster them into groups, identify a representative question for each group, and extract relevant answer snippets from the article for the representative questions. Indeed, this is the technique behind the *Frequently asked questions* feature of Google News full coverage[4]. The approach works well because the most salient aspects of a news article are reflected in commonly asked questions from the users (see Table 1):

News story	**Top asked questions**
Starbucks closed for anti-bias training	- what time will starbucks close on may 29 - why are all starbucks closed - when are starbucks closing
Belgium beat England to secure third place at the 2018 FIFA World Cup	- what time is the england game today - what channel is england vs belgium - who won 3rd place in world cup 2018
Audubon zoo closed after Jaguar escapes	- how did audubon zoo jaguar escape - where is audubon zoo - what animals were killed at the audubon zoo

Table 1. News stories and their top questions mined from anonymized query logs.

However, for the latest articles that have just been published or long-tail articles that have not been queried by many users, this mining-based approach does not work due to the lack of historical queries. To address this challenge, we propose a **learning-based approach** that first predicts important snippets from the article and then generates natural language questions with those snippets as answers. The resulting Q/A pairs can achieve the same summarization effect on latest and long-tail articles as those mined from the query logs for pop-

[4] Private communication

ular news articles. To make this learning-based approach work, it is crucial to be able to generate training examples at scale. In fact, we employ the mining-based approach to generate weak supervision data that we then leverage in the learning-based approach as training examples.

To the best of our knowledge, this is the first study to develop a learning-based approach to generate Q/A pairs as structured summaries for news articles. The rest of the paper is organized as follows. Section 2 discusses related works. Section 3 describes how we obtain Q/A pairs using a mining approach to generate large scale weak training examples. In Section 4, we tackle the core challenge of structured summarization in the absence of associated queries with two steps. First, we propose a deep learning model to predict the attention maps given a news article, Second, we propose a natural question generation model that generates questions given the snippets from the attended article. Together, this generates salient Q/A pairs for the given article. In Section 5, we compare our proposed learning approach with baselines on both an academic dataset, SQuAD [24], and a large-scale news article dataset we collected. Finally, Section 6 concludes our work.

2 Related Work

Document summarization is a major focus of NLP research and follows two main approaches: first, extractive or abstractive summarization of the article using a few natural language sentences [5, 6, 10, 13, 25], and second, extracting salient entities and relational triples from the article [1,7,21]. As discussed in Section 1, the first approach focuses on providing an overview of the article instead of capturing aspects of the story that is salient to the users and is thus complimentary to the structured summary approach we study here. The relationship extraction approach focuses on concrete attributes. For example, it will likely extract "date of closing" for the Starbucks anti-bias training story (Table 1), but it will not capture abstract notions such as "why is Starbucks closed," which is in fact central to the story. Our proposed structured summary approach aims to capture all salient aspects, both concrete and abstract. News event summarization is a line of document summarization work that is specialized to news events [9,13,16,22,27,29]. To the best of our knowledge, we are the first study to propose a mechanism for capturing salient aspects using abstractive Q/A pairs as summaries for news stories.

Open information extraction is another related area [1,7,21], where the goal is to extract relation tuples from plain text. Similarly, methods proposed here are more likely to extract "*is-CEO* (Kevin Johnson, Starbucks)", rather than the reason why Starbucks is closed for the whole day. The latter will be much more salient to the users for understanding the story.

Answering questions based on a given corpus has been studied quite extensively. For example, [3] tackles open-domain question answering using Wikipedia as the unique knowledge source. It shows that open-domain Q/A is a very challenging problem and the performance of most state-of-the-art systems drops

Question cluster	**Question summary**
- when is starbucks closed for training - what day is starbucks closed for training	Starbucks training day closing time
- why is starbucks closed today - why is starbucks closing	Starbucks closed reason
- what is anti bias training	Anti bias training meaning

Table 2. Question clusters and summaries for the story "Starbucks closed for anti-bias training"

drastically when the passage that contains the answer is not already given. In [8], the authors propose to utilize question paraphrases from the WikiAnswers corpus to improve the performance of question answering. [26] proposes a bi-directional attention flow framework for question answering, by representing the context at different levels of granularity and generating query-aware context using attention. Our work took the opposite direction, namely we identify the important answer snippets first and attempt to phrase the questions afterwards.

Generating natural questions given a text passage only recently got some attention. [31] proposes a neural encoder-decoder model that reads a text passage and generates an answer-focused question, while [4] proposes to first identify question-worthy sentences in a input text passage, and then incorporates the prediction into an existing natural question generation system. The questions our work aims to generate is much more diverse and unpredictable (Section 5.2) than those works due to the nature of news domain.

3 Structured Summarization via Mining

Document queries [12] have long been used in various search tasks to improve quality. For news articles that have been published for a while (thus enough user-issued queries have been accumulated), mining the query log for salient questions for the article is an intuitive approach that works well.

While the idea is intuitive, there are two main challenges. First, identification of representative questions from the query log. For a single question intent, there are often many semantically identical but syntactically different queries the users may have issued. It is important to avoid semantically duplicate questions in the structured summary. Second, extraction of relevant answer snippets for the representative questions, which is akin to the traditional question-and-answering task, namely given a question and a document, identifying answers from the document. The difference for structured summary, however, is that each representative question is backed by a group of similar queries, which can be leveraged collectively to identify the best answer snippet.

