Text Summarization as the Potential Technology for Intelligent Internet of Things

Lijun Wei[1,2](✉), Yun Liu[1,2], and Jian Li[1,2]

[1] School of Electronic and Information Engineering,
Beijing Jiao Tong University, Beijing, China
{18120145,liuyun,lijian}@bjtu.edu.cn

[2] Key Laboratory of Communication and Information Systems, Beijing Municipal
Commission of Education, Beijing, China

Abstract. Applying automatic text summarization technology to Internet of Things can save network cost and improve computing speed. However, the models of generated text summarization are always using sequence-to-sequence model with attention mechanism. Unluckily, this method for abstractive summarization has two main shortcomings: first, they fail to address unknown words problems, second, their generated summaries are not very readable because of repetition. In our work, our goal is to enhance the semantic coherence of summaries for original texts. In order to this end, we propose a new model that augments the traditional model in two ways. First, we apply semantic relevance to pointer-generator network to get high similarity between source texts and our summaries. Second, we change the mechanism of coverage and use it to pointer-generator network to discourage repetition. Following other works, we apply our new model to Chinese social media dataset LCSTS. Our experiments suggested that our new model outperforms current abstractive baseline systems on the dataset.

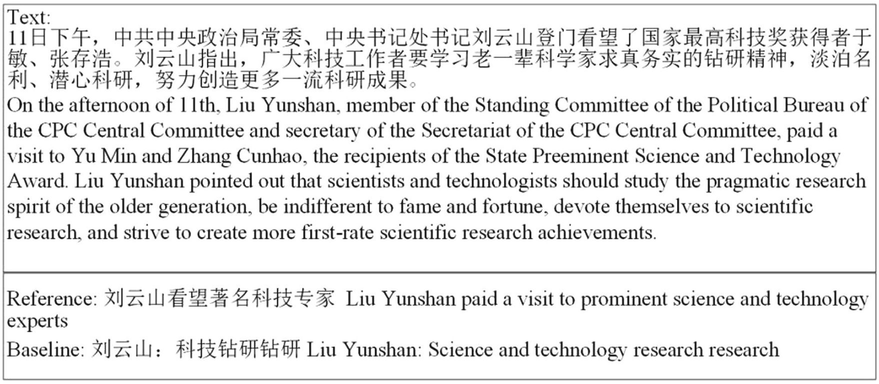
Keywords: Intelligent Internet of Things · Text summarization · Attention mechanism

1 Introduction

We use text summarization in order to get a short representation of an input text which captures the core meaning of the original. In some Internet of Things scenarios, such as smart home and intelligent robot, text summarization can compress the obtained text information to save network cost and improve computing speed. Different from the extractive text summarization [1–3], which selects elements from original text to form summaries, the aim of abstractive text summarization is to produce summaries in a generated way. Extractive text summarization performs well when the source text is long, however, it doesn't apply to short text. Recently, most abstractive text summarization is depended on seq2seq models which have attention mechanism [4–6], and this way is superior to the traditional statistical methods.

Y.-B. Lin and D.-J. Deng (Eds.): SGIoT 2020, LNICST 354, pp. 169–177, 2021.
https://doi.org/10.1007/978-3-030-69514-9_14

Unfortunately, it is shown that there are prominent shortcomings in conventional attention mechanism. Lin pointed out that the attention based seq2seq abstract text summarization model has the problems of duplication and semantic independence, which leads to poor readability and can't tell source text's core point [6]. For example, in the summary produced by traditional model in Fig. 1, because attention mechanisms take note of words with high attention scores as usual, "钻研" is still behind "钻研".



