Front. Comput. Sci., 2022, 16(6): 166333
https://doi.org/10.1007/s11704-021-0561-z

RESEARCH ARTICLE

Exploiting comments information to improve legal public opinion news abstractive summarization

Yuxin HUANG[1,2], Zhengtao YU (✉)[1,2], Yan XIANG[1,2], Zhiqiang YU[1,2], Junjun GUO[1,2]

1 Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China

2 Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Kunming 650500, China

Abstract Automatically generating a brief summary for legal-related public opinion news (LPO-news, which contains legal words or phrases) plays an important role in rapid and effective public opinion disposal. For LPO-news, the critical case elements which are significant parts of the summary may be mentioned several times in the reader comments. Consequently, we investigate the task of comment-aware abstractive text summarization for LPO-news, which can generate salient summary by learning pivotal case elements from the reader comments. In this paper, we present a hierarchical comment-aware encoder (HCAE), which contains four components: 1) a traditional sequenceto-sequence framework as our baseline; 2) a selective denoising module to filter the noisy of comments and distinguish the case elements; 3) a merge module by coupling the source article and comments to yield comment-aware context representation; 4) a recoding module to capture the interaction among the source article words conditioned on the comments. Extensive experiments are conducted on a large dataset of legal public opinion news collected from micro-blog, and results show that the proposed model outperforms several existing state-of-the-art baseline models under the ROUGE metrics.

Keywords legal public opinion news, abstractive summarization, comment, comment-aware context, case elements, bi-directional attention

1 Introduction

Legal public opinion news (LPO-news) summarization task is crucial for rapid and effective disposal of public opinion. Usually, the LPO-news contains specific case elements, which denote the prominent topics of the source article and also be the significant parts of the summary. Therefore, the core problem for the LPO-news summarization task is to identify the case elements and generate an element-aware context to assist with the decoder to generate a brief and salient summary.

Received November 23, 2020; accepted June 25, 2021

E-mail: ztyu@hotmail.com

Normally, the LPO-news generates a large number of comments on the social network, which usually pays close attention to the pivotal elements of the case. We argue that the case elements, which are the important parts of the source documents, are also the main aspects of the user comments. Intuitively, we can jointly model the source article and corresponding comments to help the model capture the important case elements and produce comment-aware context representation.

Recently, many neural sequence-to-sequence models have shown successful performance in abstractive text summarization task [1−4]. However, different from existing summarization approaches, there are two challenges to jointly modelling the source article and comments. One is how to identify the case elements from the noisy comments. Intuitively, most of the comments are informal, noisy and even do not contain any case elements. Hence, the proposed model should have the ability to filter noisy information and distinguish the crucial case elements. Another challenge is how to produce the comment-aware context by jointly considering the source article and the comments. In other words, the model should incorporate the comment information into the source article in an appropriate way to generate comments-aware context representation.

In seeking to address these challenges, this paper first introduces a high-quality LPO-news dataset, in which the sample contains a source article, a summary and several reader comments. The details of this dataset will be presented in Section 4. Then, an extended model based on seq2seq framework named hierarchical comment-aware encoder (HCAE) is proposed. A brief overview of the HCAE model is illustrated in Fig. 1.

The HCAE consists of four components in a hierarchical structure: first, two recurrent neural networks (RNNs) encoders are adopted to read the source article and comments in parallel and produce the high-level representation of each input; then, a denoising module based on selective gating mechanism is employed to filter the noisy information of comments and capture the case elements information; next, the source article and distilling comments representation are

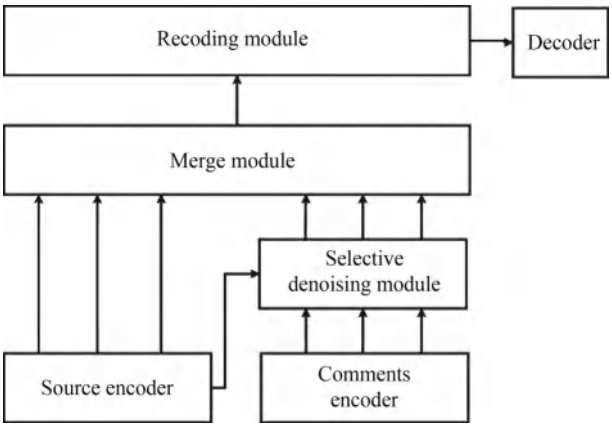

Fig. 1 The overview of HCAE model

coupled to produce the comment-aware representation; lastly, a recoding module is applied to further represent the complex interactions between the source article and comments and yield the final context representation. We show that our model significantly outperforms the existing attention-based seq2seq model on our dataset evaluated by the ROUGE metrics. Moreover, compared with several state-of-the-art baseline models, our model also achieves preferable performance. We further verify the effectiveness of each module through detailed ablation analysis.

Overall, the main contributions of this paper are summarized as follows: 1) we propose a novel model for LPO-news abstractive summarization task, in which the reader comments are utilized to produce the comment-aware representation of source article; 2) we propose a hierarchical comment-aware encoder to integrate comments information into the seq2seq model; and 3) we further conduct extensive experiments on a large dataset of LPO-news collected from micro-blog. The results demonstrate that the performance of our proposed model outperforms several state-of-the-art models.

2 Related works

Our work is mainly in the line of three directions: conventional neural summarization, comment-aware text summarization, and legal domain text summarization. We introduce them in turn below.

2.1 Conventional neural summarization

Recent works on text summarization can be grouped into two categories: extractive and abstractive methods. The extractive methods [5−8] aim at shortening the source article by selecting several significant sentences from the source article. Generally, the summaries generated by extractive methods have to originate from the source article, which limits the novelty and information coverage degree of the summaries. On the other hand, abstractive methods which compressing and rewriting the document into a short summary have achieved considerable success. Most previous works on abstractive summarization are based on seq2seq neural networks with attention mechanism, which is first introduced by Rush et al. [9]. Then, Nallapati et al. [1] further extends this model by using recurrent neural networks (RNNs) encoder-decoder architecture. This model also equips with several novel strategies including feature-rich encoder, hierarchical attention mechanism and outperforms other state-of-the-art models. To tackle the out-of-vocabulary (OOV) problem raising by the fixed vocabulary, the pointer-generator architecture is adopted which can copy words from source texts directly and generate novel words from source text via pointer mechanism [10−12]. See et al. [11] further proposes a coverage mechanism to solve the repetitive words of generated summaries. There are also several research directions such as improvement of the training strategies [13,14], improvement of the network structure of encoder and decoder [15,16] or joint training extraction and abstraction processes for the long documents [17,18].

