SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

August 2020, Vol. 63 182103:1–182103:16
https://doi.org/10.1007/s11432-019-2657-8

Topic-sensitive neural headline generation

Ayana[1,2], Ziyun WANG[1], Lei XU[1], Zhiyuan LIU[1*] & Maosong SUN[1]

[1] *State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, Beijing* 100084*, China;*
[2] *Department of Computer Information Management, Inner Mongolia University of Finance and Economics, Hohhot* 010070*, China*

Received 12 March 2019/Revised 9 August 2019 /Accepted 25 September 2019/Published online 15 July 2020

Abstract Neural models are being widely applied for text summarization, including headline generation, and are typically trained using a set of document-headline pairs. In a large document set, documents can usually be grouped into various topics, and documents within a certain topic may exhibit specific summarization patterns. Most existing neural models, however, have not taken the topic information of documents into consideration. This paper categorizes documents into multiple topics, since documents within the same topic have similar content and share similar summarization patterns. By taking advantage of document topic information, this study proposes a topic-sensitive neural headline generation model (TopicNHG). It is evaluated on a real-world dataset, large scale Chinese short text summarization dataset. Experimental results show that it outperforms several baseline systems on each topic and achieves comparable performance with the state-of-the-art system. This indicates that TopicNHG can generate more accurate headlines guided by document topics.

Keywords neural networks, sequence to sequence learning, topic, LDA, headline generation

Citation Ayana, Wang Z Y, Xu L, et al. Topic-sensitive neural headline generation. Sci China Inf Sci, 2020, 63(8): 182103, https://doi.org/10.1007/s11432-019-2657-8

1 Introduction

Text summarization, including headline generation, is an important task in natural language processing, which aims to capture essential information in a document and create an informative but brief document summary.

Most existing text summarization approaches can be divided into two categories: extractive and generative. Extractive summarization [1] simply selects a few sentences from a given document to compose a compact summary. Owing to the limitation of vocabulary and sentence structure, it is extremely difficult for extractive models to generate coherent and concise summaries. Generative summarization, on the other hand, aims at comprehending a document and generating the summary that has not necessarily appeared in the original document.

Recent years have witnessed the development of sequence-to-sequence (seq2seq) neural models [2] for natural language processing. These models typically learn distributed representations of an input sequence, and then generate an output sequence accordingly. The advantage of neural models is that they can efficiently learn a semantic mapping directly from pairs of text sequences without designing hand-crafted features.

Neural models have also shown great superiority for text summarization [3–5] and headline generation [6–9] because they can flexibly model document semantics from internal word sequences within the

[*] Corresponding author (email: liuzy@tsinghua.edu.cn)

Table 1 Examples of headlines from different topics

Topic	Headline
Finance	调查: 74% 美国经济专家认为美经济会出现衰退 (Investigation: 74% of US economic experts believe that the US economy will experience a recession)
	易尚展示尾盘炸板成交额超 7 亿, 8 月以来涨幅已超 50% (Yishang's turnover exceeded 700 million, and the increase since August has exceeded 50%)
Politics	中美经贸高级别磋商双方牵头人通话 (Leaders of China-US High-level Economic and Trade Consultation made conversation)
	新一轮中日战略对话时隔 7 年重启 (The new round of China-Japan strategic dialogue restarts after 7 years)
Legal	长沙警方侦破 “卖茶女” 特大电信网络诈骗案 (Changsha police detect “tea woman” mega telecom network fraud case)
	内蒙古侦破特大跨境开设网络赌场案 (Inner Mongolia detects a large cross-border network casino case)

document. Nevertheless, many document-level patterns may also play important roles in text summarization. Specifically, documents can usually be grouped into various topics, and documents within a certain topic are inevitably composed of topically related words or concepts and may exhibit specific summarization patterns. Table 1 shows some examples from the large scale Chinese short text summarization dataset (LCSTS) [10]. A headline from the topic finance usually includes words or concepts which are utilized to describe financial issues, and also some digital information. A headline from the topic politics often contains names of government agencies or politicians, whereas a headline from the topic legal commonly uses words which are utilized to describe criminal cases. Except for these explicit word patterns, there are also some implicit patterns. For example, a document about the finance is usually summarized including cause and effect, whereas a document about politics or legal is always summarized containing event location and result.

This paper aims to incorporate document topic information into neural models for text summarization and proposes topic-sensitive neural headline generation (TopicNHG). More specifically, the proposed model is designed as a mixed local expert [11]. Inputs are divided according to topics assigned using latent dirichlet allocation (LDA) [12], and unique “expert” networks are trained for each topic, respectively. In this way, TopicNHG can effectively identify the corresponding crucial parts in a document guided by its topic information and is expected to generate well-focused headlines.

The model is evaluated on large scale Chinese short text summarization dataset (LCSTS) [10]. Experimental results show that this model significantly outperforms most baseline systems. Moreover, it consistently performs much better than the plain baseline system on each individual topic, which verifies the significance and robustness of TopicNHG. The novelty of this study lies in the fact that our method can integrate topic information in headline generation while keeping the overall framework simple. Generally, the main contributions of the study are as follows.

- We present a topic-sensitive neural headline generation model to incorporate the topic information as prior knowledge to ensure the generated headlines more concise and dedicated to the main theme of the original document.

- In order to enable the model to better handle documents with different topics, we first utilize a “gating” model to assign a topic for each news article, and then generate a more focused headline for the news article with the corresponding “expert” network. This method can make full consideration of the features within different topics and improve the overall performance.

- We evaluate our model on Chinese headline generation task. To make sure the evaluation results more reliable, we conduct two manual evaluation methods in addition to the automatic evaluation metric. The experimental results show that our method is not only effective but also more interpretable.

The rest of this paper is organized as follows. In Section 2, the theoretical background of our work is described. In Section 3, the details of the TopicNHG are introduced. In Section 4, the dataset, implementation details, and experimental results are provided. In Section 5, related work of the neural

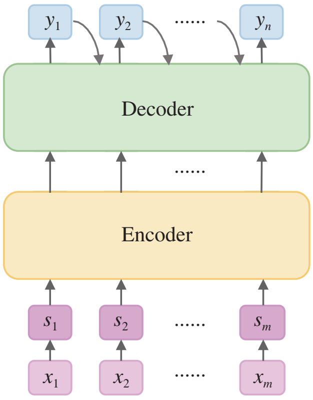

Figure 1 (Color online) NHG model architecture.

headline generation models is presented. Finally in Section 6, conclusion of this study is elaborated and the future work is revealed.

