Abstractive Text Summarization by Incorporating Reader Comments

Shen Gao,[1] **Xiuying Chen,**[1] **Piji Li,**[3] **Zhaochun Ren,**[4] **Lidong Bing,**[5] **Dongyan Zhao,**[1,2] **Rui Yan**[*1,2]

[1]Institute of Computer Science and Technology, Peking University, Beijing, China
[2]Center for Data Science, Peking University, Beijing, China
[3]Tencent AI Lab, Shenzhen, China
[4]JD.com, Beijing, China
[5]R&D Center Singapore, Machine Intelligence Technology, Alibaba DAMO Academy
{shengao, xy-chen, zhaody, ruiyan}@pku.edu.cn, pijili@tencent.com, renzhaochun@jd.com, l.bing@alibaba-inc.com

Abstract

In neural abstractive summarization field, conventional sequence-to-sequence based models often suffer from summarizing the wrong aspect of the document with respect to the main aspect. To tackle this problem, we propose the task of reader-aware abstractive summary generation, which utilizes the reader comments to help the model produce better summary about the main aspect. Unlike traditional abstractive summarization task, reader-aware summarization confronts two main challenges: (1) Comments are informal and noisy; (2) jointly modeling the news document and the reader comments is challenging. To tackle the above challenges, we design an adversarial learning model named reader-aware summary generator (RASG), which consists of four components: (1) a sequence-to-sequence based summary generator; (2) a reader attention module capturing the reader focused aspects; (3) a supervisor modeling the semantic gap between the generated summary and reader focused aspects; (4) a goal tracker producing the goal for each generation step. The supervisor and the goal tacker are used to guide the training of our framework in an adversarial manner. Extensive experiments are conducted on our large-scale real-world text summarization dataset, and the results show that RASG achieves the state-of-the-art performance in terms of both automatic metrics and human evaluations. The experimental results also demonstrate the effectiveness of each module in our framework. We release our large-scale dataset for further research[1].

Introduction

Abstractive summarization can be regarded as a sequence mapping task that the source text is mapped to the target summary, and has drawn much attention since the deep neural networks are widely applied in natural language processing field. Recently, sequence-to-sequence (seq2seq) framework (Sutskever, Vinyals, and Le 2014) has been proved effective for the task of abstractive summarization (Chopra, Auli, and Rush 2016; See, Liu, and Manning 2017) and other text generation tasks (Tao et al. 2018; Gao et al. 2019). In this paper, we use "**aspect**" to denote the topic described in a specific paragraph or a sentence of a news document, and use "**main aspect**" to denote the central

---

*Corresponding author: Rui Yan (ruiyan@pku.edu.cn)

[1]http://t.cn/EAH5JxS

Table 1: Examples of the text summarization. The text in italic style denotes the focused aspect by the good summary, while the text in bold is described by the bad summary. The text with underline is the focused aspect by reader comments.

document	On August 28, according to a person familiar with the matter, *Toyota Motor Corporation will invest 500 million U.S. dollars into the Uber*, a taxi service company, with a *valuation of up to 72 billion U.S. dollars.* The investment will focus on driverless car technology. However, its development path is not smooth. In March of this year, **a Uber driverless car hit a woman and caused her death.** In last year, Softbank also invested into Uber with a valuation of $48 billion.
comments	Toyota's investment in Uber is a wise choice.
	$500 million investment is really a lot of money!
good summary	Toyota invests $500 million into Uber with a valuation of $72 billion
bad summary	An Uber driverless car hits a passerby to death

topic which the author tends to convey to the readers. Although a document may describe an event in many different aspects, the summary of this document should always focus on the main aspect. As shown in Table 1, the good summary describes the main aspect and the bad summary describes another trivial aspect that is not the main point of the document. To focus on the main aspect, some summarization methods (Sun et al. 2018; Zhou et al. 2017; Bansal and Chen 2018) first select several sentences about the main aspect and then generate the summary. However, it is very challenging to discover which is the main aspect of the news document.

Nowadays, a great number of news comments are generated by readers to express their opinions about the event. Some comments may mention the main aspect of the document for several times. Take the case in Table 1 as an example, the focused aspect of the reader is "investment of Toyota" which is also the main aspect of this document. To be specific, we define "**reader focused aspect**" to denote the focused aspect by a reader through the comments. Intuitively, these reader comments may help the summary generator capture the main aspect of document, thereby improving the quality of the generated summary. Therefore, in this paper, we investigate a new problem setting of the task of

abstractive text summarization. We name such paradigm of extension as *reader-aware abstractive text summarization*.

The effect of comments or social contexts in document summarization have been explored by several previous works (Hu, Sun, and Lim 2008; Yang et al. 2011; Li et al. 2015; Li, Bing, and Lam 2017). Unlike these approaches that directly extract sentences from the original document (Hu, Sun, and Lim 2008; Yang et al. 2011; Li et al. 2015), we aim to generate a natural-sounding summary from scratch instead of extracting words from the document.

