A Joint Selective Mechanism for Abstractive Sentence Summarization

Junjie Fu TIANZHIYINYI@SJTU.EDU.CN
Gongshen Liu[*] LGSHEN@SJTU.EDU.CN
School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University

Editors: Jun Zhu and Ichiro Takeuchi

Abstract

Sequence-to-sequence (Seq2Seq) learning framework has been widely used in many natural language processing (NLP) tasks, including abstractive summarization and machine translation (MT). However, abstractive summarization generates the output in a *lossy manner*, in comparison with MT which is almost *loss-less*. We model this by introducing a joint selective mechanism: (i) A selective gate is added after encoding phase of the Seq2Seq learning framework, which learns to tailor the original input information and generates a selected input representation. (ii) A selection loss function is also added to help our selective gate function well, which is computed by looking at the input and the output jointly. Experimental results show that our proposed model outperforms most of the baseline models and is comparable to the state-of-the-art model in automatic evaluations.

Keywords: Abstractive Summarization, Sequence-to-sequence Model, Selective Mechanism, Selection Loss

1. Introduction

Text summarization aims to condense the original text while keeping the salient information. It can be divided into two types: extractive and abstractive. Extractive methods simply take sentences from source text to construct the final summary (Woodsend and Lapata, 2010; Cao et al., 2015; Cheng and Lapata, 2016). However, direct extraction can only copy the partial content of original source text, and if the original content has no proper parts to convey the main idea of the text, extractive methods can hardly produce high-quality summaries.

Abstractive methods read full text and try to understand it so that they can capture the key points from a higher level, and then, they create summaries by selecting words from source text or generating words which do not appear in source text. Though abstractive methods generate summaries just like humans, it is more difficult to realize in comparison with extractive ones. In this work we focus on abstractive sentence summarization task.

Recently, neural Seq2Seq models have achieved remarkable progress in this task. (Rush et al., 2015) utilize an attention-based model for generating abstractive summaries inspired by the developments in neural machine translation. (Zeng et al., 2016) produce a representation of each word by reading article twice aiming at avoiding yielding suboptimal

[*] Corresponding author.

representations. (Zhou et al., 2017) create a tailored representation of each input word by adding a selective gate after encoding phase.

Many of existing models for abstractive summarization are enlightened by MT directly or indirectly. However, though both abstractive summarization and MT can be achieved using Seq2Seq models, there are some differences between the two tasks. One of the most important differences is that the former only needs to convey the main idea of the input, and some words themselves or even their meanings will not appear in the output, while the latter needs to make an explicit alignment between the input and the output so as to get the exact translation. (Zhou et al., 2017) noticed this problem and proposed a selective mechanism to model the distilling process in abstractive summarization.

In this work, we also introduce a joint selective mechanism to model the *lossy manner* of abstractive summarization: (i) A selective gate is computed after encoding by reading the word representation directly, which is composed of the word embedding and its bidirectional hidden representation, in comparison with (Zhou et al., 2017)'s work which calculates the selective gate by incorporating the original encoder outputs and the sentence representation. (ii) A selection loss function is added which is computed by looking at the input and the output jointly to help the selective gate function well. The reason why we set it this way is that we assume joint consideration of the input and the output can make selection more accurate.

2. Related Work

Abstractive sentence summarization generates a summary at sentence level, and it can be used for sentence fusing after getting extracted sentences from document-level extractive summarization. (Rush et al., 2015) were the first to utilize neural models to the task of abstractive text summarization. They generate summaries in an encode-attend-decode paradigm, which uses an attention-based encoder and a neural network language decoder. Following their work, (Chopra et al., 2016) replaced the decoder with an RNN decoder and (Nallapati et al., 2016) designed an RNN encoder-decoder model for text summarization.