2.2 Comment-aware text summarization

There are also some previous studies to investigate how to incorporate the comments in the text summarization task and most of them are taking the form of extractive paradigm. Graph-based extractive models have been proposed to improve the sentence selection process by constructing the relation graph between comments and sentences [19,20]. In this direction, there are also two reader-comments summarization datasets are released [21,22]. However, these datasets are not suitable for the abstractive model due to the small scale. In general, the works introduced above mainly pay attention to single document summary task (SDS). Another direction is to integrate reader-aware comments into multi-document summary task(RA-MDS). Li et al. [23] presents an unsupervised compressive framework via sparse coding model to reconstruct the semantic space of the topic and generate the summary. Further, they extends a variational auto-encoders (VAEs) to construct the latent semantic jointly considering news documents and reader comments [24]. For abstractive summarization task, Gao et al. [25,26] releases a large reader-aware abstractive summarization dataset collected from micro-blog and proposes a framework of adversarial training based on sequence-to-sequence model. By reducing the distance between the reader-focused aspect and the decoder-focused aspect, the model could learn the main aspect aware representation that the reader is interested.

2.3 Legal domain text summarization

Recently, summarization for legal case documents such as court judgements or contracts is a significant research direction [27,28]. Early studies mainly pay attention to extractive text summarization methods [29]. For instance, Kumar et al. [30] employs latent dirichlet allocation model (LDA) to discover latent topics of legal texts to generate a summary. Galgani et al. [31] proposes a hybrid rule-based approach to combine several different summarization techniques for legal case reports. There are also some works based on deep learning framework, which treat the summarization task as important sentences selection task. For instance, Acharya et al. [32] creates a mini-corpus of indian legal judgments and presents a unsupervised extractive summarization method based on the capsule networks and sentence embedding. They convert the summarization task into a case-related elements

(facts, arguments, evidences and judgements) recognition task, which enhances the readability of the summary. Elnaggar et al. [33] make use of multi-task learning including translation, summarization, and classification to overcome the data scarcity problem in the legal domain. Manor et al. [34] presents a new task of summarizing legal documents in plain English and introduces an initial evaluation dataset for this task. In a word, the current research in the legal-domain mainly focuses the legal documents, which are more normative and domain-oriented compared with the news on website. However, we pay close attention to LPO-news in this paper, which has two remarkable characteristics compared with previous researches, the LPO-news contains legal words and phrases, but the expression of LPO-news is not as professional as legal documents. There are also some works for the legal-related public opinion news. Han et al. [35] proposes an extractive method for LPO-news by utilizing the case elements extracted from the input news as the domain knowledge to guide summary extraction. They further presents an abstractive summarization method for LPO-news [36], which incorporates topic information as domain knowledge to improve the generated summary quality.

3 Model

3.1 Problem formulation

We now give a formal definition of comment-aware abstractive summarization task for LPO-news. Let $X^d = \left(x_1^d, x_2^d, \ldots, x_N^d\right)$ be a source article and N is the number of words in the article and x_n^d is the nth token. For each X^d, there are also several comments $X^c = (c_1, c_2, \ldots, c_M)$ where M is the number of comments and $c_m = \left(x_{m,1}^c, x_{m,2}^c, \ldots, x_{m,L}^c\right)$ is the mth comment and $x_{m,l}^c$ denotes the l-th word in mth comment. Our goal is to find a sequence of tokens $\hat{Y} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_T)$ that best generalizes the salient information as a summary by jointly modelling the news article X^d and corresponding comments set X^c. Finally, we use the difference between the generated summary $\hat{Y}$ and ground truth summary (the title of article) Y as the training signal to optimize the model parameters.

3.2 Encoders

The input of the model consists of two parts: source article X^d and comments set X^c. For each sequence, a bi-directional long short term memory network (BI-LSTM) [37] is employed to build its corresponding high-level representation. Take the source article as an example, the BI-LSTM encoder reads the sequence $X^d = \left(x_1^d, x_2^d, \ldots, x_N^d\right)$ and constructs the hidden states $H^d = \left(h_1^d, h_2^d, \ldots, h_N^d\right)$ with:

$$h_n^d = \text{BI-LSTM}\left(x_n^d, h_{n-1}^d\right), \tag{1}$$

where h_n^d denotes the hidden state of nth time step, which is generated by combining the states from both directions of the encoder.

For the comments set, the multiple independent comments are concatenated into a sequence as

$$X^c = \left(x_{1,1}^c, \ldots, x_{1,L}^c, sep, x_{m,1}^c, \ldots, x_{m,L}^c, sep, \ldots, x_{M,L}^c\right),$$

where sep denotes the special symbol to separate the different comments [1]. Unlike the source article encoder, different comments should be encoded independently. Hence, a boundary indicator $\gamma_{m,l}$ is introduced to separate the hidden states of different comments. Especially, the value $\gamma_{m,l}$ is defined as follows:

$$\gamma_{m,l} = \begin{cases} 0, & x_{m,l}^c = \text{sep}, \\ 1, & x_{m,l}^c \neq \text{sep}. \end{cases} \tag{2}$$

Then, $\gamma_{m,l}$ is used to reset the comments hidden states:

$$h_{m,l}^c = \gamma_{m,l} h_{m,l}^c, \tag{3}$$

where $h_{m,l}^c$ represents the hidden state of lth word in m-th comment, which is obtained in a similar way generated by BI-LSTM encoder.

In this way, the encoder of different comments are initialized with the zero vector, and the final representation of comments are described as $H^c = \left(h_{1,1}^c, \ldots, h_{1,L}^c, \ldots, h_{m,l}^c, \ldots, h_{M,L}^c\right)$. Specially, we define the last time step hidden state $h_{m,L}^c$ denotes the semantic representation of mth comment. Finally, both source article representation X^d and comments representation x^c are fed to the denoising module.