Text:
11日下午，中共中央政治局常委、中央书记处书记刘云山登门看望了国家最高科技奖获得者于敏、张存浩。刘云山指出，广大科技工作者要学习老一辈科学家求真务实的钻研精神，淡泊名利、潜心科研，努力创造更多一流科研成果。
On the afternoon of 11th, Liu Yunshan, member of the Standing Committee of the Political Bureau of the CPC Central Committee and secretary of the Secretariat of the CPC Central Committee, paid a visit to Yu Min and Zhang Cunhao, the recipients of the State Preeminent Science and Technology Award. Liu Yunshan pointed out that scientists and technologists should study the pragmatic research spirit of the older generation, be indifferent to fame and fortune, devote themselves to scientific research, and strive to create more first-rate scientific research achievements.

Reference: 刘云山看望著名科技专家 Liu Yunshan paid a visit to prominent science and technology experts

Baseline: 刘云山：科技钻研钻研 Liu Yunshan: Science and technology research research

Fig. 1. A simple case got from the conventional attention-based seq2seq model. As we can see, the summary generated by the baseline contains repetition.

See and Paulus use a model with pointer-generator that uses interpolation of generation and replication probabilities to generate summaries [7, 8]. Moreover, the interpolation is controlled by a mixture coefficient that is predicted by the model. Theoretical analysis shows that the pointer-generator mechanism enables the model to get summary in a comprehensive method, which integrates the advantages of extraction summary and generative summary, in addition, it can solve the out-of-vocabulary (OOV) problem. However, in practice, compared to source text, the summaries we got have low semantic relevance.

We design a pointer-generator model with semantic relevance in order to solve the above problems, the main idea of our model is to get high relevance between generated summaries and original text. A semantic similarity evaluation factor is used in our proposed model, which can measure the correlation between original text and the generated summary. By maximize the score of similarity, our model can get high coherence between source articles and summaries during training stage. Finally, in order to reduce and avoid repetition problem, attention mechanism is introduced into our model. It can be shown that compared to current abstractive baseline systems, our model can generate better summaries which have high score.

2 Proposed Model

We describe (1) the baseline seq2seq model with attention mechanism, (2) our pointer-generator model with semantic relevance, (3) our coverage mechanism that can be added to both of the first two models in this section.

2.1 Attention-Based Seq2seq Model

The baseline model is a seq2seq model which is attention-based, and it is described in Fig. 2. Similar to that of Hu [9], our model consists of three parts, the first part is encoder, the second part is decoder and the last part is attention mechanism. The encoder can help us to condense long source texts into continuous vector representation, and the decoder can help us to generate short summary. The encoder is a single-layer bidirectional GRU that can get the sequence vector $\{h_1, h_2, h_3..., h_N\}$ from source text x. On every time step, the decoder part has a decoder state s_t, and the last character's word embedding is fed to it.

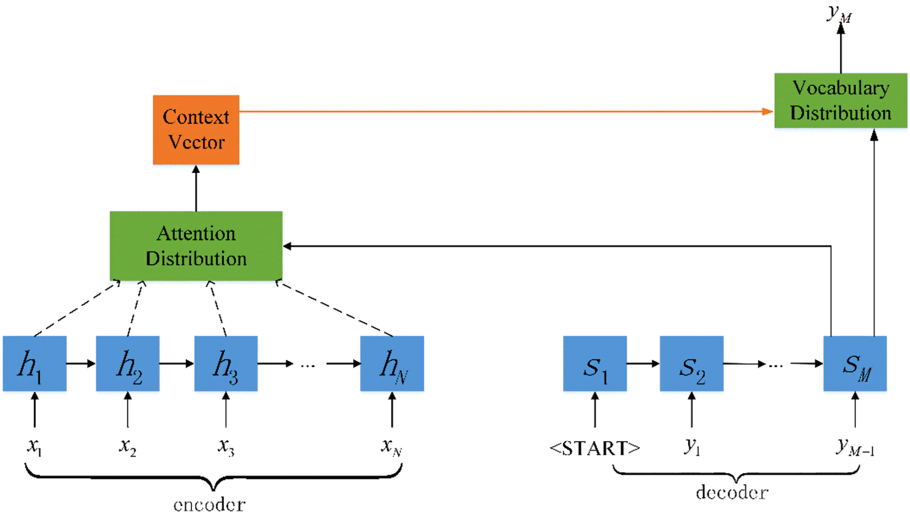


Fig. 2. Our attention-based seq2seq model. This model is made up of encoder (left), decoder (right) and attention mechanism.