2 Background

2.1 Neural headline generation model

An NHG (neural headline generation) model aims at teaching a neural model to map a short text (the length of 60–140 words) into a headline. Given an source document $\boldsymbol{x} = (x_1, x_2, \ldots, x_m)$, the NHG model takes $\boldsymbol{x}$ as input, and generates a target headline $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$ word by word whose length is $n < m$. Each source input word x_i and target output word y_j come from fixed source vocabulary $\boldsymbol{V}_x$ and target vocabulary $\boldsymbol{V}_y$ respectively. Generation probability can be formalized as

$$\Pr(\boldsymbol{y}|\boldsymbol{x};\theta) = \prod_{j=1}^{N} \Pr(\boldsymbol{y}_j|\boldsymbol{x}, \boldsymbol{y}_{<j};\theta), \tag{1}$$

where θ represents model parameters and $\boldsymbol{y}_{<j} = y_1, \ldots, y_{j-1}$ is a partial headline.

An NHG model is generally composed of three parts: an input representor that transforms each input word into a vector representation, an encoder that obtains either one single vector or several vectors to represent the original input document, and a decoder that generates each output word one by one. Figure 1 shows the general framework of the NHG model.

An input article consists of discrete words: the input representor is adopted to project each input article word x_i into its vector representation,

$$s_i = \mathrm{emb}(x_i), \tag{2}$$

where $\mathrm{emb}(\cdot)$ is the function to obtain vector representation. The simplest way is one-hot representation, which not only wastes capacity but also lacks representational ability. Word embeddings are real-valued low-dimensional vectors, which are also able to capture more semantic and syntactic relationships between words. There are also several efforts to combine other features into input representation, for example, word positions, part-of-speech tags, named-entity tags, and term frequency-inverse document frequency (TF-IDF) statistics [13].

The encoder is utilized to encode input word representations into a single vector or a sequence of encoder side hidden vectors,

$$\boldsymbol{H} = \mathrm{enc}(\boldsymbol{S}), \tag{3}$$

where $\mathrm{enc}(\cdot)$ indicates the encoder, and $\boldsymbol{S} = (s_1, s_2, \ldots, s_m)$. Common choices are bag-of-words encoders, convolutional neural network (CNN) encoders, recurrent neural network (RNN) encoders, and bi-directional RNN (BRNN) encoders. The bag-of-words encoder averages input word embeddings into

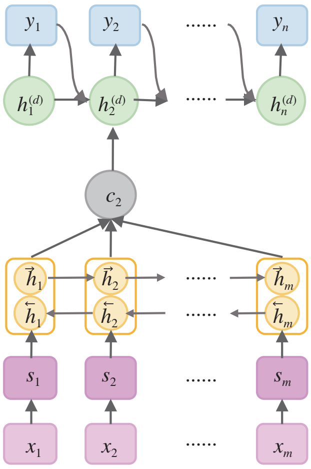

Figure 2 (Color online) The attention based NHG model architecture.

a single vector, which makes it lose the ability to consider word order information. The CNN encoder overcomes the shortcoming mentioned above by applying a convolutional layer onto word embeddings. The CNN encoder, however, remains unable to capture long-term dependencies between input words. Although the RNN encoder is able to model sequential information and is better at processing sequences with arbitrary lengths, a plain RNN encoder usually suffers from gradient vanishing or exploding problems. For this reason, variants of RNN, such as gated recurrent unit (GRU) [14] or long short term memory (LSTM), are often adopted. In a sequence, the meaning of a word is determined not only by the following words but also by the preceding ones. The Bi-RNN encoder is introduced to consider information from both forward and backward directions.

The decoder generates target headline words one by one depending on encoder outputs and the previous decoder output word,

$$y_j = \mathrm{dec}(\boldsymbol{H}, y_{j-1}), \tag{4}$$

where $\mathrm{dec}(\cdot)$ represents the decoder, y_{j-1} indicates the word generated by the decoder in the step before y_j, and $\boldsymbol{H}$ denotes encoder side hidden states. Rush et al. [6] utilized neural network language model [15] as the decoder. Various variants of RNN can also be adopted as the decoder. When generating an output word, input words usually make different contributions. The attention mechanism can model the procedure and is introduced in many natural language generation tasks, including NHG. Figure 2 shows the attention-based NHG model architecture.

2.2 LDA

A topic model tries to identify the word usage pattern in a document and attempts to assign semantic meaning to vocabulary. Topic models assume that a document is composed when words are selected from possible sets of words where each set corresponds to a topic. Though the topic model and LDA are often synonymously used, LDA is a special case of topic model introduced by Blei et al. [12]. It will be briefly introduced here.

Let $\boldsymbol{D} = \{d_1, d_2, \ldots, d_{N_d}\}$ denote a collection of N_d documents, $\boldsymbol{V} = \{w_1, w_2, \ldots, w_V\}$ indicate a fix-sized vocabulary with V words, and $z_{i,j}$ represent the topic for the j-th word in document d_i. LDA is a generative model, and it assumes that the j-th word of the i-th document is generated by taking the following steps.

(1) Choose document topic distribution θ_i for document d_i according to a Dirichlet prior which is specified with hyperparameter α.

(2) Choose topic $z_{i,j}$ for the j-th word in document d_i from Multinomial(θ_i).

(3) Given topic $z_{i,j}$, choose word distribution $\phi_{i,j}$ according to a Dirichlet prior which is specified with hyperparameter β.

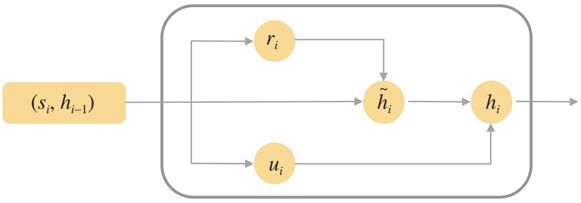

Figure 3 (Color online) Gated recurrent unit (GRU).

(4) Choose word $w_{i,j}$ from Multinomial($\phi_{i,j}$).

Gibbs sampling is adopted to teach the LDA model, and the procedure is as follows.

(1) For each word of a document in the whole corpus, randomly assign a topic x.

(2) Re-scan the corpus, and take Gibbs sampling to obtain the word topic.

(3) Repeat Step (2) till Gibbs sampling is convergent.

(4) Calculate the topic-word co-occurrence matrix.

3 Topic-sensitive NHG

Documents of similar topics may appear with similarly styled headlines. Documents can thus be naturally divided into several subsets according to their topics, and the summarization task is also divided into several subtasks. Inspired by the idea of topic-sensitive summarization, this study develops a system composed of several independent "expert" networks along with a gating model that decides which document should be handled by which "expert". The system is named TopicNHG, and it utilizes topic information of documents while keeping the overall framework simple. The study uses the LDA model [12] as the gating model and trains the conventional NHG model for individual topic "expert" networks.

