Neural Diverse Abstractive Sentence Compression Generation

Mir Tafseer Nayeem(✉), Tanvir Ahmed Fuad, and Yllias Chali

University of Lethbridge, Lethbridge, AB, Canada
mir.nayeem@alumni.uleth.ca, t.fuad@uleth.ca, chali@cs.uleth.ca

Abstract. In this work, we have contributed a novel abstractive sentence compression model which generates diverse compressed sentence with paraphrase using a neural **seq2seq** encoder decoder model. We impose several operations in order to generate diverse abstractive compressions at the sentence level which was not addressed in the past research works. Our model jointly improves the information coverage and abstractiveness of the generated sentences. We conduct our experiments on the human-generated abstractive sentence compression datasets and evaluate our system on several newly proposed Machine Translation (**MT**) evaluation metrics. Our experiments demonstrate that the methods bring significant improvements over the state-of-the-art methods across different metrics.

Keywords: Abstractive summarization · Diverse sentence compression

1 Introduction

The task of automatic text summarization aims at finding the most relevant information in a text and presenting them in a condensed form. A good summary should retain the most important contents of the original text, while being non-redundant and grammatically readable [5,15]. Summarization on the sentence level is called sentence compression. Sentence compression approaches can be classified into two categories: extractive and abstractive sentence compression. Most sentence compression models follow extractive approaches that select the most relevant information from the source sentence and generate a shorter representation of the sentence by deleting unimportant fragments which is still grammatical. On the other hand, abstractive methods, which are still a growing field, are highly complex as they need extensive natural language generation to rewrite the sentences from scratch based on the understanding of the sentences [17]. The abstractive techniques which we traditionally use are sentence compression, fusion and lexical paraphrasing [16].

2 Related Works

Recent success of neural sequence-to-sequence (**seq2seq**) models provide an effective way for text generation which achieved huge success in the case of

L. Azzopardi et al. (Eds.): ECIR 2019, LNCS 11438, pp. 109–116, 2019.
https://doi.org/10.1007/978-3-030-15719-7_14

abstractive sentence summarization. These systems have adopted techniques such as encoder-decoder with attention [2,12] models from the field of machine translation to model the sentence summarization task [8,13,19,23]. The deep neural network architectures are completely data driven hence more training data will produce good quality output sequences. Therefore, almost all the past works on sentence summarization using neural networks [4,8,19,22,26] made use of the English Gigaword dataset [14].

Unfortunately, this line of research under the term sentence compression, which can generate deletion based compressive sentences, somewhat misleadingly called abstractive summarization in some follow-up research works [13,23,26]. Our experimental results clearly demonstrate the fact that they are producing compressions by copying the source sentence words with morphological variations, no paraphrasing is involved in the process.

3 Diverse Abstractive Sentence Compression Model

Our neural **D**iverse **P**araphrastic **C**ompression model is based on Neural Machine Translation (NMT). **DPC** uses **NMT** to translate from a source sentence to an abstractive compression. Given a source sentence $\mathbf{X} = (x_1, x_2, \ldots, x_N)$, our model learns to predict its abstractive compression target $\mathbf{Y} = (y_1, y_2, \ldots, y_M)$ with diversity, where $M < N$. Inferring the target Y given the source X is a typical sequence to sequence learning problem, which can be modeled with attention-based encoder-decoder models [2,12]. As the name suggests, the basic form of an encoder-decoder model consists of two components.

Encoder. The encoder in our case is a bi-directional GRU (Bi-GRU), unlike [12] which uses uni-directional LSTM [11]. Another important modification we can do to the Bi-GRUs following [12] is stacking multiple layers on top of each other. They can extract more abstract features of the current words or sentences. However, stacking RNNs suffer from the vanishing gradient problem in the vertical direction from the output layer ($\mathbf{GRU_3}$) to the layer close to the input ($\mathbf{GRU_1}$), just as the standard RNN suffers in the horizontal direction. This causes the earlier layers of the network to be under-trained. A simple solution to this problem is to add residual connections, which has been shown to be extremely useful for the image recognition task [10]. The idea behind these networks is simply to add the output of the previous layer directly to the result of the next layer. For example, in a 3-layer stacked GRU with residual connections, the calculation at time step t would look as follows,

