Question Headline Generation for News Articles

Ruqing Zhang†, Jiafeng Guo†, Yixing Fan†, Yanyan Lan†, Jun Xu†, Huanhuan Cao*, Xueqi Cheng†
†University of Chinese Academy of Sciences, Beijing, China
†CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, China
*ByteDance Inc, Beijing, China
zhangruqing@software.ict.ac.cn,{guojiafeng,fanyixing,lanyanyan,junxu,cxq}@ict.ac.cn
caohuanhuan@bytedance.com

ABSTRACT

In this paper, we introduce and tackle the Question Headline Generation (QHG) task. The motivation comes from the investigation of a real-world news portal where we find that news articles with question headlines often receive much higher click-through ratio than those with non-question headlines. The QHG task can be viewed as a specific form of the Question Generation (QG) task, with the emphasis on creating a natural question from a given news article by taking the entire article as the answer. A good QHG model thus should be able to generate a question by summarizing the essential topics of an article. Based on this idea, we propose a novel dual-attention sequence-to-sequence model (DASeq2Seq) for the QHG task. Unlike traditional sequence-to-sequence models which only employ the attention mechanism in the decoding phase for better generation, our DASeq2Seq further introduces a self-attention mechanism in the encoding phase to help generate a good summary of the article. We investigate two ways of the self-attention mechanism, namely global self-attention and distributed self-attention. Besides, we employ a vocabulary gate over both generic and question vocabularies to better capture the question patterns. Through the offline experiments, we show that our approach can significantly outperform the state-of-the-art question generation or headline generation models. Furthermore, we also conduct online evaluation to demonstrate the effectiveness of our approach using A/B test.

CCS CONCEPTS

• **Computing methodologies → Natural language generation**;

KEYWORDS

Question headline generation; self-attention mechanism

ACM Reference Format:
Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, Jun Xu, Huan-huan Cao, Xueqi Cheng. 2018. Question Headline Generation for News Articles

CIKM '18, October 22–26, 2018, Torino, Italy

ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/3269206.3271711

Table 1: Online click-through ratios (%) from five news channels in a real-world news portal. We randomly sampled 1000 news articles with question and non-question headlines respectively from each news channel.

	question headline	non-question headline
Society	15.23	11.07
Sports	13.24	12.16
Travel	16.29	15.02
Science	14.79	12.96
Health	16.73	15.19
Avg	15.26	13.28

In *The 27th ACM International Conference on Information and Knowledge Management (CIKM '18), October 22–26, 2018, Torino, Italy.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3269206.3271711

1 INTRODUCTION

In modern content-consumption market, especially news portals or social media applications, articles with a catchy headline often better attract users' attention and receive more clicks [6, 31, 36]. A typical catchy headline, during our investigation over a popular news portal in China, is the question headline (i.e., a headline in the question form). As shown in Table 1, we compare the click-through ratios over news articles with question headlines and non-question headlines from five news channels (i.e., society, sports, travel, science and health). We find that in average the news articles with question headlines receive higher click-through ratios than those with non-question headlines, and the gap is quite significant in some channels, e.g., society or science. For example, the click-through ratio of a news article with headline "战狼2豆瓣评分并不低，网友对电影评论太过疯狂。(The rating of the Wolf Warriors 2 is not low in Douban, but comments are so crazy.[1])" is 4.21%, while a similar news article with headline "如何看待豆瓣网友大肆吐槽战狼2？(What is your opinion on the crazy comments on the Wolf Warriors 2 in Douban?)" is 24.57%. As we can see, the non-question headline tends to directly present the main topic of an article, while the question headline turns it into a question which may arouse users' curiosity, make them think, and encourage them to click to find the answer. Such effects of question headlines have also been discussed in previous studies [1, 22, 23], which showed that when used properly, question headlines will create an almost irresistible draw to prospective readers.

[1]Wolf Warriors 2 is a 2017 Chinese action film, and Douban is a Chinese social networking service website allowing users to create comments related to films, books and music.

Due to the above observations, we propose to generate a question headline for a given news article and introduce the Question Headline Generation (QHG) task in this work. In previous literature, Rus et al. [44] has defined the Question Generation (QG) task, which aims to automatically generate questions from some form of input. The input could vary from information in a database to a deep semantic representation to raw text. The QHG task introduced in our work can be viewed as a specific form of the QG task, with the emphasis on creating a natural question from a given news article by taking the entire article (which could contain multiple paragraphs) as the answer.

Based on the above definition, there are two lines of research highly related to our problem, i.e., headline generation and question generation. Models on headline generation can be categorized into two folds, i.e., extractive methods and abstractive methods. Extractive methods produce the headline by extracting a sentence from the original text [5, 13]; while abstractive methods aim to generate the headline based on the understanding of the input text [39, 45]. In abstractive methods, the headline generation task can be formulated as a sequence-to-sequence (Seq2Seq) learning problem and neural models have been widely applied to solve it [10, 50]. For question generation, early work mostly adopted rule-based methods based on parse trees [17] or question templates [27]. Recently neural Seq2Seq models have also been employed to enable end-to-end learning to generate questions [14].

Without loss of generality, the QHG task can be formulated as a sequence-to-sequence learning problem. Given an input news article, which can be viewed as a sequence of words, we aim to produce a question headline. In this work, we introduce a novel dual-attention sequence-to-sequence (DASeq2Seq) model to solve this problem. Different from existing Seq2Seq models applied in headline or question generation which only employ the attention mechanism in the decoding phase for better generation, our DASeq2Seq further employ a self-attention mechanism in the encoding phase to obtain better article representation. The key idea is that a good QHG model should be able to generate a question by summarizing the essential topics of an article, while a good summarization need to identify those important sentences in an article rather than treat each sentence equally.

Specifically, we introduce two types of self-attention mechanism, namely global self-attention and distributed self-attention. The global self-attention, in a conceptual way, is an analogy to the expectation-maximization (EM) algorithm. We estimate the importance of each sentence through attention with some initial global representation (E-step), and use the importance to further aggregate sentence representations into the article representation (M-step). The distributed self-attention mechanism, on the other hand, is an analogy to a graph-based algorithm (e.g., TextRank [32]). The importance of a sentence is estimated based on the relationship between its own representation and those of the rest sentences. By using self-attention, we attempt to obtain a better representation of the article for the follow-up decoding phase. Note our self-attention mechanisms are quite different from those proposed in previous work [26, 48] in that we make use of correlation between sentences and provide good interpretation for each mechanism. In the decoding phase, besides the attention mechanism for generation, we further introduce a question vocabulary to capture question expressions explicitly and employ a vocabulary gate over both generic and question vocabularies to better learn the question patterns.

For experiments, we collected a large-scale real-world news collection with roughly 350,000 news articles with question headlines from our news portal, and we make the dataset publicly available for academic research[2]. We compared our model with several state-of-the-art methods using both automatic and human-based evaluations. The results demonstrate that our model can perform significantly better on the QHG task than existing methods. We also conducted online evaluation through A/B test. The results show that news articles with our generated question headline can receive much higher click-through ratio than that with the original non-question headline.

2 RELATED WORK

In this section, we briefly review the two lines of related work, i.e., headline generation and question generation.

2.1 Headline Generation

Broadly speaking, headline generation can be viewed as a text summarization problem, with the constraint that only a short sequence of words is allowed to generate to preserve the essential topics of a document. Existing methods on headline generation can be categorized into two folds, i.e., extractive methods and abstractive methods.