3.1 Expert network encoder

Since the RNN is better at processing sequential information and variants of plain RNN can tackle the issue of gradient vanishing and exploding problems, one of the RNN variants is adopted here as an encoder. Specifically, the GRU-RNN, which is originally proposed for neural machine translation (NMT), is used as the encoder. As shown in Figure 3, the GRU is equipped with the update gate u_i and the reset gate r_i to adaptively capture dependencies of the input sequence. The reset gate determines whether to ignore previous hidden states, and the update gate controls how many of the previous hidden states will be passed on. The GRU updates the i-th hidden state as follows:

$$\begin{aligned} r_i &= \sigma\big(W_r s_i + U_r h_{i-1}\big), \\ u_i &= \sigma\big(W_u s_i + U_u h_{i-1}\big), \\ \tilde{h}_i &= \tanh\big(W_h s_i + U_h(r_i \odot h_{i-1})\big), \\ h_i &= u_i \odot h_{i-1} + (1 - u_i) \odot \tilde{h}_i, \end{aligned}$$

where the h_i and $\tilde{h}_i$ are generated hidden state and candidate activation. $\sigma(\cdot)$ is the sigmoid function. $\odot$ indicates element-wise multiplication. $W_r, W_u, W_h \in \mathbb{R}^{H\times D}$ and $U_r, U_u, U_h \in \mathbb{R}^{H\times H}$ are weighting matrices and D and H denote the dimensions of word embeddings and hidden states respectively.

By reading in input words from start to end, the obtained hidden state only considers its preceding words. However, the exact meaning of the word is influenced by its context. Hence, BRNN [16] is used to obtain hidden states considering not only the preceding words but also the following ones. BRNN processes the input document in with two separate GRU-RNNs, one in forward direction and one in backward direction in order to obtain forward hidden states $(\overrightarrow{h}_1, \ldots, \overrightarrow{h}_m)$ and backward hidden states $(\overleftarrow{h}_1, \ldots, \overleftarrow{h}_m)$. Then for each position i, the final hidden state is the concatenation of its forward and backward hidden states: $h_i = \overrightarrow{h}_i \oplus \overleftarrow{h}_i$, whereby operator $\oplus$ indicates concatenation.

3.2 Expert network decoder

Headline words are generated according to (4). Because the model performs better when the decoder adopts the same RNN variant as the encoder, and an attention mechanism [17] can bring significant improvement to the model [18], the attention-based GRU-RNN is adopted as the decoder. The decoder calculates the j-th headline word as follows:

$$\begin{aligned}
r_j &= \sigma\big(W_r y_{j-1} + U_r h_{j-1}^{(d)} + C_r c_j\big),\\
u_j &= \sigma\big(W_u y_{j-1} + U_u h_{j-1}^{(d)} + C_u c_j\big),\\
\tilde{h}_j^{(d)} &= \tanh\big(W_h y_{j-1} + U_h (r_j \odot h_{j-1}^{(d)}) + C_h c_j\big),\\
h_j^{(d)} &= u_j \odot h_{j-1}^{(d)} + (1-u_j) \odot \tilde{h}_j^{(d)},
\end{aligned}$$

where the $C_r, C_u, C_h \in \mathbb{R}^{H\times 2H}$ are weighting matrices and the rest of the notations are identical to those of GRU-RNN in the encoder. The first decoder hidden state is set as $h_0^{(d)} = \tanh(W_f \overleftarrow{h}_1)$ with $W_f \in \mathbb{R}^{H\times H}$. Combining the preterite generated headline words, the context vector c_j contains attention information of the input document,

$$c_j = \sum_{i=1}^{m} \alpha_{ji} h_i, \tag{5}$$

where h_i is the encoder hidden state, α_{ji} indicates the contribution weight of the i-th source word $\mathbf{x_i}$ when generating the j-th output word. α_{ji} is computed as follows:

$$\begin{aligned}
\alpha_{ji} &= \frac{\exp(e_{ji})}{\sum_{k=1}^{m} \exp(e_{jk})},\\
\boldsymbol{e}_{ji} &= \rho(h_{j-1}^{(d)}, h_i),
\end{aligned}$$

where $\rho(\cdot)$ is a scoring function.

3.3 Expert network training

The model parameters of NHG can be estimated using large-scale document-headline pairs. Given the training set $D = \{(\boldsymbol{x}^{(1)}, \boldsymbol{y}^{(1)}), \ldots, (\boldsymbol{x}^{(T)}, \boldsymbol{y}^{(T)})\}$, the training objective is to maximize the log-likelihood over the training set D,

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \big\{\mathcal{L}(\theta)\big\}, \tag{6}$$

where

$$\begin{aligned}
\mathcal{L}(\theta) &= \sum_{t=1}^{T} \log \Pr(\boldsymbol{y}^{(t)} | \boldsymbol{x}^{(t)}; \theta)\\
&= \sum_{t=1}^{T} \sum_{j=1}^{n^{(t)}} \log \Pr(y_j^{(t)} | \boldsymbol{x}^{(t)}, \boldsymbol{y}_{<j}^{(t)}; \theta),
\end{aligned}$$

where $(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)})$ indicates the t-th training pair of training set D, $n^{(t)}$ denotes the length of the t-th target headline $\boldsymbol{y}^{(t)}$. This objective can be achieved by minimizing the cross-entropy loss at each decoding step.

3.4 Topic assignment

LDA [12] is used to identify the latent topic distribution feature of a document. Given a document set $D = \{d_1, d_2, \ldots, d_n\}$, LDA makes a maximum likelihood estimation on

$$\Pr(D|\alpha, \beta) = \prod_{d\in D} \Pr(d|\alpha, \beta), \tag{7}$$

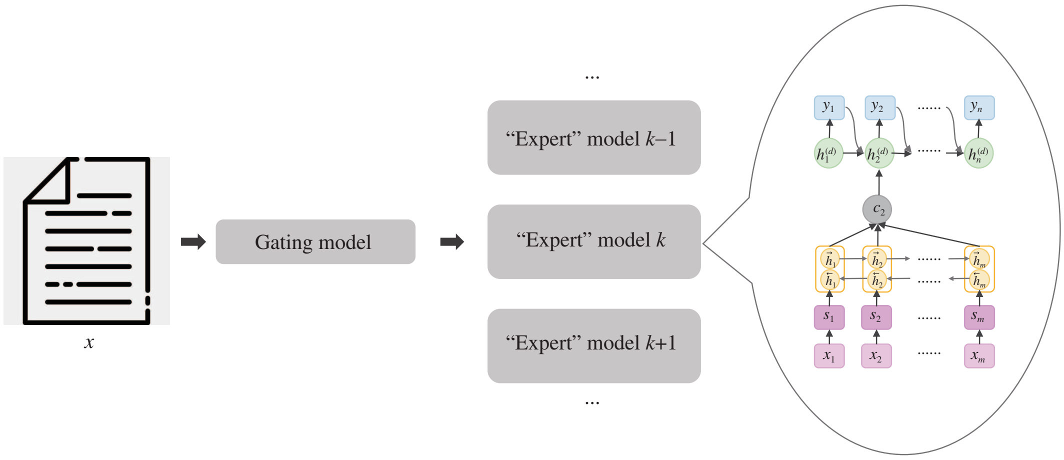

Figure 4 (Color online) Topic-sensitive neural headline generation model. A gating model will perform topic assignment and select the most appropriate local “expert” network.

where α, β are the Dirichlet priors on per-document topic distribution and per-topic word distribution respectively. Afterward, $\Pr(t|\alpha, \beta, d)$ can be used to infer the topic of the document. According to the observation of the dataset LCSTS, a single topic can, in most cases, sufficiently reflect the main theme of a short text. Hence, for the sake of simplicity, only one topic for each input sequence is selected $l = \operatorname{argmax}_t \Pr(t|\alpha, \beta, d)$. The process also serves as the gating model [11] to find an appropriate local expert for topic-sensitive headline generation.