Generally, existing text summarization approaches confront two challenges when addressing reader-aware summarization task. The first challenge is that reader comments are very noisy and informative. Not all the information provided by the comments is useful when modeling the reader focused aspects. Therefore, it is crucial to make the model own the ability of capturing main aspect and filtering noisy information when incorporating reader comments. The second challenge is how to generate summaries by jointly modeling the main aspect of document and the reader focused aspect revealed by comments. Meanwhile, the model should not be sensitive to the diverse unimportant aspects introduced by some reader comments. Thus, simply absorbing all the reader aspect information to directly guide the model to generate summary is not feasible, as it will make the generator lose the ability of modeling the main aspect.

In this paper, we propose a summarization framework named *reader-aware summary generator* (RASG) that incorporates reader comments to improve the summarization performance. Specifically, a seq2seq architecture with attention mechanism is employed as the basic summary generator. We first calculate alignment between the reader comments words and document words, and this alignment information is regarded as reader attention representing the "**reader focused aspect**". Then, we treat the decoder attention weights as the focused aspect of the generated summary, *a.k.a.,* "**decoder focused aspect**". After each decoding step, a supervisor is designed to measure the distance between the reader focused aspect and the decoder focused aspect. Given this distance, a goal tracker provides the goal to the decoder to induce it to reduce this distance. The training of our framework RASG is conducted in an adversarial way. To evaluate the performance of our model, we collect a large amount of document-summary pairs associated with several reader comments from social media website. Extensive experiments conducted on this dataset show that RASG significantly outperforms the state-of-the-art baselines in terms of ROUGE metrics and human evaluations.

To sum up, our contributions can be summarized as follows:

• We propose a reader-aware abstractive text summarization task. To solve this task, we propose an end-to-end learning framework to conduct the reader attention modeling and reader-aware summary generation.

• We design a supervisor as well as a goal tracker to guide the generator to focus on the main aspect of the document.

• To reduce the noisy information introduced by the reader comments, we propose a denoising module to identify which comments are helpful for summary generation automatically.

• We release a large scale abstractive text summarization dataset associated with reader comments. Experimental results on this dataset demonstrate the effectiveness of our proposed framework.

Related Work

Text summarization can be classified into extractive and abstractive methods. Extractive methods (Jadhav and Rajan 2018; Narayan, Cohen, and Lapata 2018) read the article and get the representations of the sentences and article to select sentences. However, summaries generated by extractive methods always suffer from redundancy problem. Recently, with the emergence of neural network models for text generation, a vast majority of the literature on summarization is dedicated to abstractive summarization (Bansal and Chen 2018; Ma et al. 2018b; Zhou et al. 2018). On the text summarization benchmark dataset CNN/DailyMail, the state-of-the-art abstractive methods outperform the best extractive method in terms of ROUGE score. Most methods for abstractive text summarization are based on the sequence-to-sequence model (Sutskever, Vinyals, and Le 2014), which encodes the source texts into the semantic representation with an encoder, and generates the summaries from the representation with a decoder. To tackle the out-of-vocabulary problem, some researchers employ the copy mechanism to copy some words from the input document to summary (Gu et al. 2016; See, Liu, and Manning 2017). To capture the main aspect of document, Chen *et al.* (2018) propose to select salient sentences and then rewrite these sentences to a concise summary. This approach achieves the state-of-the-art of text summarization on CNN/DailyMail benchmark dataset. Unlike document summarization that needs to encode a long text, social media summarization usually reads short and noisy text and has become a popular task these days. After Hu *et al.* (2015) propose a short text summarization dataset on social media and many researchers follow this task. Lin *et al.* (2018) propose a seq2seq based model which uses an CNN to refine the representation of source context. Wang *et al.* (2018) use convolutional seq2seq model to summarize text and use the policy gradient algorithm to directly optimize the ROUGE score. However, these summarization models do not utilize the reader's comments in generating summaries.

To consider the reader's comments into text summarization, the reader-aware summarization is proposed and it mainly takes the form of extractive approaches. Graph-based method has been used for comment oriented summarization task such as (Hu, Sun, and Lim 2007; 2008), where they identify three relations (topic, quotation, and mention) by which comments can be linked to one another. Recently, Nguyen *et al.* (2016) publish a small extractive sentence-comment dataset which can not be used to train neural models due to its small size. Li *et al.* (2015) propose an unsupervised compressive multi-document summarization model using sparse coding method. Following previous work, there are some models (Li, Bing, and Lam 2017; Li et al. 2017) using variational auto-encoder to model the

latent semantic of original article and reader comments. Different from our abstractive summarization task, these related works are all based on extractive or compressive approaches.

Problem Formulation

Before presenting our approach for the reader-aware summarization, we first introduce our notations and key concepts.

To begin with, for a document $X^d = \{x_1^d, x_2^d, \ldots, x_{T^d}^d\}$, we assume there is a comment set $X^c = \{c_1, c_2, \ldots, c_{T^c}\}$ where $c_i = \{x_{i,1}^c, x_{i,2}^c, \ldots, x_{i,T_i}^c\}$ is the i-th comment, x_i^d denotes the i-th word in document X^d, and $x_{i,j}^c$ denotes the j-th word in i-th comment sentence c_i. Given the document X^d, the summary generator reads the comments X^c, then generates a summary $\hat{Y} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{T^{\hat{Y}}}\}$. Finally, we use the difference between generated summary $\hat{Y}$ and ground truth summary Y as the training signal to optimize the model parameters.