Extractive methods produce the headline by extracting a sentence from the original text. Early works include cue phrases [28], positional indicators [15], lexical occurrence statistics [30] and probabilistic measures for token salience [46]. Later, methods using sophisticated post-extraction strategies, such as revision [19] and grammar-based generation [41], have also been presented. More recently, Dorr et al. [13] proposed the Hedge Trimmer algorithm to create a headline by making use of handcrafted linguistic rules. Woodsend et al. [52] proposed a joint content selection and surface realization model which formulated the headline generation problem at the phrase level. Colmenares et al. [11] proposed a sequence-prediction technique which models the headline generation problem as a discrete optimization task in a feature-rich space. Recently neural Seq2Seq models have also been investigated for the extractive task [34].

Abstractive methods, on the other hand, aim to generate a headline based on the understanding of the input text. Banko et al. [4] viewed the task as a problem analogous to statistical machine translation for content selection and surface realization. Xu et al. [53] extracted features from Wikipedia to select keywords, and then employed keyword clustering methods to construct a headline. Recently, the task is formulated as a Seq2Seq learning problem and neural models have been widely adopted to solve it. For example, Rush et al. [45] trained an attention-based summarization (ABS) model, which used the lead (first) sentence of each article as the input for headline generation. Chopra et al. [10] extended this work with an attentive recurrent neural network (RNN) framework, and incorporated the position information of words. Nallapati et al. [35] introduced various effective techniques in the RNN seq2seq

[2]The dataset is available at https://github.com/daqingchong/QHGCorpus

framework. There have been some recent studies [2, 50] which also used the lead sentences as the input to generate headlines. In [51], Tan et al. proposed to identify the most important sentences in the input text by some existing extractive methods for headline generation. Different from these previous efforts, we introduce the self-attention mechanism into the Seq2Seq model to obtain a better article representation for question headline generation.

2.2 Question Generation

The Question Generation (QG) task has been defined by Rus et al. [44], which aims to automatically generate questions from some form of input. There have been two specific subtasks of QG introduced in previous literature: Question Generation from Sentences (QGS) and Question Generation from Paragraphs (QGP). The QGS task aims to generate questions of a specified type from a single sentence; The QGP task aims to generate a list of questions from a given input paragraph. In the QGP task, the generated questions should be at three scope levels: broad (entire input paragraph), medium (one or more sentences), and specific (phrase or less), where the scope is defined by the portion of the paragraph that answers the question. The QHG task introduced in our work is related to the QGP task but with some clear differences. Firstly, the input of our QHG task is a news article with multiple paragraphs, while the input of the QGP task is a single paragraph from Wikipedia, OpenLearn, and Yahoo!Answers. Secondly, our QHG task only requires a broad-level question to be generated as the headline. Finally, our QHG task provide a large dataset with tens of thousands articles while the QGP task only provides 60 paragraphs.

Early works on question generation often adopted a rule-based approach. Several previous works processed documents as individual sentences using syntactic [17, 21] or semantic parsing [29, 40], then reformulated questions using hand-crafted rules over parse trees. These traditional approaches often generate questions with high word overlap with the original text. An alternative approach is to use generic question templates whose slots can be filled with entities from the document [7, 27]. These approaches comprise pipelines of several independent components which are often difficult to achieve the optimal performance.

Recently, neural Seq2Seq models have enabled end-to-end learning of question generation systems. Serban et al. [47] trained a neural system to convert knowledge base triples (subject, relation, object) into natural language questions. Mostafazadeh et al. [33] used a neural architecture to generate questions from images rather than text. Du et al. [14] proposed to directly map a sentence from a text passage to a question using a traditional Seq2Seq model with decoder attention. Our model shares similar ideas with recent neural Seq2Seq models, but further introduces a self-attention mechanism in the encoding phase to obtain better article representations as well as a vocabulary gating in the decoding phase to better capture question patterns.

3 BACKGROUND

For better description of our model, we first briefly introduce the basic idea of the Seq2Seq model and the attention mechanism which have been widely adopted in headline generation and question generation.

3.1 Sequence-to-Sequence Model

The Seq2Seq model employs the encoder-decoder framework for text generation. The encoder is used for encoding the input text into a representation vector and the decoder is used to generate the output according to the input representation.

Formally, let $X = (x_1, \ldots, x_M)$ denote the input text which is a sequence of M words, and $Y = (y_1, \ldots, y_N)$ denote the output. The Seq2Seq model takes X as input, encodes it into a vector representation $\mathbf{c}$, and uses $\mathbf{c}$ to decode the output Y. Recurrent Neural Network (RNN) [43] is natural to be the encoder for text inputs due to its ability to deal with varied-length inputs. The idea of RNN is to perform the same task for every element of a sequence, with the output being dependent on the previous computations, i.e.,

$$\mathbf{h}_i = f(x_i, \mathbf{h}_{i-1}), \tag{1}$$

$$\mathbf{c} = g(\{\mathbf{h}_1, \ldots, \mathbf{h}_M\}), \tag{2}$$

where $\mathbf{h}_i$ is the hidden state for the input word x_i, f is the dynamics function, $\mathbf{c}$ is the so-called context vector, and g is a function to calculate $\mathbf{c}$ from hidden states. A typical instance of g is choosing the last state: $g(\{\mathbf{h}_1, \ldots, \mathbf{h}_M\}) = \mathbf{h}_M$. In practice it is found that gated RNN alternatives such as LSTM [18] and GRU [9] often perform much better than vanilla ones.

The decoder is used to generate the output sequence given the input representation $\mathbf{c}$. The decoder generates one word every step based on the input representation and previously generated words:

$$\mathbf{h}_{y_t} = f(y_{t-1}, \mathbf{h}_{y_{t-1}}, \mathbf{c}), \tag{3}$$

$$p(y_t | y_{<t}, X) = \phi(y_{t-1}, \mathbf{h}_{y_t}), \tag{4}$$

where $\mathbf{h}_{y_t}$ is the hidden state at time t of the decoder, y_t is the predicted target symbol at t (through function $\phi(\cdot)$ which is a single-layer feed-forward neural network) with $y_{<t}$ denoting the history $\{y_1, \ldots, y_{t-1}\}$.

3.2 The Attention Mechanism

The attention mechanism [3] was first introduced into Seq2Seq models to release the burden of compressing the entire source into a fixed-length vector as context. Many successful applications show the effectiveness of the attention mechanism [24, 45]. The attention mechanism uses a dynamically changing context $\mathbf{c}_t$ in the decoding phase. A natural option (or rather "soft attention") is to represent $\mathbf{c}_t$ as a weighted sum of the source hidden states $\{\mathbf{h}_1, \ldots, \mathbf{h}_M\}$, i.e.,