$$h_{1,t} = \mathbf{BiGRU_1}(e(x_t), h_{1,t-1}) + e(x_t)$$

$$h_{2,t} = \mathbf{BiGRU_2}(h_{1,t}, h_{2,t-1}) + h_{1,t}$$

$$h_{3,t} = \mathbf{BiGRU_3}(h_{2,t}, h_{3,t-1}) + h_{2,t}$$

where, the $h_{1,t} \in \mathbb{R}^n$ encodes all content seen so far at time t from layer 1, which is computed from h_{t-1} and $e(x_t)$, where $e(x_t) \in \mathbb{R}^m$ is the m-dimensional

embedding of the current word x_t. Therefore, we use the idea of residual networks for building our encoder decoder model **DPC** to perform the abstractive compression generation task which is illustrated in Fig. 1. The initial hidden states of the encoder are set to zero vectors, i.e., $\overrightarrow{h^S_{1,1}} = 0$, $\overleftarrow{h^S_{1,N}} = 0$. In our **DPC** model, the encoder transforms the source sentence $\mathbf{X}$ into a sequence of hidden states $(\mathbf{h}^S_{3,1}, \mathbf{h}^S_{3,2}, \ldots, \mathbf{h}^S_{3,N})$ with a stacked residual network.

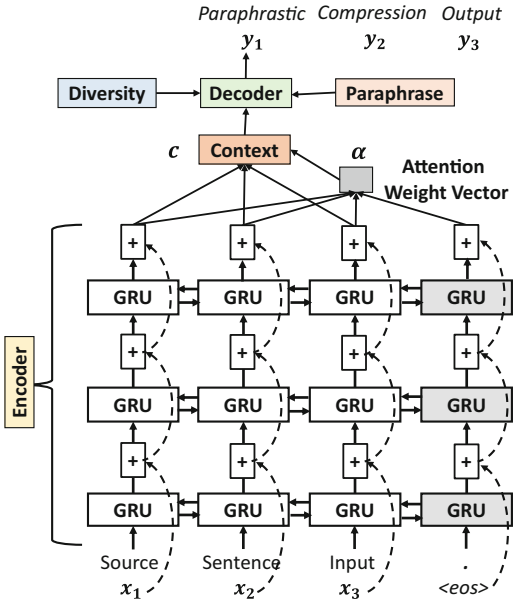

Fig. 1. Neural **D**iverse **P**araphrastic **C**ompression Generation Model

Decoder and Attender. The decoder uses a simple GRU with attention to generate one word y_{t+1} at a time in the target sentence $\mathbf{Y}$.

$$P(\mathbf{Y}|\mathbf{X}) = \prod_{t=1}^{M} P(y_t|y_{1:t-1}, \mathbf{X}) \tag{1}$$

We use the $(\cdot)$ *dot* attention mechanism [12] due to its efficiency and which is simple to implement. The dot attention mechanism is actually the dot product between two hidden vectors.

Decoder with Copying Mechanism. At each generation step of the decoder, the output word is selected according to the probability distribution over the whole target vocabulary in the softmax layer, which is the most time and capacity-consuming part of the system. Therefore, we limit our vocabularies to

be the top 60K most frequent words. The infrequent words were removed from the vocabulary and were replaced with the symbol UNK, meaning unknown word. However, it has been observed that the infrequent words are usually proper nouns or named-entities that have an impact on the meaning of the sentence. Therefore, we used the **COPYNET** model proposed by [9] which can integrate the regular way of word generation in the decoder with the new copying mechanism which can choose words or subsequences in the input sequence and put them at appropriate places in the output sequence. Please refer to the original paper by [9] for more detail.

Paraphrasing in Context. Our model implicitly learned how to paraphrase and can eventually generate paraphrases from the data itself. Moreover, to ensure complete paraphrasing we also impose an explicit edit operation. The pre-edit paraphrasing operation is applied to the source sentence before giving it to the model. We use the 60K most frequent words as our model vocabulary, out of almost 300K unique words from the whole training set. We create an alignment table for the words outside the vocabulary to the words inside the vocabulary using **fastText** [3] embedding. The word-to-word alignment has been done by calculating the cosine distance between **fastText** average word vectors. We found an alignment table of almost 8K words outside of the vocabulary to words inside with $CosDistance \geq 0.7$ (e.g., pricey ⇒ expensive, detested ⇒ hated). Our model tries to replace the out of vocabulary words with the words inside using the alignment table before sequence generation.