3.3 Denoising module

The goal of this module is to reduce the influence of noise comments and pay attention to the significant case elements. We argue that there are two characteristics should be considered in the selection of case elements: the case elements should be a significant part of the comments, but also should be closely related to the source article. Hence, a dual-channel selective denoising module is designed, as shown in Fig. 2. The denoising module contains two gates: comment-to-comment (C2C) gate and source-to-comment (S2C) gate. Essentially, the effect of denoising module is similar to soft case elements selection, which selects the vital case elements by assign different weights to different words of comments.

Inspired by the selective mechanism [3,38] in abstractive summarization task, the C2C gate is first adopted to capture the highlights information of each comment by utilizing the m-th comment final representation $h_{m,L}^c$:

$$g_{m,l}^c = \sigma\left(\mathbf{W}_{c1} h_{m,L}^c + \mathbf{W}_{c2} h_{m,l}^c + b_c\right), \tag{4}$$

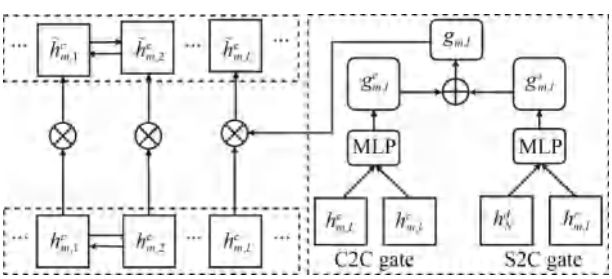

Fig. 2 The architecture of dual-channel selective denoising module

[1] In this paper, the "||" is adopted as the separator.

where $g^c_{m,l}$ denotes the relative importance of lth word in m-th comment compared with $h^c_{m,L}$, $\mathbf{W}_{c1}$, $\mathbf{W}_{c2}$ and b_c are learning parameters and σ is sigmod activation function.

Besides, the case elements in the comments should be closely related to the source article, therefore, the S2C gate is applied to filter the comments by the source article representation h^d_N as follows:

$$g^s_{m,l} = \sigma\left(\mathbf{W}_{s_{11}} h^d_N + \mathbf{W}_{s2} h^c_{m,l} + \boldsymbol{b}_s\right), \tag{5}$$

where $g^s_{m,l}$ measures the semantic similarity between the lth word in mth comment and source article representation h^d_N. $\mathbf{W}_{s1}$, $\mathbf{W}_{s2}$ and b_s are learning parameters and σ is also sigmod activation function.

Finally, the final gate weight $g_{m,l}$ is computed from a linear combination of $g^c_{m,l}$ and $g^s_{m,l}$ by following Equation:

$$g_{m,l} = \left(\mu_{m,l} g^c_{m,l} + (1-\mu_{m,l}) g^s_{m,l}\right), \tag{6}$$

here $\mu_{m,l} \in [0,1]$ is a soft switch to adjust the relative importance of two gate weights $g^c_{m,l}$ and $g^s_{m,l}$. There are two ways to get the parameter $\mu_{m,l}$: the first is to consider $\mu_{m,l}$ as a fixed hyper-parameter and gain the optimal value by manual adjustment; another alternative way is learned by the neural network automatically. In this paper, we choose the latter method to calculate $\mu_{m,l}$ as follows:

$$\mu_{m,l} = \sigma\left(\boldsymbol{w}^T \left[g^c_{m,l}; g^s_{m,l}\right]\right), \tag{7}$$

where the vector $\boldsymbol{w}$ is learnable parameters and σ is sigmod function. Finally, the gate weight $g_{m,l}$ is applied to calculate the final representation of comments as:

$$\tilde{h}^c_{m,l} = h^c_{m,l} \odot g_{m,l}, \tag{8}$$

where $\odot$ is element-wise multiplication operation. After the denoising module, we obtain the distilling representation of comments $\tilde{H}^c = \left(\tilde{h}^c_1, \tilde{h}^c_2, \ldots, \tilde{h}^c_\kappa\right)$ [2)].

3.4 Merge module

The core idea behind merge module is to couple the comments and source article to generate the comment-aware context representation. Let $Z = f\left(H^s, \tilde{H}^c\right)$ be the merge representation, and $f(\cdot)$ denotes the merge function which is responsible for joint modelling the interaction of source article H^s and comments $\tilde{H}^c$. In this work, we investigate three representative choices for function $f(\cdot)$.

3.4.1 Concatenation

Intuitively, we can directly concatenate the comments and source article representations. First, the max-pooling [39] operation is applied to the comments sequence:

$$\begin{aligned} \tilde{h}^c_{\max} &= \text{max-pooling}\left(\tilde{H}^c\right), \\ z_n &= \tanh\left(\mathbf{W}_d \left[h^d_n; \tilde{h}^c_{\max}\right]\right), \end{aligned} \tag{9}$$

where the vector $\tilde{h}^c_{\max}$ denotes the final representation of all comments. The final merged representation z_n is calculated by concatenating the $\tilde{h}^c_{\max}$ to the source document. $\mathbf{W}_d$ is trainable parameters, tanh is the activation function, and [;] denotes the concatenate operation.

3.4.2 Selective gate

Another choice $f(\cdot)$ is selective gate, where the comments representation $\tilde{h}^c_{\max}$ is supposed to be credible to generate gate weights jointly with h^d_n. The gate weights are used to control the source article information flow to the above layer. In other words, the merged representation is produced by using the comment information to filter the article representation:

$$\begin{aligned} sg_n &= \sigma\left(\mathbf{W}_{sg}\left[h^d_n; h^c_{\max}\right] + b_{sg}\right), \\ z_n &= h^d_n \odot sg_n. \end{aligned} \tag{10}$$

3.4.3 Bi-directional attention

Recently, Seo et al. [40] proposed a novel bidirectional attention flow mechanism (BiDAF) to construct a query-aware representation in machine comprehension (MC), which can modeling complex interactions between context and query. Inspired by this work, we apply a bi-directional attention mechanism to merge the source article and comments. As shown in Fig. 3, we consider the comments as the query and the source article as the context to construct the source-to-comment (S2C) attention and comment-to-source (C2S) attention.

