Multi-Reward Reinforced Summarization with Saliency and Entailment

Ramakanth Pasunuru and **Mohit Bansal**
UNC Chapel Hill
{ram, mbansal}@cs.unc.edu

Abstract

Abstractive text summarization is the task of compressing and rewriting a long document into a short summary while maintaining saliency, directed logical entailment, and non-redundancy. In this work, we address these three important aspects of a good summary via a reinforcement learning approach with two novel reward functions: ROUGESal and Entail, on top of a coverage-based baseline. The ROUGESal reward modifies the ROUGE metric by up-weighting the salient phrases/words detected via a keyphrase classifier. The Entail reward gives high (length-normalized) scores to logically-entailed summaries using an entailment classifier. Further, we show superior performance improvement when these rewards are combined with traditional metric (ROUGE) based rewards, via our novel and effective multi-reward approach of optimizing multiple rewards simultaneously in alternate mini-batches. Our method achieves the new state-of-the-art results on CNN/Daily Mail dataset as well as strong improvements in a test-only transfer setup on DUC-2002.

1 Introduction

Abstractive summarization, the task of generating a natural short summary of a long document, is more challenging than the extractive paradigm, which only involves selection of important sentences or grammatical sub-sentences (Jing, 2000; Knight and Marcu, 2002; Clarke and Lapata, 2008; Filippova et al., 2015). Advent of sequence-to-sequence deep neural networks and large human summarization datasets (Hermann et al., 2015; Nallapati et al., 2016) made the abstractive summarization task more feasible and accurate, with recent ideas ranging from copy-pointer mechanism and redundancy coverage, to metric reward based reinforcement learning (Rush et al., 2015; Chopra et al., 2016; Ranzato et al., 2015; Nallapati et al., 2016; See et al., 2017).

A good abstractive summary requires several important properties, e.g., it should choose the most salient information from the input document, be logically entailed by it, and avoid redundancy. Coverage-based models address the latter redundancy issue (Suzuki and Nagata, 2016; Nallapati et al., 2016; See et al., 2017), but there is still a lot of scope to teach current state-of-the-art models about saliency and logical entailment. Towards this goal, we improve the task of abstractive summarization via a reinforcement learning approach with the introduction of two novel rewards: 'ROUGESal' and 'Entail', and also demonstrate that these saliency and entailment skills allow for better generalizability and transfer.

Our ROUGESal reward gives higher weight to the important, salient words in the summary, in contrast to the traditional ROUGE metric which gives equal weight to all tokens. These weights are obtained from a novel saliency scorer, which is trained on a reading comprehension dataset's answer spans to give a saliency-based probability score to every token in the sentence. Our Entail reward gives higher weight to summaries whose sentences logically follow from the ground-truth summary. Further, we also add a length normalization constraint to our Entail reward, to importantly avoid misleadingly high entailment scores to very short sentences.

Empirically, we show that our new rewards with policy gradient approaches perform significantly better than a cross-entropy based state-of-the-art pointer-coverage baseline. We show further performance improvements by combining these rewards via our novel multi-reward optimization approach, where we optimize multiple rewards simultaneously in alternate mini-batches (hence avoiding complex scaling and weighting issues in

Proceedings of NAACL-HLT 2018, pages 646–653
New Orleans, Louisiana, June 1 - 6, 2018.

reward combination), inspired from how humans take multiple concurrent types of rewards (feedback) to learn a task. Overall, our methods achieve the new state-of-the-art on the CNN/Daily Mail dataset as well as strong improvements in a test-only transfer setup on DUC-2002. Lastly, we present several analyses of our model's saliency, entailment, and abstractiveness skills.

2 Related Work

Earlier summarization work was based on extraction and compression-based approaches (Jing, 2000; Knight and Marcu, 2002; Clarke and Lapata, 2008; Filippova et al., 2015), with more focus on graph-based (Giannakopoulos, 2009; Ganesan et al., 2010) and discourse tree-based (Gerani et al., 2014) models. Recent focus has shifted towards abstractive, rewriting-based summarization based on parse trees (Cheung and Penn, 2014; Wang et al., 2016), Abstract Meaning Representations (Liu et al., 2015; Dohare and Karnick, 2017), and neural network models with pointer-copy mechanism and coverage (Rush et al., 2015; Chopra et al., 2016; Chen et al., 2016; Nallapati et al., 2016; See et al., 2017), as well as reinforce-based metric rewards (Ranzato et al., 2015; Paulus et al., 2017). We also use reinforce-based models, but with novel reward functions and better simultaneous multi-reward optimization methods.