$$\mathbf{c}_t = \sum_{i=1}^{M} \alpha_{ti} \mathbf{h}_i, \tag{5}$$

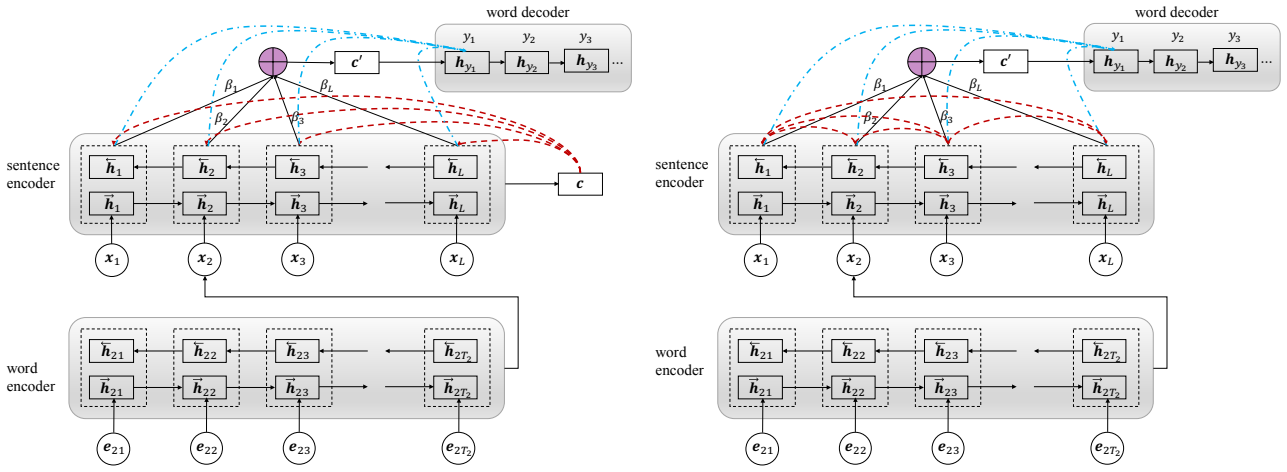
where α_{ti} indicates how much the i-th word x_i from the source input contributes to generating the t-th word, and is usually computed as:

$$\alpha_{ti} = \frac{\exp(\mathbf{h}_i \cdot \mathbf{h}_{y_{t-1}})}{\sum_{j=1}^{M} \exp(\mathbf{h}_j \cdot \mathbf{h}_{y_{t-1}})}, \tag{6}$$

where $\mathbf{h}_{y_{t-1}}$ represents the RNN hidden state (just before emitting y_t) of the decoder.

4 OUR APPROACH

In this section, we present the Dual-Attention Sequence-to-Sequence Model (DASeq2Seq), a novel Seq2Seq model designed for the Question Headline Generation task.



(a) DASeq2Seq with global self-attention mechanism

(b) DASeq2Seq with distributed self-attention mechanism

Figure 1: Dual-Attention Sequence-to-Sequence model (DASeq2Seq) with two types of self-attention mechanism. Red colored lines stand for the self-attention mechanism in the encoder, and blue colored lines stand for the attention mechanism in the decoder.

4.1 Overview

Formally, given a news article $D = \{s_1, \ldots, s_L\}$ with L sentences, where each sentence s_i contains a sequence of T_i words w_{it} ($t \in [1, T_i]$), DASeq2Seq aims to generate a question headline Y for the news article D.

Basically, the DASeq2Seq employs the encoder-decoder framework for the task. In the encoding phase, since news articles are usually quite long, DASeq2Seq utilizes the hierarchical encoder framework as previous practice [24]. Moreover, in order to learn a better representation of the news article, DASeq2Seq employs a self-attention mechanism to identify the importance of different sentences. We introduce two types of self-attention mechanism, namely global self-attention and distributed self-attention. In the decoding phase, DASeq2Seq employs a similar attention mechanism as traditional Seq2Seq approaches [3] as well as a vocabulary gating scheme to control the question headline generation.

4.2 Encoder

The goal of the encoder is to map the input news article D into a compact vector representation that can capture its essential topics. In DASeq2Seq, we adopt a hierarchical encoder framework, where we use a word encoder to encode the words of a sentence s_i, and use a sentence encoder to encode the sentences of a news article D.

As depicted in Figure 1, each word w_{it} is represented by its distributed representation $\mathbf{e}_{it}$ which is mapped by a word embedding matrix $\mathbf{E}$. We then use a bi-directional GRU as both the word and sentence encoder, which summarizes not only the preceding words/sentences, but also the following words/sentences. The implementation of the GRU is parameterized as:

$$\begin{aligned} \mathbf{z}_t &= \sigma(\mathbf{W}_z \mathbf{x}_t + \mathbf{U}_z \mathbf{h}_{t-1} + \mathbf{b}_z), \\ \mathbf{r}_t &= \sigma(\mathbf{W}_r \mathbf{x}_t + \mathbf{U}_r \mathbf{h}_{t-1} + \mathbf{b}_r), \\ \mathbf{h}_t &= (1 - \mathbf{z}_t) \circ \tanh(\mathbf{W}_h \mathbf{x}_t + \mathbf{U}_h(\mathbf{h}_{t-1} \circ \mathbf{r}_t) + \mathbf{b}_h) + \mathbf{z}_t \circ \mathbf{h}_{t-1}, \end{aligned} \tag{7}$$

where $\mathbf{x}_t$, $\mathbf{h}_t$, $\mathbf{z}_t$, and $\mathbf{r}_t$ are the input vector, output vector, update gate vector and reset gate vector respectively. $\mathbf{W}_z$, $\mathbf{W}_r$, $\mathbf{W}_h$, $\mathbf{U}_z$, $\mathbf{U}_r$, $\mathbf{U}_h$, $\mathbf{b}_z$, $\mathbf{b}_r$ and $\mathbf{b}_h$ are parameter matrices and vectors.

The forward GRU in word encoder reads the words in the i-th sentence s_i in the left-to-right direction, resulting in a sequence of hidden states $(\overrightarrow{\mathbf{h}_{i1}}, \ldots, \overrightarrow{\mathbf{h}_{iT_i}})$. The backward GRU reads s_i in the reversed direction and outputs $(\overleftarrow{\mathbf{h}_{i1}}, \ldots, \overleftarrow{\mathbf{h}_{iT_i}})$. We obtain the hidden state for a given word w_{it} by concatenating the forward and backward hidden states, i.e., $\mathbf{h}_{it} = [\overrightarrow{\mathbf{h}_{it}} || \overleftarrow{\mathbf{h}_{it}}]$. Then, we concatenate the last hidden states of the forward and backward passes as the representation of the sentence s_i, denoted as $\mathbf{x}_i = [\overrightarrow{\mathbf{h}_{iT_i}} || \overleftarrow{\mathbf{h}_{i1}}]$. A sentence encoder is used to sequentially receive the embeddings of sentences $\{\mathbf{x}_i\}_{i=1}^{L}$ in a similar way. The hidden state of each sentence is given by $\mathbf{h}_i = [\overrightarrow{\mathbf{h}_i} || \overleftarrow{\mathbf{h}_i}]$, where $\overrightarrow{\mathbf{h}_i}$ and $\overleftarrow{\mathbf{h}_i}$ are the forward and backward hidden states of the sentence encoder respectively .

Based on the representation of each sentence, a simple method to obtain the news article representation $\mathbf{c}$ is to directly concatenate or aggregate the hidden states of sentences. However, such method may not be optimal since it assumes that each sentence is equally important to the article. Ideally, we need a way to identify the important sentences that can reveal the essential topics of the article, and aggregate the sentence representations according to their importance to obtain a good article representation. Here we introduce two types of self-attention mechanism for this purpose in the encoder, namely global self-attention and distributed self-attention.

