IMPROVING LONG TEXT UNDERSTANDING WITH KNOWLEDGE DISTILLED FROM SUMMARIZATION MODEL

Yan Liu[1], *Yazheng Yang*[2], *Xiaokang Chen*[3]

[1]The Chinese University of Hong Kong,
[2]Department of Computer Science, Hong Kong University,
[3]School of Intelligence Science and Technology, Peking University,

ABSTRACT

Long text understanding is important yet challenging for natural language processing. A long article or document usually contains many redundant words that are not pertinent to its gist and sometimes can be regarded as noise. With recent advances of abstractive summarization, we propose our *Gist Detector* to leverage the gist detection ability of a summarization model and integrate the extracted gist into downstream models to enhance their long text understanding ability. Specifically, Gist Detector first learns the gist detection knowledge distilled from a summarization model, and then produces gist-aware representations to augment downstream models. We evaluate our method on three different tasks: long document classification, distantly supervised open-domain question answering, and non-parallel text style transfer. The experimental results show that our method can significantly improve the performance of baseline models on all tasks.

Index Terms— long text understanding, distillation, gist detection

1. INTRODUCTION

Recently, deep learning has developed rapidly [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. Transformer-based models are prevalent [13, 14, 15, 16, 17, 18] across numerous NLP tasks[19], but have difficulty in processing long texts due to the quadratic complexity of input text length[20]. Unlike short texts, long texts intrinsically contain many noisy words irrelevant to their gist. Although recent works have achieved promising results, few of them pay attention to measuring whether each part of the text is salient or negligible. Abstractive summarization is a classic NLP task which aims to compress and rewrite a source text into a short version while retaining its main information [21, 22]. With this optimization objective, a well-trained summarization model has the potential to detect the gist of long texts. Figure 1 shows an example from the *CNN/Daily Mail* [23] dataset, where the blue shading intensity represents the importance weight extracted from a well-trained summarization model. As we can see, the summarization model learns to focus on gist-relevant parts while neglecting irrelevant ones. Intuitively, the gist detection ability can improve long text understanding through making models aware of salient parts of long texts.

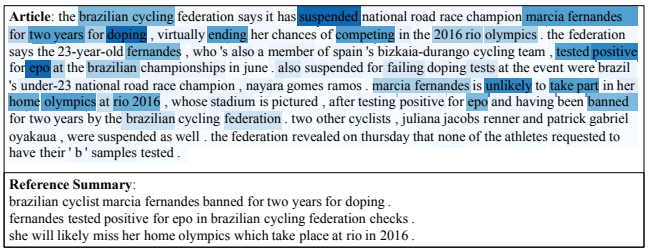

Fig. 1. An example from the CNN/Daily Mail dataset. The shading intensity represents the importance weight extracted from a well-trained summarization model.

In this paper, we propose to leverage the gist detection ability of a summarization model and integrate the distilled gist information into downstream models to enhance their long text understanding ability. However, there remain two challenges: First, it is time-consuming to extract salient information from a large summarization model for each training sample. Second, the summarization model produces salient information at each decoding step, while long text understanding models produce a single representation.

To solve these challenges, we propose our Gist Detector to transfer the gist information from a summarization model to downstream long text understanding models. Specifically, Gist Detector is first trained to reproduce the gist information from the summarization model, then provides the gist-aware representation as supplementary to augment long text understanding models. We train our Gist Detector with knowledge distillation mechanism, where a summarization model with an encoder-decoder architecture is the teacher model and Gist Detector with a fewer-layers' encoder is the student model. The student model is trained with the average attention distribution over all decoding steps produced by the teacher model as "soft target". Since Gist Detector is a non-autogressive model and much smaller than the summarization model, the process of gist extraction can be significantly efficient. Then, we integrate the gist information extracted by our distilled