Attention mechanism is used to inform our model to find the correct place which is used to get next word [4]. The context vector c_t is equal to the weighted sum of the hidden states of the encoder

$$c_t = \sum_{i=1}^{N} \alpha_{ti} h_i \tag{1}$$

where h_i is the hidden state of the i th input x, α_{ti} is the probability of x_i on t step:

$$\alpha_{ti} = \frac{e^{g(s_t,h_i)}}{\sum_{j=1}^{N} e^{g(s_t,h_i)}} \tag{2}$$

The correlation score between the decoder hidden state s_t and that of encoder is $g(s_t, h_i)$. The context vector c_t and the decoder state s_t are linked, and it is the input of the two linear layers, then we can get the vocabulary distribution P_{vocab}, which is a probability distribution to predict words w:

$$P(w) = P_{vocab}(w) \tag{3}$$

During training stage, the loss for the whole input sequence is:

$$\text{loss} = \frac{1}{T}\sum\nolimits_{t=0}^{T} \text{loss}_t \tag{4}$$

In which loss_t is the loss for time step t:

$$\text{loss}_t = -\log P\left(w_t^*\right) \tag{5}$$

2.2 Attention-Based Seq2seq Model

Our semantic relevance-based pointer-generator network (depicted in Fig. 3) contains five components: encoder, decoder, attention mechanism, generation probability calculation and a similarity function. Our model is a hybrid among basic attention-based seq2seq model [10], a pointer network [11], and a semantic relevance [12].

Our model can get summary in a generated way or an abstracted way. The generated way can get novel words that are in the vocabulary. The abstracted way can get important sentences that are in the source text. We calculated the model's context vector c_t and attention distribution α_t in Sect. 2.1. In addition, the generation probability p_{gen} is applied to represents whether getting the summary in a generated way or getting it in an abstract way. When this scalar is bigger than 0.5, we get more information in a generated way. It is calculated through a linear cell from three inputs, the first is context vector c_t, the second is decoder state s_t and the third is decoder input y_{t-1}:

$$p_{gen} = \sigma\left(w_{c*}^T c_t^* + w_s^T s_t + w_y^T y_{t-1} + b_{ptr}\right) \tag{6}$$

where vectors w_{c*}, w_s, w_y and b_{ptr} are parameters which can be learned, the function σ is a sigmoid function. Then we use p_{gen} to get the final words distribution P_{final}:

$$P_{final}(w) = p_{gen}P_{vocab}(w) + \left(1 - p_{gen}\right)\sum_{i:w_i=w} \alpha_i \tag{7}$$

The pointer-generator network can solve OOV problem easily. Suppose that w is a word that does not appear in the vocabulary, then $P_{vocab}(w)$ is zero, we can get the word through pointing.

We adapt similarity function to our model and we can receive a high semantic coherence between long original texts and short summaries. In order to get high semantic relevance, we select maximize the score computed by similarity function as our training object. Original text's semantic vector V_t is equal to the last output h_N. Previous work has proved that we can get the summary's semantic vector by simply subtracting h_N from s_M:

$$V_s = s_M - h_N \tag{8}$$

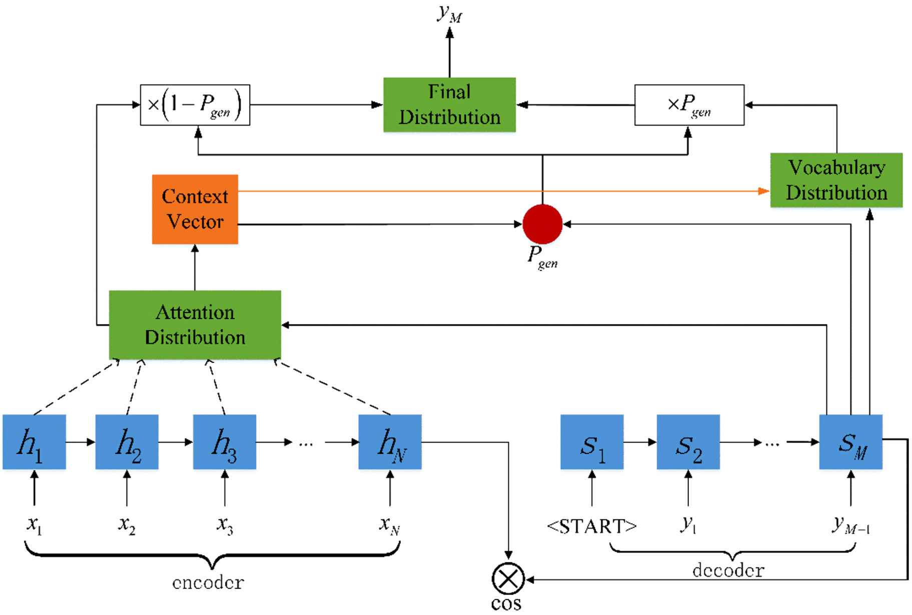


Fig. 3. Pointer-generator model with semantic relevance computing unit. It is made up of encoder, decoder, attention mechanism, generation probability calculation and cosine similarity function.

Cosine similarity is a measure of similarity between two non-zero vectors in the same space, and we use it to measure the similarity relevance:

$$\cos(V_s, V_t) = \frac{V_s \cdot V_t}{\|V_s\|\|V_t\|} \tag{9}$$

During training stage, we need to add similarity score to our loss function:

$$\text{loss}_t = -\log P_{final}\left(w_t^*\right) - \lambda \cos(V_s, V_t) \tag{10}$$

where λ is a hyperparameter [13].

2.3 Coverage Mechanism

Repetition is one of the big problems in abstractive text summarization [7], moreover, it is a universal question for sequence-to-sequence models [14, 15]. We use the coverage model [7] to solve the repetition problem. The idea is that we use the attention distribution to track what has been covered so far, and penalize the networks that participate in the same part again. On each time step t of the decoder, the coverage vector co_t is the mean of the sum of all the attention distributions α_j to data:

$$co_t = \sum_{j=0}^{t-1} \alpha_j \Big/ t - 1 \tag{11}$$

Then, we introduce an additional loss term to penalize any overlap between the new attention distribution α^t and the coverage vector co_t:

$$\text{covloss}_t = \sum_i \min\left(\alpha_i^t, co_i^t\right) \tag{12}$$

Finally, we add the coverage loss to the loss function (10):

$$\text{loss}_t = -\log P_{final}\left(w_t^*\right) - \lambda_1 \cos(V_s, V_t) + \lambda_2 \sum_i \min\left(\alpha_i^t, co_i^t\right) \tag{13}$$

where λ_1 and λ_2 are hyper-parameters.

3 Experiment

Following the previous work, we evaluate our model on the main social media datasets in China [9]. We first present the experimental datasets, evaluation indicators, and experimental details. Second, we compare our model with the baseline systems.

3.1 Dataset

LCSTS is a large-scale Chinese short text summarization dataset collected from Sina Weibo, a well-known Chinese social media website, consisting of over 2.4 million text-summary pairs [9]. The summaries are created manually, and the source texts are less than 140 Chinese characters. In addition, the dataset is divided into three parts, and all the text-summary pairs are manually scored, with associated scores ranging from 1 to 5. We just select pairs with scores more than 2, leaving 8,685 pairs in PART II and 725 pairs in PART III. In experiment, PART I is used for training, PART II for validation and PART III for testing.