(Zeng et al., 2016) tackled the problem that the encoders only read the history of the sequence when computing the representation of a word by proposing a read-again mechanism. This mechanism reads the input twice and produces a second read representation according to the first read representation. (Zhou et al., 2017) created a selective gate network to generate a second level representation of each word, which can help capture highlights in the original. (Lin et al., 2018) proposed a global encoding framework to control the information flow from the encoder to the decoder. (Ma et al., 2018) took a summary autoencoder to supervise the learning of the Seq2Seq model. Our work is enlightened by (Zhou et al., 2017)'s work mostly, but we consider the relation between the input sentence and its summary comprehensively, in consideration of the fact that one can hardly learn to select the highlights in a sentence if he has never seen a number of original long sentences and their summaries jointly before.

3. RNN-Based Seq2Seq Attentional Model

The RNN-based Seq2Seq attentional model has been a commonly used baseline for the task of abstractive summarization, which consists of an encoder and an attentional decoder.
Encoder: Let $x = \{x_1, x_2, ..., x_n\}$ be a sequence of input words, and the goal of the encoder is to generate a representation of the input sequence. Here we utilize a bidirectional RNN to construct the encoder, which is implemented using gated recurrent unit (GRU) (Cho et al., 2014):

$$\overrightarrow{h}_i = \text{GRU}(x_i, \overrightarrow{h}_{i-1}) \quad (1)$$

$$\overleftarrow{h}_i = \text{GRU}(x_i, \overleftarrow{h}_{i-1}) \quad (2)$$

By using bidirectional GRU, we can get the forward outputs $(\overrightarrow{h}_1, \overrightarrow{h}_2, ..., \overrightarrow{h}_n)$ and the backward outputs $(\overleftarrow{h}_1, \overleftarrow{h}_2, ..., \overleftarrow{h}_n)$. For each word x_i, its representation h_i is constructed by simply concatenating the bidirectional outputs:

$$h_i = \overrightarrow{h}_i \oplus \overleftarrow{h}_i \quad (3)$$

where operator $\oplus$ indicates concatenation.
Attentional Decoder: A unidirectional GRU with attention mechanism is used as the decoder, and on each time step t, the decoder generates current hidden state s_t by reading the word embedding of the previous word w_{t-1} and the previous context vector c_{t-1}. The decoder hidden state is initialized using the last backward encoder hidden state:

$$s_t = \text{GRU}(w_{t-1}, c_{t-1}, s_{t-1}) \quad (4)$$

$$s_0 = \tanh(\mathbf{W}_d \overleftarrow{h}_1 + b) \quad (5)$$

The context vector c_t at each time step is a weighted sum of the encoder outputs and the weights can be computed through attention mechanism (Bahdanau et al., 2014):

$$e_i^t = v^\top \tanh(\mathbf{W}_e h_i + \mathbf{U}_e s_t) \quad (6)$$

$$\alpha^t = \text{softmax}(e^t) \quad (7)$$

$$c_t = \sum_{i=1}^{n} \alpha_i^t h_i \quad (8)$$

where $\mathbf{W}_e$ and $\mathbf{U}_e$ are weight matrices.

Finally, the probability of the next word is generated by passing a linear transforming of the word embedding of the previous word w_{t-1}, the context vector c_t and the current hidden state s_t through a softmax layer, and a maxout layer (Goodfellow et al., 2013) is also used after getting o_t as (Zhou et al., 2017) does.

$$o_t = \mathbf{W}_o w_{t-1} + \mathbf{U}_o c_t + \mathbf{V}_o s_t \quad (9)$$

$$m_t = [\max\{o_{t,2j-1}, o_{t,2j}\}]_{j=1,...,d}^\top \quad (10)$$

$$p(y_t|y_1, ..., y_{t-1}) = \text{softmax}(\mathbf{W}_m m_t) \quad (11)$$

where $\mathbf{W}_o$, $\mathbf{U}_o$, $\mathbf{V}_o$ and $\mathbf{W}_m$ are weight matrices. o_t is a $2d$-dimensional vector.