4.2.1 Global Self-attention Mechanism. The underlying idea of the global self-attention mechanism is that a sentence is important in an article if it can strongly support the major topics of the article, while the major topics of an article can be represented by its important sentences. This is like a chicken-and-egg problem so the global self-attention employs a way similar to the expectation-maximization

(EM) algorithm. As shown in Figure 1(a), we first assume that each sentence is equally important (E-step) and obtain an initial global representation of an article $\mathbf{c}$ by simply concatenating the last hidden state of the forward and backward pass in the sentence encoder, i.e., $\mathbf{c} = [\overrightarrow{\mathbf{h}_L}||\overleftarrow{\mathbf{h}_1}]$ (M-step). We then use the initital $\mathbf{c}$ to re-estimate the importance of each sentence through attention (E-step). The importance score (or weight) β_i of the sentence s_i is given by

$$\hat{\beta}_i = tanh(\mathbf{h}_i^{\mathrm{T}} \cdot \mathbf{Q} \cdot \mathbf{c}), \tag{8}$$

$$\beta_i = \exp(\hat{\beta}_i) / \sum_j \exp(\hat{\beta}_j), \tag{9}$$

where $\mathbf{Q}$ is a parameter matrix to be learned and the softmax function ensures all the weights sum up to 1. Finally, we obtain the representation of the article $\mathbf{c}'$ by using the weighted sums of hidden states $\{\mathbf{h}_1, \ldots, \mathbf{h}_L\}$ (M-step):

$$\mathbf{c}' = \sum_{k=1}^{L} \beta_k \mathbf{h}_k. \tag{10}$$

The representation $\mathbf{c}'$ is then used as the initial hidden state of the decoder.

4.2.2 Distributed Self-attention Mechanism. The underlying idea of the distributed self-attention mechanism is that a sentence is important in an article if it is highly related to many important sentences. In a conceptual way, the distributed self-attention mechanism is an analogy to graph-based extractive summarization models such as TextRank [32] and LexRank [16], which are based on the PageRank [37] algorithm. These graph-based models have shown strong ability in identifying important sentences in an article. As shown in Figure 1(b), we compute the importance score (or weight) of a sentence based on the relationship between hidden states of the target sentence and all the rest sentences.

Specifically, the importance score (or weight) β_i of the sentence s_i is computed by

$$\hat{\beta}_i = \sum_{j=1,\ldots,L, j\neq i} \eta(\mathbf{h}_i, \mathbf{h}_j), \tag{11}$$

$$\beta_i = \exp(\hat{\beta}_i) / \sum_j \exp(\hat{\beta}_j), \tag{12}$$

where η can be defined in different ways such as $\eta(\mathbf{a}, \mathbf{b}) = \mathbf{a}^{\mathrm{T}}\mathbf{b}$, $\eta(\mathbf{a}, \mathbf{b}) = \mathbf{a}^{\mathrm{T}}\mathbf{U}\mathbf{b}$ or some non-linear function like $\eta(\mathbf{a}, \mathbf{b}) = tanh(\mathbf{a}^{\mathrm{T}}\mathbf{U}\mathbf{b})$. Here, we adopt $\eta(\mathbf{a}, \mathbf{b}) = \mathbf{a}^{\mathrm{T}}\mathbf{U}\mathbf{b}$ as a trade-off between model capacity and computational complexity, where $\mathbf{U}$ is a parameter matrix to be learned. Similar to the global self-attention, we then sum up the GRU hidden states $\mathbf{h}_i$ according to the weight β_i to get a vector representation $\mathbf{c}'$ of the input article, and use $\mathbf{c}'$ as the initial state of the decoder.

4.3 Decoder

The goal of the decoder is to generate a question headline Y given the hidden representation of the input news article. As we known, a question usually contains some question words such as "WHO", "WHAT" and "WHERE" and some generic words such as "he","music" and "home". To better capture the question patterns, we distinguish the question words from those generic words and employ two vocabularies in the decoding phase, namely question vocabulary and generic vocabulary respectively. Note that these two vocabularies have no intersection. As shown in Figure 2, to generate a word at each time step, the decoder learns two key abilities jointly: (1) to predict which vocabulary should be used; and (2) to pick a specific word from that vocabulary.

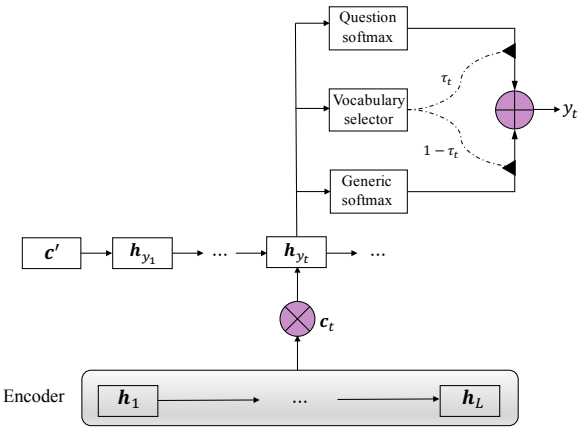


Figure 2: A simple depiction of the Vocabulary Gate (VoG) architecture in the decoder of DASeq2Seq. The final word prediction will be performed via a softmax output layer, either over the question vocabulary or over the generic (non-question) vocabulary. The vocabulary selector decides which vocabulary to use.

We employ a Vocabulary Gate (VoG) for the vocabulary selection in step (1), which is defined by

$$\tau_t = \sigma(\mathbf{W}_K^s \cdot \mathbf{h}_{y_t} + \mathbf{W}_K^y \cdot y_{t-1} + \mathbf{W}_K^c \cdot \mathbf{c}_t + \mathbf{b}_K), \tag{13}$$

where $\sigma(\cdot)$ stands for the sigmoid function $\sigma(x) = \frac{1}{exp(-x)+1}$, and $\mathbf{W}_K^s, \mathbf{W}_K^y, \mathbf{W}_K^c$ and $\mathbf{b}_K$ are parameters. To pick up a specific word from the selected vocabulary in step (2), we use two different softmax output layers, one for the question vocabulary ($P_q(y_t = w_q)$) and one for the generic vocabulary ($P_g(y_t = w_g)$) which are defined by

$$P_q(y_t = w_q) = softmax(\mathbf{W}_Q^s \cdot \mathbf{h}_{y_t} + \mathbf{W}_Q^y \cdot y_{t-1} + \mathbf{b}_Q), \tag{14}$$

$$P_g(y_t = w_g) = softmax(\mathbf{W}_G^s \cdot \mathbf{h}_{y_t} + \mathbf{W}_G^y \cdot y_{t-1} + \mathbf{b}_G), \tag{15}$$

where $\mathbf{W}_Q^s, \mathbf{W}_Q^y, \mathbf{W}_G^s, \mathbf{W}_G^y, \mathbf{b}_Q$ and $\mathbf{b}_G$ are parameters. $\mathbf{h}_{y_t}$ is the hidden state at step time t of the decoder, with the initial state defined by the article representation $\mathbf{c}'$. Meanwhile, the decoder also employs an attention mechanism over all the sentence representations $\{\mathbf{h}_1, \ldots, \mathbf{h}_L\}$ to form a context vector $\mathbf{c}_t$ along with each decoding phase for better generation. Thus, $\mathbf{h}_{y_t}$ can be obtained by

$$\mathbf{h}_{y_t} = f(y_{t-1}, \mathbf{h}_{y_{t-1}}, \mathbf{c}_t), \tag{16}$$

$$\mathbf{c}_t = \sum_{i=1}^{L} \alpha_{ti} \mathbf{h}_i, \tag{17}$$

$$\mathbf{h}_{y_0} = \mathbf{c}', \tag{18}$$

where f is a GRU unit, and α_{ti} is computed as in Equation 6. The final output distribution $\mathbf{o}_t$ over all the words (i.e., both generic and question words) is the concatenation of the two vectors, namely,

$$\mathbf{o}_t = \begin{bmatrix} \tau_t \times P_q(y_t = w_q) \\ (1-\tau_t) \times P_g(y_t = w_g) \end{bmatrix}. \tag{19}$$

Table 2: Data statistics: #s denotes the number of sentences and #w denotes the number of Chinese words.