3.5 TopicNHG model

Given an input $\mathbf{x}$, the gating model first determines which “expert” network should perform NHG; a normal training/inference process is then performed on that specific local “expert” network with $\mathbf{x}$. The entire framework of the TopicNHG model is shown in Figure 4. For a short text $\mathbf{x}$, summary $\mathbf{y}$, and topic label l, the TopicNHG model aims to maximize the likelihood,

$$\Pr(\mathbf{y}, l|\mathbf{x}, \theta, \theta_t) = \Pr(l|\mathbf{x}, \theta_t) \Pr(\mathbf{y}|\mathbf{x}, l, \theta), \tag{8}$$

where θ_t represents the parameters of the LDA model, and θ represents the parameters of expert seq2seq models. k ($k >= 1$) different “expert” networks have the same structure but different parameter settings (i.e., k different encoders, decoders, and attention layers). Though the scale of our model is k times to a conventional seq2seq model, the actual training time is far less than k times in comparison to training the seq2seq model, because the training set is divided into k parts, with each “expert” network containing fewer training samples, and leading to a significant reduction in the number of iterations needed for training.

4 Experiments

This section introduces the experimental setup for model performance assessment, particularly datasets, implementation details, evaluation protocol, and baseline systems used for comparison.

4.1 Dataset

Experiments are conducted on the LCSTS [10] to evaluate the performance of TopicNHG. LCSTS consists of short news articles with headlines from Sina Weibo[1], a Chinese microblog service. LCSTS collects news messages from verified organizations such as China Daily to guarantee data quality. LCSTS contains 3

[1] The website is http://weibo.com

Table 2 art.avg.tok and head.avg.tok refer to average token numbers of articles and headlines on PART-I respectively

LCSTS	PART-I	PART-II	PART-III	art.avg.tok	head.avg.tok
	2400591	10666	1106	103.68	17.86

parts, and each document-headline pair in PART-II and PART-III has a human-labeled score, indicating the relevance between the news article and the summary. Scores range from 1 to 5, whereby 1 means "least relevant", and 5 means "most relevant". Data in PART-II are labeled by 1 annotator, and data in PART-III are labeled by 3 annotators. In the experiment, PART-I is used as training data, and PART-III with scores higher than or equal to 3 as test data. Table 2 shows the data statistics.

4.2 Experimental settings

Word-level LDA models are trained on documents in the training data. Jieba[2)] is used for Chinese word segmentation, and LDA is trained using Mallet toolkit[3)]. When training seq2seq networks, it is found that the word-level model had poor performance and generated lots of UNKs (UNK denotes the unknown word) owing to limited vocabulary size. Hence, character-based models are used instead (i.e., Chinese characters are used as input when training and testing). To improve the generalization of the model, each "expert" network is initialized using the parameters trained in a conventional NHG model given all training cases for 1 million iterations. Afterward, each network is fine-tuned using only the documents for each topic to make the model topic-specific for 0.05 million iterations.

4.3 Models for comparison

We compare our model with the following systems and state-of-the-art methods (since datasets are broadly used, only the results are extracted from the original papers).

- RNN-context(W) and RNN-context(C) [10] are seq2seq architectures. (W) and (C) indicate that a model is word-based and character-based respectively. "Context" indicates that the model incorporates the attention mechanism to capture context information.
- CopyNet [7] integrates a copying mechanism into seq2seq architecture so that certain words from the input could be replicated to the output headline.
- RNN-distract [19] proposes a framework in which the attention mechanism can keep track of the passive input context and attention weight vectors, and distract attention from historical information in decoding steps.
- DRGD [20] integrates variational auto-encoders into the decoder of the seq2seq framework, to model the latent structure information implied in target headlines. As a result, the generated target headlines have better quality.
- MRT(C) [18] is also a character-based seq2seq architecture that adopts the ROUGE-based minimum risk training as the training procedure while its other settings remain the same as in the model developed in this study.
- Reinforced [21] shares the same inspiration as ours and integrates topic information into a neural headline generation model. Specifically, the authors take the CNN as the basic model architecture, and incorporate multi-step attention and topic-aware attention mechanism when calculating the decoder side output distribution, then train the model with ROUGE-based reinforcement learning. The topic information appears in the form of topic embeddings and participates in the calculation of topic-aware attention.

4.4 Evaluation metric

4.4.1 *ROUGE*

ROUGE [22] automatically measures summary quality by comparing computer-generated summaries to standard summaries created by humans. ROUGE is a common evaluation metric in document under-

[2)] https://pypi.python.org/pypi/jieba/

[3)] http://mallet.cs.umass.edu/

standing conference, which is a large-scale summarization evaluation sponsored by national institute of standards and technology (NIST). The basic idea of ROUGE is to count the number of overlapping units such as overlapped n-grams, word sequences, and word pairs between computer-generated summaries and standard summaries.

In this study, we consider two types of ROUGE: ROUGE-N and ROUGE-L. ROUGE-N counts N-grams, while ROUGE-L counts the longest common sub-sequences. Suppose $\mathbf{y}'$ is the generated summary, and $\mathbf{y}$ is the standard summary, ROUGE-N is defined as follows:

$$\text{ROUGE-N} = \frac{\sum_{\text{gram}_\text{N} \in \mathbf{y}} M(\text{gram}_\text{N})}{\sum_{\text{gram}_\text{N} \in \mathbf{y}} C(\text{gram}_\text{N})}, \tag{9}$$

where N indicates the type of N-gram (e.g., uni-gram and bi-gram, corresponding to ROUGE-1 and ROUGE-2), $M(\text{gram}_\text{N})$ is the number of n-grams matched between $\mathbf{y}'$ and $\mathbf{y}$, and $C(\text{gram}_\text{N})$ is the total number of n-grams in $\mathbf{y}$.