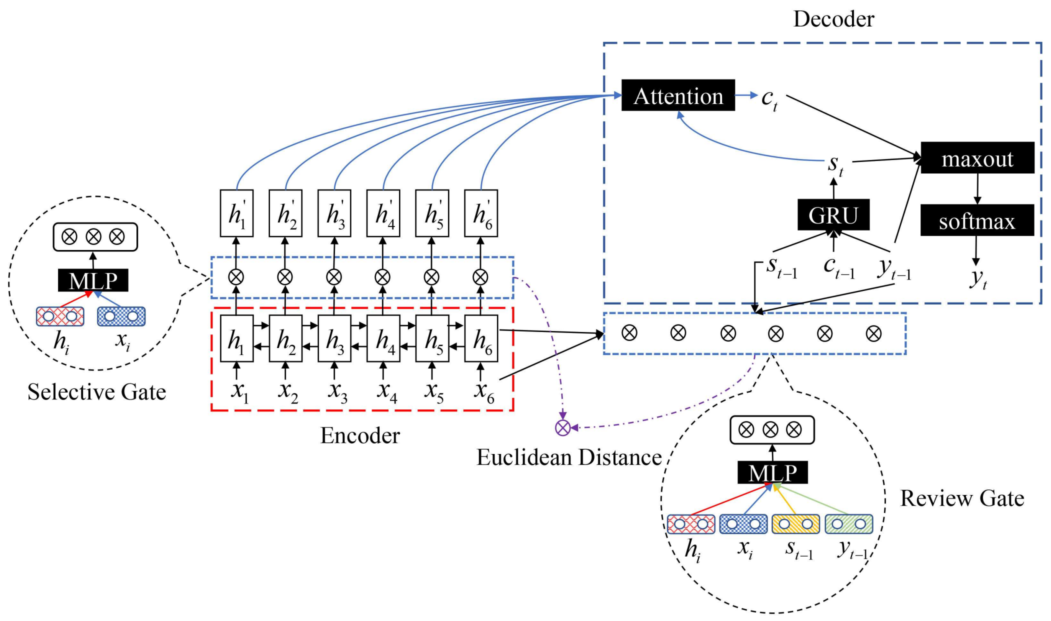
Decoder
Attention
c_t
s_t
maxout
softmax
GRU
s_{t-1}
c_{t-1}
y_{t-1}
y_t
h'_1 h'_2 h'_3 h'_4 h'_5 h'_6
h_1 h_2 h_3 h_4 h_5 h_6
x_1 x_2 x_3 x_4 x_5 x_6
MLP
h_i x_i
Selective Gate
Encoder
Euclidean Distance
Review Gate
h_i x_i s_{t-1} y_{t-1}

Figure 1: An overview of our proposed model. Each time a review gate is computed towards the original encoder outputs and the Euclidean distance between the selective gate and the mean of the review gate at each time step will be treated as an extra loss function.

4. Proposed Model

In this section, we introduce our proposed model which is based on the RNN baseline model described above. Figure 1 shows an overview of our proposed model.

4.1. Selective Mechanism

In consideration of the *lossy manner* existing in the task of abstractive summarization, a selective mechanism is introduced to model it. (Zhou et al., 2017) has proposed a selective mechanism before, and in their model, the selective mechanism combines each encoder output and the input sentence representation generated by concatenating the last forward hidden state and backward hidden state together to compute a selective gate, and then a new encoder output can be created by element-wise multiplication between the original encoder output and the selective gate. This process is just like letting the network know which words in the input are important for generating a summary only by reading the input text, while in this paper, we let the network learn to select by looking at the input and the output jointly, and the selective gate is computed as follows:

$$g_i = \sigma(\mathbf{W}_g h_i + \mathbf{U}_g u_i) \tag{12}$$

$$h'_i = h_i \odot g_i \tag{13}$$

where $\mathbf{W}_g$, $\mathbf{U}_g$ are weight matrices, σ denotes sigmoid activation function, and $\odot$ is element-wise multiplication. Here we combine the word embedding of the input word u_i and its corresponding original encoder output h_i to compute a gate vector g_i. The word embedding u_i represents the meaning of the word itself and the original encoder output h_i represents potentially large input context, and both are taken into consideration for selection.