Pairs	331,608
Vocabulary #w	902,635
Article: max #s	355.0
Article: avg #s	12.2
Article: avg #w	300.8
Article sentence: avg #w	24.7
Question headline: avg #w	9.4

Note that, during the training, we provide the model with the specific information wherever the target word is from the question vocabulary or generic vocabulary, and thus we do not need to sample.

4.4 Model Learning

We employ maximum likelihood estimation (MLE) to learn our DASeq2Seq model. Specifically, we define the loss function as the cross entropy error between the predicted token distribution $\mathbf{o}_t$ and the gold distribution $\mathbf{p}_t$ in the training corpus $\mathcal{D}$. Furthermore, we apply another cross entropy error over the vocabulary gate, emphasizing the selection of the vocabularies. The loss over one news article and question headline pair < D, Y > is then defined as

$$\mathcal{L}(\theta) = -\sum_{t=1}^{|Y|} \mathbf{p}_t \log(\mathbf{o}_t) - \sum_{t=1}^{|Y|} q_t \log(\tau_t), \tag{20}$$

where τ_t denotes the vocabulary selection probability, and $q_t \in \{0, 1\}$ is the true vocabulary choice of each word in Y. We use the Adam [20] gradient-based optimization method to learn the model parameters θ.

5 OFFLINE EXPERIMENTS

In this section, we conduct offline experiments to verify the effectiveness of our proposed model.

5.1 Data Collection

For evaluation purposes, we build a novel QHG dataset, i.e., a collection of 350,000 <news article, question headline> pairs, from a real-world news portal. These articles and their corresponding question headlines are manually written by professional editors, thus suitable to be viewed as gold standard for our task.

The pre-process of the dataset is as follows: We clean all non-text contents (e.g., pictures and videos) and noise (e.g., HTML tags), and employ the Jieba Chinese word segmenter[3] to tokenize the pairs. We leave out the articles that have less than 20 Chinese words or more than 2000 Chinese words, and whose headlines have less than 3 Chinese words or more than 25 Chinese words. Different from previous work [45], we do not filter the dataset by using the word overlap between the headlines and the lead sentences of articles, which ensures the dataset to be more realistic. After cleaning, there are 331,608 pairs left. The detailed statistics of the dataset are shown in Table 2. We randomly divide the dataset into a training set (80%), a development set (10%), and a test set (10%).

[3]https://pypi.python.org/pypi/jieba

5.2 Implementation Details

We implement our model in Tensorflow[4]. We use one layer of bi-directional GRU for word and sentence encoder respectively and another uni-directional GRU for decoder, with the GRU hidden unit size set as 600 in the encoder and 1200 in the decoder. The dimension of word embeddings is 300. We use pretrained word2vec[5] vectors trained on the same corpus to initialize the word embeddings, and the word embeddings will be further fine-tuned during training. The parameters of Adam are set as in [20]. The learnable parameters (e.g., the parameters $\mathbf{Q}$, $\mathbf{U}$, $\mathbf{W}_Q^s$ and $\mathbf{W}_Q^y$) are uniformly initialized in the range $[-0.1, 0.1]$.

We keep the 60,000 most frequently occurring words in our experiments. We select 144 question words (such as "哪里(WHERE)", "什么(WHAT)", "何时(WHEN)" and "是否(IF)") to form the question vocabulary, and the rest words make up the generic vocabulary. The question vocabulary covers question words in 94.5% question headlines in the corpus. All the other words outside the question and generic vocabularies are replaced by the special <UNK> symbol, and all digits are replaced by the # symbol. Lastly, "<eos>" is appended at the end of each sentence to indicate the end of the sentence, while "<eod>" is appended at the end of each article to indicate the end of the whole article.

For training, we use a mini-batch size of 64 and news articles with similar length (in terms of the number of sentences in the input news articles) are organized to be a batch [54]. Dropout with probability 0.2 is applied between vertical GRU stacks and gradient clipping is adopted by scaling gradients when the norm exceeded a threshold of 5. The sentence decoder stops when it generates the "<eos>" token. We run our model on a Tesla K80 GPU card, and we run the training for up to 16 epochs, which takes approximately one day. We select the model that achieves the lowest perplexity on the development set. All hyper-parameters of our model are also tuned using the development set. We report results on the test set. We refer to our DAseq2seq model with global and distributed self-attention mechanisms as DASeq2Seq$_{global}$ and DASeq2Seq$_{dist}$ respectively.

5.3 Baselines

To verify the effectiveness of our model on the QHG task, we first implement some variants of our model by removing the self-attention mechanism and the vocabulary gating scheme, and adopting different model architectures, including:

- **Seq2Seq**$_{flat}$ is a basic Seq2Seq model using a flat encoder structure, which concatenates all the sentences of a news article as the input. It can be viewed as the adaption of the model in [49].
- **Seq2Seq**$_{flat+att}$ extends Seq2Seq$_{flat}$ by adding word-level attention in the decoding phase. It can be viewed as the adaption of the model proposed in [14].
- **Seq2Seq**$_{hie+att}$ employs a hierarchical encoder structure and uses sentence-level attention in the decoding phase. It can be viewed as the adaption of the model in [24].

Furthermore, we also apply several state-of-the-art question generation or headline generation models to the QHG task.

[4]https://www.tensorflow.org/
[5]https://code.google.com/archive/p/word2vec/

Table 3: Automatic evaluation results of different models by ROUGE, BLEU and METEOR on the QHG dataset. Two-tailed t-tests demonstrate the improvements of our models to all the baselines are statistically significant (‡ indicates p-value < 0.01).

Model	ROUGE-1	ROUGE-2	ROUGE-L	BLEU-1	BLEU-2	BLEU-3	METEOR
IR_{BM25}	2.95	0.23	1.79	1.61	0.87	0.38	2.28
PREFIX	24.94	10.75	21.22	13.53	9.18	6.87	8.84
ABS	25.91	13.15	23.95	24.01	15.29	11.57	11.39
Distraction	25.27	13.23	23.17	23.46	14.66	11.05	10.93
Seq2Seq_{flat}	18.73	9.36	17.22	17.65	8.38	5.98	7.28
$\text{Seq2Seq}_{flat+att}$	24.23	13.49	22.36	21.75	12.62	9.06	10.10
$\text{Seq2Seq}_{hie+att}$	27.29	14.57	25.24	25.66	15.59	11.62	11.62
$\text{DASeq2Seq}_{global}$ (no VoG)	28.63‡	15.14	26.11‡	26.76‡	16.57‡	12.15	12.45‡
DASeq2Seq_{dist} (no VoG)	28.86‡	15.51‡	26.41‡	26.87‡	16.92‡	12.57‡	12.78‡
DASeq2Seq_{glob}	29.51‡	15.65‡	26.63‡	27.24‡	17.35‡	12.92‡	**12.84**‡
DASeq2Seq_{dist}	**29.85**‡	**16.42**‡	**27.23**‡	**27.28**‡	**17.88**‡	**13.09**‡	**13.10**‡

- **IR_{BM25}** [45] stands for an information retrieval (IR) baseline, which indexes the question headlines in the training set, and searches the best matching question headline for the input news article using BM-25 [42] scoring function.
- **PREFIX** [45] simply uses the first sentence as the headline.
- **ABS** [45] is the attention bag-of-words encoder based sentence summarization model[6], which uses the lead (first) sentence of each article as the input for headline generation.
- **Distraction** [8] uses a new attention mechanism by distracting the historical attention in the decoding steps[7].