Diversity-Promoting Beam Search. Most of the generated outputs using standard beam search are lexically similar and they are different at only some small parts of the text, such as punctuation and stopwords. A solution to this problem is to force the beam-search decoder to generate more diverse outputs. In this work, we use a technique proposed by [1] for enforcing the diversity between beams. This work has shown to be effective for creating diverse image captions, machine translation and visual question generation. The authors divide K beams into G groups and control the diversity between these groups while expanding each beam. They modify the log probability of each predicted word on group G_i (except the first group G_1) as follows:

$$score(w_t^i \mid \mathbf{X}, W_{t-1}^i) = logP(w_t^i \mid \mathbf{X}, W_{t-1}^i) + \lambda \cdot \Delta(P_{i-1}) \tag{2}$$

where, w_t^i is the candidate word of group G_i at time-step t, P_{i-1} is the list of last generated tokens from previous groups, λ is the diversity factor and Δ is the dissimilarity of current group with previous groups[1]. The beam size of the decoder was set to be 10. We present our N-best ($N = 5$) model generated output in Table 1.

[1] We use Hamming Diversity due to its simplicity and efficiency as *Delta* function.

Table 1. Our **DPC** model generated output (**CR** means Compression Ratio and highlighted words indicate paraphrasing in context).

Source sentence	It is the right message, sent while it is still early enough to do something constructive about the disappointing quality of the work so far
Reference (*Best*)	It is the right message to send to correct the disappointing quality of work so far. **(CR: 0.36)**
Output#1	***This*** message is the right message. **(CR: 0.76)**
Output#2	It is the right message, sent while it is still early enough to do something ***suitable***. **(CR: 0.44)**
Output#3	It is the right message, sent while it is still early enough to do something ***faster*** about the work. **(CR: 0.24)**
Output#4	***This*** message is the right message, sent while it is still early enough to do something ***useful*** about the work so far. **(CR: 0.12)**
Output#5	It is the right message, sent while it is still early enough to do something ***faster*** about the work so far. **(CR: 0.16)**

4 Experiments

4.1 Datasets

For training set, we use a parallel corpus which was constructed from the Annotated English Gigaword dataset [14]. We use the script released by [19] to generate **3.8M** sentence-summary pairs as training set. For validation and test set, we use MSR-ATC dataset [24]. We filtered out the compressions which involve multiple source sentences. The final validation and test set contains 271 and 459 pairs of single sentence abstractive compression with maximum of five human rewrite variations.

4.2 Evaluation Metric

We evaluate our system automatically using various automatic metrics such as **BLEU** [18], **SARI** [25] and **METEOR-E** [21]. **Compression Ratio (CR)** is a measure of how terse a compression. We define **Copy Rate** as how many tokens are copied to the abstract sentence from the source sentence without paraphrasing. Lower copy rate score means more paraphrasing is involved in the output abstract sentence. Copy rate of 100% means no paraphrasing.

4.3 Performance Comparison and Discussion

We compare our model with the systems which include both deletion-based and near abstractive models. **ILP**, an integer linear programing approach for

Table 2. Performance of different systems compare to our proposed model.

Model	Information coverage		Abstractiveness		
	BLEU	SARI	METEOR-E	CR	Copy rate
T3 [7]	11.1	25.7	0.22	**0.75**	90.6
ILP [6]	54.7	38.1	0.36	0.29	99.5
Seq2Seq [8]	53.8	35.5	0.34	0.39	99.7
NAMAS [19]	38.7	36.6	0.31	0.24	99.8
PG + C [20]	45.5	37.3	0.37	0.21	99.3
SEASS [26]	44.6	38.5	0.35	0.34	99.6
DPC (ours)	**54.9**	**39.3**	**0.41**	0.47	**84.5**

sentence compression which involves word deletion [6]; **T3**, a tree-to-tree transduction model for abstractive sentence compression [7]; **seq2seq**, a neural model for deletion-based compression [8]; and **NAMAS**, a neural model for abstractive compression and sentence summarization [19]. The output generated by the above mentioned systems were collected from [24]. Moreover, we also compare our system with [20] which uses Pointer Generator Networks and Coverage Mechanism and with [26] which uses a selective gate network and an attention equipped decoder to tackle sentence summarization task.

We take the identical test set of [24] for comparison. We use the generated output directly from the baseline models using their settings to compare with our model across the metrics discussed earlier. For fair comparison, we add all the top ($N = 5$) candidates in the evaluation process. The results of different baseline systems across different evaluation metrics are presented in Table 2. Our model balances the information coverage (BLUE, SARI) and complete abstractiveness (METEOR-E, Copy Rate), instead of over compressing the generated sentences (Compression Ratio (**CR**)). As our model is generating diverse paraphrastic compression, we obtain a higher BLEU score compare to all the models presented in Table 2. We get a slightly higher score in terms of SARI because of the multiple human references. The Copy Rate scores of the baseline systems other than T3 clearly indicates that they are doing completely compression, no paraphrasing is involved. Lower copy rate means more new words were generated in the output sentences. We also get a higher score in METEOR-E metric because of the lexical substitution operations.