In this section, we describe how we leverage existing techniques to design a mining approach for structured summarization. Throughout the section, we will follow the examples in Tables 2 and 3, which illustrate the two main tasks for the mining approach, question summarization and answer snippet extraction, respectively.

Question summary	**Answer snippet**
Starbucks training day closing time	Starbucks is closing more than 8,000 stores **Tuesday afternoon** for anti-bias training ...
Starbucks closed reason	... Tuesday afternoon **for anti-bias training**, a strategy some believe can keep racism at bay ...
Anti bias training meaning	... which offers training on **unconscious bias** and gave Starbucks input on its program ...

Table 3. Question summary and corresponding answer snippet for the news story "Starbucks closed for anti-bias training"

3.1 Question Clustering and Summarization

There are many benefits of leveraging documents queries for structured summarization. For example, document queries can counter the bias inherent in the article content: while the article author often injects their bias into the article (especially when they have a strong opinion on the underlying news story), queries issued by a large number of users, in an aggregated fashion, are less prone to any individual's bias. However, document queries are also challenging to work with because of their inherent noise and the multiple ways for users to express the same semantic intent.

The first challenge is that most document queries are single entities or short phrases and only $\sim 1\%$ of any article's accumulated document queries are in question format. When they are present, however, those questions are more specific and thus more useful in capturing important aspects in a story. For example, for the story "Starbucks closed for anti-bias training" in Table 2, the top (single-entity) queries are "starbucks" and "anti-bias", which are useful for knowing what is being talked about but difficult for users to understand what exactly happened. On the other hand, the top *question queries* are "when is starbucks closed" and "why is starbucks closed," which represent the aspects that are most interesting to the users. We extract question queries from all queries associated with a news article using a simple pattern, i.e., any query starting with *what, when, how, why, who, where, which, etc.*

The second challenge is that document queries contain many near duplicates since different users phrase the same question in different ways. We address this challenge by using hierarchical agglomerative clustering to cluster the question queries as shown, again, in Table 2. For the similarity measure between each pair of question queries $\mathrm{sim}(q_i, q_j)$, we take a weighted average of the word embeddings to derive a single vector for each query q_i, q_j and the weights are the inverted word frequency. The similarity between two queries are computed using cosine similarity. The word embedding is a 300-dimension vector we borrow from fastText [19].

The third challenge is readability. Question clusters are great for identifying salient aspects that users collectively consider as important, but a list of questions are not easily consumable by the readers. To improve the readability, we further generate *question summary*. Intuitively, for each question-query cluster,

Question summary	**Question cluster**
blockbuster last location	- where is the last blockbuster store located - where is the last open blockbuster
mlb trading deadline	- when is mlb trade deadline - when is trading deadline in mlb
winner of pacquiao fight	- who won the pacquiao fight last night - who won pacquiao fight

Table 4. Extracted question summary from queries for some example question clusters.

we pick a non-question query that is most similar to all question queries within the cluster. Anecdotally, as shown in Table 2, most of the "when ..." questions are summarized using "... time/date," and the "why ..." questions are summarized using "... reason." Note that we can also pick a representative query that is itself a question—we choose to have a non-question representative because the pool of non-question queries is bigger and a summary that is not a question can be used in more product features than a summary that is itself a question.

Specifically, for each question cluster C_q and question queries $q_1, q_2, \ldots, q_k \in C_q$, we find the closest non-question query q^* by: $q^* = \arg\max_{q \in C_{nq}} \sum_{i=1}^{k} \mathrm{sim}(q, q_i)$, where C_{nq} is the set of non-question queries, and $\mathrm{sim}(q, q_i)$ is the cosine similarity between the weighted average word embeddings of q, q_i as described in the clustering stage. In Table 4, we list examples of the question summary we automatically identified using this approach. In practice we found this approach can summarize the question clusters fairly well.

3.2 Answer Snippets Extraction

Identifying questions only partially fulfills the structured summary—it is also important to pair the questions with the correct answer snippets from the article so the users can grasp the whole story. We accomplish this in two stages. First, for each question in the question cluster, we apply a state-of-the-art question-and-answering model, QANet [30], to extract the corresponding answer snippet from the article. While QANet achieved an 84.5% Exact Match score on SQuAD v1.1 dataset[5], it has a much lower accuracy in our open domain scenario: $\sim 60\%$ based on our evaluation over a random sample of 100 (question-query, news-article) pairs.