5.4 Evaluation Metrics

We use both automatic evaluation and human evaluation to measure the quality of question headlines generated by our model and the baselines.

For automatic evaluation, we use ROUGE [25], BLEU [38] and METEOR [12], which have been proved strongly correlated with human evaluations. ROUGE is commonly employed to evaluate n-grams recall of the summaries with gold-standard sentences as references. ROUGE-1, ROUGE-2 and ROUGE-L recall scores measure the uni-gram, bi-gram and longest-common substring similarities, respectively. BLEU measures the average n-gram precision on a set of reference sentences, with a penalty for overly short sentences. BLEU-n is the BLEU score that uses up to n-grams for counting co-occurrences. METEOR is a recall-oriented metric, which calculates the similarity between generations and references by considering synonyms, stemming and paraphrases.

For human evaluation, following the procedure in [14], we consider two modalities: 1) Naturalness, which indicates whether the headline is a grammatically correct and fluent question; and 2) Difficulty, which measures the headline-article syntactic divergence and the reasoning needed to answer the question. We randomly sampled 100 <news article, question headline> pairs generated from the best performing baseline and our model. We asked three professional native speakers to rate the pairs in terms of the modalities mentioned above on a 1~5 scale (5 for the best). We also asked the native speakers to rank the question headlines generated by different models and the ground truth (Human), according to the overall quality.

Table 4: Human evaluation results for question headlines on the QHG dataset. Naturalness and Difficulty are rated on a 1~5 scale (5 for the best). Best% is the ratio of the best score (5) in the Naturalness and Difficulty modalities. Avg. rank is the average ranking of the three question headlines. Two-tailed t-tests demonstrate the improvements of our model compared to $\text{Seq2Seq}_{hie+att}$ are statistically significant (‡ indicates p-value < 0.01).

	Naturalness	Difficulty	Best%	Avg. rank
$\text{Seq2Seq}_{hie+att}$	3.41	2.57	20.30	2.70
DASeq2Seq_{dist}	**3.72**	**3.01**‡	**32.25**‡	**2.19**‡
Human	4.89	4.72	84.37	1.11

5.5 Results and Analysis

Table 3 shows the automatic evaluation results for all the models. We can observe that: (1) IR_{BM25} performs poorly, indicating that memorizing the training set is not sufficient for the QHG task. (2) The *PREFIX* model performs pretty well, showing that the first sentence of a news article is often a good summary. However, although the ROUGE score is good for the *PREFIX* model, usually the model will not produce a question headline. (3) By building a Seq2Seq model based on the lead sentence of an article, the *ABS* model can learn to generate question headline and achieve much better results than the *PREFIX* model. (4) *Distraction* also perform well by distracting the model to different input content to better grasp the overall meaning of input articles.

When we look at the three variants of our model, we find that: (1) The performance of Seq2Seq_{flat} is relatively poor, indicating that encoding the semantics of an article through long time steps is difficult. This baseline is more appropriate for encoding short

[6]https://github.com/facebookarchive/NAMAS
[7]https://github.com/lukecq1231/nats

Table 5: An example from the test QHG data. G is the true headline. H is the output of the Seq2Seq$_{hie+att}$ model. D is the output of our DASeq2Seq$_{dist}$ model. S1 to S16 are the sentences in the article.

S1: 常听人说补铁要多吃菠菜,这种说法不完全正确，但也有一定的科学道理。
We've all heard that it is necessary to eat more spinach for iron supplement. That's not entirely true, but makes some sense theoretically.
S2 . . . **S5** . . .
S6: 菠菜中的铁含量在蔬菜中是比较高的，达#<UNK>/#<UNK>（跟瘦猪肉差不多，#<UNK>/#<UNK>），但是植物性食物中的铁是以非血红素铁的形式存在，其吸收明显受到草酸、植酸、膳食纤维、多酚类物质等膳食因素的抑制。
The iron in spinach is highest in vegetables, which achieves #<UNK>/#<UNK> (it is similar to lean pork, #<UNK>/#<UNK>). However, the iron in plant-based foods is in the form of non-haem iron and the absorption is controlled by oxalic acid, phytic acid, dietary fiber, polyphenol, and so on.
. . . **S7** . . . **S9** . . .
S10: 菠菜中的草酸不但干扰菠菜中铁的吸收，甚至还会干扰其他食物中非血红素铁的吸收。
The oxalic acid in spinach can disturb not only the absorption of iron, but also the absorption of non-haem iron in other foods.
S11: 所以吃菠菜非但不能补铁，反而有可能加重缺铁。
Therefore, spinach cannot help supplement iron, but may even increase the risk of lossing iron.
. . . **S12** . . . **S16**
G: 多吃菠菜可以补铁吗？Can eating more spinach help supplement iron?
H: 吃什么补铁最好？What vegetables can help supplement iron?
D: 多吃菠菜真的能补铁吗？Can eating more spinach help supplement iron?

text. (2) The results of Seq2Seq$_{flat+att}$ show that word-level attention in the decoding phase is effective and can improve the performance significantly. (3) By introducing a hierarchical structure in the encoder and using sentence-level attention in the decoder, the Seq2Seq$_{hie+att}$ model is able to achieve the best performance among all the baseline methods.

Finally, we find that our DASeq2Seq model can outperform all the baseline methods significantly. The better results of our models over Seq2Seq$_{hie+att}$ demonstrate the effectiveness of the dual-attention and vocabulary gating mechanisms, which can identify the importance of different sentence for better question generation. Among the two of our models, DASeq2Seq$_{dist}$ performs better than DASeq2Seq$_{global}$, indicating that "voting" by all the other sentences is more valid than "deciding" by some initial global representation of the article. Moreover, the improvement of DASeq2Seq with VoG over that without VoG suggests that the vocabulary gating mechanism does help in modeling question patterns. The improvement of DASeq2Seq without VoG over Seq2Seq$_{hie+att}$ also suggests that the self-attention mechanism can help obtain a better representation of the articel for question generation.