We first construct a similarity matrix $\mathbf{S} \in \mathbb{R}^{N \times K}$ to measure the similarity between nth article words and kth comment words. Each element of $\mathbf{S}$ is calculated as follows:

$$S_{n,k=}\left(\mathbf{W}_s \left[h^d_n; \tilde{h}^c_k\right]\right), \tag{11}$$

where h^d_n and $\tilde{h}^c_k$ are the representation of n-th article and kth comment words respectively, and [;] denotes the concatenate operation. $\mathbf{W_s}$ is learnable parameter.

Then, the similarity matrix $\mathbf{S}$ is used to produce bi-directional attention. On one hand, we consider the comments as query and construct the source-to-comment (S2C) attention to identify which words in comment are most relevant to the nth word in the source article:

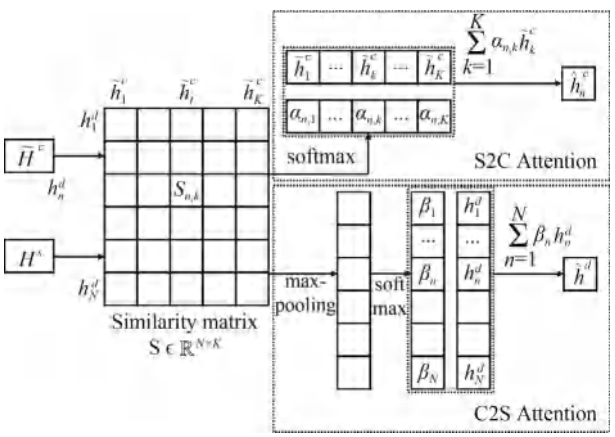

Fig. 3 The structure of the bi-directional attention module

[2)] For convenience, we treat the representation of comments as a sequence of length $K = M * L$ and the subscriptions m is omitted.

$$\alpha_n = \text{softmax}(\mathbf{S}_{n,:}),$$
$$\hat{h}_n^c = \sum_{t=1}^{T} \alpha_{n,k} \tilde{h}_k^c, \tag{12}$$

where $\mathbf{S}_{n,:}$ denotes the nth row vector of $\mathbf{S}$, the softmax function is performed across the nth row of similarity matrix to calculate the normalized attention weights of each comment words by nth source article words. Then $\hat{h}_n^c$ is calculated to represent the semantic representation of nth source article words by considering all the comment words. On another hand, the comment-to-source (C2S) attention is calculated to signify which article words have the closest similarity to each of the comment words and be critical for capturing the case elements, the C2S attention is calculated as follow:

$$\beta = \text{softmax}(\text{max-pooling}_{col}(\mathbf{S})),$$
$$\hat{h}^d = \sum_{n=1}^{N} \beta_n h_n^d, \tag{13}$$

where the attention distribution on the source article words $\beta \in \mathbb{R}^N$ is obtained by performing the max-pooling function across the column of the similarity matrix $\mathbf{S}$, then β is applied to weighted sum the article representation and produce the attention vector $\hat{h}_d$. In the end, the attended vector $\hat{h}_d$ is repeated N times across the column and record the final matrix as $\hat{H}_d$.

Finally, the comment-aware representation z_n is obtained by concatenating the attention vector and the output of the article encoder as:

$$z_n = \left(\mathbf{W}_h \left[\hat{h}^d; \hat{h}_n^c\right]\right), \tag{14}$$

where $\mathbf{W}_h$ is the learnable parameter, and Z denotes the merged representation of article and comments.

3.4.4 Recoding module

The role of the recoding module is to model the interaction among the article words conditioned on the comments. Intuitively, we can treat Z as a sequence of representations, and recurring all the representations with a BI-LSTM network:

$$\hat{z}_n = \text{BI-LSTM}(z_n, \hat{z}_{n-1}). \tag{15}$$

Then, $\hat{Z} = (\hat{z}_1, \hat{z}_2, \cdots, \hat{z}_N)$ is treated as the final context representation for the decoder.

3.5 Decoder

We use a conventional LSTM decoder with an attention mechanism to generate the summary [11,41]. At time step t, the decoding state s_t is calculated by the previously generated token y_{t-1}, the previous decoder state s_{t-1}, and the context vector c_{t-1}, as shown in Eq.(16):

$$s_t = \text{LSTM}(s_{t-1}, y_{t-1}, c_{t-1}). \tag{16}$$

Given the encoder hidden states $\hat{z}_n$ and the previous decoder hidden state s_{t-1}, the attention distribution $\tau_{t,n}$ over the input tokens and the context vector c_t are computed as follows:

$$\tau_{t,n} = \text{softmax}(\tanh(\mathbf{W}_a [\hat{z}_n; s_{t-1}])),$$
$$c_t = \sum_{n=1}^{N} \tau_{t,n} \hat{z}_n. \tag{17}$$

Lastly, the decoder state s_t and context vector c_t are combined and then we obtain the output distribution over the target vocabulary at time step t as:

$$p(y_t) = \text{softmax}\left(\mathbf{W}_p \tanh(\mathbf{W}_o [s_t; c_t])\right), \tag{18}$$

where $\mathbf{W}_o, \mathbf{W}_p$ are learnable parameters, [;] denotes the concatenated operation and tanh is the activation function.

We optimize the negative log-likelihood loss function between the generated summary $\hat{Y}$ and the ground-truth Y :

$$\mathcal{L} = -\sum \log p_\theta\left(\hat{Y} | X^d, X^c\right), \tag{19}$$

where the loss function is equivalent to maximizing the conditional probability of summary $\hat{Y}$ given parameters θ, source article X^d and corresponding comments X^c.

4 Experiments

To verify the performance of our model, a large legal public opinion summarization dataset is constructed. We describe our data collection process, experimental setup, and the baseline models in this section and present the results and detailed analysis in Section 5.

4.1 Data collection

The article-summary-comments pair are collected as training sample from MicroBlog to construct the dataset (search LawCommentSummary in github). In order to ensure that the samples belong to the legal field, we first select several MicroBlog sites (e.g., Peng Mei news, The Beijing News), which are mainly concerned with legal public opinion domain. Then the collected samples are labelled by using a domain correlation analysis algorithm [42] to distinguish whether they belong to the legal domain or not. The collected dataset is then cleaned by excluding three kinds of noise samples: 1) the words of article are fewer than 10; 2) the words of summary are fewer than 5; 3) the articles that do not have any comment. We also remove the sentences identical to the summary from the article. Furthermore, some standardized text preprocessing steps are applied, including tokenization, special character cleaning, replacing all digit characters with #, and replacing word seen less than five times with UNK. These steps result in a dataset with a total of 10k samples. We further randomly split the dataset into a training (109,301), a development (1,000) and a test set (1,000). We show the brief statistics information of our dataset in Table 1.