The main challenge we encounter, is that in some cases the questions are not directly answerable from the article content. This is expected because compared to datasets where the provided passage (or context) is known to contain the answer, our question queries have a weaker association with the document content: users might have clicked on a document without knowing whether their questions can be answered. Fortunately, we have many paraphrased questions for each question cluster. Instead of using QANet to identify answer snippets just for one representative question, we can apply QANet on all paraphrasing questions in the cluster and pick the answer snippet that have the highest aggregated

[5] `https://rajpurkar.github.io/SQuAD-explorer/`

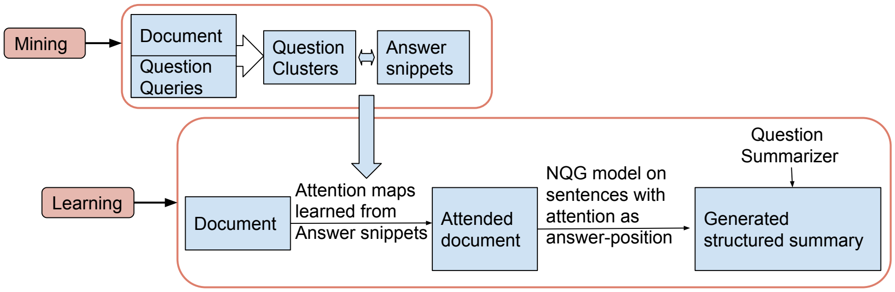

Mining
Document
Question Queries
Question Clusters
Answer snippets
Learning
Document
Attention maps learned from Answer snippets
Attended document
NQG model on sentences with attention as answer-position
Question Summarizer
Generated structured summary

Fig. 1. Overview of the learning based structured summarization system.

confidence score for all question queries in the cluster. We found this extra step improves the answer snippet accuracy by a large margin for the mining approach (from 60% to 75%+), enabling us to leverage the data for learning.

3.3 Results from the Mining Approach as Weak Supervision Data

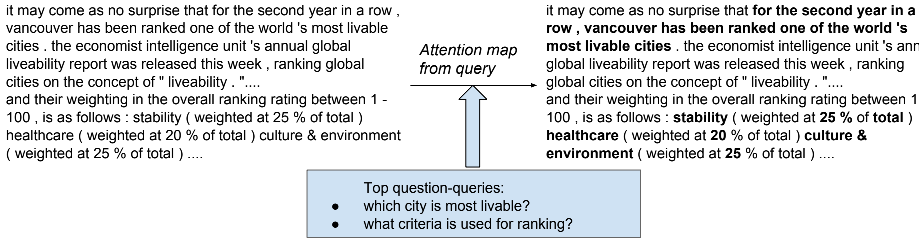
While the mining approach can be quite effective for articles with accumulated document queries, it does not address the challenge of producing structured summary for news articles in practice. The reason is that most news articles are consumed when they are just published and not enough document queries have been accumulated for the mining approach to be effective. Furthermore, long tail news articles, i.e., ones that do not have a large audience, also have very few accumulated document queries for the mining approach to be effective.

As a result, we do not consider our technical contributions on the mining approach as main contributions to the paper. Instead, we designed this mining approach with the main goal of using the results from this approach as weak supervision data, which we can subsequently use for a learning based approach that requires a substantial amount of the training data. We describe this learning based approach next.

4 Structured Summarization via Learning

As motivated in Section 3.3, for structured summary to work in practice (i.e., on fresh and long tail news articles), a more general approach is to summarize the document from its content only, without using any document queries. In this section, we describe a weakly-supervised system that utilizes the training data generated in Section 3 to produce structured summary for documents without associated document queries.

Figure 1 shows an overview of the system. Given a news article, the system predicts the *document attention map* (i.e., important answer snippets) using a model trained from prior associations of popular news articles, their question-query clusters and corresponding answer snippets. Intuitively, this can be considered as the reverse process of answer snippet extraction as described in Sec-


it may come as no surprise that for the second year in a row , vancouver has been ranked one of the world 's most livable cities . the economist intelligence unit 's annual global liveability report was released this week , ranking global cities on the concept of " liveability . "....
and their weighting in the overall ranking rating between 1 - 100 , is as follows : stability (weighted at 25 % of total) healthcare (weighted at 20 % of total) culture & environment (weighted at 25 % of total) ....

it may come as no surprise that **for the second year in a row , vancouver has been ranked one of the world 's most livable cities** . the economist intelligence unit 's anr global liveability report was released this week , ranking global cities on the concept of " liveability . "....
and their weighting in the overall ranking rating between 1 100 , is as follows : **stability** (weighted at **25 %** of **total**) **healthcare** (weighted at **20** % of total) **culture & environment** (weighted at **25** % of total) ....

Top question-queries:
- which city is most livable?
- what criteria is used for ranking?

Fig. 2. Example document attention map built on a news article from its question-query clusters.

tion 3.2. The attention map specifies the attended positions (i.e., answer snippets), for which a *natural question generation* (NQG) model is then used to automatically generate questions to form the question-and-answer pairs. Finally, a *question summarizer*, which is trained using the question cluster data (Section 3.1), consolidates the questions and summarizes the resulting representative questions into readable format.

4.1 Document Attention Map Prediction

A document attention map model predicts which parts of a document the users are paying attention. The answer snippet extraction process (Section 3.2) we described earlier enables us to generate the training corpus of (document, attention map) pairs at scale. We further improve the attention map quality by being very selective on choosing the answer position—for the set of all answers A to each question cluster, an answer position p is chosen only if:

$$S(A, p) = \frac{\sum_{a_i \in A, p \in a_i} s(a_i)}{\sum_{a_j \in A} s(a_j)} > 0.5$$

Intuitively, the aggregation score simulates majority voting, i.e., an answer position will be counted only if at least half of the paraphrased questions point to an answer that contains that position. Here $s(a_i)$ indicates the confidence score for answer a_i as computed by QANet [30]. Figure 2 illustrates an example document attention map.

