Highlight-Transformer: Leveraging Key Phrase Aware Attention to Improve Abstractive Multi-Document Summarization

Shuaiqi LIU, Jiannong Cao, Ruosong Yang, and Zhiyuan Wen
Department of Computing, The Hong Kong Polytechnic University
{cssqliu, csjcao, csryang, cszwen}@comp.polyu.edu.hk

Abstract

Abstractive multi-document summarization aims to generate a comprehensive summary covering salient content from multiple input documents. Compared with previous RNN-based models, the Transformer-based models employ the self-attention mechanism to capture the dependencies in input documents and can generate better summaries. Existing works have not considered key phrases in determining attention weights of self-attention. Consequently, some of the tokens within key phrases only receive small attention weights. It can affect completely encoding key phrases that convey the salient ideas of input documents. In this paper, we introduce the Highlight-Transformer, a model with the highlighting mechanism in the encoder to assign greater attention weights for the tokens within key phrases. We propose two structures of highlighting attention for each head and the multi-head highlighting attention. The experimental results on the Multi-News dataset show that our proposed model significantly outperforms the competitive baseline models.

1 Introduction

Abstractive Multi-Document Summarization (MDS) offers the challenge of generating a comprehensive summary of multiple related documents. It requires summarization models to capture the salient content from input documents. Compared with the previous RNN-based models for abstractive MDS, the Transformer-based models (Gehrmann et al., 2018; Liu et al., 2018; Liu and Lapata, 2019a; Li et al., 2020b) employ the self-attention mechanism to capture the dependencies in input documents, and they can generate better summaries.

Calculating attention weights is a crucial step in the self-attention mechanism. Input documents usually contain some key phrases that convey the salient ideas of input documents. However, existing works have not considered key phrases in determining attention weights of self-attention. Key phrases usually comprise multiple tokens, which should be highly related and serve as a complete grammatical unit in input documents. When testing Transformer-based models, we observe some of the tokens within key phrases only receive small attention weights, which can affect completely encoding key phrases and the salient ideas they convey.

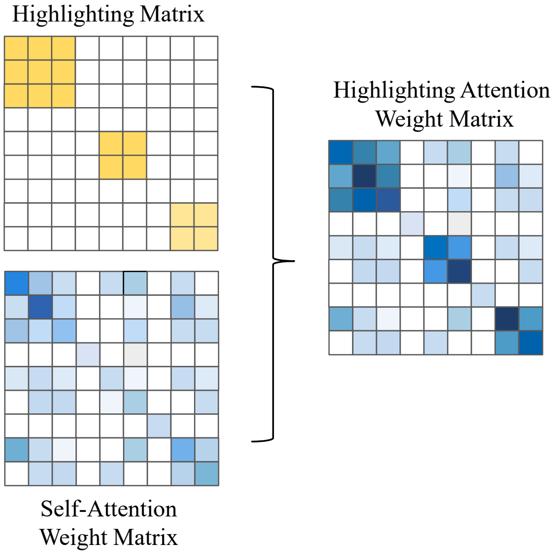
Highlighting Matrix
Highlighting Attention Weight Matrix
Self-Attention Weight Matrix

Figure 1: The highlighting mechanism assigns greater attention weights for tokens within key phrases indicated by the highlighting matrix.

In this paper, we propose the Highlight-Transformer, an abstractive summarization model with the highlighting mechanism in the encoder. As depicted in Figure 1, the highlighting mechanism assigns greater attention weights for tokens within key phrases. Furthermore, the highlighting mechanism mainly comprises three parts: the highlighting matrix, the highlighting attention, and the multi-head highlighting attention.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 5021–5027
August 1–6, 2021. ©2021 Association for Computational Linguistics

Our work is inspired by previous studies in education and psychology that indicate key phrases are important for people to understand (Rello et al., 2014; Hargreaves and Crabb, 2016) and summarize (Benzer et al., 2016; Chou, 2012) the given documents. Highlighting key phrases can help people with dyslexia improve comprehension (Rello et al., 2014; Hargreaves and Crabb, 2016). Their findings can be instructive to improve the self-attention mechanism.

We build a highlighting matrix for each input token sequence to indicate key phrases' positions in the attention weight matrix and phrases' importance values. We propose two structures of highlighting attention for each head to adjust attention weights according to the phrase importance. After comparing the effects of adopting the highlighting attention in the different numbers of heads and layers, we discover that adopting it in a subset of heads surpass adopting it in all heads. Experimental results on the Multi-News dataset (Fabbri et al., 2019) exhibit that our proposed model significantly improves the ROUGE scores (Lin, 2004) of generated summaries.