The Proposed RASG Model

Overview

In this section, we propose our *reader-aware summary generator*, abbreviated as RASG. The overview of RASG is shown in Figure 1 which can be split into four main parts:

• *Summary generator* is a seq2seq based architecture with attention and copy mechanisms.

• *Reader attention* module learns a semantic alignment between each word in document and comments, thus captures the reader focused aspect.

• *Supervisor* measures the semantic gap between decoder focused aspect and reader focused aspect. There is also a discriminator which uses convolutional neural network to extract features and then distinguishes how similar is decoder focused aspect to reader focused aspect.

• *Goal tracker* utilizes the semantic gap learned by supervisor and the features extracted learned by the discriminator to set a goal, which is further utilized as a more specific guidance for summary generator to produce better summary.

Summary generator

At the beginning, we use an embedding matrix e to map one-hot representation of each word in the document X^d and comments X^c to a high-dimensional vector space. We denote $e(x)$ as the embedding representation of word x. From these embedding representations, we employ a bi-directional recurrent neural network (Bi-RNN) to model the temporal interactions between words:

$$h_t^d = \text{Bi-RNN}_d(e(x_t^d), h_{t-1}^d), \tag{1}$$

where h_t^d denotes the hidden state of t-th step in Bi-RNN for document X^d. We denote the final hidden state $h_{T^d}^d$ of Bi-RNN$_d$ as the vector representation of the document X^d. Following (See, Liu, and Manning 2017; Ma et al. 2018a), we choose the long short-term memory (LSTM) as the Bi-RNN cell.

Then we apply a linear transform layer on the input document vector representation $h_{T^d}^d$ and use the output of this

layer as the initial state of decoder LSTM, shown in Equation 2. In order to reduce the burden of compressing document information into initial state s_0, we use the attention mechanism (Bahdanau, Cho, and Bengio 2015) to summarize the input document into context vector f_{t-1} dynamically and we will show the detail of these in the following sections. We then concatenate the context vector f_{t-1} with the embedding of previous step output $e(y_{t-1})$ and feed this into decoder LSTM, shown in Equation 3. We use the notion $[\cdot; \cdot]$ as the concatenation of two vectors.

$$s_0 = W_d h_{T^d}^d + b_d, \tag{2}$$

$$s_t = \text{LSTM}\left(s_{t-1}, [f_{t-1}; e(y_{t-1})]\right). \tag{3}$$

At t-th decoding step, we use the decoder state s_{t-1} to attend to each the document states h^d and resulting in the attention distribution $\alpha_t \in \mathbb{R}^{T^d}$, shown in Equation 5. Then we use the attention distribution α_t to weighted sum the document states as the context vector f_{t-1}.

$$\alpha'_{t,i} = W_a^\intercal \tanh\left(W_s s_t + W_h h_i^d\right), \tag{4}$$

$$\alpha_{t,i} = \exp\left(\alpha'_{t,i}\right) / \sum_{j=1}^{T^d} \exp\left(\alpha'_{t,j}\right), \tag{5}$$

$$f_{t-1} = \sum_{i=1}^{T^d} \alpha_{t,i} h_i^d. \tag{6}$$

Finally, an output projection layer is applied to get the final generating distribution P_v over vocabulary, as shown in Equation 7. We concatenate goal vector g_t, gap content d_t, and the output of decoder LSTM s_t as the input of the output projection layer. The goal vector g_t represents the goal of current generation step, the gap content d_t denotes the semantic gap between generated summary and reader focused document and we will show the details of these variables in the following sections.

$$P_v = \text{softmax}\left(W_v[s_t; g_t; d_t] + b_v\right), \tag{7}$$

In order to handle the out-of-vocabulary (OOV) problem, we equip the pointer network (Gu et al. 2016; Vinyals, Fortunato, and Jaitly 2015; See, Liu, and Manning 2017) with our decoder, which makes our decoder capable to copy words from the source text. The design of the pointer network is the same as the model used in (See, Liu, and Manning 2017), thus we omit this procedure in our paper due to the limited space. We use the negative log-likelihood as the loss function:

$$\mathcal{L}_g = -\sum_{t=1}^{T^{\hat{Y}}} \log P_v(y_t). \tag{8}$$

Denoising module

Due to the fact that reader comments are a kind of informal text, they may consist of many noisy information, and not all the comments are helpful for generating better summaries. Consequently, we employ a **denoising module** to distinguish which comments are helpful. First, we employ an Bi-RNN$_c$ to model the comment word embeddings:

$$h_{i,t}^c = \text{Bi-RNN}_c(h_{i,t-1}^c, e(x_{i,t}^c)), \tag{9}$$

where $h_{i,t}^c$ denotes the hidden state of t-th word in i-th comment c_i. Next, we use average-pooling operation over these
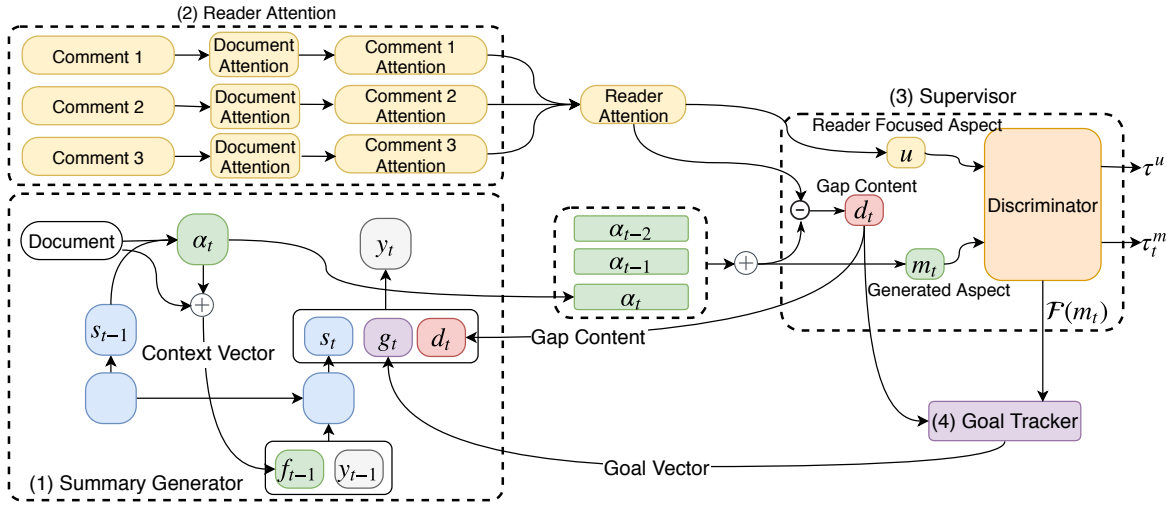
Figure 1: Overview of RASG. We divide our model into four parts: (1) *Summary generator* generates a summary to describe the main aspect of document. (2) *Reader attention* module models the readers attention of document. (3) *Supervisor* models the gap of focused document aspect between generated summary and reader comments. (4) *Goal tracker* sets a goal of summary generator according to gap given by supervisor.

hidden states to produce a vector representation a_i of i-th comment, shown in Equation 10. Finally, we apply a linear transform with sigmoid function to predict whether the comment is useful, and the sigmoid output $\hat{\beta}_i \in (0, 1)$ also can be seen as a salience score of i-th comment given the document representation $h_{T^d}^d$.

$$a_i = \text{avg}(\{h_{i,1}^c, h_{i,2}^c, \ldots, h_{i,T_i^c}^c\}), \qquad (10)$$

$$\hat{\beta}_i = \text{sigmoid}(W_s[a_i; h_{T^d}^d] + b_i), \qquad (11)$$

To train the denoising module, we use the cross entropy loss to supervise this procedure.

$$\mathcal{L}_d = -\sum_{t=1}^{T^c} \hat{\beta}_i \log(\beta_i). \qquad (12)$$

where $\beta_i \in \{0, 1\}$ is the ground truth salience score of comments. $\beta_i = 1$ denotes the i-th comment c_i is helpful for generating summary and vice-versa.

Reader attention modeling

To model the reader focused aspect, we first calculate the word alignment of reader comments towards the document. We use the embeddings of words in document and comments to calculate the semantic alignment score. Precisely, $\gamma_{i,j,k}$ is the alignment socre between the i-th document word x_i^d and the k-th word in the j-th comment $x_{j,k}^c$, as shown in Equation 13:

$$\gamma_{i,j,k} = e(x_i^d)^\top e(x_{j,k}^c), \qquad (13)$$

$$\delta_{i,j} = \max(\{\gamma_{i,j,1}, \ldots, \gamma_{i,j,T_j^c}\}), \qquad (14)$$

In Equation 14, we use a max-operation over the alignment $\gamma_{i,j,\cdot}$ to signify whether the i-th word of document is focused by the j-th comment. We regard the alignment score $\delta_{i,j}$ as the reader attention weight for the j-th reader comment to the i-th document word.

In order to reduce the interference caused by the noisy comments, we employ the comment salience score $\hat{\beta}_j$ obtained from the denoising module to weighted combine the j-th reader attention $\delta_{i,j}$, as shown in Equation 15. It means that noisy comments will contribute less in the procedure of reader attention modeling.

$$\epsilon_i' = \sum_{j=1}^{T^c} \delta_{i,j} \hat{\beta}_j, \qquad (15)$$

$$\epsilon_i = \exp\left(\epsilon_i'\right) / \sum_{j=1}^{T^d} \exp\left(\epsilon_j'\right). \qquad (16)$$

Finally, we get the reader attention $\epsilon_i \in \mathbb{R}$ for i-th document word after a softmax function as shown in Equation 16.

Supervisor

To model the semantic gap between the generated summary and the reader focused aspects, we design a supervisor module. First, for the decoder, we need to know which aspect in document has been focused by our summary generator in the past decoding steps. We sum up the latest k attention distributions $\{\alpha_t, \alpha_{t-1}, \ldots, \alpha_{t-k+1}\}$ and result in $\nu_t \in \mathbb{R}^{T^d}$ as the focus distribution of generated summary over T^d document words, shown in Equation 17. Then we use ν_t to weighted sum the document hidden states $h_\cdot^d$ and result in m_t:

$$\nu_t = 1/k \sum_{i=0}^{k-1} \alpha_{t-i}, \qquad (17)$$

$$m_t = \sum_{i=1}^{T^d} \nu_{t,i} h_i^d, \qquad (18)$$

where m_t represents the focused aspect by the latest k decoding steps, *a.k.a.,* decoder focused aspect.