After getting new encoder outputs h'_i, an input context vector can be computed through attention mechanism using h'_i and current decoder hidden state s_t.

4.2. Selection Loss Function

For a person who knows many words' meanings and can read long sentences, he may hardly know how to select the highlights of the original long sentences easily unless he has seen many summaries of other long sentences before. He needs to read the summary and review the original text to learn what information will be salient for summarization. Thus we compute a review gate at each time step as follows:

$$r_{i,t} = \sigma(\mathbf{W}_r h_i + \mathbf{U}_r u_i + \mathbf{V}_r s_{t-1} + \mathbf{Q}_r w_{t-1}) \tag{14}$$

where $\mathbf{W}_r$, $\mathbf{U}_r$, $\mathbf{V}_r$ and $\mathbf{Q}_r$ are weight matrices. The embedding w_{t-1} is the output of the last decoding time step, which represents the meaning of the word itself. The hidden state s_{t-1} is the hidden state of the last decoding time step, which represents the potential context before generating the word w_{t-1}.

Computing the review gate this way is just like a person reading the summary and reviewing the input text so as to learn to select. The mean of the review gate at each time step is calculated to get the final review gate r_i, and then the Euclidean distance between the selective gate and the final review gate is treated as an extra loss function:

$$r_i = \frac{1}{m}\sum_{t=2}^{m+1} r_{i,t} \tag{15}$$

$$d(g, r) = \frac{1}{n}\sum_{i=1}^{n} |r_i - g_i| \tag{16}$$

where m is the length of the output sequence and n is the length of the input sequence. The review gate of the first time step $r_{i,1}$ is ignored, because it is generated using the SOS token.

5. Objective Function

We adopt commonly used negative log-likelihood (NLL) loss function as well as our proposed selection loss function as our learning goal:

$$L = -p(y|x, \theta) + \lambda d(g, r) \tag{17}$$

where $p(y|x)$ is the conditional probability of summaries given input texts. $d(g, r)$ is the Euclidean distance between the selective gate and the final review gate. Minimizing $d(g, r)$ can make the selection more accurate because the review gate considers the source input and the target output comprehensively. λ is a tunable hyperparameter to balance the selection loss and the NLL loss.

6. Experiments

6.1. Datasets

We conduct our experiments on English dataset and Chinese dataset both.

English Dataset: For training, we use the Annotated English Gigaword corpus (Napoles et al., 2012) as with (Rush et al., 2015). The parallel corpus is produced by pairing the first sentence and the headline in the news article with heuristic rules. The training set and the development set can be constructed with the script[1]. The script helps preprocess the data using various basic text normalization, including tokenization, lower-casing, replacing all digit characters with #, and replacing word types seen less than 5 times with a UNK tag. Finally we get 3.8M sentence-summary pairs for training and 189K pairs for the development set.

For testing, we use the same test sets as (Zhou et al., 2017), including the English Gigaword, DUC 2004 and the Microsoft Research Abstractive Text Compression test sets. The English Gigaword test set are originally used by (Rush et al., 2015), which contains 1951 sentence-summary pairs after removing pairs with empty titles. As this test set still has some invalid lines, an internal test set with 2000 pairs is also used which is sampled by (Zhou et al., 2017). The DUC 2004 test set (Over et al., 2007) includes 500 pairs, and each pair is composed of one source sentence and four reference summaries. The Microsoft Research Abstractive Text Compression (Toutanova et al., 2016) test set contains 785 source sentences with multiple manually-created summaries, and we denote this test set as MSR-ATC in the following.