Model. The overall architecture of the model is illustrated in Figure 3. We take word embeddings from fastText [19] and Part-Of-Speech Tags[6] as input features to the model and use layers described below to obtain predictions for positive (attended places) and negative (unattended places) classes.

Context layer. We place a bi-directional LSTM on top of the feature vectors, and concatenate the hidden states in both directions for each word i: $h_i = [\overrightarrow{h_i}, \overleftarrow{h_i}] \in \mathbb{R}^{2d}, i \in 1, ..., N$, where N is the total number of words in the context, and d is the dimension of the one-directional hidden state $\overrightarrow{h_i}$.

[6] https://nlp.stanford.edu/software/tagger.html

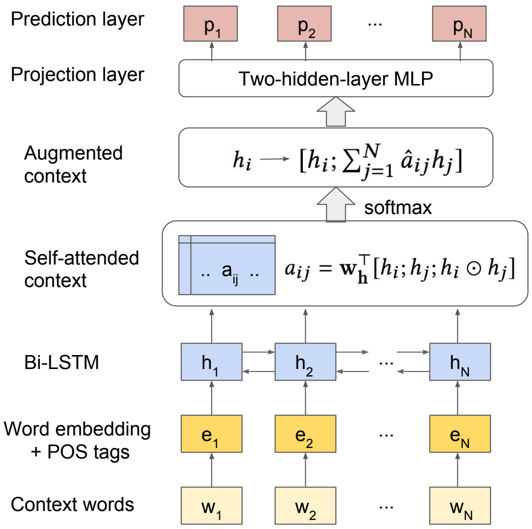


Fig. 3. Model architecture for predicting attention maps.

Self-attention layer. To augment the weights of important words in a context, we use a self-attention layer to attend the context to itself. The attention weight a_{ij} between each pair of hidden state representations (h_i, h_j) is computed using: $a_{ij} = \mathbf{w}_\mathbf{h}^\top [h_i; h_j; h_i \odot h_j]$, where $h_i \odot h_j$ is the element-wise product between two vectors. $h_i, h_j, h_i \odot h_j$ are concatenated together, and $\mathbf{w_h}$ is a trainable weight vector. The resulting attention matrix is denoted as $A \in \mathbb{R}^{N \times N}$. We mask the diagonal elements in A using a very large negative number (since the attention between a word and itself is not useful) and compute the softmax over the attention weights in each row, we denote the resulting matrix as $\hat{A}$. The attended-context is then given by: $\mathbf{H}_a = \hat{A}\mathbf{H}$, where $\mathbf{H} \in \mathbb{R}^{N \times 2d}$ is a matrix with row i being the hidden state representation h_i. We concatenate the context matrix $\mathbf{H}$ and the attended-context $\mathbf{H}_a$ as the augmented context representation $[\mathbf{H}; \mathbf{H}_a] \in \mathbb{R}^{N \times 4d}$, i.e., each hidden state representation h_i is augmented as $[h_i; \sum_{j=1}^{N} \hat{a}_{ij} h_j]$.

Output layers. Finally, we place a two-hidden-layer feed-forward network with ReLU activations on top of the augmented context representation to get the logits $\hat{p}_i$, which is a two-dimension vector representing the prediction for negative and positive classes, respectively.

Weighted cross-entropy loss. We apply a weighted cross-entropy loss function to balance positive and negative classes since attended places are usually a small fraction of the whole document context:

$$\text{loss} = -(1 - w_p) y \log(p) - w_p (1 - y) \log(1 - p),$$

where $p = \text{softmax}(\hat{p}_i)$, and w_p is automatically set based on the fraction of positive classes in the data.

4.2 Natural Question Generator

Given an answer position learned from the attention map model and the context surrounding it, we further train a sequence-to-sequence model to generate natural questions. As an example, for the answer "1724" in passage *The Old Truman Brewery, then known as the Black Eagle Brewery, was founded in* ***1724****,* we can generate the following question: *When was the Black Eagle Brewery founded?* The answer position is crucial here, namely, if the answer is "The Old Truman Brewery," then the question should be *Which brewery was founded in 1724?*[7]

Training data. We use the SQuAD [24] dataset as the training data. For each annotated passage, we generate the training pairs by first processing the passage using the PTBTokenizer[8] and obtaining the sentence segmentations. For each question-and-answer pair within the passage, we then generate (sentences, question) pairs by taking the sentence that contains the answer, the entire answer and answer_start position annotated in the dataset.

Model. The overall model architecture is described in Figure 4. We take word embeddings, POS tags (categorical), and answer positions (binary indicator of 1 or 0) as input features to the model, for each word in the input sentences. We use the conventional encoder/decoder framework to generate the questions, where the **encoder** layer is a bi-directional LSTM similar to the context layer in the attention map model. We concatenate the hidden states in both directions as hidden-state representation $h_i^s \in \mathbb{R}^{d_s}$ for each source token i. The **decoder** layer is much more sophisticated and we describe that in details next.