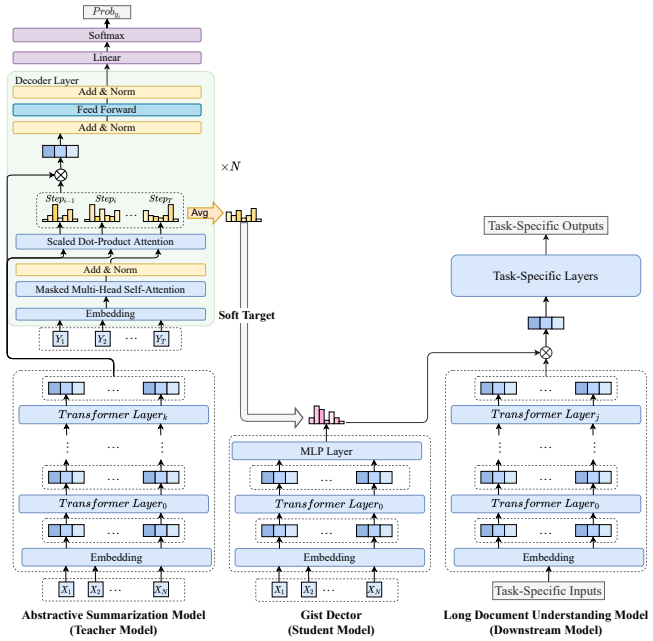

Fig. 2. Gist Detector is trained to reproduce the salient information from the teacher model, and provides the salient-aware representations as supplementary to augment the downstream model.

Gist Detector into downstream models with a fuse module, effectively enhancing their long text understanding ability.

To evaluate the effectiveness of our method, we conduct extensive experiments on three tasks: long document classification, distantly supervised open-domain question answering (DS-QA) and non-parallel text style transfer. Experimental results reveal that our method effectively augments different baseline models with better long text understanding ability, thus achieving significant performance improvement on all downstream tasks.

2. METHODOLOGY

In this paper, we propose our Gist Detector to leverage the gist detection ability of summarization model, and transfer gist information into downstream long text understanding models. We first introduce the architecture of Gist Detector (§ 4.1). During training, we use the knowledge distillation mechanism to transfer the gist detection ability from a well-trained summarization model (teacher model) to Gist Detector (student model) (§ 2.2). Then, we integrate gist information extracted by Gist Detector into downstream models (§ 2.3). The much smaller model size and the non-autogressive architecture reduce the time-consuming problem, and the generated single gist-aware representation overcomes the mismatch problem.

2.1. Gist Detector Architecture

As shown in middle part of Figure 2, Gist Detector has an encoder architecture, which learns the importance weight of each word in the source sequence from the summarization model, and produces this information for downstream models. There are many possible network architectures for Gist Detector. We implement our Gist Detector with several Transformer encoder layers [24], and show that the a simple distilled Gist Detector can successfully benefit the long document understanding models.

Specifically, the input $\{x_1, ..., x_N\}$ is firstly mapped into embeddings $\{\mathbf{e}_1, ..., \mathbf{e}_N\}$, then fed into a four-layer transformer encoder and obtain the representations $\mathbf{H} = \{\mathbf{h}_1, ..., \mathbf{h}_N\}$. Then, a two-layer MLP followed by a softmax function is applied to produce the the probability distribution over the input text $\mathbf{p} = \{p_1, ..., p_N\}$, which reveals the importance of each word in the source sequence.

2.2. Training with knowledge distillation

We leverage the knowledge distillation mechanism to train Gist Detector (student model) with the salient information extracted from the abstractive summarization model (teacher model). Different from the typical knowledge distillation, which uses the teacher's predictive distribution over the target classes as the soft target, we assume the attention distribution extracted from the decoding process reveals the salient information of the source text, and use the teacher's attention distribution as the soft target. The student model learns to reproduce the attention distribution for each training sample.