We further show the statistics related to the comments. As depicted in Fig. 4(a), we notice that each article has about 8.2 comments, 80% of articles have fewer than 10 comments. The

Table 1 Data statistics for our dataset. #(x) denotes the number of x, e.g., #(examples) is the number of samples of corresponding datasets. AvgArticleLen is the average input article length and AvgSummLen is the average summary length

Dataset	Training set	Validation set	Test set
#(examples)	109,301	1,000	1,000
#(articleWords)	7.74M	71.9K	71.2K
#(summWords)	1.32M	12.1K	12.2K
AvgArticleLen	70.8	71.86	71.21
AvgSummLen	12.08	12.11	12.20

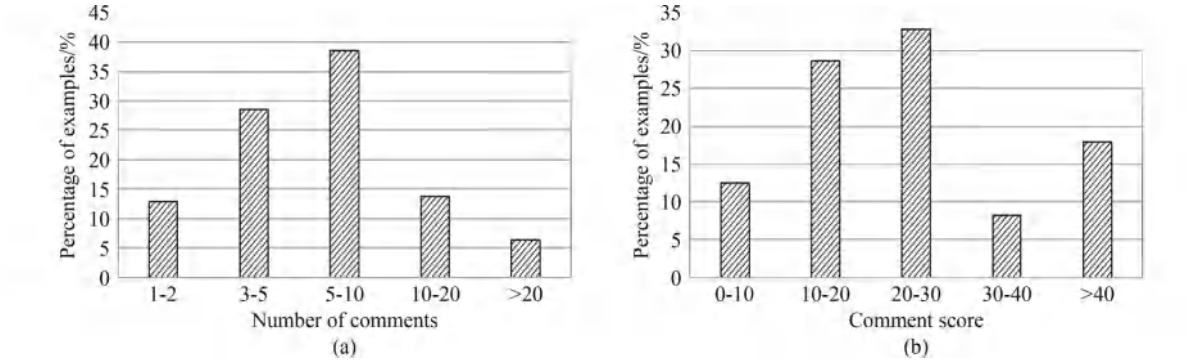

Fig. 4 Statistics of comments in LPO-news corpus. (a) The distribution of comments; (b) The distribution of comment scores

average comment length is about 200 words, 76% of the articles have comments words between 50 and 500.

In addition, we also estimate the quality of the comments using the ROUGE score (ROUGE-1) between the comments and the source document, which can measure how many words in the comments originate from the source document. In other words, if the words in the comment is covered by the source document, indicating that the comment is discussed around the case elements of the source document. Otherwise, it means that the comment is not related to the original document, which belongs to the low-quality comment. We brief this Rouge score as *comment score* to avoid naming conflicts with conventional ROUGE scores. As shown in Fig. 4(b), 60% comments achieve more than 20 comment score, indicating that most of the comments are related to the source document. Therefore, it should be feasible to mine useful information from comments to promote the performance of the model. However, we also notice that about 10% of the comments only scored below 10, indicating that the comments are noisy, and it's necessary to design an effective denoising module to utilize the comment.

4.2 Evaluation metrics

Following previous researches, we choose ROUGE scores to evaluate the performance of our model, which was first proposed by Lin [43]. ROUGE scores determine the quality of summarization by counting the number of overlapping units (i.e., *n*-grams, word sequence, and word pairs) between generated summaries and gold summaries. In this paper, all models are evaluated by using pyrouge script, which provides precision, recall, and F-score for these measures. Finally, the F-scores of ROUGE-1, ROUGE-2, and ROUGE-L (which respectively measure the overlapping of word, bigram and longest common sequence between the reference summary and the summary to be evaluated) are adopted as the evaluation metrics in following reported experimental results. Briefly, We abbreviate them as RG-1, RG-2 and RG-L, respectively.

4.3 Parameters settings

We implement our experiments using Pytorch [44] framework on a NVIDIA P100 GPU. For all experiments below, we employ 2-layer BI-LSTM for all encoders with hidden size to be 100; 1-layer LSTM for the decoder and set the hidden size to be 200. We train both the models for 20 epochs with mini-batch size of 32 and also use the typical teach-forcing strategy (the gold summarizes is used in training stage) during training stage. The encoder and decoder share the vocabulary and the size is set to 30K. Unknown words beyond the vocabulary are replaced with UNK. We use a pre-trained legal domain 100-dimensional word vectors which are provided by Hu [45] to initialize all word embedding layers in our model and empirically found that it can improve the performance of our model by RG-1 score. For the dropout operation, we set the *rate* = 0.5. Adam optimizer [46] with hyper-parameter β_1 = 0.9, β_2 = 0.999 and $\xi = 1e-3$ is used for stochastic optimization. Learning *rate* is fixed to $1e-3$. Gradient clipping is used with a maximum gradient norm of 5.0. We also apply a special training policy, the gradient decays with a rate of 0.9 from epoch 10. Note that the standard beam search strategy is adopted with beam size five during inference stage, and decoding process is stopped whenever *¡eos¿* token is predicted, or the output exceeds the maximum length of 30.

4.4 Baseline models

In order to investigate the performance of our proposed model, we introduce several competitive baselines to compare:

- RA Zeng et al. [15] proposes a read-again mechanism that reads the source article multiple times to improve the encoder representation.
- SEASS Zhou et al. [3] proposes a selective gate mechanism which built upon an attention-based encoder-decoder framework for the purpose of distilling salient information from source articles.
- CGU [47] A convolutional gated unit is used to refine the representation of the source sequence.
- PG [11] We run the baseline pointer-generator model, which includes the pointer and coverage mechanism and achieves state-of-the-art performance on short text summarization task.
- KIGN [48] extracts key information from the source document and integrates them into the decoding process by utilizing a multi-view attention mechanism.
- KCS [49] further improves the KIGN by introducing a co-selective module to incorporate the information of the source document and keywords.
- S2S_LSTM We also implement an LSTM based sequence-to-sequence model as our baseline and denote it as S2S.