Decoder: We use an attention-based decoder [2, 18] with the copy mechanism [10, 25]. For a target state h_j^t, the attention weights over source hidden states h_i^s are computed by $a_{ij} = \text{softmax}(h_i^s \mathbf{W}_{s,t} h_j^t)$, where $\mathbf{W}_{s,t}$ is a trainable matrix placed between source hidden states and target hidden states. The attentional hidden state $\tilde{h}_j^t$, which is used for generating the current token given a sequence of previously generated tokens and inputs, is given by:

$$\tilde{h}_j^t = \tanh(\mathbf{W}_c[\texttt{context}_j; h_j^t])),$$

where $\mathbf{W_c}$ is a trainable matrix, and context_j represents the current context for h_j^t, i.e., the attention-weighted source hidden states, $\texttt{context}_j = \sum_i a_{ij} h_i^s$.

We further project $\tilde{h}_j^t$ to a D-dimension vector $\mathbf{g}$ with D being the size of the generation vocabulary $\mathcal{G}$, the attention-decoder gives a probability distribution on the generation vocabulary:

$$\mathbf{g} = \mathbf{\tilde{H}^t} \mathbf{W}_g + \mathbf{b}_g,$$

where $\mathbf{\tilde{H}^t}$ is a matrix with each row being $\tilde{h}_j^t \in \mathbb{R}^{d_t}$, $\mathbf{W}_g \in \mathbb{R}^{d_t \times D}, \mathbf{b}_g \in \mathbb{R}^D$ are trainable weights in the projection layer, and d_t is the dimension of the attentional target hidden states from the decoder.

[7] Note there can be multiple questions with the same answer snippet, for example, another question candidate could be: *Under which name is the Black Eagle Brewery also known?* Our learning based approach can learn those diverse questions provided that the training data captures the same diversity.

[8] https://nlp.stanford.edu/software/tokenizer.shtml

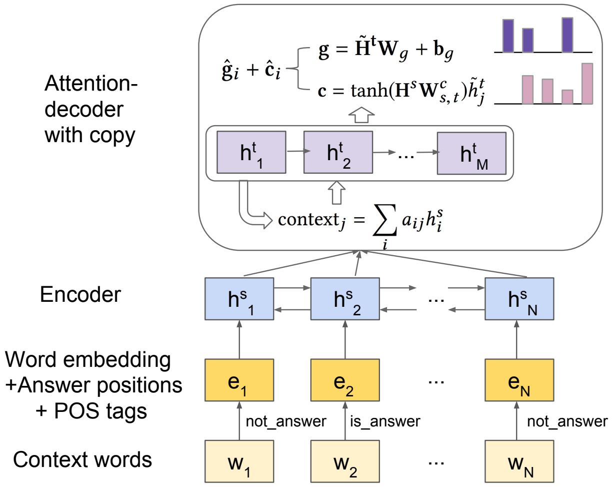

Attention-decoder with copy
$\hat{\mathbf{g}}_i + \hat{\mathbf{c}}_i$
$\mathbf{g} = \tilde{\mathbf{H}}^t\mathbf{W}_g + \mathbf{b}_g$
$\mathbf{c} = \tanh(\mathbf{H}^s\mathbf{W}^c_{s,t})\tilde{h}^t_j$
h^t_1
h^t_2
...
h^t_M
$\text{context}_j = \sum_i a_{ij}h^s_i$
Encoder
h^s_1
h^s_2
h^s_N
Word embedding +Answer positions + POS tags
e_1
e_2
e_N
not_answer
is_answer
not_answer
Context words
w_1
w_2
w_N

Fig. 4. Model architecture for natural question generation.

We augment the score by adding another probability distribution indicating whether a token in the target sequence should be copied from the tokens in the source sequence:

$$\mathbf{c} = \tanh(\mathbf{H}^s\mathbf{W}^c_{s,t})\tilde{h}^t_j,$$

where $\mathbf{H}^s \in \mathbb{R}^{N\times d_s}$ is a matrix with row i being the hidden state representation h^s_i from the encoder. $\mathbf{W}^c_{s,t} \in \mathbb{R}^{d_s\times d_t}$ is again a trainable matrix as the weights for copying a source token to the current target state, and d_s, d_t are the dimension of the hidden states from the encoder and decoder, respectively. The resulting vector $\mathbf{c} \in \mathbb{R}^N$ is a copy-score vector with each element c_i being the weight of copying source token i to the current target state $\tilde{h}^t_j$, $i \in 1, \dots, N$ where N is the total number of words in the input context.

Finally, we extend the vocabulary to be $\mathcal{G} \cup \mathcal{C}$, where $\mathcal{C}$ denotes the copy vocabulary (i.e., all the tokens from the each input sentence). The augmented score for each token i is then given by $\hat{\mathbf{g}}_i + \hat{\mathbf{c}}_i$, where $\hat{\mathbf{g}}$ and $\hat{\mathbf{c}}$ are vectors produced by projecting $\mathbf{g}, \mathbf{c}$ to the extended vocabulary, i.e., $\hat{\mathbf{g}}_i = \mathbf{g}_i$ if token $i \in \mathcal{G}$ and $\hat{\mathbf{g}}_i = 0$ otherwise. Similarly, $\hat{\mathbf{c}}_i = \mathbf{c}_i$ if token $i \in \mathcal{C}$, and $\hat{\mathbf{c}}_i = 0$ otherwise. Note for some tokens the score will be augmented as $\mathbf{g}_i + \mathbf{c}_i$ if token i is in both vocabularies.