Chinese Dataset: Our Chinese dataset is Large Scale Chinese Short Text Summarization Dataset (LCSTS) (Hu et al., 2015). This dataset contains more than 2.4M text-summary pairs, which are constructed from a popular Chinese social media website named Sina Weibo[2]. There are three parts of it, and we use PART I as training set, PART II as development set and PART III as test set. PART II and PART III are manually annotated with relevant scores ranged from 1 to 5, and we only keep pairs with scores no less than 3 following (Ma et al., 2017).

6.2. Experimental Setting

For English dataset, as the dataset has already been preprocessed, we simply extract words from the training set to construct vocabularies, resulting in a 120K size source vocabulary and a 69k size target vocabulary. The embedding size and the GRU hidden size are set to 300 and 512 respectively. We use dropout (Srivastava et al., 2014) with probability 0.4. For Chinese dataset, in consideration of the risk of word segmentation mistakes (Xu and Sun, 2016), we train the Chinese dataset in character-level. The vocabulary size is limited to 4000 for both the source input and the target output. The settings of the embedding size, the GRU hidden size and the dropout probability are the same as the settings of English dataset. The parameter λ is 1. Following previous work, we evaluate the test performance with the commonly used ROUGE-1, ROUGE-2, ROUGE-L (Lin, 2004). A beam search

1. https://github.com/facebookarchive/NAMAS
2. https://weibo.com/

optimization (Koehn, 2004) is also used for decoding and the beam size is set to 5 for the following experiments.

6.3. Baselines

We compare our proposed model with following baselines (baselines for the English dataset and the Chinese dataset are both listed):
ABS: (Rush et al., 2015) use an attentive CNN encoder and neural network language decoder for abstractive summarization.
ABS+: (Rush et al., 2015) further tune their model using additional dataset, leading to improvements on the test performance.
Luong-NMT: A two-layer LSTM neural machine translation model proposed by (Luong et al., 2015).
SEASS: (Zhou et al., 2017) propose a selective mechanism after encoding, which helps produce a second sentence representation.
RNN and RNN context: Two Seq2Seq baselines with bidirectional GRU encoder and unidirectional GRU decoder provided by (Hu et al., 2015).
CopyNet: (Gu et al., 2016) incorporate copying mechanism into Seq2Seq learning framework to solve out-of-vocabulary problem.
SRB: (Ma et al., 2017) propose a model which tries to maximize the semantic similarity between texts and generated summaries.
DRGD: (Li et al., 2017) propose a deep recurrent generative decoder model which combines the decoder with a variational autoencoder.
CGU: (Lin et al., 2018) proposed a global encoding framework to control the information flow from the encoder to the decoder.
superAE: (Ma et al., 2018) took a summary autoencoder to supervise the learning of the Seq2Seq model.
s2s+att: We also implement a RNN-based Seq2Seq attentional model as our baseline and denote it as "s2s+att".

6.4. Results

We refer to our proposed Joint Selective Mechanism model as JSM. Table 1 shows the evaluation results on the English Gigaword test set used by (Rush et al., 2015). From Table 1 we can learn that our JSM model outperforms all the baseline models in ROUGE-1 and ROUGE-L and is comparable to the best baseline in terms of ROUGE-2. As for the SEASS baseline model, we both use a selective mechanism, but our JSM model outperforms it in all evaluation metrics. Table 2 demonstrates the evaluation results on the internal English Gigaword test set sampled by (Zhou et al., 2017). Our JSM model achieves 47.32 ROUGE-1, 24.96 ROUGE-2 and 44.12 ROUGE-L, which owns an advantage of 0.46 ROUGE-1, 0.38 ROUGE-2 and 0.59 ROUGE-L compared with the SEASS model.