Our contribution is threefold:

- We present the highlighting mechanism that assigns greater attention weights for the tokens within key phrases.
- We propose the multi-head highlighting attention and two structures of highlighting attention for each head to combine attention weights with the phrase importance.
- Our proposed model significantly outperforms the competitive baseline models on the Multi-News dataset.

2 Related Work

Previous encoder-decoder models (Rush et al., 2015; Nallapati et al., 2016; Paulus et al., 2018; Chopra et al., 2016) equipped with the attention mechanism (Bahdanau et al., 2015) have achieved great performance on abstractive summarization. However, they were found to miss some important content in input documents (Li et al., 2018; Xu et al., 2020). How to retain the key information of input documents in the generated summaries has received increasing attention in the past few years. Some previous works focus on improving the copy mechanism. Gehrmann et al. (2018) utilize the attention masks to restrict copying phrases from the selected parts of an input document. Xu et al. (2020) explicitly guide the copy process with the centrality of each source word. Several papers also explore the potential of enhancing the encoder. Li et al. (2018, 2020a) extend the pointer-generator-based models (See et al., 2017) with a separate LSTM-based encoder to get the keywords' representation and then combine it with the sentence representation. In this work, we explore the potential of leveraging phrase importance as guidance to adjust attention weights in the multi-head self-attention of the Transformer encoder.

3 Model

In this section, we present the Highlight-Transformer, a model with the highlighting mechanism. We introduce its three main components: the highlighting matrix, the highlighting attention for each head, and the multi-head highlighting attention. We focus on the encoder part, and our decoder follows the CopyTransformer model used in (Gehrmann et al., 2018; Fabbri et al., 2019). Each input example of our proposed model includes the source articles, the articles' key phrases, and the phrases' importance values. The automatic key phrases extraction method we used will be introduced in section 4.1.

3.1 Highlighting Matrix

The first step of the highlighting mechanism is to build a highlighting matrix for each input example. It can indicate key phrases' positions in the attention weight matrix and the phrases' importance values. The concatenated source articles can be represented as an input sequence $(t_1, ..., t_n)$ containing n tokens. We use $(p_1, ..., p_k)$ and $(v_1, ..., v_k)$ to denote key phrases and their importance values. For each input example, We build the highlighting matrix $H \in \mathbb{R}^{n \times n}$ with the same shape as the self-attention weight matrix. Assuming a phrase p_r contains b tokens in the input sequence $p_r = (x_a, ..., x_{a+b})$, the phrase's importance value v_r is added to the elements $H_{i,j}$, where $i = a, \ldots, a+b, j = a, \ldots, a+b$, in the highlighting matrix. The phrases can be overlapping or nested, and the token t_i may be contained in c phrases $(p_r, ..., p_{r+c})$, whose importance values are $(v_r, ..., v_{r+c})$. The element H_{ii} is assigned as the sum of the c phrases' importance values.

3.2 Highlighting Attention

The highlighting attention is the key component in our proposed model for adjusting attention weights according to the phrase importance. For the head *m*, the Transformer model (Vaswani et al., 2017) adopts the scaled dot-product attention that operates on a query Q, a key K, and a value V:

$$\mathrm{Attention}(Q, K, V) = W^m V \tag{1a}$$

$$W^m = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}}) \tag{1b}$$

where $W^m \in \mathbb{R}^{n \times n}$, and d_k is the dimensionality of key. In the encoder layers, queries, keys, and values come from the output of the previous layer.

The highlighting matrix H can be used to determine which elements in the attention weight matrix should be increased. We propose two structures of highlighting attention, namely the weighted highlighting attention and the additive highlighting attention, to adjust attention weights according to the phrase importance.