4.3 Question Summarizer

Intuitively, in document attention map prediction and natural question generation, the learning based approach is a reverse process to the mining based approach: instead of mapping existing questions to snippets in the articles as answers, we learn where the important answers are and generate the questions from the answers and the context they are in. The two approaches, however, share

Dataset	# Articles	# QA pairs	# Question clusters
SQuAD	536	107,785	NA
News	91,675	3,096,289	458,375

Table 5. Dataset statistics for SQuAD and News

the same direction in question summarizer, both aim to consolidate the semantically equivalent questions and produce a readable summary of the questions. In the mining based approach, the summary comes from the non-question query q^* that is closest to all the question queries $\{q_1, q_2, \ldots, q_k\}$ in the question cluster C_q (Section 3.1). This is the process we leverage to generate training data for the learning based question summarizer at scale. Specifically, we construct each training pair as $\langle \{q_1, q_2, \ldots, q_k\}, q^* \rangle$, where $\{q_1, q_2, \ldots, q_k\}$ is the concatenation of all questions in cluster C_q, with an delimiter symbol $\langle s \rangle$.

The model architecture is similar to the sequence-to-sequence model we used for natural question generation as described in Figure 4, where the input sequence is now multiple question queries concatenated via $\langle s \rangle$. Furthermore, augmentations that are specific to question generation are removed, e.g., answer positions. We skip the detailed model description due to lack of space. Section 5.3 will show examples of the question summarization.

5 Experiments

We conduct extensive experiments on all three components/models of the learning based approach, namely document attention map prediction, natural question generation, and question summarization. Table 5 lists the characteristics of the two datasets.

SQuAD v1.1 [24]: The Stanford Question Answering Dataset is a reading comprehension dataset consisting of $107,785$ question-answer pairs posed by crowd workers on 536 Wikipedia articles. One limitation of this dataset is that the number of articles is relatively small and most of the questions posed by crowd workers are trivia questions and do not necessarily capture the important aspects presented in the article. As a result, this dataset can be used for learning natural question generation but is not very useful for learning document attention maps.

News. We collected a large set of news articles with their associated anonymized question queries from the Google search engine repository. We performed several filtering tasks: 1) removing articles that are too short (< 50 words) or too long (> 500 words, for efficiency consideration); 2) removing question queries that are too long (> 20 words) or too infrequently issued by the users (< 5 impressions); 3) for query clusters, only those have at least 3 valid question queries are considered; 4) removing articles with < 5 valid query clusters. Eventually we collected $91,675$ news articles as the input to our system, paired with $\sim$ 3M question-and-answer pairs and $\sim$ 460K query clusters.

The two datasets vary greatly in the number of articles and average number of Q/A pairs per article. SQuAD is specifically designed for question-and-answering

Method	Precision	Recall	AUC
Random	49.43	50.08	0.500
All-positive	50.00	100.00	0.500
MLP-only	57.44	67.17	0.590
+ Bi-LSTM	68.94	**78.07**	0.731
+ Self-attended context	**70.15**	74.22	**0.734**
Ablation experiments			
w/o pre-trained embedding	67.36	65.03	0.701
w/o POS tags	69.74	74.88	0.731

Table 6. Performance of document attention map prediction on the News data (test).

task and the number of pairs per article is large. As a result, the answer positions in the article are more “question-worthy” rather than *important* or *interesting* to the users. The News dataset, on the other hand, has a much smaller average number of question clusters per article (~ 5), most of which correspond to the most important aspects in the article since they are mined from actual user queries after proper anonymization.

5.1 Document Attention Map Prediction

We use the News dataset to evaluate the performance of our document attention map prediction model (Section 4.1). The evaluation data is generated as described in Section 4.1. In total we have $91,675$ (news-article, attention-map) pairs and we split the entire dataset into 90% training (82,501), 5% development (4,587), and 5% test (4,587). The input texts are lower-cased and tokenized[9] for processing. The word embeddings are the 300-dimension vectors from fastText [19] and the vocabulary size is $134K$, which consists of the most frequent words in our corpus, plus an $\langle unk \rangle$ token for all unknown words. In the experiments we use a 2-layer bidirectional LSTMs with 512 hidden units, and a dropout probability of 0.2 is applied to all LSTM cells. The two hidden layers in the output layer are set to size 512, 512, respectively. A mini-batch size of 256 examples is used and during training we use the Adam optimizer [14] with a learning rate of 0.001. An exponential decay of 0.95 is also applied to the learning rate for each epoch. The hyper-parameters are chosen based on the best performance on the development set.

Results. The results on the test set are listed in Table 6. Since the class probability is imbalanced we use the *Area Under the ROC Curve* (AUC) to evaluate the proposed methods. Because the problem is really new, we design our own baseline methods. As naive baselines, Random (by randomly highlighting a word) and All-positive (by predicting positive for all positions) both achieve an AUC score around 0.5. The MLP-only method is a stronger baseline that lays a multi-layer perceptron output layer directly on top of the input feature vectors and it achieves an AUC of 0.590.

[9] https://nlp.stanford.edu/software/tokenizer.shtml

Our proposed model, with additional layers for bi-directional LSTM, self-attended context, and augmented contexts, achieves the best performance of 0.734 AUC, significantly higher than the baselines. Results from two ablation experiments demonstrate that the pre-trained embedding improves the performance substantially (AUC increases from 0.701 to 0.734) and the POS-tag feature improves the performance slightly (AUC increases from 0.731 to 0.734).