We report the ROUGE recall score for the DUC 2004 test set following (Zhou et al., 2017). As shown in Table 3, our JSM model outperforms the SEASS model with 0.55 ROUGE-1, 0.34 ROUGE-2 and 0.56 ROUGE-L improvements. However, the DRGD model achieves the highest evaluation results with 31.79 ROUGE-1, 10.75 ROUGE-2 and 27.48 ROUGE-L, and we think the fact that the reference summaries of the DUC 2004 test set

Table 1: ROUGE F1 scores on the English Gigaword test set.

Model	**RG-1**	**RG-2**	**RG-L**
ABS(beam)	29.55	11.32	26.42
ABS+(beam)	29.76	11.88	26.96
Luong-NMT(beam)	33.10	14.45	30.71
SEASS(greedy)	35.48	16.50	32.93
SEASS(beam)	36.15	17.54	33.63
DRGD(beam)	36.27	17.57	33.62
CGU(beam)	36.30	**18.00**	33.80
s2s+att(greedy)	33.53	14.94	30.86
s2s+att(beam)	34.62	15.93	31.86
JSM(greedy)	35.66	16.85	33.35
JSM(beam)	**36.40**	17.85	**34.00**

Table 2: ROUGE F1 scores on the internal English Gigaword test set.

Model	**RG-1**	**RG-2**	**RG-L**
ABS(beam)	37.41	15.87	34.70
SEASS(greedy)	45.27	22.88	42.20
SEASS(beam)	46.86	24.58	43.53
s2s+att(greedy)	42.61	20.12	39.65
s2s+att(beam)	44.60	22.23	41.54
JSM(greedy)	46.03	23.63	43.07
JSM(beam)	**47.32**	**24.96**	**44.12**

are capped at 75 bytes has an impact on the evaluation of our model. The results on the MSR-ATC test set is demonstrated in Table 4, and our JSM model performs significantly better in terms of ROUGE-1, ROUGE-2 and ROUGE-L scores compared to the SEASS model.

Table 5 presents the evaluation results on the LCSTS test set. Our JSM model achieves the score of 37.90 ROUGE-1, 24.68 ROUGE-2 and 35.06 ROUGE-L while the highest CGU baseline model outperforms our JSM model a lot. We can see that our JSM model performs worse on the Chinese dataset compared to the English dataset. One possible reason is that our JSM model does not have a good adaptability on the Chinese dataset when trained in character-level.

Table 3: ROUGE recall scores on the DUC 2004 test set.

Model	**RG-1**	**RG-2**	**RG-L**
ABS(beam)	26.55	7.06	22.05
ABS+(beam)	28.18	8.49	23.81
Luong-NMT(beam)	28.55	8.79	24.43
SEASS(greedy)	28.68	8.55	25.04
SEASS(beam)	29.21	9.56	25.51
DRGD(beam)	**31.79**	**10.75**	**27.48**
s2s+att(greedy)	27.69	8.08	24.00
s2s+att(beam)	28.46	9.07	24.61
JSM(greedy)	28.79	8.96	25.22
JSM(beam)	29.76	9.90	26.07

Table 4: ROUGE F1 scores on the MSR-ATC test set.

Model	**RG-1**	**RG-2**	**RG-L**
ABS(beam)	20.27	5.26	17.10
SEASS(greedy)	19.77	6.44	17.36
SEASS(beam)	25.75	10.63	22.90
s2s+att(greedy)	19.36	5.63	17.03
s2s+att(beam)	24.34	9.16	21.62
JSM(greedy)	21.43	7.62	19.28
JSM(beam)	**27.98**	**12.90**	**25.36**

7. Discussion

In this section, we first discuss the effectiveness of the modified selective gate and the selection loss and then we compare the difference between our model and (Zhou et al., 2017)'s model through case analysis.