- S2S_Transformer We also implement a self-attention based baseline model [50] based on Open-NMT framework [51] which has been shown better performance than LSTM. We denote it as S2S_Transformer. For the hyper-parameter setting of the transformer, we use the recommended configuration of Gigaword dataset (See OpenNMT in github).

5 Results and analysis

5.1 Research questions

We list four research questions to guide the experiments: RQ1: Does our model outperform other baseline models? RQ2: Is every module effective? RQ3: How about the performance of the different merge approaches? RQ4: Does the HCAE capture the vital case elements from noisy comments?

5.2 Overall performance

For research question RQ1, we examine the performance of HCAE in terms of ROUGE, and the results are presented in Table 2. In particular, we choose bidirectional attention (introduced in Section 3.4.3) as the merge approach of HCAE, which achieves the best performance among the three merge approaches. It is worth noticing that our model significantly outperforms all baseline models both in RG-1, RG-2, and RG-L. We argue that its superior performance stems from the better context representation obtained from the jointly modelling the article and comments. It is also noted that the performance of HCAE has been significantly improved compared with the S2S_LSTM model (+2.95 RG-1, +0.95 RG-2, +2.26 RG-L), which suggests the effectiveness of integrating the comment information into the S2S model to catch the case elements information. We also implement a Transformer model based on self-attention which has been proved to be more effective than LSTM. Yet, the transformer model may not be suitable for our dataset and has not achieved obvious performance improvement compared with LSTM model, even a slight decline in terms of RG-1 (−0.29). The possible reason is that the source article in our dataset is relatively short (about 70 words), the transformer model has no significant advantage over LSTM in short text. On the other hand, compared with three strong baseline models that focus on the improvement of the encoder including RA, SEASS, and CGU, our model also achieves better performance by incorporating comments to improve the ability of encoder. Furthermore, we compare with the PG model which has obtained the state-of-the-art performance on abstractive summarization task, our model also gained +0.81 improvement in term of RG-1. In addition, our model get a slight boost compared with the recent keywords-guide model KIGN and KCS, which extract words directly from the source document. Our conjecture is that the extracted keywords are spliced into a sequence for encoding, in which the irrelevant contexts of keyword are introduced, and it is difficult to generate high-quality representations. However, the keywords (case elements) learned from the comments can obtain better context representation and produce better performance by using the dual-channel selective denoising mechanism. Therefore, it is safe to conclude that integrating comments into the S2S framework has a significant contribution to the increase of ROUGE scores, and can yield higher quality context representation.

Table 2 Full-length ROUGE F1 evaluation results on the test set. All the ROUGE scores have a 95% confidence interval of at most ±0.5 as calculated by the official ROUGE script

Models	RG-1	RG-2	RG-L
RA	28.15	11.85	27.01
SEASS	28.54	12.11	27.35
CGU	28.37	12.34	27.31
PG	29.99	12.39	27.90
KIGN	30.26	12.31	28.04
KCS	30.41	12.18	28.27
S2S_LSTM	27.85	11.71	26.34
S2S_Transformer	27.56	11.58	27.18
HCAE	**30.80**	**12.66**	**28.60**

5.3 Ablation study

Next, we turn to the research question RQ2. To identify the performance of each module, we conduct ablation experiments and the ROUGE scores are listed in Table 3. What needs a special explanation is that the HCAE w/o Merging model directly concatenates the comments and source article to form a long sequence as the input of the recoding module. As shown in Table 3, all ablation models obtain lower scores compared with HCAE (full) model, which demonstrates that each module is effective and necessary. It is noteworthy that the HCAE w/o Merging shows a serious decline of 29.85%, 39.02% and 32.90% in terms of RG-1, RG-2, and RG-L, respectively. This also indicates the merging module plays a crucial role in the encoder process. As for the effectiveness of denoising module, there is a decrease of 18.64% from HCAE (full) to HCAE w/o Denosing in terms of RG-1, which reveals that the denoising module can filter the noise in the comments and assist with the model generate better context representation. Another interesting finding is that compared with the previous two modules, the recoding module has less impact (7.3% decrement in terms of RG-1 compared with HACE(full) model) on the performance of the model. We hypothesis the recoding module can further optimize the representation of input to obtain the interaction relation among the article and the comments.

5.4 Performance of different merge approaches

In Section 3.4, we also explore three alternative approaches to couple the representation of comments and source article. The comparison results are presented in Table 4. The most interesting finding is that, compared with the S2S_LSTM baseline model, all merge approaches achieve performance improvement, even though the selective gate method still gain

Table 3 Full-length ROUGE F1 evaluation results of different ablation models on the test set

Models	RG-1	RG-2	RG-L
w/o Denosing	25.06	10.37	24.33
w/o Merging	21.65	7.72	19.19
w/o Recoding	28.65	11.79	27.81
HCAE	**30.80**	**12.66**	**28.60**

Table 4 Full-length ROUGE F1 results of different merge approaches on the test set

Merge approaches	RG-1	RG-2	RG-L
Concatenation	30.46	12.14	28.32
Selective Gate	29.14	11.65	28.18
Bi-Directional Attention	**30.80**	**12.66**	**28.60**

an increment of +7.0% in terms of RG-L. This demonstrates that it's effective to integrate the comment into the S2S framework. Another important finding is that bi-directional attention achieves the best performance, while the performance of the selective gate mechanism is poor among the three merge approaches. We hold the superior performance of the former stems from the ability to better model the complex interaction of source article and corresponding comments. Yet, the possible reason for the poor performance of the latter is that the significant information in the source article may be filtered out due to the inaccurate comments.

5.5 Performance of different number of comments

The main contribution of our work is to utilize the user comments to force the summary model focusing on the main aspect of the source document. Therefore we further estimate the influence of different number of comments on the performance of the summary model. The results are listed in Table 5.