ROUGE-L is formalized as

$$\text{ROUGE-L} = \frac{(1+\beta^2)R_\text{L}P_\text{L}}{R_\text{L}+\beta^2 P_\text{L}}, \tag{10}$$

where β is the harmonic factor between recall R_L and precision P_L, which are defined as

$$R_\text{L} = \frac{\text{Lcs}(\mathbf{y}',\mathbf{y})}{\text{Len}(\mathbf{y})}, \quad P_\text{L} = \frac{\text{Lcs}(\mathbf{y}',\mathbf{y})}{\text{Len}(\mathbf{y}')}, \tag{11}$$

where $\text{Lcs}(\mathbf{y}',\mathbf{y})$ is the length of the longest common subsequence between $\mathbf{y}'$ and $\mathbf{y}$, and $\text{Len}(\mathbf{y})$ is the length of $\mathbf{y}$.

4.4.2 *Manual evaluation*

NHG is one of the neural sequence generation tasks, so the generated words can be arbitrary words instead of words from the original input. When the automatic evaluation metric ROUGE is used as the only way to evaluate system performance, the results can be misleading [23–25]. It is, therefore, more desirable to utilize manual evaluation in addition to ROUGE.

The question-answering (QA) based method was adopted to evaluate the ability of the system to capture salient information for a long time [23, 24, 26–28]. Chen et al. [24] treated each text passage as a small knowledge base and made a large number of questions to identify all content points in a text passage. Narayan et al. [25] manually created several questions for each text passage and asked the subjects to answer the questions after reading system outputs. The method from Narayan et al. [25] is more suitable for this experimental setting since headlines are usually very short. Particularly, one question was manually created for each data pair based on the reference headline. The subjects were then asked to evaluate whether the system-generated headlines were able to answer the questions without accessing original news articles. The scoring mechanism was as follows: a correct answer is marked with score 1, a partially correct answer gets score 0.5, and an incorrect answer obtains score 0. The final system score is the average of all its scores. Twenty articles were randomly selected from the LCSTS test set to conduct the QA-based evaluation. Questions for reference headlines were written without looking at original articles.

The QA-based evaluation makes an effort to judge how well a system is able to capture salient information. Aside from informativeness, a headline should also possess other qualities: (1) it should have a good flow while reading; (2) it should be concise; (3) it should correlate to the original news article; (4) it should correlate to the reference headline. System ranking was used as the way to evaluate the above-mentioned qualities. Each time, a subject was presented with a news article, its corresponding reference headline, and system-generated headlines. After carefully reading the article, the subject was asked to rank the headline regarding fluency, conciseness, relation to the original input, and relation to the reference headline.

Table 3 The model performance with different topic numbers

Topic numbers k	ROUGE-1	ROUGE-2	ROUGE-L
1	34.7	22.9	32.5
2	38.7	26.3	36.1
3	**38.9**	**26.5**	**36.4**
4	38.7	26.4	36.3
5	38.4	26.6	36.1
6	38.7	26.5	36.0
7	38.3	25.8	35.6
8	38.7	26.1	35.9
9	38.4	26.0	35.8
10	38.7	26.1	36.0

Two manual evaluations were conducted upon a small subdataset composed of twenty randomly selected articles from the LCSTS test set. Ten subjects, all native Chinese speakers, were asked to participate in the study. They were separated into two groups in order to accomplish two experiments since the requirements of the two experiments were different. To make the evaluation fair enough, the order of articles and the order of headlines were all randomized for each subject. The systems for comparison included the baseline, MRT(C), DRGD, and TopicNHG.

4.5 Topic number effect

As introduced in Subsections 3.5 and 2.2, the first step of the TopicNHG model is to assign a topic to each training pair. The number of topic k should be predefined in LDA. It is obvious that different topic number settings affect the performance of the TopicNHG model. This subsection presents how the model behaves with different topic number settings. The k was set as $1, 2, \ldots, 10$, and Table 3 shows ROUGE evaluation results (when $k = 1$, the model would degenerate to the baseline model). There are three major observations. Firstly, the model obtains the lowest ROUGE score when the topic number is set as 1; in other words, when the model degenerates to the baseline model, the model performs the worst. This confirms that assigning a certain topic to an input and generating headlines with the corresponding expert network would consistently improve model performance. Secondly, model performance does not continuously rise with the growth of the number of topics. It is hence important to choose an appropriate number of topics through experiments. Thirdly, when the topic number is set as 3, the TopicNHG model performs best on all three evaluation metrics. Thus, $k = 3$ was chosen as the topic number in the following experiments.

Topic keywords were also a point of interest in the study and Table 4 shows top-15 keywords of each topic when the topic number was set as 3. The keywords in each category generally show semantic similarity or are commonly used to describe events in the same category. For example, in the first topic, the keywords “互联网” (Internet), “上市公司” (listed company), and “阿里巴巴” (Alibaba) commonly appear in financial news reports. The keywords in the other two topics also exhibit the same property. Hence, Topic-1, Topic-2, and Topic-3 are manually marked as finance, politics, and legal respectively, for the sake of expression simplicity. Table 5 shows the number of instances belonging to each topic, and the data for each topic remains balanced.

4.6 Main results

4.6.1 *ROUGE evaluation results*

The proposed model is compared with the conventional seq2seq model (baseline) and several other models, as introduced in Subsection 4.3. Table 6 shows the ROUGE F-measure results of these models. The model architectures of each system are shown in Table 7 to help understand the model performance. The proposed model outperforms the baseline one with 4 points improvement on all three indexes. This shows the significance of introducing topic information. The RNN_context(W) and RNN_context(C) have the

Table 4 Top-15 words with highest probability in each topic

Topic	Keywords		
Topic-1 (Finance)	互联网 (Internet)	人民币 (RMB)	亿美元 (billions of dollars)
	房地产 (real estate)	消费者 (consumer)	投资者 (investor)
	董事长 (chairman)	上市公司 (listed company)	证监会 (CSRC)
	有限公司 (limited company)	IPO	阿里巴巴 (Alibaba)
	智能手机 (smart phone)	运营商 (operator)	创业板 (start up board)
Topic-2 (Politics)	负责人 (person in charge)	进一步 (further)	国务院 (State Council)
	北京市 (Beijing)	习近平 (Jinping Xi)	公务员 (civil servant)
	发改委 (NDRC)	办公室 (office)	委员会 (committee)
	毕业生 (graduate)	有限公司 (limited company)	李克强 (Keqiang Li)
	幼儿园 (kindergarten)	工作人员 (staff)	高速公路 (high way)
Topic-3 (Legal)	嫌疑人 (suspect)	工作人员 (staff)	公交车 (bus)
	出租车 (Taxi)	派出所 (police station)	为什么 (why)
	公安局 (Public Security Bureau)	有期徒刑 (fixed term imprisonment)	支付宝 (Alipay)
	年轻人 (young people)	一个月 (one month)	身份证 (ID card)
	被告人 (defendant)	人民法院 (People's Court)	银行卡 (bank card)