7.1. Effectiveness of the Modified Selective Gate and the Selection Loss

We train our model with different values of the hyperparameter λ, including 0, 0.01, 0.1 and 1. We also incorporate the selection loss with (Zhou et al., 2017)'s selective mechanism to make a comprehensive comparison, and the λ is set to 1. Table 6 shows the results on the English Gigaword test set with different values of the λ. All of the results are produced using beam search decoding. For the SEASS model, we find the model with the selection loss performs worse than the original model, and one possible explanation is that the guidance of the selection loss conflicts with the guidance of the source sentence representation. Without the selection loss, the modified selective gate is inferior to the selective gate of the SEASS model, but with the hyperparameter λ increasing within certain range, the performance of

Table 5: ROUGE F1 scores on the LCSTS test set.

Model	**RG-1**	**RG-2**	**RG-L**
RNN(W)	17.70	8.50	15.80
RNN(C)	21.50	8.90	18.60
RNN context(W)	26.80	16.10	24.10
RNN context(C)	29.90	17.40	27.20
SRB	33.30	20.00	30.10
CopyNet(W)	35.00	22.30	32.00
CopyNet(C)	34.40	21.60	31.30
DRGD	36.99	24.15	34.21
CGU	**39.40**	**26.90**	**36.50**
superAE	39.20	26.00	36.20
s2s+att	34.90	21.28	32.02
JSM	37.90	24.68	35.06

Table 6: ROUGE F1 scores on the English Gigaword test set with different values of the λ.

Model	**RG-1**	**RG-2**	**RG-L**
SEASS(λ=0)	36.15	17.54	33.63
SEASS(λ=1)	35.58	17.14	32.98
JSM(λ=0)	35.37	17.36	32.81
JSM(λ=0.01)	35.63	17.22	33.18
JSM(λ=0.1)	36.06	17.35	33.41
JSM(λ=1)	36.40	17.85	34.00

the modified selective gate improves. In fact We also train our model with the λ setting to 10, but the results is so bad (even worse than the baseline s2s+att model) that we do not list them in Table 6.

7.2. Case Study

As shown in Table 7, two examples are chosen to compare the summaries generated by our JSM model and the SEASS model. From the cases we can see that both models can figure out the general idea of the source texts, but our model can capture the key points more accurate benefitting from a joint selective learning between the input and the output during training. For example, the golden summary for S(1) is "*un senior official says peace process in middle east hopeful with immense challenges.*", and it points out two key points "*hopeful*" and "*challenges*". From the source text, we can observe that "*challenges*" is more important than "*hopeful*" because of the word "*although*". Our result points out "*challenges*" while the result of the SEASS model pays more attention to "*hopeful*".

Table 7: Examples of the generated summaries.

S(1): un under-secretary-general for political affairs ibrahim gambari said on wednesday that although the future of the peace process in the middle east is hopeful, it still faces immense challenges.
Golden: un senior official says peace process in middle east hopeful with immense challenges.
SEASS: un official optimistic about future of mideast peace process.
JSM: **un official says future of mideast peace process still faces challenges.**

S(2): a russian plane filled with coal miners slammed into a mountain on a norwegian island near the arctic circle on thursday, apparently killing all ### people aboard.
Golden: UNK russian plane crashes in arctic rescue team finds no sign of.
SEASS: ### killed in russian plane crash.
JSM: **russian plane crashes into mountain near arctic circle.**

Following (Li et al., 2017), we visualize our selective gate to demonstrate how it works. Fig. 2 shows the first derivative heat map of the selective gate of S(1). Although the words "*hopeful*" and "*challenges*" are both lit up, the word "*still*" makes our model select the relevant information of "*challenges*" finally.

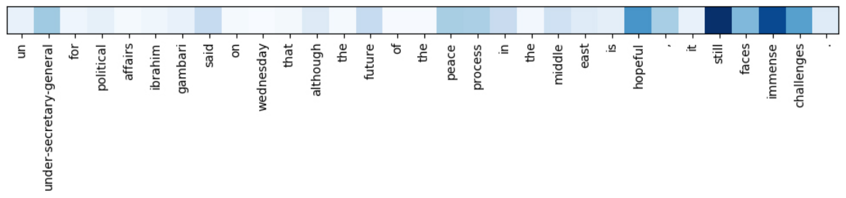

Figure 2: First derivative heat map of the output with respect to the selective gate. In this input sentence, the words "*hopeful*", "*still*", and "*challenges*" are highlighted in the heat map. The ground truth summary is "*un senior official says peace process in middle east hopeful with immense challenges*" and the output summary of our model is "*un official says middle east peace process still faces challenges*".