As shown in Table 5, We present the results with different comments selection strategies, where **Random** denotes randomly selecting comments, and **Comment Score** represents the comments are selected according to the comment score. **Num** means the amount of selections. First, it can be noticed that compared with the S2S_LSTM model (without using any comments), the model performance declines slightly when only one comment is randomly chosen, while the *Comment Score* strategy achieves improvements of 0.08, 0.21, and 0.43 in terms of RG-1, RG-2, and RG-L, respectively. We conjecture this happens because our proposed of learning case elements from comments depends on the quality of the comments, high-quality comments are beneficial to element extraction and summary generation. Furthermore, we can see that the *Random* strategy achieves lower performance when fewer comments are used, however, with the increase of the comments, the performance is gradually increasing. These results testify our hypothesis that when there are only a few randomly selected comments, it's difficult for our model to learn useful information from irrelevant comments. However, although introducing more comments may bring noisy data, useful comments may also be selected, which are useful for the summary model. In addition, this also verifies the effectiveness of our proposed denoising model in Section 3.3, which can well distill useful information from noisy comments. For *comment score* strategy, we can find that choosing fewer high-quality comments can achieve better performance. For instance, when the comment number is set 5, our model obtains the best performance of 31.24 in terms of RG-1, which even better than HCAE model (using all comments). In addition, it can be found that the performance gap of different comment selection strategies gradually narrows when more comments are chosen. In short, it is evident from the table that high-quality comments are useful for the summary model, and the number of the comment is not as important as the quality.

Table 5 Full-length ROUGE F1 results of different merge approaches on the test set

Num	Random			Comment score		
	RG-1	RG-2	RG-L	RG-1	RG-2	RG-L
1	26.24	10.77	25.69	27.93	11.92	26.77
3	28.86	11.06	26.47	30.28	12.50	28.09
5	29.57	11.64	27.15	31.24	12.92	29.17
10	30.11	12.37	28.42	30.98	12.90	28.78
20	30.24	12.63	28.39	30.69	12.75	28.46

6 Conclusion

In this paper, we propose a novel model of integrating comments into sequence-to-sequence framework to capture the vital case elements for the LPO-news abstractive summarization task. Experimental results demonstrated that the introduction of comments indeed enhance the model ability of producing better context representation for legal news summarization task and achieve state-of-the-art performance on the collected legal public opinion news dataset.

There are several promising future directions. First, the extraction of case elements in this work can be regarded as a soft feature selection process, thus utilizing explicit case elements to guide the summary generation may lead to better performance. Another consideration improvement may be given that the case elements information learning from the reader comments can be directly used as the domain knowledge to guide the decoding process instead of using in encoder.

Acknowledgements This work was supported by the National Key Research and Development Program of China (2018YFC0830105, 2018YFC0830101, 2018YFC0830100); the National Natural Science Foundation of China (Grant Nos. 61972186, 61762056, 61472168); the Yunnan Provincial Major Science and Technology Special Plan Projects (202002AD080001); the General Projects of Basic Research in Yunnan Province (202001AT0 70046, 202001AT070047).

References

1. Nallapati R, Zhou B W, Santos D C, Guçehre Ç, Xiang B. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In: Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning. 2016, 280–290
2. Gu J T, Lu Z D, Li H, Li V O. Incorporating copying mechanism in sequence-to-sequence learning. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2016, 1631−1640
3. Zhou Q Y, Yang N, Wei F R, Zhou M. Selective encoding for abstractive sentence summarization. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017, 1095−1104
4. Xu H Y, Wang Z Q, Zhang Y F, Weng X L, Wang Z J, Zhou G D. Document structure model for survey generation using neural network. Frontiers of Computer Science, 2021, 15(4): 1–10
5. Jadhav A, Rajan V. Extractive summarization with SWAP-NET: Sentences and words from alternating pointer networks. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2018, 142–151