Specifically, the soft target $\mathbf{q} = \{q_1, ..., q_N\}_{n=1}$ is calculated as the geometric mean of the attention distribution over all decoding steps:

$$q_n = \frac{\sum_t a_{n,t}}{T} \tag{1}$$

, where T is the total decoding steps. Finally, the optimization objective is the cross entropy between the predicted probability distribution $\mathbf{p}$ of the student model and the soft target $\mathbf{q}$ from the teacher model:

$$L_{KD} = -\sum_{(x,y)} \sum_{n=1}^{N} q_n log(p_n)) \tag{2}$$

2.3. Integration of salient information

To enhance the long document understanding ability of the downstream model, we extract the salient information from the well-trained Gist Detector, and integrate it into the downstream model with a fuse module.

Specifically, for each long text $\{x_1, ..., x_N\}$ as the input, the Gist Detector produces the probability distribution

Methods	Appeal	Baby	Books	Camera	DVD	Electronics	Health	IMDB	Kitchen	Magazines	MR	Music	Software	Sports	Toys	Video	Overall
ASP-MTL	87.0	88.2	84.0	89.2	85.5	86.8	88.2	85.5	86.2	92.2	76.7	82.5	87.2	85.7	88.0	84.5	86.1
S-LSTM	85.8	86.3	83.4	90.0	85.5	83.3	86.5	87.2	84.5	93.8	76.2	82.0	87.8	85.8	85.3	86.8	85.6
Meta-MTL	87.0	88.0	87.5	89.7	88.0	89.5	90.3	88.0	91.3	91.0	77.0	86.3	88.5	86.7	88.5	88.3	87.9
BiLSTM	84.8	84.5	78.8	86.3	80.7	81.9	83.0	79.8	82.1	90.5	76.1	79.6	85.4	80.3	83.9	81.1	82.4
BiLSTM+GD	87.6	88.5	86.7	90.8	87.8	89.6	88.2	88.1	90.7	94.6	78.2	86.4	90.3	87.1	88.3	88.5	88.2
- w/o KD	86.8	86.4	81.7	89.1	82.9	82.3	84.5	81.2	85.0	91.7	76.5	82.9	87.8	84.9	85.4	84.6	84.6

Table 1. Document classification results across 16 domains of FDU-MTL datasets. GD denotes our Gist Detector method. KD denotes the knowledge distillation training.

$\mathbf{p} = \{p_1, ..., p_N\}$ over the input text, revealing the importance weights of each word. Given the context representation of the long document understanding model $\mathbf{c} = \sum_n^N \mathbf{s_n}$, we fuse the context representation $\mathbf{c}$ with the importance weights $\mathbf{p}$ as:

$$\mathbf{c}' = (1 - \lambda)\mathbf{c} + \lambda \sum_t p_t \mathbf{s}_t \tag{3}$$

, where $\lambda \in [0, 1]$ is a tunable hyperparameter. As for the downstream model that predict scores for each word of the input text, such as extractive QA models, we fuse the prediction scores $\{r_1, ..., r_N\}$ with the importance weights $\{p_1, ..., p_N\}$:

$$r'_t = (1 - \lambda')r_t + \lambda p_t \tag{4}$$

Note that we use the importance weight rather than the context representation as the salient information, since it contains much less parameters and alleviates the impact of domain-specific information.

3. EXPERIMENTS

3.1. Distillation

Firstly, We train an ensemble of 8 abstractive summarization models with Transformer-based encoder-decoder architecture as the teacher model on *CNN/Daily Mail*. The average ROUGE F_1 scores [25] of the teacher model are 38.6, 16.3 and 35.4 for ROUGE-1, ROUGE-2 and ROUGE-L respectively. We follow the same setup and use the scripts provided by [26] to pre-process the *CNN/Daily Mail* dataset. We use the 100 dimensional filters with width of 5 for CNN to capture the character embeddings. We select the 300d GloVe pre-trained word embedding and share the same word embedding weight between encoder and decoder. The hidden size of Transformer is 512. We use the Adam optimizer [27] with learning rate of 0.0004, $\beta_1 = 0.9$, $\beta_2 = 0.999$. The dropout rate and batch size are set to 0.35 and 16, respectively. To avoid the gradient explosion problem, we apply the gradient norm clipping with a maximum gradient norm of 2.0.