The weighted highlighting attention mainly modifies Equation (1b) to calculate the attention weight matrix W^m for the head *m*. The highlighting matrix H is multiplied by a scalar α, named the brightness factor. The product will be added to the input of the softmax function.

$$W^m = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}} + \alpha H) \tag{2}$$

Since the softmax function applies the exponential function to each input element and divides them by the sum of all these exponentials, the above additive operation can be identical to calculating the weighted average.

$$\mathrm{softmax}(z_i + b_i) = \frac{e^{b_i} e^{z_i}}{\sum_{j=1}^{n} e^{b_j} e^{z_j}} \quad i = 1, \ldots, n \tag{3}$$

The additive highlighting attention is also designed to adjust the attention weight matrix W^m. In Equation (4a), the product of the highlighting matrix H and the scalar α is normalized by the softmax function[1] and added to the original attention weight matrix W_a^m calculated by Equation (1b). And then, elements in W_b^m will be normalized to ensure the sum of the attention weights equals one along the dimension where the softmax conducts.

$$W_b^m = W_a^m + \mathrm{softmax}(\alpha H_m) \tag{4a}$$

$$W^m_{:,\,j} = \frac{W^m_{b\;:,\,j}}{||W^m_{b\;:,\,j}||_1} \quad j = 1, \ldots, n \tag{4b}$$

3.3 Multi-Head Highlighting Attention

In our proposed model, the encoder with d_{model} consists of N layers and h heads. Each encoder layer has two sub-layers: the multi-head highlighting attention layer and the position-wise fully connected feed-forward network. We proposed the multi-head highlighting attention mechanism, which employs the highlighting attention on p highlighted heads and the scaled dot-product attention on the rest of $(h - p)$ normal heads.

$$\begin{aligned} &\mathrm{MultiHead}(Q, K, V) = \mathrm{Heads}W^o \\ &\mathrm{Heads} = \mathrm{Concat}(\mathrm{Head}_1, ..., \mathrm{Head}_h) \\ &\mathrm{Head_i} = \mathrm{Attention}(Q, K, V) \end{aligned} \tag{5}$$

where the projection is a parameter matrix $W^o \in \mathbb{R}^{hd_v \times d_{model}}$. The matrix Head_i is calculated by Equation (1a). The attention weight matrix W of the highlighted heads can be calculated by Equation (2) or (4), and that of the normal heads can be calculated by Equation (1b). The results on all heads will be concatenated and then projected through a feed-forward layer.

4 Experiments

4.1 Data Preparation

We train and evaluate our model on a MDS dataset named Multi-News (Fabbri et al., 2019), in which each example includes multiple news articles about the same event and a human-written summary collected from the website newser.com.

Following the setting in (Fabbri et al., 2019), we truncate each input article to 500/S tokens for the example with S news articles and concatenate the truncated articles into a single document. For each example, we first filter out stopwords and select candidate phrases from these truncated source articles. And then, we use the library named scikit-learn to calculate the candidate phrases' tf-idf values (Salton and Buckley, 1988) as their importance values. These candidate phrases are sorted in descending order of their importance values. We only select the top-10 bigrams or trigrams as key phrases

[1] Since the number of key phrases is limited, and the highlighting matrix can be sparse, we mask the zero elements and only conduct the softmax operation on the nonzero elements.

in each example since we observe longer phrases are sparse and more likely to be compressed in summary. Each input example of our proposed model includes the source articles, key phrases together with their L2 normalized tf-idf values.

4.2 Experimental Setting

We adopt a 4-layer encoder and a 4-layer decoder to build our proposed model, in which each layer has eight attention heads. Both the word embedding size and hidden size are set as 512. The maximum size of the vocabulary is set as 50000. Besides, we implement our model with the framework named OpenNMT-py (Klein et al., 2017).

The optimizer is Adam (Kingma and Ba, 2015) with learning rate 2, β_1=0.9, and β_2=0.998. Learning rate warmup is adopted to linearly increase the learning rate over the first 8,000 steps and then decrease it as the setting in (Vaswani et al., 2017). In addition, the brightness factor α in the highlighting attention also progressively decreases at the end of each epoch. Following the setting in (Fabbri et al., 2019), we also apply label smoothing (Szegedy et al., 2016) with smoothing factor 0.1 and dropout (Srivastava et al., 2014) with probability 0.2.

During testing, we use beam search with a beam size of 5. We also use trigram blocking to reduce repetitions. Our models are trained and evaluated on one NVIDIA Tesla P100 GPU.

4.3 Baselines

We compare our proposed Highlight-Transformer model with the following extractive and abstractive summarization methods.

LexRank and TextRank (Erkan and Radev, 2004; Mihalcea and Tarau, 2004) are two graph-based ranking methods that can be used for extractive summarization.

A **tf-idf**-based extractive summarization method (Christian et al., 2016) is evaluated to compare with introducing tf-idf score into our abstractive method.