Table 5 Number of instances in different topics

Topic	PART-I	PART-II	PART-III
Finance	954423	2586	213
Politics	717374	3985	509
Legal	728794	4095	384
Total	2400591	10666	1106

Table 6 ROUGE F-measure (%) on LCSTS

	ROUGE-1	ROUGE-2	ROUGE-L
RNN_context(W)	26.8	16.1	24.1
RNN_context(C)	29.9	17.4	27.2
CopyNet	34.4	21.6	31.3
RNN_distract	35.2	22.6	32.5
DRGD	37.0	24.2	34.2
MRT(C)	38.2	25.2	35.4
Reinforced	**45.1**	**33.1**	**42.7**
Baseline	34.7	22.9	32.5
TopicNHG	**38.9**	**26.5**	**36.4**

same model architecture as the baseline model and TopicNHG, but a different decoding strategy makes them underperform. CopyNet is a word-based model which is able to copy out-of-vocabulary words from the original input. The RNN_distract, DRGD, and MRT(C) utilize different techniques to improve model performance, respectively. The techniques they adopt inevitably increase model complexity. Although they perform superior to the baseline model, they still perform inferior to the TopicNHG model. This further proves that TopicNHG can improve model performance with less complexity. However, our model does not achieve as good performance as reinforced on all three metrics. We suspect that it may be caused by the model design of their work, which includes a fully convolutional model for the sequence to sequence learning with multi-step attention [29] and the reinforcement learning based self-critical sequence training method [30], that boosts the model performance.

The proposed model is also compared to the baseline model on each topic, and the result is shown in Table 8. The proposed model consistently outperforms the baseline one on each topic, demonstrating the robustness of TopicNHG. The improvement rates of TopicNHG compared to the baseline model on each topic are also different. While the topics of "finance" and "politics" outperform the baseline system for about 3 points on ROUGE-1 and ROUGE-L, and for 2 points on ROUGE-2, the "legal" topic outperforms

Table 7 Model architectures corresponding to Table 6 a)

	Extra input	Encoder	Decoder	Attention	Topic	Others	Training
RNN_context(W)	–	RNN	RNN	Traditional attention	–	–	–
RNN_context(C)							
CopyNet						Copy	
RNN_distract						–	
DRGD							
MRT(C)							mrt
Reinforced	pos+topic	CNN	CNN	Multi-step Attention	Yes		rl
Baseline	–	RNN	RNN	Traditional attention	–		–
TopicNHG					Yes		

a) In the "extra input" column, "–" means a model only takes word embeddings as the encoder input, pos and topic mean the word position information and the topic embedding information are taken as extra inputs, respectively. Copy denotes copy mechanism, mrt stands for minimum risk training, and rl means reinforcement learning

Table 8 The ROUGE F-measure (%) comparison between the Baseline and the TopicNHG on each topic

	ROUGE-1		ROUGE-2		ROUGE-L	
	Baseline	TopicNHG	Baseline	TopicNHG	Baseline	TopicNHG
Finance	35.9	38.9	24.2	26.9	33.6	36.8
Politics	35.1	38.5	23.5	26.1	33.1	36.1
Legal	33.7	**39.4**	21.8	**26.6**	31.4	**38.2**

Table 9 Evaluation results on system ranking and QA-based evaluations

Model	First	Second	Third	Fourth	QA-Based
Baseline	0.23	0.13	0.24	**0.40**	57.5
DRGD	0.26	**0.33**	0.24	0.17	59.5
MRT(C)	0.16	0.27	**0.28**	0.30	69.5
TopicNHG	**0.35**	0.27	0.24	0.14	70.5

the baseline system for about 6, 5, and 7 points on ROUGE-1, ROUGE-2, and ROUGE-L, respectively. It is common knowledge that a neural model demands large-scale training data for better performance. However, despite the fewest training data among each of the three topics (according to Table 5), the improvements of the "legal" topic are the greatest. This observation further proves the effectiveness of the significance of introducing topic information.

4.6.2 *Manual evaluation results*

The QA-based evaluation results are shown in the last column of Table 9[4]. The subjects answered 70.5% of questions correctly based on headlines generated by TopicNHG. The headlines produced by MRT(C), DRGD, and the baseline model provide answers for 69.5%, 59.5% and 57.5% of the questions, respectively. The rest of Table 9 shows how the subjects ranked each system 1st, 2nd, and so on. TopicNHG is ranked best, followed by DRGD, MRT(C), and the baseline model, which are mostly ranked 2nd, 3rd, and 4th. The two manual evaluations, the QA-based evaluation, and system ranking, show the same pattern as ROUGE scores in Table 6.

4.6.3 *Case study*

We present randomly picked examples of generated headlines in Table 10 to verify the performance of the proposed model compared with the baseline system. The original examples in Chinese and the corresponding English translations are shown in Table 10. From these examples, we find that all the systems are able to capture the main topic of the original input. Among them, the TopicNHG usually generates more informative and diverse headlines comparing to others, which could prove the effectiveness

[4] We are grateful to Piji LI for providing us with the output of their systems.