Then we train Gist Detector with Transformer-based encoder architecture using knowledge distillation mechanism. We use 100d GloVe for word embedding, 50d for character embedding, the hidden size for the Transformer encoder is 256. We take the same optimization setting as that of the teacher model.

3.2. Integration into Downstream Tasks

Finally, we transfer the salient information from the well-trained Gist Detector to downstream models of three long text understanding tasks: document classification, distantly supervised open-domain question answering (DS-QA) and non-parallel text style transfer.

3.2.1. Document Classification:

We take the BiLSTM model as our baseline model for document classification task that concatenates the final state values of forward and backward pass as the context representation vector, then feeds it into a MLP to predict the label. We initialize the word embedding with the 300d GloVe. The hidden size of BiLSTM is set as 256. The layer number of BiLSTM and MLP are both set to 2. We take the Adam as optimizer with lr = 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, 0.35 dropout and train for 6 epochs. The λ in § 2.3 is set to be 0.5 while integrating the BiLSTM model with our Gist Detector.

3.2.2. Distantly Supervised Open-Domain QA:

We use the OpenQA model[28] as our baseline model for distantly supervised open-domain question answering task, which applies a selector to filter passages, then a precise reader to extract the potential answers, finally aggregates these results to predict the final answer. We evaluate our method on two high-quality datasets, *TriviaQA* (open-domain setting)[29] and *SearchQA*[30] with two metrics including ExactMatch (EM) and F1 scores. We keep the same setup of hyper-parameters and training settings as that in OpenQA while some important details are as follow. We combine the passage selector with Gist Detector as introduced in § 2.3 and the λ is set as 0.5. We feed the $\mathbf{c}'$ through a linear function followed by multiplication with the question vector to produce the score for filtering passages and add it to the original score produced by the OpenQA selector to predict the final passage score. For the reader, we directly add the predicted score of answer span with the probability distribution $\mathbf{p}$ produced by Gist Detector as introduced in § 2.3 to produce the final score, where the λ' is set as 0.2.

3.2.3. Text Style Transfer:

As for the non-parallel text style transfer task, the model aims to compress gist of texts into fixed-size vectors separated from pure style information. We select Cross-aligned AE[31] and Adversarially Regularized Autoencoder (ARAE)[32] as our baseline models. We follow the setup of [31] but remain reviews whose length are between 70 and 150 rather than not exceeding 15, and eventually obtain 350K, 280K non-parallel data from Amazon and Yelp reviews respectively. We keep

QA Models	TriviaQA		SearchQA	
	EM	F1	EM	F1
BiDAF [37]	-	-	28.6	34.6
AQA [38]	-	-	40.5	47.4
R^3 [39]	47.3	53.7	49.0	55.3
Re-Ranker [40]	50.6	57.3	57.0	63.2
TraCRNet [41]	-	-	52.9	65.1
OpenQA [28]	48.7	56.3	58.8	64.5
OpenQA + GD	50.3	57.6	59.5	65.1
- w/o GD in selector	49.2	56.5	59.0	64.8
- w/o GD in reader	49.4	57.1	59.2	64.8

Table 2. EM and F1 results on the TriviaQA (open-domain setting) and SearchQA datasets.