5.2 Natural Question Generation

We use the SQuAD dataset to demonstrate the performance of natural question generation (Section 4.2). The SQuAD dataset (which is randomly split into 90% for training, 5% for development, and 5% for test in our experimental setting) has the ground truth questions as well as the correct answer positions annotated for each passage. We employ n-gram matching score (BLEU-4 [23] and ROUGE-L [17]) as the metrics. We use beam search with beam width 10 for generating questions from the decoder. We use a generation vocabulary size of 30K, and a copy vocabulary including all the source tokens. The combined vocabulary size is $80K$. We also experimented with different generation vocabulary sizes and the results were similar. The hidden unit sizes for the encoder and the decoder are set to 512 and 256, respectively. A dropout probability of 0.2 is applied to all LSTM cells. The training examples are sorted by input sequence length and divided into mini-batches of 128 examples each.

Results. Table 7 shows the results. The prior state-of-art model, which we adapt from the machine translation community to our problem as baseline, uses a sequence-to-sequence model with the attention decoder [18]. It achieves a BLEU-4 score of 6.83 and ROUGE-L score of 30.42. By incorporating the copy mechanism and answer positions into our model, the BLEU-4 and ROUGE-L scores can be improved significantly, reaching 11.79 and 38.96, respectively. The big improvements from adding the answer positions shows the importance of highlighting the right answers and demonstrates the value of our idea, namely using the mining-based approach to generate large scale training data for the learning-based approach. Finally, adding the POS tags leads to additional slight improvements, reaching BLEU-4 and ROUGE-L scores of 12.46 and 39.79, respectively. For completeness, we include the BLUE-4 result from [4] in the last row, even though it is focused on read comprehension, not news summary.

We do not have golden labels from the News dataset for this task. As anecdotal evidences of how well the natural question generation model works on

Method	**BLEU-4**	**ROUGE-L**
Seq2seq with attention decoder [18]	6.87	30.42
Seq2seq + copy	8.31	32.92
Seq2seq + copy + ans_pos	11.79	38.96
Seq2seq + copy + ans_pos + POS_tags	**12.46**	**39.79**
Neural generation model from [4]	11.50	n/a

Table 7. Performance comparison on the SQuAD dataset (test set).

Input sentence	**Generated question**
PC James Dixon , **39** , who starred in Sky TV 's Road Wars...	How old was James Dixon ?
By Tuesday , **it was downgraded to a post-tropical cyclone** .	What happened to the cyclone ?
Wilson is also a partner in a venture to bring the NBA back to Seattle .	Who is a partner to bring the NBA back to Seattle ?
Major tourists attractions , including the **Toronto Zoo , Ripley 's Aquarium of Canada , the CN Tower**...	What are some of the top attractions ?

Table 8. Example generated questions on the News dataset.

news given predicted document attention maps, Table 8 shows some examples. In general, we observe that the topics on the News dataset are more diverse and the answer types are more open-ended. For example, there are a few "what happened" (2nd example) questions and answers with long-spans (2nd and 5th examples), compared to answers from the SQuAD dataset which are usually shorter and in most cases single entities.

5.3 Question Summarizer

We use the News dataset to evaluate the performance of question summarizer (Section 4.3). Similar to Section 5.1, we split the data into 90% training, 5% development and 5% test. As described in Section 4.3, the reference summaries used for evaluation come from the non-question queries that are closest to all the question queries in the question cluster.

Method	**BLEU-2**	**BLEU-3**	**BLEU-4**	**ROUGE-L**
Seq2seq with attention decoder [18]	34.7	25.3	19.1	45.5
Seq2seq + copy + POS_tags	**41.8**	**31.5**	**24.4**	**52.3**

Table 9. Performance of question summarizer on the News dataset (test set).

Results. Table 9 shows the performance of our model compared against the same state-of-art sequence-to-sequence with attention decoder model as we used in Section 5.2 but adapted for this task. Our proposed model improves the performance substantially through the copy mechanism, achieving BLEU-4 and ROUGE-L scores of 24.4 and 52.3, respectively. Note that for question summarizer, the output is usually much shorter than the output of natural question generator, thus both BLEU-4 and ROUGE-L scores are higher than the results in Table 7. For comparison we also attached the BLEU-2 and BLEU-3 scores in the table. Table 10 shows a few example generated summaries. By training from large amount of samples, the model is able to summarize question clusters in a more concise way, and is sometimes capable of correcting the reference summary (top non-question query), e.g., in the last example, "winner" is a better summary than "score".

Input query cluster	**Generated summary**	**Reference summary**
how can i watch the golden knights game $\langle s \rangle$ where to watch golden knights $\langle s \rangle$ how to watch the golden knights game tonight $\langle s \rangle$	watch golden knights game	bars to watch golden knights
which bishop offered to resign $\langle s \rangle$ bishops who resigned $\langle s \rangle$ bishops who resign $\langle s \rangle$ chile bishops who resigned $\langle s \rangle$	bishops that resigned	bishops that resigned
who won stanley cup 2018 $\langle s \rangle$ who won the stanley cup 2018 $\langle s \rangle$	winner of stanley cup 2018	stanley cup 2018 score

Table 10. Example question summaries on the News dataset. The input to the model are all question queries in the same cluster, concatenated using the $\langle s \rangle$ symbol.