Table 10 Examples of generated headlines

Weibo	央行今日将召集大型商业银行和股份制银行开会, 以应对当前的债市风暴. 消息人士表示, 央行一方面旨在维稳银行间债券市场, 另一方面很可能探讨以丙类户治理为重点的改革内容. 此次债市风暴中, 国家审计署扮演了至关重要的角色. (The PBOC will convene today commercial banks and joint-stock banks to cope with the current bond market storm. According to sources, the central bank is aiming at maintaining the inter-bank bond market on the one hand and is likely to explore reforms focusing on class C account governance on the other. The National Audit Office played a vital role in this bond market storm.)
Reference	媒体称央行今日召集银行开会应对当前债市风暴 (The media said the PBOC today convened a bank meeting to cope with the current bond market storm)
Baseline	央行今日将召集大型商业银行开会 (The PBOC will convene a meeting of large commercial banks today)
MRT(C)	央行今日将召集大型商业银行开会债市风暴 (The PBOC will today convene a large commercial bank a bond market storm)
DRGD	央行今日召集大型商业银行开会 (The PBOC today convened a meeting of large commercial banks)
TopicNHG	央行或将召集银行开会以应对债市风暴 (The PBOC may convene banks a meeting to cope with the bond market storm)
Weibo	欠债还钱, 自古天理. 为什么有那么一些业主, 自己可以大手大脚随便挥霍, 就是不肯把仨瓜俩枣的工钱支付给农民工. 欠薪的本质是什么? 是对劳动权利的蔑视, 是对劳动者的不尊重, 是对劳动法的粗暴践踏. (To liquidate a debt is perfectly justified since ancient times. Why are there so many owners who can squander as much as they want, and refuse to pay the scanty wages to migrant workers? What is the nature of wage arrears? It is contempt for labor rights, disrespect for workers, and a gross violation of labor law.)
Reference	党报批欠薪业主: 自己挥霍就不肯把工钱给农民工 (Party newspapers criticized owners wage arrears: they squander, but they will not give money to migrant workers)
Baseline	农民工欠薪的本质是什么? (What is the nature of migrant workers wage arrears)
MRT(C)	农民工欠薪的本质 (The nature of migrant workers wage arrears)
DRGD	人民日报对劳动法的蔑视是对劳动者的不尊重 (The contempt of People's Daily for labor law is a disrespect for workers)
TopicNHG	人民日报批欠薪: 自己挥霍就不肯把钱给农民工 (People's Daily criticized wage arrears: they squander, but they will not give money to migrant workers)
Weibo	昨晚, 中联航空成都飞北京一架航班被发现有多人吸烟. 后因天气原因, 飞机备降太原机场. 几名乘客在舱门边吸烟被发现. 有乘客要求重新安检, 机长决定继续飞行, 引起机组人员与未吸烟乘客冲突. 目前中联航空正联系机组进行核实. (Last night, many were found to have smoked on a flight from China United Airlines Chengdu to Beijing. Later, due to the weather, the aircraft was ready to drop on Taiyuan Airport. Several passengers were found smoking by the hatch. The captain decided to continue the flight after few passengers asked for a re-screening, causing a clash between the crew and the non-smoking passengers. At present, China Union Airlines is contacting the crew for verification.)
Reference	成都飞北京航班多人吸烟机组人员与未吸烟乘客冲突 (Many were found to have smoked on a flight from Chengdu to Beijing, there was a clash between the crew and the non-smoking passengers)
Baseline	飞机被发现多人吸烟乘客将继续飞行 (The plane was found many people smoked and the passengers would continue the flight)
MRT(C)	成都飞北京一航班有多人吸烟 (There were many people had smoked on a flight from Chengdu to Beijing)
DRGD	成都飞北京航班多人吸烟乘客要求安检 (Many were found to have smoked on a flight from Chengdu to Beijing, the passengers asked for a re-screening)
TopicNHG	飞机上多人吸烟机组人员与未吸烟乘客冲突 (Many were found to have smoked on a flight, there was a clash between the crew and the non-smoking passengers)

Table 11 Headlines generated by various "expert" networks on the same input.

Weibo	不允许挪用的社保金被拿去投资, 导致亏损308万元无法收回, 广东东源县法院副院长经上级授意伪造判决书, 对"窟窿"资金进行"依法核销". 近日, 经手伪造判决书的县法院副院长刘伟华已被检察机关立案侦查. (The social security fund not allowed to embezzle is taken into investment, resulting in a loss of 3.08 million RMB which can not be recovered, the vice president of the court of Dongyuan Guangdong has forged the judgement under the superior's order. The capital hole are "written off according to law". Recently, the country court vice president Weihua Liu, who handled the forged verdict, has been investigated by procuratorial organ.)
Reference	广东东源县法院副院长经上级授意伪造判决书 (The vice president of Dongyuan Guangdong court forged judgement under the superior's order)
Baseline	广东东源县法院副院长被立案侦查 (The vice-president of the Court of Dongyuan Guangdong is put on file for investigation)
Finance	广东官员挪用社保基金 (A Guangdong officer embezzles social security funds)
Politics	广东东源县法院被指伪造判书 (The vice-president of the Court of Dongyuan Guangdong is accused of forging judgement)
Legal	广东东源县法院副院长伪造判决书被立案侦查 (The vice-president of the Court of Dongyuan Guangdong forged a judgment and is put on file for investigation)

of TopicNHG. However, we also find the TopicNHG, and other systems as well would suffer from the common weak spot of neural generation systems: generating fake output words. For example, in the second example, the "人民日报 (People's Daily) " has not appeared in the original input, but it appeares in the outputs of system DRGD and TopicNHG.

In addition to the examples from different systems, we also present examples from different topics. Table 11 shows an example of topic Legal, since short texts related to topic Legal are most significantly improved according to Table 8. This example illustrates the original input document, reference headline, headline generated by the baseline model, and headline generated by three different "expert" networks. Among the system-generated headlines, the "expert" network of topic legal provides the most informative headline. Interestingly, the headlines generated by different "expert" networks pay attention to different aspects of the original document. While topic politics concentrates on a forging judgment, finance focuses on embezzling social security funds, and legal headline not only focuses on the forging judgment but also investigation. This example shows that an "expert" network can be more dedicated to the main theme of the original document with more accurate information.

5 Related work

End-to-end NHG has attracted considerable attention in recent years. A lot of work has been done to improve various aspects of model performance, as will be explained below.

- **Toward the unknown words.** Considering the computing and capacity complexity, most NHG systems maintain fix-sized input and output vocabularies according to word frequency. The words that are not included in the vocabularies are usually replaced as a special token "UNK", which consequently affects model performance. CopyNET proposed by Gu et al. [7] can either directly "copy" words from the original input or generate target vocabulary words at each decoding time step. CopyNET can enlarge output vocabulary with input words, and is also consistent with human behavior: people tend to repeat words when communicating. Cao et al. [31] and Gulcehre et al. [32] built their models sharing the same motivation with [7].

- **Online learning.** Existing studies on NHG usually adopt the same architecture in which the decoder starts to generate output words after the encoder reads in the whole input sequence. An online model for NHG was proposed by Yu et al. [33]. In each decoding time step, the model calculates the corresponding prediction probability and transition probability to decide either to generate an output word or to read in an input word.

- **Toward the characteristics of the headline.** Although encoder-decoder architecture brings huge benefits to abstractive headline generation, the great similarity of basic model architecture between NHG systems and neural machine translation systems still remains unsatisfying. The following works aim to improve NHG systems considering unique headline characteristics. Length limit or compression ratio are important criteria for a summarization system, and the same is true for a headline generation system. Kikuchi et al. [34] attempted to train a model, which is able to generate target headlines with exact length requirements. Most NHG systems share the same attention-based encoder-decoder architecture. However, there is no explicit aligning relationship between input words and output words in the headline generation task. Zhou et al. [8] proposed a solution by additionally incorporating a selective gate network, which is used to filter out redundant information in input words. When writing a headline, there are customary routines: people tend to include information regarding “Who”, “What” and “When” in the headline. Miao and Blunsom [35] and Li et al. [36] both designed a model architecture that captures such latent structure of target headlines.
- **Output quality.** NHG systems usually suffer from problems such as missing salient information and generating repeated and extra words [18]. Cao et al. [37] designed a framework that is able to focus on the input to alleviate the above-mentioned problems. Ayana et al. [18] leveraged the minimum risk training [38] to improve model performance and output fluency. Li et al. [39] build an actor-critic framework that can discriminate the quality of a generated headline. Cao et al. [9] proposed to combine the traditional NHG framework with a template-based summarization method.
- **Topic information** Many previous studies related to multi-document summarization have been focused on improving the summarization performance by introducing topic information [40–43]. Although the topic information has been proven to be important in summarization community, it has not been well studied in NHG, till Wang et al. [21] proposed a topic-aware convolutional sequence to sequence model with reinforcement learning for headline generation.