QA Models	TriviaQA			SearchQA		
	Hit@1	Hit@3	Hit@5	Hit@1	Hit@3	Hit@5
OpenQA	43.4	51.5	54.5	59.1	68.7	76.3
OpenQA + GD	49.1	57.7	63.1	65.3	73.4	79.6

Table 3. Performance of passage selection on TriviaQA and SearchQA development set. Hit@N represents the proportion of related passages being ranked in top-N.

the same setup of hyper-parameters and training settings as that of Cross-aligned AE and ARAE. We combine the content vector with our Gist Detector as introduced in § 2.3, and the λ is set to be 0.5. To evaluate the model, we use 4 automatic metrics: (i) Acc: the accuracy of successfully changing the style into the target style measured by a pre-trained classifier. Following [31], we use the TextCNN model as the classifier that achieves the accuracy of 94.2% and 95.7% on Amazon and Yelp respectively. (ii) Cosine: we follow the setup of [33] to measure the content preservation with cosine similarity. (iii) Entity: we use the proportion of noun entities to measure the content consistency between source and generated texts. (iv) PPL: the fluency of generated texts measured by a pre-trained language model on corresponding datasets.

4. RESULTS AND ANALYSIS

4.1. Results on Document Classification

We evaluate our our method across 16 domains on FDU-MTL datasets[34]. As shown in Table 1, augmented with our Gist Detector, the baseline BiLSTM model obtains significant performance improvement on all of the 16 domains and outperforms prior approaches (ASP-MTL [34], S-LSTM [35], Meta-MTL [36]) with 88.2 overall accuracy. An ablation study shows that if we use Gist Detector with random initializad parameters, the overall performance drops 3.6. It indicates that both the additional parameters from Gist Detector and the gist detection ability distilled from the summarization model contributes to the performance improvement.

4.2. Results on DS-QA

We evaluate our method on TriviaQA (open-domain setting) [29] and SearchQA [30] datasets with ExactMatch (EM) and F1 score metrics. As shown in Table 2, Augumented with our Gist Detector, the baseline OpenQA model performs much better on both two datasets. An ablation study shows that integration of salient information into both the selector and the reader leads to the best performance. Table 3 shows the passage selection performance of our method. We find that with Gist Detector, the selector filters passages much more precisely, thus our QA system can aggregate information among fewer passages and make faster answer predictions.

Models	Amazon				Yelp			
	Acc	Cosine	Entity	PPL	Acc	Cosine	Entity	PPL
Cross-aligned AE	84.7%	0.46	26.13	34.67	89.5%	0.53	26.63	28.46
ARAE	86.2%	0.57	31.37	36.36	89.3%	0.61	32.46	29.18
ARAE + GD	91.0%	0.71	47.56	24.15	93.4%	0.73	49.04	21.43

Table 4. Automatic evaluation results on Amazon and Yelp text style transfer datasets.

Models	Amazon			Yelp		
	Acc	Correlation	Fluency	Acc	Correlation	Fluency
Cross-aligned AE	56.4%	2.4	3.0	58.2%	2.7	3.1
ARAE	73.6%	2.8	3.3	74.1%	3.1	3.5
ARAE + GD	78.2%	3.7	3.5	78.6%	3.9	3.8

Table 5. Human evaluation on accuracy, content correlation and fluency of the generated text.

4.3. Results on Text Style Transfer

We further evaluate our method on the Amazon and Yelp text style transfer dataset [31]. The automatic evaluation results from Table 4 shows that with our Gist Detector, the baseline model ARAE[32] can achieve significantly higher transfer accuracy, better content preservation, better noun entity preservation and much more fluency. It indicates that the Gist Detector helps the model detect and compress more important information from long texts. Moreover, we conduct human evaluation to further evaluate the quality of the style transfer models. We randomly select 1000 examples ($500/500$ positive/negative), and employ people to judge whether texts are converted to the target style, and to evaluate content correlation ($0-5$, 5 for the most correlative) and fluency ($0-5$, 5 for the most fluent). As shown in Table 5, Gist Detector can significantly improve the baseline model's performance on all evaluation metrics.