Table 4 shows the results of the human evaluation. We can see that our DASeq2Seq$_{dist}$ outperforms the best performing baseline Seq2Seq$_{hie+att}$ in all modalities. The results imply that our model

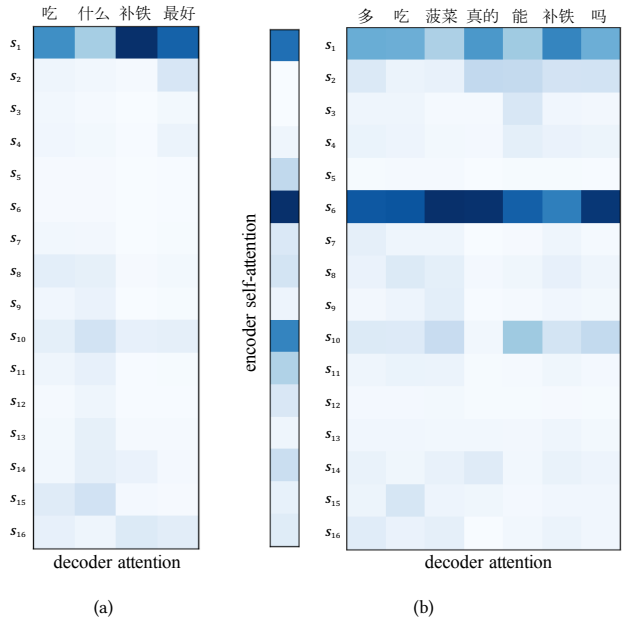


Figure 3: (a) is the heatmap of the sentence-level decoder attention weight matrix for the example in Table 5, generated by Seq2Seq$_{hie+att}$. (b) is the heatmap of the sentence-level encoder self-attention and decoder attention weight matrix for the example in Table 5, generated by DASeq2Seq$_{dist}$. Darker color indicates higher weight.

can generate fluent and grammatically correct questions (i.e., Naturalness) which better summarize the major topics of the input news article (i.e., Difficulty) than the baseline Seq2Seq$_{hie+att}$.

5.6 Case Study

To better understand what can be learned by our model, we conduct some case studies. We take one news article from the test data as an example. As shown in Table 5, this article talks about the reason why spinach cannot help supplement iron for human beings, with a question headline "多吃菠菜可以补铁吗？(Can eating more spinach help supplement iron?)". There are 16 sentences distributed over 6 paragraphs in this article, and due to the limited space, we only show some key sentences. We show the generated question headline from our model as well as that from the best baseline model Seq2Seq$_{hie+att}$. Meanwhile, we also depict the learned decoder attention weights over sentences of Seq2Seq$_{hie+att}$ in Figure 3(a), and the learned encoder and decoder attention weights of DASeq2Seq$_{dist}$ in Figure 3(b) to help analysis.

As we can see, when generating the question headline, the Seq2Seq$_{hie+att}$ pays too much attention to the lead sentence while ignores most of the rest. The lead sentence further makes the decoder focus on "补铁(supplement iron)" and generate a question about what vegetables can help supplement iron. On the contrary, by using self-attention in the encoding phase, our model finds that the most informative sentences are **S6**, **S1**, and **S10**. This in turn guides the decoder to pay attention to these informative sentences and generate a much better question headline which is more consistent with the ground-truth. Note here we also run some well-known

Table 6: Automatic evaluation results of different models by ROUGE, BLEU and METEOR on the New York Times (NYT) Annotated corpus. Two-tailed t-tests demonstrate the improvements of our models to all the baseline models are statistically significant (‡ indicates p-value < 0.01).

Model	ROUGE-1	ROUGE-2	ROUGE-L	BLEU-1	BLEU-2	BLEU-3	METEOR
PREFIX	11.85	5.29	10.50	5.64	2.11	0.72	7.95
ABS	28.29	15.49	27.35	23.74	16.94	13.58	16.41
Seq2Seq$_{hie+att}$	33.98	17.15	32.74	27.86	19.04	14.38	21.89
DASeq2Seq$_{glob}$ (no VoG)	35.15‡	17.70‡	33.64‡	28.94‡	20.18‡	15.46‡	22.73‡
DASeq2Seq$_{dist}$ (no VoG)	**35.24**‡	**17.81**‡	**33.75**‡	**29.48**‡	**20.41**‡	**15.71**‡	**22.89**‡

extractive summarization methods, namely Luhn [28], TextRank [32] and LexRank [16], on this example article. The three most important sentences they extract are [**S6**, **S9**, **S10**], [**S6**, **S7**, **S10**], and [**S6**, **S11**, **S12**] respectively. This also verifies to some extent that our model with self-attention mechanism can identity the key sentences in the input article.

5.7 Evaluation on Headline Generation

The previous experiments have demonstrated the effectiveness of our model on the QHG task. A natural question is whether the proposed model can perform equally well on headline generation in non-question form, which can be well evaluated on some publicly available dataset. In this section, we apply the DASeq2Seq model on the general headline generation task by simply removing the vocabulary gating scheme, and verify its effectiveness on a public benchmark collection, i.e., the New York Times (NYT) Annotated corpus[8].

The corpus contains over 1.8 million articles published by the New York Times between January 1, 1987 and June 19, 2007. Most headlines are in non-question form, with the average length of 6.6 words. We leave out the articles whose headlines have less than 3 words or more than 15 words, and whose articles have less than 20 words or more that 2000 words, reducing the corpus to 1.58 million articles. We randomly sample 2000 articles to form the development and test set respectively, and the other articles are used as the training data. We keep the 60,000 most frequently occurring words and other words are replaced with the <UNK> symbol. We use pre-trained GloVe vectors[9] for the initialization of word embeddings and the word embeddings will be further fine-tuned during training.

Results on the general headline generation are shown in Table 6. As we can see, the relative order of different models on this task is quite consistent with that on the previous QHG task. By using the dual-attention mechanism, our DASeq2Seq models can significantly outperform the state-of-the-art headline generation baselines on the public benchmark collection.

6 ONLINE ANALYSIS

Beyond the offline experiments, we further conduct online evaluation to verify whether the news articles with our generated question headlines can attract users' attention and receive higher click-through ratio. Specifically, we use the online A/B Test. We randomly sampled 200 real-time news articles in one day, whose original headlines are in non-question form. We then generated question headlines for these news articles using our DASeq2Seq$_{dist}$ model. We delivered these news articles with either question or non-question headlines randomly to different real users in our mobile news portal, and collected the online user clicks in the following week. Table 7 shows the statistics of the collected online user click performance.

Table 7: Online user click performance in the A/B Test. Each news article has two headlines: question headline and non-question headline.

	question headline	non-question headline
Total user impression	216,963	205,879
Total user click	30,952	20,273
Avg click-through ratio	7.98%	6.02%

We can find that question headlines do draw more clicks than non-question headlines for the same news articles. The question headlines improve the click-through ratio over non-question headlines by around 32.56%. From the perfomance gap on absolute user clicks, we can see that question headlines do be worth thousands or even tens of thousands clicks.

7 CONCLUSION AND FUTURE WORK

In this paper we introduced a challenging task to automatically generate question headlines for news articles. To tackle this problem, we developed a novel DASeq2Seq model with a dual-attention mechanism and a vocabulary gating scheme, which can better capture the major topics of the original news article for question generation. We considered two ways of the self-attention mechanism, namely the global self-attention and the distributed self-attention. The offline experimental results demonstrated that our model can outperform all the state-of-the-art baselines on the QHG task significantly. The online evaluation verifies the effectiveness of our model on improving the click-through ratio by generating question headlines for news articles.

As we know, the question headline is only one specific form of the catchy headlines. There could be many other forms, such as using humorous words, showing unique rationale or making an exaggerated claim. In the future work, we would like to extend our model to produce catchy headlines in those other forms.