6 Conclusion and future work

This paper proposes topic-sensitive NHG, which introduces document-level topic information into the NHG model. Precise manual evaluation investigated the performance of the model in addition to the automatic evaluation metric ROUGE. Automatic evaluation results, as well as manual results and analysis, verify that the topic carries important information for headline generation tasks. The proposed TopicNHG model is simple and significantly outperforms baseline systems. There are two directions for future work: (1) explore methods that can deal with fine-grained topics; (2) extend the model to support NHG for documents with multiple topics in order to alleviate potential errors in topic assignment.

References

1 Edmundson H P. New methods in automatic extracting. J ACM, 1969, 16: 264–285

2 Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks. In: Proceedings of the 28th Conference on Neural Information Processing Systems, Montreal, 2014. 3104–3112

3 Cheng J, Lapata M. Neural summarization by extracting sentences and words. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, 2016. 484–494

4 Nallapati R, Zhai F, Zhou B. Summarunner: a recurrent neural network based sequence model for extractive summarization of documents. In: Proceedings of the 31st AAAI Conference on Artificial Intelligence, San Francisco, 2017

5 Tan J, Wan X, Xiao J. Abstractive document summarization with a graph-based attentional neural model. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, 2017. 1171–1181

6 Rush A M, Chopra S, Weston J. A neural attention model for abstractive sentence summarization. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, 2015. 379–389

7 Gu J, Lu Z, Li H, et al. Incorporating copying mechanism in sequence-to-sequence learning. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, 2016. 484–494

8 Zhou Q, Yang N, Wei F, et al. Selective encoding for abstractive sentence summarization. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, 2017. 1095–1104

9 Cao Z, Li W, Li S, et al. Retrieve, rerank and rewrite: soft template based neural summarization. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, 2018. 152–161

10 Hu B, Chen Q, Zhu F. LCSTS: a large scale chinese short text summarization dataset. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, 2015. 1967–1972

11 Jacobs R A, Jordan M I, Nowlan S J, et al. Adaptive mixtures of local experts. Neural Comput, 1991, 3: 79–87

12 Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation. J Machine Learning Res, 2003, 3: 993–1022

13 Nallapati R, Zhou B, dos Santos C. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In: Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, Berlin, 2016. 280–290

14 Cho K, van Merrienboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, 2014. 1724–1734

15 Mikolov T, Karafiát M, Burget L, et al. Recurrent neural network based language model. In: Proceedings of the Eleventh Annual Conference of the International Speech Communication Association, Chiba, 2010. 1045–1048

16 Schuster M, Paliwal K K. Bidirectional recurrent neural networks. IEEE Trans Signal Process, 1997, 45: 2673–2681

17 Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. In: Proceedings of the International Conference on Learning Representations, San Diego, 2015

18 Ayana, Shen S Q, Lin Y K, et al. Recent advances on neural headline generation. J Comput Sci Technol, 2017, 32: 768–784

19 Chen Q, Zhu X, Ling Z, et al. Distraction-based neural networks for document summarization. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence, New York, 2016

20 Li P, Lam W, Bing L, et al. Deep recurrent generative decoder for abstractive text summarization. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, 2017. 2091–2100

21 Wang L, Yao J, Tao Y, et al. A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization. In: Proceedings of the International Joint Conferences on Artifical Intelligence, Stockholm, 2018

22 Lin C-Y. ROUGE: a package for automatic evaluation of summaries. In: Proceedings of Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL 2004, Barcelona, 2004

23 Schluter N. The limits of automatic summarisation according to rouge. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, 2017. 41–45

24 Chen P, Wu F, Wang T, et al. A semantic qa-based approach for text summarization evaluation. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, New Orleans, 2018

25 Narayan S, Cohen S B, Lapata M. Ranking sentences for extractive summarization with reinforcement learning. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, 2018. 1747–1759

26 Morris A H, Kasper G M, Adams D A. The effects and limitations of automated text condensing on reading comprehension performance. Inf Syst Res, 1992, 3: 17–35

27 Mani I, Klein G, House D, et al. SUMMAC: a text summarization evaluation. Nat Lang Eng, 2002, 8: 43–68

28 Clarke J, Lapata M. Discourse constraints for document compression. Comput Linguistics, 2010, 36: 411–441

29 Gehring J, Auli M, Grangier D, et al. Convolutional sequence to sequence learning. 2017. ArXiv: 1705.03122

30 Rennie S J, Marcheret E, Mroueh Y, et al. Self-critical sequence training for image captioning. 2016. ArXiv: 1612.00563

31 Cao Z, Luo C, Li W, et al. Joint copying and restricted generation for paraphrase. In: Proceedings of the 31st AAAI Conference on Artificial Intelligence, San Francisco, 2017

32 Gulcehre C, Ahn S, Nallapati R, et al. Pointing the unknown words. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, 2016. 484–494

33 Yu L, Buys J, Blunsom P. Online segment to segment neural transduction. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, 2016. 1307–1316

34 Kikuchi Y, Neubig G, Sasano R, et al. Controlling output length in neural encoder-decoders. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, 2016. 1328–1338

35 Miao Y, Blunsom P. Language as a latent variable: discrete generative models for sentence compression. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, 2016. 319–328

36 Li P, Lam W, Bing L, et al. Deep recurrent generative decoder for abstractive text summarization. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, 2017. 2091–2100

37 Cao Z, Wei F, Li W, et al. Faithful to the original: fact aware neural abstractive summarization. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, New Orleans, 2018

38 Shen S, Cheng Y, He Z, et al. Minimum risk training for neural machine translation. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, 2016. 1683–1692

39 Li P, Bing L, Lam W. Actor-critic based training framework for abstractive summarization. 2018. ArXiv: 1803.11070

40 Celikyilmaz A, Hakkani-Tür D. Discovery of topically coherent sentences for extractive summarization. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, 2011. 491–499

41 Li J, Li S. A novel feature-based bayesian model for query focused multi-document summarization. Trans Assoc Comput Linguist, 2013, 1: 89–98

42 Li Y, Li S. Query-focused multi-document summarization: combining a topic model with graph-based semi-supervised learning. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin, 2014. 1197–1207

43 Bairi R, Iyer R, Ramakrishnan G, et al. Summarization of multi-document topic hierarchies using submodular mixtures. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, 2015. 553–563