5. CONCLUSION

In this paper, we propose Gist Detector to learn gist detection ability from a summarization model with knowledge distillation mechanism. We integrate the gist information detected by distilled Gist Detector into different downstream models to enhance their long document understanding ability. Experimental results show that our method significantly improves the performance of all baseline models for different tasks that require long text understanding. Future work will involve finding better strategies to integrate our gist detector into more tasks and processing longer sequences.

6. REFERENCES

[1] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang, "Context autoencoder for self-supervised representation learning," *arXiv preprint arXiv:2202.03026*, 2022.

[2] Xiaokang Chen, Jiahui Chen, Yan Liu, and Gang Zeng, "D^3etr: Decoder distillation for detection transformer," *arXiv preprint arXiv:2211.09768*, 2022.

[3] Qiang Chen, Xiaokang Chen, Jian Wang, Haocheng Feng, Junyu Han, Errui Ding, Gang Zeng, and Jingdong Wang, "Group detr: Fast detr training with group-wise one-to-many assignment," *arXiv preprint arXiv:2207.13085*, vol. 1, no. 2, 2022.

[4] Jiaxiang Tang, Xiaokang Chen, Jingbo Wang, and Gang Zeng, "Not all voxels are equal: Semantic scene completion from the point-voxel perspective," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, vol. 36, pp. 2352–2360.

[5] Xiaokang Chen, Fangyun Wei, Gang Zeng, and Jingdong Wang, "Conditional detr v2: Efficient detection transformer with box queries," *arXiv preprint arXiv:2207.08914*, 2022.

[6] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang, "Conditional detr for fast training convergence," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3651–3660.

[7] Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang, "Semi-supervised semantic segmentation with cross pseudo supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2613–2622.

[8] Xiaokang Chen, Kwan-Yee Lin, Jingbo Wang, Wayne Wu, Chen Qian, Hongsheng Li, and Gang Zeng, "Bi-directional cross-modality feature propagation with separation-and-aggregation gate for rgb-d semantic segmentation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI*. Springer, 2020, pp. 561–577.

[9] Xiaokang Chen, Kwan-Yee Lin, Chen Qian, Gang Zeng, and Hongsheng Li, "3d sketch-aware semantic scene completion via semi-supervised structure prior," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4193–4202.

[10] Xiaokang Chen, Yajie Xing, and Gang Zeng, "Real-time semantic scene completion via feature aggregation and conditioned prediction," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 2830–2834.

[11] Jiaxiang Tang, Xiaokang Chen, Jingbo Wang, and Gang Zeng, "Compressible-composable nerf via rank-residual decomposition," *arXiv preprint arXiv:2205.14870*, 2022.

[12] Jiaxiang Tang, Xiaokang Chen, Jingbo Wang, and Gang Zeng, "Point scene understanding via disentangled instance mesh reconstruction," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*. Springer, 2022, pp. 684–701.

[13] Yan Liu, Xiaokang Chen, and Qi Dai, "Parallel sentence-level explanation generation for real-world low-resource scenarios," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[14] Yan Liu, Xiaokang Chen, Yan Gao, Zhe Su, Fengji Zhang, Daoguang Zan, Jian-Guang Lou, Pin-Yu Chen, and Tsung-Yi Ho, "Uncovering and quantifying social biases in code generation," *arXiv preprint arXiv:2305.15377*, 2023.

[15] Yan Liu, Yan Gao, Zhe Su, Xiaokang Chen, Elliott Ash, and Jian-Guang Lou, "Uncovering and categorizing social biases in text-to-sql," *arXiv preprint arXiv:2305.16253*, 2023.

[16] Daoguang Zan, Bei Chen, Zeqi Lin, Bei Guan, Wang Yongji, and Jian-Guang Lou, "When language model meets private library," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, Eds., Abu Dhabi, United Arab Emirates, Dec. 2022, pp. 277–288, Association for Computational Linguistics.