3.2 Evaluation Metric

For evaluation metrics, we adopt ROUGE scores [16], which is widely used for summarization evaluation [17–19]. By calculating the overlapping units, the ROUGE metrics compare generated summary with the reference summary. Following previous work, we use ROUGE-1 (overlap of unigram), ROUNGE-2 (overlap of bigrams) and ROUNGE-L (longest common subsequence) as our evaluation metrics.

3.3 Experimental Details

We implement our experiments in TensorFlow [20]. The vocabularies are extracted from the training sets, and the summaries and the source contents share the same vocabularies. We split the Chinese sentences into characters to mitigate the risk of word segmentation errors. In order to covering most of the common characters, we trim the vocabulary to 50,000.

For all experiments, our model has 256-dimentional hidden states and 128-dimentional word embeddings. Unlike [10], we do not use transfer learning and the pre-trained word embeddings. Instead, they are learned during training. We use Adagrad [21] for training with a learning rate of 0.15 and an initial accumulator value of 0.1. The batch size is 32, and we do not use dropout [22] on this dataset. Following the previous work, we implement the beam search with a beam size 4 and gradient pruning with a maximum gradient norm of 2. To obtain our final coverage model with semantic relevance, we set $\lambda_1 = 0.0001$ and $\lambda_2 = 1$ (as described in Eq. 13).

3.4 Results

We compare our proposed model with the following baseline models:

Simple Seq2seq is the basic sequence-to-sequence model for abstractive summarization. The encoder is a bidirectional GRU and the decoder is a unidirectional GRU [9].

Attention-based Seq2seq is a sequence-to-sequence model with attention mechanism. The main difference between Attention-based Seq2seq and Simple Seq2seq is that attention mechanism is added to the first one, so it can pay different attention to the source words on each time step [9].

SRB is an encoder-decoder model that takes semantic relevance into account. This model adds a similarity function to attention-based sequence-to-sequence model in order to make sure that there is high semantic relevance between source texts and generated summaries [6].

Table 1. Comparison with baseline models on LCSTS test set.

Model	ROUGE-1	ROUGE-2	ROUGE-L
Simple Seq2seq (W[1]) [9]	17.7	8.5	15.8
Simple Seq2seq (C[2]) [9]	21.5	8.9	18.6
Attention-based Seq2seq (W) [9]	26.8	16.1	24.1
Attention-based Seq2seq (C) [9]	29.9	17.4	27.2
SRB (C) [6]	33.3	20.0	30.1
PGC (C)	38.5	20.3	32.4
PGCS (C)	39.1	20.2	33.3

[1] Word level
[2] Character level.

We denote PGC as our proposed pointer-generator network with coverage mechanism, and PGCS as our pointer-generator network with semantic relevance and coverage mechanism. Table 1 shows the results of our experiments. We can see that the ROUGE-1, ROUGE-2, and ROUGE-L scores keep rising among the models. Compared with SRB, the ROUGE-1, ROUGE-2, and ROUGE-L scores of PGCS improves 17.42%, 1% and

10.63% respectively, which shows that pointer-generator network and coverage mechanism play an important role in getting more coherent summaries. In addition, from the comparison of PGC and PGCS ROUGE scores, we can notice that the semantic relevance unit in the model improves the quality of generated summaries.

4 Related Work

The encoder-decoder architecture is the basic framework of our proposed model. Sutskever first proposed sequence-to-sequence model and used it for neural machine translation [23]. Bahdanau proposed attention mechanism, which allows the model to automatically select a part from the primitive text [4]. Rush first applied attention mechanism to text summarization task and the model performs better than the state-of-the-art sequence-to-sequence models [5]. Vinyals described pointer network than can learn the conditional probability of an output sentence in a new way [11]. Coverage mechanism is first applied to neural machine translation, then See used it for text summarization and it solves the problem of repetition at some level. Weber modified the pointer-generator network and the new model can control the amount of copying [24]. Ma pro-posed a neural model based on semantic relevance that can improve the semantic relevance between the source text and the generated summaries [12]. Ma used the semantic representation of standard summary to supervise the learning of that of source text [6].