[8] https://catalog.ldc.upenn.edu/LDC2008T19
[9] http://nlp.stanford.edu/projects/glove

8 ACKNOWLEDGMENTS

This work was funded by the 973 Program of China under Grant No. 2014CB340401, the National Natural Science Foundation of China (NSFC) under Grants No. 61425016, 61472401, 61722211, 61872338 and 20180290, the Youth Innovation Promotion Association CAS under Grants No. 20144310, and 2016102, and the National Key R&D Program of China under Grants No. 2016QY02D0405.

REFERENCES

[1] Michael Alley, Madeline Schreiber, Katrina Ramsdell, and John Muffo. 2006. How the design of headlines in presentation slides affects audience retention. *Technical communication* 53, 2 (2006), 225–234.
[2] Shiqi Shen Ayana, Zhiyuan Liu, and Maosong Sun. 2016. Neural headline generation with minimum risk training. *arXiv preprint arXiv:1604.01904* (2016).
[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.
[4] Michele Banko, Vibhu O Mittal, and Michael J Witbrock. 2000. Headline generation based on statistical translation. In *ACL*. Association for Computational Linguistics, 318–325.
[5] Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*. ACM, 335–336.
[6] Abhijnan Chakraborty, Bhargavi Paranjape, Sourya Kakarla, and Niloy Ganguly. 2016. Stop clickbait: Detecting and preventing clickbaits in online news media. In *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on*. IEEE, 9–16.
[7] Yllias Chali and Sina Golestanirad. 2016. Ranking Automatically Generated Questions Using Common Human Queries.. In *INLG*. 217–221.
[8] Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, and Hui Jiang. 2016. Distraction-Based Neural Networks for Document Summarization. In *IJCAI*.
[9] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*.
[10] Sumit Chopra, Michael Auli, Alexander M Rush, and SEAS Harvard. 2016. Abstractive Sentence Summarization with Attentive Recurrent Neural Networks. In *HLT-NAACL*. 93–98.
[11] Carlos A Colmenares, Marina Litvak, Amin Mantrach, and Fabrizio Silvestri. 2015. HEADS: Headline Generation as Sequence Prediction Using an Abstract Feature-Rich Space.. In *HLT-NAACL*. 133–142.
[12] Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*. 376–380.
[13] Bonnie Dorr, David Zajic, and Richard Schwartz. 2003. Hedge trimmer: A parse-and-trim approach to headline generation. In *HLT-NAACL*. Association for Computational Linguistics, 1–8.
[14] Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to Ask: Neural Question Generation for Reading Comprehension. In *ACL*.
[15] HP Edmundson. 1964. Problems in automatic abstracting. *Commun. ACM* 7, 4 (1964), 259–263.
[16] Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *JAIR* 22 (2004), 457–479.
[17] Michael Heilman and Noah A Smith. 2010. Good question! statistical ranking for question generation. In *HLT-NAACL*. 609–617.
[18] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
[19] Hongyan Jing and Kathleen R McKeown. 1999. The decomposition of human-written summary sentences. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 129–136.
[20] Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.
[21] Girish Kumar, Rafael E Banchs, and Luis Fernando D'Haro Enriquez. 2015. Revup: Automatic gap-fill question generation from educational texts.
[22] Linda Lai and Audun Farbrot. 2014. What makes you click? The effect of question headlines on readership in computer-mediated communication. *Social Influence* 9, 4 (2014), 289–299.
[23] Paul LaRocque. 2003. *Heads You Win: An Easy Guide to Better Headline and Caption Writing*. Marion Street Press, Inc.
[24] Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. 2015. A hierarchical neural autoencoder for paragraphs and documents. In *ACL*.
[25] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, Vol. 8. Barcelona, Spain.
[26] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In *ICLR*.
[27] David Lindberg, Fred Popowich, John Nesbit, and Phil Winne. 2013. Generating natural language questions to support learning on-line. (2013).
[28] Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development* 2, 2 (1958), 159–165.
[29] Prashanth Mannem, Rashmi Prasad, and Aravind Joshi. 2010. Question generation from paragraphs at UPenn: QGSTEC system description. In *Proceedings of QG2010*. 84–91.
[30] Betty A Mathis, James E Rush, and Carol E Young. 1973. Improvement of automatic abstracts by the use of structural analysis. *Journal of the Association for Information Science and Technology* 24, 2 (1973), 101–109.
[31] Gerardo Atienza Merino and Leonor Varela Lema. 2008. Needs and demands of policy-makers. *HEALTH TECHNOLOGY ASSESSMENT AND HEALTH POLICY-MAKING IN EUROPE* (2008), 137.
[32] Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order into Text.. In *EMNLP*, Vol. 4. 404–411.
[33] Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Margaret Mitchell, Xiaodong He, and Lucy Vanderwende. 2016. Generating natural questions about an image. In *ACL*.
[34] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents. In *AAAI*.
[35] Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023* (2016).
[36] Juntiga Nasunee. 2004. An analysis of catchy words and sentences in Volkswagen beetle advertisements in the United States. *Unpublished master' s project. Srinakharinwirot University* (2004).
[37] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank citation ranking: Bringing order to the web*. Technical Report. Stanford InfoLab.
[38] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*. ACL, 311–318.
[39] Daniele Pighin, Marco Cornolti, Enrique Alfonseca, and Katja Filippova. 2014. Modelling Events through Memory-based, Open-IE Patterns for Abstractive Summarization.. In *ACL*. 892–901.
[40] David Lindberg Fred Popowich and John Nesbit Phil Winne. 2013. Generating Natural Language Questions to Support Learning On-Line. *ENLG* (2013), 105.
[41] Dragomir R Radev and Kathleen R McKeown. 1998. Generating natural language summaries from multiple on-line sources. *Computational Linguistics* 24, 3 (1998), 470–500.
[42] Stephen E Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR*. Springer-Verlag New York, Inc., 232–241.
[43] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. 1988. Learning representations by back-propagating errors. *Cognitive modeling* 5, 3 (1988), 1.
[44] Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Cristian Moldovan. 2010. The first question generation shared task evaluation challenge. In *INLG*.
[45] Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *EMNLP*.
[46] Gerard Salton, Amit Singhal, Mandar Mitra, and Chris Buckley. 1997. Automatic text structuring and summarization. *Information Processing & Management* 33, 2 (1997), 193–207.
[47] Iulian Vlad Serban, Alberto García-Durán, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, and Yoshua Bengio. 2016. Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus. In *ACL*.
[48] Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2018. Disan: Directional self-attention network for rnn/cnn-free language understanding. In *AAAI*.
[49] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*. 3104–3112.
[50] Sho Takase, Jun Suzuki, Naoaki Okazaki, Tsutomu Hirao, and Masaaki Nagata. 2016. Neural Headline Generation on Abstract Meaning Representation.. In *EMNLP*. 1054–1059.
[51] Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. From Neural Sentence Summarization to Headline Generation: A Coarse-to-Fine Approach. In *IJCAI*.
[52] Kristian Woodsend, Yansong Feng, and Mirella Lapata. 2010. Generation with quasi-synchronous grammar. In *EMNLP*. Association for Computational Linguistics, 513–523.
[53] Songhua Xu, Shaohui Yang, and Francis Chi-Moon Lau. 2010. Keyword Extraction and Headline Generation Using Novel Word Features.. In *AAAI*. 1461–1466.
[54] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J Smola, and Eduard H Hovy. 2016. Hierarchical Attention Networks for Document Classification.. In *HLT-NAACL*. 1480–1489.