[17] Daoguang Zan, Ailun Yu, Bo Shen, Jiaxin Zhang, Taihong Chen, Bing Geng, Bei Chen, Jichuan Ji, Yafen Yao, Yongji Wang, and Qianxiang Wang, "Can programming languages boost each other via instruction tuning?," 2023.

[18] Daoguang Zan, Bei Chen, Fengji Zhang, Dianjie Lu, Bingchao Wu, Bei Guan, Yongji Wang, and Jian-Guang Lou, "Large language models meet nl2code: A survey," 2023.

[19] Yan Liu, Sanyuan Chen, Yazheng Yang, and Qi Dai, "Mpii: Multi-level mutual promotion for inference and interpretation," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 7074–7084.

[20] Maor Ivgi, Uri Shaham, and Jonathan Berant, "Efficient long-text understanding with short-text models," *arXiv preprint arXiv:2208.00748*, 2022.

[21] Tiezheng Yu, Zihan Liu, and Pascale Fung, "Adaptsum: Towards low-resource domain adaptation for abstractive summarization," *CoRR*, vol. abs/2103.11332, 2021.

[22] Feng Nan, Ramesh Nallapati, Zhiguo Wang, Cícero Nogueira dos Santos, Henghui Zhu, Dejiao Zhang, Kathleen R. McKeown, and Bing Xiang, "Entity-level factual consistency of abstractive text summarization," *CoRR*, vol. abs/2102.09130, 2021.

[23] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang, "Abstractive text summarization using sequence-to-sequence rnns and beyond," in *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, 2016.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017.

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

[26] Abigail See, Peter J Liu, and Christopher D Manning, "Get to the point: Summarization with pointer-generator networks," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017.

[27] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[28] Yankai Lin, Haozhe Ji, Zhiyuan Liu, and Maosong Sun, "Denoising distantly supervised open-domain question answering," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018.

[29] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer, "Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 1601–1611.

[30] Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho, "Searchqa: A new q&a dataset augmented with context from a search engine," *arXiv preprint arXiv:1704.05179*, 2017.

[31] Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola, "Style transfer from non-parallel text by cross-alignment," in *Advances in neural information processing systems*, 2017, pp. 6830–6841.

[32] Junbo Jake Zhao, Yoon Kim, Kelly Zhang, Alexander M. Rush, and Yann LeCun, "Adversarially regularized autoencoders," in *Proceedings of the 35th International Conference on Machine Learning,ICML 2018*.

[33] Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan, "Style transfer in text: Exploration and evaluation," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[34] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang, "Adversarial multi-task learning for text classification," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017.

[35] Yue Zhang, Qi Liu, and Linfeng Song, "Sentence-state lstm for text representation," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018.

[36] Junkun Chen, Xipeng Qiu, Pengfei Liu, and Xuanjing Huang, "Meta multi-task learning for sequence modeling," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[37] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi, "Bidirectional attention flow for machine comprehension," in *5th International Conference on Learning Representations, ICLR 2017*.

[38] Christian Buck, Jannis Bulian, Massimiliano Ciaramita, Wojciech Gajewski, Andrea Gesmundo, Neil Houlsby, and Wei Wang, "Ask the right questions: Active question reformulation with reinforcement learning," in *6th International Conference on Learning Representations, ICLR 2018*.

[39] Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerry Tesauro, Bowen Zhou, and Jing Jiang, "R 3: Reinforced ranker-reader for open-domain question answering," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[40] Shuohang Wang, Mo Yu, Jing Jiang, Wei Zhang, Xiaoxiao Guo, Shiyu Chang, Zhiguo Wang, Tim Klinger, Gerald Tesauro, and Murray Campbell, "Evidence aggregation for answer re-ranking in open-domain question answering," in *6th International Conference on Learning Representations, ICLR 2018*.

[41] Mostafa Dehghani, Hosein Azarbonyad, Jaap Kamps, and Maarten de Rijke, "Learning to transform, combine, and reason in open-domain question answering," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019*.