5 Conclusion

In this paper, we propose an architecture that can get the summary in an automatically copied or generated way. The similarity computing unit can improve the semantic relevance between source text and generated summaries. Our coverage mechanism solves the problem of repetition to some extent. In addition, experimental results show that our PGCS (pointer-generator network with coverage and semantic relevance mechanism) outperforms the baseline models. As a result, when our PGCS is applied to many intelligent scenarios of Internet of Things, we can get more benefits about computing and network transmission.

Acknowledgments. This work is supported by the National Natural Science Foundation of China under Grant 61701019.

References

1. Dragomir, R., Timothy, A., Sasha, B.: MEAD - a platform for multidocument multilingual text summarization. In: Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC (2004)
2. Kristian, W., Mirella, L.: Automatic generation of story highlights. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL, pp. 565–574 (2010)

3. Cheng, J., Lapata, M.: Neural summarization by extracting sentences and words. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL, vol. 1: Long Papers (2016)
4. Dzmitry, B., Kyunghyun, C., Yoshua, B.: Neural machine translation by jointly learning to align and translate. CoRR, abs/1409.0473 (2014)
5. Alexander, M., Sumit, C., Jason, W.: A neural attention model for abstractive sentence summarization. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP, pp. 379–389 (2015)
6. Ma, S., Sun, X., Lin, J., Wang, H.: 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, vol. 2: Short Papers (2018)
7. Abigail, S., Peter, J., Christopher, D.: Get to the point: Summarization with pointer-generator networks. arXiv: 1704.04368 (2017)
8. Romain, P., Caiming, X., Richard, S.: A deep reinforced model for abstractive summarization. arXiv:1705.04304 (2017)
9. Hu, B., Chen, Q., Zhu, F.: LCSTS: a large scale Chinese short text summarization dataset. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP, pp. 1967–1972 (2015)
10. Ramesh, N., Bowen, Z., Cicero, D.: Abstractive text summarization using sequence-to-sequence RNNs and beyond. In: Computational Natural Language Learning (2016)
11. Oriol, V., Meire, F., Navdeep, J.: Pointer networks. In: Neural Information Processing Systems (2015)
12. Ma, S., Sun, X., Xu, J.: 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, ACL, pp. 635–640 (2017)
13. Wang, W., Chang, B.: Graph-based dependency parsing with bidirectional LSTM. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL, vol. 1: Long Papers (2016)
14. Tu, Z., Lu, Z., Liu, Y.: Modeling coverage for neural machine translation. In: Association for Computational Linguistics, ACL (2016).
15. Mi, H., Sankarab, B., Wang, Z.: Coverage embedding models for neural machine translation. In: Empirical Methods in Natural Language Processing (2016)
16. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Text Summarization Branches Out (2014)
17. Chen, X., Chan, Z., Gao, S.: Learning towards abstractive timeline summarization. In: Proceedings of the Twenty-Eighth International Joint Conference on Artifificial Intelligence, IJCAI-19, pp. 4939–4945 (2019)
18. Gao, S.,Chen, X., Li, P.: How to write summaries with patterns? learning towards abstractive summarization through prototype editing (2019)
19. Lin, J., Sun, X., Ma, S.: Global Encoding for Abstractive Summarization (2018)
20. Abadi, M., Barham, P., Jianmin, C.: Tensorflflow: a system for large-scale machine learning. OSDI **16**, 265–283 (2016)
21. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. **12**, 2121–2159 (2011)
22. Srivastava, N., Hinton, G., Krizhevsky, A.: Dropout: a simple way to prevent neural networks from overfifitting. J. Mach. Learn. Res. **15**, 1929–1958 (2014)
23. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Annual Conference on Neural Information Processing Systems, pp. 3104–3112 (2014)
24. Weber, N., Shekhar, L., Balasubramanian, N.: Controlling Decoding for More Abstractive Summaries with Copy-Based Networks (2018)