Next, we use the reader attention $\epsilon_\cdot$ to weighted sum the document hidden states $h_\cdot^d$:

$$u = \sum_{i=1}^{T^d} \epsilon_i h_i^d, \qquad (19)$$

where u represents the reader focused aspect.

For encouraging the decoder focused aspect become similar to the reader focused aspect, we employ an CNN based discriminator to signify the difference between the decoder focused aspect m_t and the reader focused aspect u. Then we can use this difference to guide the decoder focus on the reader focused aspect. Typically, the discriminator is a binary classifier which can be decomposed into a convolutional feature extractor $\mathcal{F}$ shown in Equation 20 and a sigmoid classification layer shown in Equation 21 and 22.

$$\mathcal{F}(x) = \text{relu}(W_c \otimes x), \tag{20}$$

$$\tau_t^m = \text{sigmoid}(W_f \mathcal{F}(m_t) + b_f), \tag{21}$$

$$\tau^u = \text{sigmoid}(W_f \mathcal{F}(u) + b_f), \tag{22}$$

where $\otimes$ denotes the convolutional operation, trainable parameter W_c denotes the convolutional kernel, and τ_t^m and τ^u are both the classification probabilities.

Note that a token generated at time t will influence not only the gradient received at that time but also the gradient at subsequent time steps. Intuitively, the decoding attention ν_t of latter decoding step is more similar to the attention of final summary than the earlier steps. Thus we propose to define the cumulative loss with a discount factor $\varphi \in (0, 1]$ as the loss functions. Note that the training objective for discriminator can be interpreted as maximizing the log-likelihood for classification, whether the input x in Equation 20 comes from reader focused aspect or from decoder focused aspect.

$$\mathcal{L}_c^d = \sum_{t=1}^{T^d} \varphi^{T^d - t}(\log \tau^u + \log(1 - \tau_t^m)), \tag{23}$$

$$\mathcal{L}_c^g = \sum_{t=1}^{T^d} \varphi^{T^d - t}(\log(1 - \tau_t^m)). \tag{24}$$

In order to model the gap between reader focused aspect and decoder focused aspect, we subtract the reader attention $\epsilon \in \mathbb{R}^{T^d}$ by $\nu_t \in \mathbb{R}^{T^d}$ resulting in attention difference $\zeta_t \in \mathbb{R}^{T^d}$, shown in Equation 25. Then we use the attention difference ζ_t to sum up the document hidden states $h_\cdot^d$:

$$\zeta_t = \epsilon - \nu_t, \quad d_t = \sum_{i=1}^{T^d} \zeta_{t,i} h_i^d. \tag{25}$$

where d_t denotes the semantic of unfocused document aspects by summary generator, *a.k.a.,* gap content. To encourage the summary generator focus on the unfocused document aspects, we feed the gap content d_t to the generator, as shown in Equation 7.

Goal tracker

Since the discriminator only provides a scalar guiding signal τ_t^m at each decoding step, it becomes relatively less informative when the sentence length $T^{\hat{Y}}$ goes larger. Inspired by LeakGAN (Guo et al. 2018), the proposed RASG framework allows discriminator to provide additional information, denoted as goal vector g_t. In view of there is certain relationship between the goal of current decoding step and previous steps, we need to model the temporal interactions between the goal of each step. More specifically, we introduce a **goal tracker** module, an LSTM that takes the extracted feature vector $\mathcal{F}(m_t)$ and gap content d_t as its input at each step t and outputs a goal vector g_t:

$$g_t = \text{LSTM}(g_{t-1}, [\mathcal{F}(m_t); d_t]). \tag{26}$$

In order to achieve higher consistency of reader focused aspect, we feed the goal vector g_t into the generator to guide the generation of the next word, as shown in Equation 7.

Model training

As our model is trained in an adversarial manner, we re-split the parameters in our model into two parts: (1) *generation module* including the parameters of summary generator, reader attention module and goal tracker; (2) *discriminator module* including the parameters of CNN classifier. As for training generation module, we sum up the loss function of denoising module $\mathcal{L}_d$, cross entropy between ground truth $\mathcal{L}_g$ and the result of discriminator $\mathcal{L}_c^g$, as shown in Equation 27. We use the $\mathcal{L}$ to optimize the parameters of generation module.

$$\mathcal{L} = \mathcal{L}_g + \mathcal{L}_d + \mathcal{L}_c^g, \tag{27}$$

Next, we train the discriminator module to maximize the probability of assigning the correct label to both generated aspect m_t and reader focused aspect u. More specifically, we optimize the parameters of discriminator module according to the loss function $\mathcal{L}_c^d$ calculated in Equation 23.

Experimental Setup

Research questions

We list four research questions that guide the experiments: **RQ1**: Does RASG outperform other baselines? **RQ2**: What is the effect of each module in RASG? **RQ3**: Does RASG capture useful information from noisy comments? **RQ4**: Can goal tracker give a helpful guidance to decoder?