8. Conclusion

In this paper, we propose a joint selective mechanism for abstractive sentence summarization which selects the important information by reading the input and the output jointly. The selective gate is trained according to a selection loss function which is calculated from the Euclidean distance between the selective gate and the mean of the review gate at each time step. Experimental results show our proposed model outperforms most of the baseline models and is comparable to the state-of-the-art model benefitting from the joint selective mechanism.

Acknowledgments

This research work has been funded by the National Natural Science Foundation of China (Grant No.61772337, U1736207 and 61472248), the SJTU Shanghai Songheng Content Analysis Joint Lab, and program of Shanghai Technology Research Leader (Grant No.16XD1424400).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming Zhou. Ranking with recursive neural networks and its application to multi-document summarization. In *AAAI*, pages 2153–2159, 2015.

Jianpeng Cheng and Mirella Lapata. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 484–494. Association for Computational Linguistics, 2016.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Sumit Chopra, Michael Auli, and Alexander M Rush. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, 2016.

Ian Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1319–1327, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL http://proceedings.mlr.press/v28/goodfellow13.html.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 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)*, pages 1631–1640, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P16-1154.

Baotian Hu, Qingcai Chen, and Fangze Zhu. LCSTS: A large scale chinese short text summarization dataset. *CoRR*, abs/1506.05865, 2015. URL http://arxiv.org/abs/1506.05865.

Philipp Koehn. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *Machine Translation: From Real Users To Research, Conference of the Association for Machine Translation in the Americas, Amta 2004, Washington, Dc, Usa, September 28-October 2, 2004, Proceedings*, pages 115–124, 2004.

Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. Deep recurrent generative decoder for abstractive text summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2091–2100, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D17-1222.

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

Junyang Lin, Xu SUN, Shuming Ma, and Qi Su. Global encoding for abstractive summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 163–169. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/P18-2027.

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL https://aclweb.org/anthology/D/D15/D15-1166.

Shuming Ma, Xu Sun, Jingjing Xu, Houfeng Wang, Wenjie Li, and Qi Su. Improving semantic relevance for sequence-to-sequence learning of chinese social media text summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 635–640. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-2100. URL http://www.aclweb.org/anthology/P17-2100.

Shuming Ma, Xu SUN, Junyang Lin, and Houfeng WANG. Autoencoder as assistant supervisor: Improving text representation for chinese social media text summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 725–731. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/P18-2115.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*, 2016.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. Annotated gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 95–100. Association for Computational Linguistics, 2012.

Paul Over, Hoa Dang, and Donna Harman. Duc in context. *Information Processing & Management*, 43(6):1506–1520, 2007.

Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, 2015.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Kristina Toutanova, Chris Brockett, Ke M Tran, and Saleema Amershi. A dataset and evaluation metrics for abstractive compression of sentences and short paragraphs. 2016.

Kristian Woodsend and Mirella Lapata. Automatic generation of story highlights. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 565–574. Association for Computational Linguistics, 2010.

Jingjing Xu and Xu Sun. Dependency-based gated recursive neural network for chinese word segmentation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–572. Association for Computational Linguistics, 2016. doi: 10.18653/v1/P16-2092. URL http://www.aclweb.org/anthology/P16-2092.

Wenyuan Zeng, Wenjie Luo, Sanja Fidler, and Raquel Urtasun. Efficient summarization with read-again and copy mechanism. *arXiv preprint arXiv:1611.03382*, 2016.

Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. Selective encoding for abstractive sentence summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1095–1104, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL http://aclweb.org/anthology/P17-1101.