6 Conclusion

In this paper, we propose to summarize news articles in a structured way by using question-and-answer pairs. We propose an unsupervised approach by clustering question queries of historical popular news articles, extracting answer snippets of each question query in the cluster, and consolidating the questions into readable summaries, to produce the structured summary. This mining based approach enables us to generate training corpus for a learning based approach that allows us to perform structured summarization for cases where document queries are not present or scarce (e.g., newly published or long-tail articles). We proposed three predictive models. First, a model to predict the document attention map given a news article. Second, a model to generate natural language questions given the attended positions as answer positions in the article. Finally, a question summarizer to provide readable and succinct query summary. We show that this learning based approach produces meaningful structured summaries to capture important aspects of news articles.

References

1. Angeli, G., Premkumar, M.J., Manning, C.D.: Leveraging linguistic structure for open domain information extraction. In: ACL (2015)
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: ICLR (2015)
3. Chen, D., Fisch, A., Weston, J., Bordes, A.: Read wikipedia to answer open-domain questions. In: ACL (2017)
4. Du, X., Cardie, C.: Identifying where to focus in reading comprehension for neural question generation. In: EMNLP (2017)
5. Erkan, G., Radev, D.R.: Centroid-based summarization of multiple documents: sentence extraction, utilitybased evaluation, and user studies. In: NAACL-ANLP Workshop on Automatic Summarization (2000)
6. Erkan, G., Radev, D.R.: Lexrank: Graph-based lexical centrality as salience in text summarization. In: JAIR (2004)
7. Fader, A., Soderland, S., Etzioni, O.: Identifying relations for open information extraction. In: EMNLP (2011)

8. Fader, A., Zettlemoyer, L., Etzioni, O.: Paraphrase-driven learning for open question answering. In: ACL (2013)
9. Feng, X., Huang, L., Tang, D., Qin, B., Ji, H., Liu, T.: A language-independent neural network for event detection. In: ACL (2016)
10. Gu, J., Lu, Z., Li, H., Li, V.O.: Incorporating copying mechanism in sequence-to-sequence learning. In: ACL (2016)
11. Höffner, K., Walter, S., Marx, E., Usbeck, R., Lehmann, J., Ngonga Ngomo, A.C.: Survey on challenges of question answering in the semantic web. Semantic Web **8**(6) (2017)
12. Joachims, T.: Optimizing search engines using clickthrough data. In: KDD (2002)
13. Kedzie, C., Diaz, F., McKeown, K.: Real-time web scale event summarization using sequential decision making. In: IJCAI (2016)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
15. Kolomiyets, O., Moens, M.F.: A survey on question answering technology from an information retrieval perspective. Information Sciences **181**(24), 5412–5434 (2011)
16. Koutra, D., Bennett, P.N., Horvitz, E.: Events and controversies: Influences of a shocking news event on information seeking. In: WWW (2015)
17. Lin, C.Y.: Rouge: a package for automatic evaluation of summaries. In: ACL (July 2004), `https://www.microsoft.com/en-us/research/publication/rouge-a-package-for-automatic-evaluation-of-summaries/`
18. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: EMNLP (2015)
19. Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., Joulin, A.: Advances in pre-training distributed word representations. In: LREC (2018)
20. Nenkova, A., McKeown, K.: A survey of text summarization techniques. In: Mining Text Data, pp. 43–76. Springer (2012)
21. Nguyen, D.B., Abujabal, A., Tran, K., Theobald, M., Weikum, G.: Query-driven on-the-fly knowledge base construction. In: VLDB (2017)
22. Nguyen, T.H., Cho, K., Grishman, R.: Joint event extraction via recurrent neural networks. In: NAACL (2016)
23. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: ACL (2002)
24. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: Squad: 100,000+ questions for machine comprehension of text. In: EMNLP (2016)
25. See, A., Liu, P., Manning, C.: Get to the point: Summarization with pointer-generator networks. In: ACL (2017)
26. Seo, M., Kembhavi, A., Farhadi, A., Hajishirzi, H.: Bidirectional attention flow for machine comprehension. In: ICLR (2017)
27. Shen, C., Liu, F., Weng, F., Li, T.: A participant-based approach for event summarization using twitter streams. In: NAACL-HLT (2013)
28. Upstill, T.: The new Google News: AI meets human intelligence (2018), `https://www.blog.google/products/news/new-google-news-ai-meets-human-intelligence/`
29. Walker, C., Strassel, S., Medero, J., Maeda, K.: ACE 2005 Multilingual Training Corpus (Feb 2006), `https://catalog.ldc.upenn.edu/ldc2006t06`
30. Yu, A.W., Dohan, D., Luong, M.T., Zhao, R., Chen, K., Norouzi, M., Le, Q.V.: QANet: Combining local convolution with global self-attention for reading comprehension. In: ICLR (2018)
31. Zhou, Q., Yang, N., Wei, F., Tan, C., Bao, H., Zhou, M.: Neural question generation from text: A preliminary study. In: NLPCC (2017)