Dataset

We collect the document-summary-comments pair data from Weibo which is the largest social network website in China, and users can read a document and post a comment about the document on this website. Each sample of data contains a document, a summary and several reader comments. Most comments are about the readers' opinion of their focused aspect in the document. In order to train the denoising module, we should give a ground truth label β_i for i-th comment. When there is at least one common word in summary and comment, we regard such comment is helpful for generating summary. Accordingly, we give the $\beta_i = 1$ to i-th comment when it contains at least one common word and give $\beta_i = 0$ when it does not. In total, our training dataset contains 863826 training samples. The average length of document is 67.08 words, average length of comment is 16.61 words and average length of summary is 16.56 words. The average comments number of a document is 9.11.

Evaluation metrics

For evaluation metrics, we adopt ROUGE score (Lin 2004) which is widely applied for summarization evaluation (Sun et al. 2018; Chen et al. 2018). The ROUGE metrics compare generated summary with the reference summary by computing overlapping lexical units, including ROUGE-1 (unigram), ROUGE-2 (bi-gram) and ROUGE-L (longest common subsequence).

Table 2: Ablation models for comparison.

Acronym	Gloss
RASG w/o DM	RASG w/o **D**enoising **M**odule
RASG w/o G	RASG w/o **G**ap content
RASG w/o GT	RASG w/o **G**oal **T**racker
RASG w/o GTD	RASG w/o **G**oal **T**racker **D**iscriminator

Table 3: ROUGE scores comparison between baselines.

	ROUGE-1	ROUGE-2	ROUGE-L
S2S	23.86	9.86	23.83
S2SR	24.70	10.00	24.50
CGU	27.32	11.36	25.49
RASG	**30.33**	**12.39**	**27.16**
LEAD1	5.51	1.71	4.94
TextRank	13.5	4.55	11.46

Table 4: ROUGE scores of different ablation models.

	ROUGE-1	ROUGE-2	ROUGE-L
RASG w/o DM	27.29	11.01	24.64
RASG w/o G	28.03	11.24	25.28
RASG w/o GT	22.75	9.17	20.80
RASG w/o GTD	19.30	6.95	17.70
RASG	**30.33**	**12.39**	**27.16**

ROUGE-2 and ROUGE-L respectively. It is worth noticing that the baseline model S2SR achieves better performance than S2S which demonstrates the effectiveness of incorporating reader focused aspect in summary generation. However when compared with RASG, S2SR achieves lower performance in terms of all ROUGE score. Thus, simply adding the reader focused aspect into generation procedure is not a good reader-aware summarization method.

Comparison methods

In order to prove the effectiveness of each module in RASG, we conduct some ablation models introduced in Table 2.

To evaluate the performance of our proposed dataset and model, we compare it with the following baselines:
(1) **S2S**: Sequence-to-sequence framework (Sutskever, Vinyals, and Le 2014) has been proposed for language generation task. (2) **S2SR**: We simply add the reader attention $\epsilon.$ on attention distribution $\alpha_{t,.}$ in each decoding step. (3) **CGU**: Lin *et al.* (2018) propose to use the convolutional gated unit to refine the source representation, which achieves the state-of-the-art performance on social media text summarization dataset. (4) **LEAD1**: LEAD1 is a commonly used baseline (Nallapati, Zhai, and Zhou 2017; See, Liu, and Manning 2017), which selects the first sentence of document as the summary. (5) **TextRank**: Mihalcea *et al.* (2004) propose to build a graph, then add each sentence as a vertex and use link to represent semantic similarity. Sentences are sorted based on final scores and a greedy algorithm is employed to select summary sentences.

Implementation details

We implement our experiments in TensorFlow (Abadi et al. 2016) on an NVIDIA P40 GPU. The word embedding dimension is set to 256 and the number of hidden units is 512. We set the $k = 5$ in the Equation 17 and $\varphi = 0.5$ in Equation 23 and 24. We use Adagrad optimizer (Duchi, Hazan, and Singer 2010) as our optimizing algorithm. We employ beam search with beam size 5 to generate more fluency summary sentence.

Experimental Results

Overall performance

For research question **RQ1**, we examine the performance of our model in terms of ROUGE. Table 3 lists performances of all comparisons in terms of ROUGE score. We see that RASG achieves a 11.0%, 9.1% and 6.6% increment over the state-of-the-art method CGU in terms of ROUGE-1,

Ablation study

Next, we turn to research question **RQ2**. We conduct ablation tests on the usage of denoising module, supervisor as well as the goal tracker and the ROUGE score result is shown in Table 4. The discriminator provides the scalar training signal $\mathcal{L}_c^g$ for generator training and the feature vector $\mathcal{F}(m_t)$ for goal tracker. Consequently, there is an increment of 17.51% from RASG w/o GTD to RASG w/o GT in terms of ROUGE-L, which demonstrates the effectiveness of discriminator. As for the effectiveness of goal tracker, compared with RASG and RASG w/o GT, RASG w/o GTD offers a decrease of 45.23% and 17.88% in terms of ROUGE-1, respectively. This demonstrates that the goal tracker with the feature from discriminator plays an important role in producing better summary. However, using the goal tracker without the feature extracted by the discriminator does not help improve the performance of the summary generator, shown by the performance of RASG w/o GTD. Finally, RASG w/o DM offers a decrease of 10.22% compared with RASG in terms of ROUGE-L, which demonstrates the effectiveness of denoising module.