5 Conclusion and Future Work

In this paper, we have designed a new abstractive compression generation model at the sentence level which jointly performs diverse sentence compression and paraphrasing. We have imposed several operations to this architecture to reduce the extractiveness of abstractive sentence level output summaries.

Acknowledgements. The research reported in this paper was supported by the Natural Sciences and Engineering Research Council (**NSERC**) of Canada - discovery grant and the University of Lethbridge.

References

1. Vijayakumar, A.K., et al.: Diverse beam search: decoding diverse solutions from neural sequence models. In: AAAI 2018, February 2018
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: ICLR 2015 (2015)
3. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Trans. Assoc. Comput. Linguist. **5**, 135–146 (2017)
4. Cao, Z., Li, W., Li, S., Wei, F.: Retrieve, rerank and rewrite: soft template based neural summarization. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 152–161. Association for Computational Linguistics (2018)
5. Chali, Y., Tanvee, M., Nayeem, M.T.: Towards abstractive multi-document summarization using submodular function-based framework, sentence compression and merging. In: Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, 27 November–1 December 2017, Volume 2: Short Papers, pp. 418–424 (2017)
6. Clarke, J., Lapata, M.: Global inference for sentence compression: an integer linear programming approach. JAIR **31**, 399–429 (2008)
7. Cohn, T., Lapata, M.: Sentence compression as tree transduction. JAIR **34**(1), 637–674 (2009)
8. Filippova, K., Alfonseca, E., Colmenares, C., Kaiser, L., Vinyals, O.: Sentence compression by deletion with LSTMs. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (2015)
9. Gu, J., Lu, Z., 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), Berlin, Germany, pp. 1631–1640, August 2016
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, pp. 770–778. IEEE Computer Society (2016)
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
12. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Empirical Methods in Natural Language Processing (EMNLP), Lisbon, Portugal, pp. 1412–1421, September 2015
13. Nallapati, R., Zhou, B., dos Santos, C., glar Gulçehre, Ç., Xiang, B.: Abstractive text summarization using sequence-to-sequence RNNs and beyond. CoNLL 2016, p. 280 (2016)
14. Napoles, C., Gormley, M., Van Durme, B.: Annotated gigaword. In: Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-Scale Knowledge Extraction, AKBC-WEKEX 2012, Stroudsburg, PA, USA, pp. 95–100 (2012)
15. Nayeem, M.T., Chali, Y.: Extract with order for coherent multi-document summarization. In: Proceedings of TextGraphs@ACL 2017: The 11th Workshop on Graph-based Methods for Natural Language Processing, Vancouver, Canada, 3 August 2017, pp. 51–56 (2017)

16. Nayeem, M.T., Chali, Y.: Paraphrastic fusion for abstractive multi-sentence compression generation. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, 06–10 November 2017, pp. 2223–2226 (2017)
17. Nayeem, M.T., Fuad, T.A., Chali, Y.: Abstractive unsupervised multi-document summarization using paraphrastic sentence fusion. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 1191–1204. Association for Computational Linguistics (2018)
18. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL 2002, Stroudsburg, PA, USA, pp. 311–318 (2002)
19. 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, pp. 379–389, September 2015
20. 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), Vancouver, Canada, pp. 1073–1083, July 2017
21. Servan, C., Berard, A., Elloumi, Z., Blanchon, H., Besacier, L.: Word2Vec vs DBnary: augmenting METEOR using vector representations or lexical resources? In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, Osaka, Japan, pp. 1159–1168, December 2016
22. Song, K., Zhao, L., Liu, F.: Structure-infused copy mechanisms for abstractive summarization. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 1717–1729. Association for Computational Linguistics (2018)
23. Suzuki, J., Nagata, M.: Cutting-off redundant repeating generations for neural abstractive summarization. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, Valencia, Spain, pp. 291–297, April 2017
24. Toutanova, K., Brockett, C., Tran, K.M., Amershi, S.: A dataset and evaluation metrics for abstractive compression of sentences and short paragraphs. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, Texas, pp. 340–350, November 2016
25. Xu, W., Napoles, C., Pavlick, E., Chen, Q., Callison-Burch, C.: Optimizing statistical machine translation for text simplification. Trans. Assoc. Comput. Linguist. **4**, 401–415 (2016)
26. Zhou, Q., Yang, N., Wei, F., 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), Vancouver, Canada, pp. 1095–1104, July 2017