BertExt (Liu and Lapata, 2019b) stacks inter-sentence Transformer layers on top of the pre-trained BERT (Devlin et al., 2019). We fine-tune this model on the Multi-News training set.

PG and PG-MMR are the pointer-generator (PG) network based summarization models reported by (Lebanoff et al., 2018).

Hi-MAP (Fabbri et al., 2019) extends the PG network into a hierarchical network. The attention distribution of tokens is multiplied by the MMR score of the sentence to which they belong.

Method	R-1	R-2	R-SU
LexRank	38.27	12.70	13.20
TextRank	38.44	13.10	13.50
tf-idf	38.68	12.09	13.54
BertExt	44.27	15.09	17.44
PG	41.85	12.91	16.46
PG-MMR	40.55	12.36	15.87
Hi-MAP	43.47	14.89	17.41
BertAbs	42.21	15.14	16.33
SAGCopy	43.98	15.21	17.65
CopyTransformer	43.57	14.03	17.37
Highlight (Weighted)	**44.62**	**15.57**	**18.06**
Highlight (Additive)	44.29	15.46	17.73

Table 1: Evaluation results on the Multi-News test set.

Highlight (Weighted)	R-1	R-2	R-SU
1/4 Heads 1/2 Layers	**44.62**	**15.57**	**18.06**
1/2 Heads 1/2 Layers	44.25	15.37	17.84
All Heads 1/2 Layers	44.18	15.12	17.70
1/4 Heads All Layers	44.32	15.16	17.82
1/2 Heads All Layers	44.41	15.50	17.84
All Heads All Layers	44.21	15.11	17.72

Table 2: Evaluation results on highlighting different numbers of heads and layers.

SAGCopy (Xu et al., 2020) adds the word centrality score to the linearly transformed hidden state when calculating the copy distribution.

BertAbs (Liu and Lapata, 2019b) adopts the pre-trained BERT as the encoder. A decoder with six Transformer layers is initialized randomly. We fine-tune this model on the Multi-News training set.

CopyTransformer (Gehrmann et al., 2018; Fabbri et al., 2019) adds the copy mechanism (See et al., 2017) to a 4-layer Transformer model. The decoder of our model follows its architecture.

4.4 Results and Discussion

We report the ROUGE F_1 (Lin, 2004) scores, including the overlap of unigrams (R-1), bigrams (R-2), and skip bigrams with a max distance of four words (R-SU). The results of LexRank, TextRank, PG, PG-MMR, Hi-MAP, and CopyTransformer follow Fabbri et al. (2019).

As shown in Table 1, the Highlight-Transformer significantly outperforms these baseline models on all metrics, which proves the effectiveness of the highlighting mechanism. Compared with the additive highlighting attention, the weighted high-

	Win	Lose	Tie	Kappa
Informativeness	46.5%	21.5%	32.0%	0.664
Fluency	29.5%	26.0%	44.5%	0.639
Non-Redundancy	27.5%	25.5%	47.0%	0.624

Table 3: Human evaluation results. "Win" represents the generated summary of our proposed model is better than that of CopyTransformer in one aspect.

lighting attention is more favorable.

We also compare the effects of adopting the weighted highlighting attention in different numbers of heads and layers in the encoder of our proposed model. The results on the test set of Multi-News are summarized in Table 2. It reveals that adopting it in a quarter of the heads and half of the layers achieves the best performance. We discover that adopting highlighting attention in a subset of heads surpasses adopting it in all heads. Besides, applying the multi-head highlighting attention on all layers of the encoder is also not optimal.

Multi-head attention in the Transformer model (Vaswani et al., 2017) is designed for jointly attending to information from different representation sub-spaces. Voita et al. (2019) find the heads in Transformer model trained on the neural machine translation dataset have specialized functions and focus on different types of information. Adopting the highlighting attention in all heads and layers will affect the Transformer-based model to encode other types of useful information and lead to performance degradation.

In addition to automatic evaluation, we performed a human evaluation to compare the generated summaries in terms of informativeness (the coverage of information from input documents), fluency (content organization and grammatical correctness), and non-redundancy (less repetitive information). We randomly selected 50 samples from the test set of the Multi-News dataset. Four annotators are required to compare two models' generated summaries that are presented anonymously. We also assess their agreements by Fleiss' kappa (Fleiss, 1971). The human evaluation results in Table 3 exhibit that the Highlight-Transformer significantly outperforms the CopyTransformer in terms of informativeness and is comparative in terms of fluency and non-redundancy.