Denoising ability

Next, we turn to research question **RQ3**. Due to the fact that the denoising module is learned in a supervised way, there is a ground truth label associated with each comment. Thus when the predict salience score $\hat{\beta}_i > 0.5$ we classify it as a helpful comment and vice-versa. As the denoising module can be regarded as a binary classifier to classify each comment to $\hat{\beta}_i = 1$ or $\hat{\beta}_i = 0$, we calculate the classification recall score of comments to measure the performance of this module. The recall curve is shown in Figure 2. As the training progresses, the recall score is on a steady upward curve which proves the improved performance of denoising module. To conclude, the denoising module can give a meaningful salience score for the subsequent process.
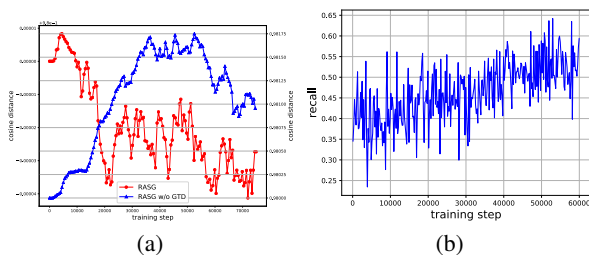
Figure 2: (a) Cosine distance between decoding attention and reader attention. (b) Recall score of denoising module.

Table 5: Consistency and fluency comparison by human evaluation.

	Fluency		Consistency	
	mean	variance	mean	variance
S2S	2.17	0.24	1.98	0.28
CGU	2.20	0.26	2.08	0.29
RASG	2.65▲	0.21	2.48▲	0.26

Analysis of goal tracker

In this section, we turn to research question **RQ4**. The main purpose of employing goal tracker is to help the summary generator utilize the reader focused aspect. Intuitively, we want to know whether the summary generator follows the goal set by the goal tracker. Therefore, we calculate the cosine distance between decoder attention $\nu_{T\hat{Y}} \in \mathbb{R}^{T^d}$ (in Equation 17) and reader attention $\epsilon \in \mathbb{R}^{T^d}$ (in Equation 16). In Figure 2, we compare the cosine distance between the ablation model RASG w/o GTD and RASG. RASG observes a decrease of cosine distance and conversely, the RASG w/o GTD observes an increment of cosine distance. The fact that RASG can narrow the cosine distance proves that goal tracker and discriminator can lead the generator to follow the reader focused aspect.

Human evaluation

We ask three highly-educated Ph.D. students to rate 100 generated summaries of different models according to consistency and fluency. These annotators are all native speakers. The rating score ranges from 1 to 3 and 3 is the best. We take the average score of all summaries as the final score of each model, as shown in Table 5. It can be seen that RASG outperforms other baseline models in both sentence fluency and consistency by a large margin. We calculate the kappa statistics in terms of fluency and consistency, and the score is 0.33 and 0.29 respectively. To prove the significance of the above results, we also do the paired student t-test between our model and CGU model (row with shaded background), the p-value are 0.0017 and 0.0012 for fluency and consistency respectively.

document	徐麟表示，中央网信办将提供政策支持建立健全国有资本进入培育互联网企业，完善互联网企业国内上市等相关政策；通过在新闻网站核发新闻记者证，开展从业人员教育培训等措施，努力打造新媒体平台的国家队和主力军。详见长微博(Lin Xu said that the Central Network Office will provide policy support to establish a nationwide capital to enter the cultivation of internet companies, improve the policies of domestic listing of internet companies; through the press on the news website to issue a press card, carry out education and training for practitioners, and strive to create new the national team and the main force of the media platform. See Long Weibo for details.)
comments	国有资本必须介入(State-owned capital must be involved.)
	中央网信办不要把举措落实在文件上，要切实的执行！(The Central Network Office should not implement the measures on the documents and must implement them in a practical way !)
	我觉得网信办要好好治治这些害群之马了(I feel that the Central Network Office should cure these black sheep.)
reference	网信办副主任：建立健全国有资本进入培育互联网企业(Deputy director of the Central Network Office: Establishing state-owned capital to cultivate internet enterprises)
S2S	徐麟：努力打造新媒体平台的国家队和国家队(Lin Xu : National team and national team working hard to build a new media platform)
CGU	长微博：中央网信办将提供政策支持建立健全国有资本(Long Weibo: The Central Network Office will provide policy support to establish state-owned capital.)
RASG	中央网信办：努力打造新媒体平台的国家队(Central Network Office: Working hard to build a national team of new media platform.)

Figure 3: Examples of the generated summary by RASG and other models.

Case analysis

Figure 3 shows a document and its corresponding summaries generated by different methods. We can observe that S2S does generate fluent summary. However, the generated aspect is contradictory to the focused aspect of reader or ground truth summary. Meanwhile, RASG overcomes this shortcoming by using goal vector and gap content given by goal tracker and supervisor at training stage, and produces the summary that is not only fluent but also consistent with main aspect of document.