6. Wang H, Wang X, Xiong W H, Yu M, Guo X X, Chang S Y, Wang W Y. Self-supervised learning for contextualized extractive summarization. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019, 2221−2227
7. Cho S W, Lebanoff L, Foroosh H, Liu F. Improving the similarity measure of determinantal point processes for extractive multi-document summarization. 2019, arXiv preprint arXiv: 1906.00072
8. Zhao W X, Wen J R, Li X M. Generating timeline summaries with social media attention. Frontiers of Computer Science, 2016, 10(4): 702–716
9. 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. 2015, 379–389
10. Vinyals O, Fortunato M, Jaitly N. Pointer networks. Advances in neural information processing systems, 2015, 2692−2700
11. See A, Liu P J, Manning C D. Get to the point: Summarization with pointer-generator networks. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017, 1073−1083
12. Song K Q, Zhao L, Liu F. Structure-infused copy mechanisms for abstractive summarization. 2018, arXiv preprint arXiv: 1806.05658
13. Zhang X X, Lapata M. Sentence simplification with deep reinforcement learning. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. 2017, 584–594
14. Pasunuru R, Bansal M. Multi-reward reinforced summarization with saliency and entailment. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). 2018, 646–653
15. Zeng W Y, Luo W J, Fidler S, Urtasun R. Efficient summarization with read-again and copy mechanism. 2016, arXiv preprint arXiv: 1611.03382
16. Xia Y C, Tian F, Wu L J, Lin J X, Qin T, Yu N H, Liu T Y. Deliberation networks: Sequence generation beyond one-pass decoding. Advances in Neural Information Processing Systems, 2017, 1784−1794
17. Chen Y C, Bansal M. Fast abstractive summarization with reinforce-selected sentence rewriting. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2018, 675–686
18. Hsu W T, Lin C K, Lee M Y, Min K R, Tang J, Sun M. A unified model for extractive and abstractive summarization using inconsistency loss. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2018, 132–141
19. Hu M S, Sun A X, Lim E P. Comments-oriented document summarization: understanding documents with readers' feedback. In: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. 2008, 291–298
20. Yang Z, Cai K K, Tang J, Zhang L, Su Z, Li J Z. Social context summarization. In: Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval. 2011, 255–264
21. Nguyen M T, Tran C X, Tran D V, Nguyen M L. Solscsum: A linked sentence-comment dataset for social context summarization. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. 2016, 2409−2412
22. Nguyen M T, Lai D V, Do P K, Tran D V, Le Nguyen M. Vsolscsum: Building a vietnamese sentence-comment dataset for social context summarization. In: Proceedings of the 12th Workshop on Asian Language Resources (ALR12). 2016, 38–48
23. Li P J, Bing L D, Lam W, Li H, Liao Y. Reader-aware multi-document summarization via sparse coding. In: Proceedings of Twenty-Fourth International Joint Conference on Artificial Intelligence. 2015
24. Li P J, Bing L D, Lam W. Reader-aware multi-document summarization: An enhanced model and the first dataset. In: Proceedings of the Workshop on New Frontiers in Summarization. 2017, 91–99
25. Gao S, Chen X Y, Li P J, Ren Z C, Bing L D, Zhao D Y, Yan R. Abstractive text summarization by incorporating reader comments. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2019, 6399−6406
26. Gao S, Chen X Y, Ren Z C, Zhao D Y, Yan R. From standard summarization to new tasks and beyond: Summarization with manifold information. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20. 2020, 4854−4860
27. Bhattacharya P, Hiware K, Rajgaria S, Pochhi N, Ghosh K, Ghosh S. A comparative study of summarization algorithms applied to legal case judgments. In: Proceedings of European Conference on Information Retrieval. 2019, 413–428
28. Jain D, Borah M D, Biswas A. Summarization of legal documents: Where are we now and the way forward. Computer Science Review, 2021, 40: 100388
29. Hachey B, Grover C. Extractive summarisation of legal texts. Artificial Intelligence and Law, 2006, 14(4): 305–345
30. Kumar R, Raghuveer K. Legal document summarization using latent dirichlet allocation. Int. J. of Computer Science and Telecommunications, 2012, 3: 114–117
31. Galgani F, Compton P, Hoffmann A. Combining different summarization techniques for legal text. In: Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data. 2012, 115–123
32. Acharya H R, Bhat A D, Avinash K, Srinath R. Legonet-classification and extractive summarization of indian legal judgments with capsule networks and sentence embeddings. Journal of Intelligent & Fuzzy Systems, 2020(Preprint): 1–10
33. Elnaggar A, Gebendorfer C, Glaser I, Matthes F. Multi-task deep learning for legal document translation, summarization and multi-label classification. In: Proceedings of the 2018 Artificial Intelligence and Cloud Computing Conference. 2018, 9–15
34. Manor L, Li J J. Plain English summarization of contracts. In: Proceedings of the Natural Legal Language Processing Workshop 2019. 2019, 1–11
35. Han P Y, Gao S X, Yu Z T, Huang Y X, Guo J J. Case-involved public opinion news summarization with case elements guidance. Journal of Chinese Information Processing, 2020, 34(5): 56–63
36. Huang Y X, Yu Z T, Guo J J, Yu Z Q, Xian Y T. Legal public opinion news abstractive summarization by incorporating topic information. International Journal of Machine Learning and Cybernetics, 2020: 1–12
37. Hochreiter S, Schmidhuber J. Lstm can solve hard long time lag problems. Advances in neural information processing systems, 1997, 473–479
38. Wang K, Quan X J, Wang R. BiSET: Bi-directional selective encoding with template for abstractive summarization. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019, 2153−2162
39. Kalchbrenner N, Grefenstette E, Blunsom P. A convolutional neural network for modelling sentences. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2014, 655–665
40. Seo M, Kembhavi A, Farhadi A, Hajishirzi H. Bidirectional attention flow for machine comprehension. 2016, arXiv preprint arXiv: 1611.01603
41. Gulcehre C, Ahn S, Nallapati R, Zhou B W, Bengio Y. Pointing the unknown words. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2016, 140–149
42. Zhang Y, Yu Z T, Mao C L, Huang Y X, Gao S X. Correlation analysis of law-related news combining bidirectional attention flow of news title and body. Journal of Intelligent & Fuzzy Systems, (Preprint): 1–13
43. Lin C Y. ROUGE: A package for automatic evaluation of summaries.

Text Summarization Branches Out, 2004, 74–81
44. Adam P, Sam G, Soumith C, Gregory C, Edward Y, Zachary D, Ze-Ming L, Alban D, Luca A, Adam L. Automatic differentiation in pytorch. In: Proceedings of Neural Information Processing Systems. 2017
45. Hu Z K, Li X, Tu C C, Liu Z Y, Sun M S. Few-shot charge prediction with discriminative legal attributes. In: Proceedings of the 27th International Conference on Computational Linguistics. 2018, 487–498
46. Kingma D P, Ba J. Adam: A method for stochastic optimization. 2014, arXiv preprint arXiv: 1412.6980
47. Lin J Y, Sun X, Ma S M, Su Q. Global encoding for abstractive summarization. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2018, 163–169
48. Xu W R, Li C L, Lee M H, Zhang C. Multi-task learning for abstractive text summarization with key information guide network. EURASIP Journal on Advances in Signal Processing, 2020, 2020: 1–11
49. Li H R, Zhu J N, Zhang J J, Zong C Q, He X D. Keywords-guided abstractive sentence summarization. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 8196−8203
50. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser L, Polosukhin I. Attention is all you need. Advances in Neural Information Processing Systems 30, 2017, 5998-6008
51. Klein G, Kim Y, Deng Y T, Nguyen V, Senellart J, Rush A. OpenNMT: Neural machine translation toolkit. In: Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers). 2018, 177–184

Yuxin Huang is a PhD candidate in computer science at Kunming university of Science and Technology, China. His research interests include natural language processing, text summarization, machine translation, etc.

Zhengtao Yu received the PhD degree in computer application technology from Beijing Institute of Technology, China in 2005. Now he is a professor and PhD supervisor at Kunming University of Science and Technology, China and the director of Yunnan Key Laboratory of Artificial Intelligence. His research interests include natural language processing, machine translation and information retrieval, etc.

Yan Xiang received the MS degree from Wuhan University, China in 2001. She is currently a PhD candidate in computer science at Kunming University of Science and Technology, China. Her research interests include medical image processing, natural language processing, sentiment classification, and text mining, etc.

Zhiqiang Yu is a PhD candidate in computer science at Kunming university of Science and Technology, China. His research interests include natural language processing, neural machine translation, etc.

Junjun Guo received the PhD degree from Xi'an Jiao Tong University, China in 2016. Now He is an associate professor at Kunming University of Science and Technology, China. His research interests include natural language processing, machine translation, etc.