The ablation study aims to validate the effectiveness of individual components in our proposed model. In Table 4, "w/o highlight attn" refers to the CopyTransformer model used in (Gehrmann et al., 2018; Fabbri et al., 2019). The results confirm that incorporating the highlighting attention is beneficial for multi-document summarization, and the decreasing brightness factor α also benefits our model's performance. Besides, we tried replacing the self-attention weight matrices in a quarter of the heads and half of the layers with the highlighting matrices. The performance degradation reveals that it is important to combine the attention weights with the phrase importance instead of directly replacing the attention weights.

	R-1	R-2	R-SU
Highlight-Transformer	**44.62**	**15.57**	**18.06**
w/o brightness	44.38	15.44	17.90
w/o highlight attn	43.57	14.03	17.37
w/o self-attention	42.54	14.40	16.54

Table 4: Ablation study on the Multi-News test set. "brightness" denotes the brightness factor α and "highlight attn" denotes the highlighting attention.

5 Conclusion

In this paper, we introduce the Highlight-Transformer, a novel summarization model with the highlighting mechanism in the encoder. The highlighting mechanism assigns greater attention weights for the tokens within key phrases, and it comprises three main parts: the highlighting matrix, the highlighting attention, and the multi-head highlighting attention. Specifically, a block diagonal highlighting matrix is built for each input token sequence to indicate key phrases' positions and phrases' importance values. For each head, we propose and compare two structures of highlighting attention. Furthermore, we also compare the effects of adopting the weighted highlighting attention in different numbers of heads and layers in the encoder of our proposed model. The experimental results exhibit the effectiveness of our proposed model. We intend to incorporate more phrase-level and sentence-level information into Transformer-based summarization models and evaluate them on different datasets in future work.

Acknowledgments

This work is supported by the Hong Kong RGC Collaborative Research Fund with project code C6030-18G and Hong Kong Red Swastika Society Tai Po Secondary School with project code P20-0021.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Ahmet Benzer, Ayşegül Sefer, Zeyneb Ören, and Sümeyye Konuk. 2016. A student-focused study: Strategy of text summary writing and assessment rubric. *Education & Science/Egitim ve Bilim*, 41(186).

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. 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, San Diego, California. Association for Computational Linguistics.

Mu-hsuan Chou. 2012. Implementing keyword and question generation approaches in teaching efl summary writing. *English Language Teaching*, 5(12):36–41.

Hans Christian, Mikhael Pramodana Agus, and Derwin Suhartono. 2016. Single document automatic text summarization using term frequency-inverse document frequency (tf-idf). *ComTech: Computer, Mathematics and Engineering Applications*, 7(4):285–294.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium. Association for Computational Linguistics.

Sandra Hargreaves and Jamie Crabb. 2016. *Study Skills for Students with Dyslexia: Support for Specific Learning Differences (SpLDs)*. Sage.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Logan Lebanoff, Kaiqiang Song, and Fei Liu. 2018. Adapting the neural encoder-decoder framework from single to multi-document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4131–4141, Brussels, Belgium. Association for Computational Linguistics.

Chenliang Li, Weiran Xu, Si Li, and Sheng Gao. 2018. Guiding generation for abstractive text summarization based on key information guide network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 55–60, New Orleans, Louisiana. Association for Computational Linguistics.

Haoran Li, Junnan Zhu, Jiajun Zhang, Chengqing Zong, and Xiaodong He. 2020a. Keywords-guided abstractive sentence summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8196–8203.

Wei Li, Xinyan Xiao, Jiachen Liu, Hua Wu, Haifeng Wang, and Junping Du. 2020b. Leveraging graph to improve abstractive multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6232–6243, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*.

Yang Liu and Mirella Lapata. 2019a. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081, Florence, Italy. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019b. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Romain Paulus, Caiming Xiong, and R. Socher. 2018. A deep reinforced model for abstractive summarization. *ArXiv*, abs/1705.04304.

Luz Rello, Horacio Saggion, and Ricardo Baeza-Yates. 2014. Keyword highlighting improves comprehension for people with dyslexia. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pages 30–37.

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

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. 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)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Song Xu, Haoran Li, Peng Yuan, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. Self-attention guided copy mechanism for abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1355–1362, Online. Association for Computational Linguistics.