Conclusion

In this paper, we propose a new framework named *reader-aware summary generator* (RASG) which aims to generate summaries for document from social media incorporating the reader comments. In order to capture the reader focused aspect, we design a reader attention component with a denoising module to capture the alignment between comments and document. We employ a supervisor to measure the semantic gap between generated summary and reader focused aspect. A goal tracker uses the information of semantic gap and the feature extracted by the discriminator to produce a goal vector to guide the summary generator. In our experiments, we have demonstrated the effectiveness of RASG and have found significant improvements over state-of-the-art baselines in terms of ROUGE and human evaluations.

Moreover, we have verified the effectiveness of each module in RASG for improving the summarization performance.

Acknowledgments

References

Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. 2016. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, 265–283.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Bansal, M., and Chen, Y.-C. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *ACL*.

Chen, X.; Gao, S.; Tao, C.; Song, Y.; Zhao, D.; and Yan, R. 2018. Iterative document representation learning towards summarization with polishing. In *EMNLP*, 4088–4097.

Chopra, S.; Auli, M.; and Rush, A. M. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *HLT-NAACL*.

Duchi, J. C.; Hazan, E.; and Singer, Y. 2010. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR* 12:2121–2159.

Gao, S.; Ren, Z.; Zhao, Y. E.; Zhao, D.; Yin, D.; and Yan, R. 2019. Product-aware answer generation in e-commerce question-answering. In *WSDM*.

Gu, J.; Lu, Z.; Li, H.; and Li, V. O. K. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *CoRR* abs/1603.06393.

Guo, J.; Lu, S.; Cai, H.; Zhang, W.; Yu, Y.; and Wang, J. 2018. Long text generation via adversarial training with leaked information. *CoRR* abs/1709.08624.

Hu, B.; Chen, Q.; and Zhu, F. 2015. Lcsts: A large scale chinese short text summarization dataset. In *EMNLP*.

Hu, M.; Sun, A.; and Lim, E.-P. 2007. Comments-oriented blog summarization by sentence extraction. In *CIKM*.

Hu, M.; Sun, A.; and Lim, E.-P. 2008. Comments-oriented document summarization: understanding documents with readers' feedback. In *SIGIR*.

Jadhav, A., and Rajan, V. 2018. Extractive summarization with swap-net: Sentences and words from alternating pointer networks. In *ACL*.

Li, P.; Bing, L.; Lam, W.; Li, H.; and Liao, Y. 2015. Reader-aware multi-document summarization via sparse coding. In *IJCAI*.

Li, P.; Wang, Z.; Lam, W.; Ren, Z.; and Bing, L. 2017. Salience estimation via variational auto-encoders for multi-document summarization. In *AAAI*.

Li, P.; Bing, L.; and Lam, W. 2017. Reader-aware multi-document summarization: An enhanced model and the first dataset. In *Proceedings of the Workshop on New Frontiers in Summarization*, 91–99.

Lin, J.; Sun, X.; Ma, S.; and Su, Q. 2018. Global encoding for abstractive summarization. In *ACL*.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Ma, S.; Sun, X.; Lin, J.; and Ren, X. 2018a. A hierarchical end-to-end model for jointly improving text summarization and sentiment classification. In *IJCAI*.

Ma, S.; Sun, X.; Lin, J.; and Wang, H. 2018b. Autoencoder as assistant supervisor: Improving text representation for chinese social media text summarization. In *ACL*.

Mihalcea, R., and Tarau, P. 2004. Textrank: Bringing order into text. In *EMNLP*.

Nallapati, R.; Zhai, F.; and Zhou, B. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *AAAI*.

Narayan, S.; Cohen, S. B.; and Lapata, M. 2018. Ranking sentences for extractive summarization with reinforcement learning. In *NAACL-HLT*.

Nguyen, M.-T.; Tran, C.-X.; Tran, D.-V.; and Nguyen, M.-L. 2016. Solscsum: A linked sentence-comment dataset for social context summarization. In *CIKM*.

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Sun, M.; Hsu, W. T.; Lin, C.-K.; Lee, M.-Y.; Min, K.; and Tang, J. 2018. A unified model for extractive and abstractive summarization using inconsistency loss. In *ACL*.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Tao, C.; Gao, S.; Shang, M.; Wu, W. C.; Zhao, D.; and Yan, R. 2018. Get the point of my utterance! learning towards effective responses with multi-head attention mechanism. In *IJCAI*.

Vinyals, O.; Fortunato, M.; and Jaitly, N. 2015. Pointer networks. In *NIPS*, 2692–2700.

Wang, L.; Yao, J.; Tao, Y.; Zhong, L.; Liu, W.; and Du, Q. 2018. A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization. In *IJCAI*.

Yang, Z.; Cai, K.; Tang, J.; Zhang, L.; Su, Z.; and Li, J.-Z. 2011. Social context summarization. In *SIGIR*.

Zhou, Q.; Yang, N.; Wei, F.; and Zhou, M. 2017. Selective encoding for abstractive sentence summarization. In *ACL*.

Zhou, Q.; Yang, N.; Wei, F.; and Zhou, M. 2018. Sequential copying networks. In *AAAI*.