Recognizing Textual Entailment (RTE), the task of classifying two sentences as entailment, contradiction, or neutral, has been used for Q&A and IE tasks (Harabagiu and Hickl, 2006; Dagan et al., 2006; Lai and Hockenmaier, 2014; Jimenez et al., 2014). Recent neural network models and large datasets (Bowman et al., 2015; Williams et al., 2017) enabled stronger accuracies. Some previous work (Mehdad et al., 2013; Gupta et al., 2014) has explored the use of RTE by modeling graph-based relationships between sentences to select the most non-redundant sentences for summarization. Recently, Pasunuru and Bansal (2017) improved video captioning with entailment-corrected rewards. We instead directly use multi-sentence entailment knowledge (with additional length constraints) as a separate RL reward to improve abstractive summarization, while avoiding their penalty hyperparameter tuning.

For our saliency prediction model, we make use of the SQuAD reading comprehension dataset (Rajpurkar et al., 2016), where the answer spans annotated by humans for important questions, serve as an interesting and effective proxy for keyphrase-style salient information in summarization. Some related previous work has incorporated document topic/subject classification (Isonuma et al., 2017) and webpage keyphrase extraction (Zhang et al., 2004) to improve saliency in summarization. Some recent work Subramanian et al. (2017) has also used answer probabilities in a document to improve question generation.

3 Models

3.1 Baseline Sequence-to-Sequence Model

Our abstractive text summarization model is a simple sequence-to-sequence single-layer bidirectional encoder and unidirectional decoder LSTM-RNN, with attention (Bahdanau et al., 2015), pointer-copy, and coverage mechanism – please refer to See et al. (2017) for details.

3.2 Policy Gradient Reinforce

Traditional cross-entropy loss optimization for sequence generation has an exposure bias issue and the model is not optimized for the evaluated metrics (Ranzato et al., 2015). Reinforce-based policy gradient approach addresses both of these issues by using its own distribution during training and by directly optimizing the non-differentiable evaluation metrics as rewards. We use the REINFORCE algorithm (Williams, 1992; Zaremba and Sutskever, 2015) to learn a policy p_θ defined by the model parameters θ to predict the next action (word) and update its internal (LSTM) states. We minimize the loss function $L_{\text{RL}} = -\mathbb{E}_{w^s \sim p_\theta}[r(w^s)]$, where w^s is the sequence of sampled words with w_t^s sampled at time step t of the decoder. The derivative of this loss function with approximation using a single sample along with variance reduction with a bias estimator is:

$$\nabla_\theta L_{\text{RL}} = -(r(w^s) - b_e)\nabla_\theta \log p_\theta(w^s) \quad (1)$$

There are several ways to calculate the baseline estimator; we employ the effective SCST approach (Rennie et al., 2016), as depicted in Fig. 1, where $b_e = r(w^a)$, is based on the reward obtained by the current model using the test time inference algorithm, i.e., choosing the arg-max word w_t^a of the final vocabulary distribution at each time step t of the decoder. We use the joint cross-entropy and reinforce loss so as to optimize the non-differentiable evaluation metric as reward

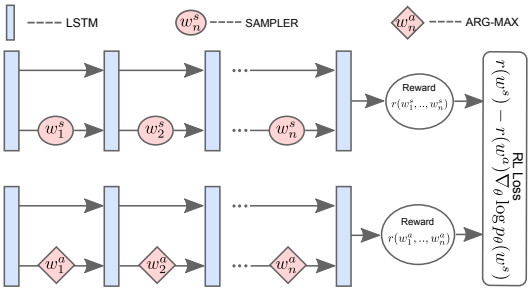

Figure 1: Our sequence generator with RL training.

while also maintaining the readability of the generated sentence (Wu et al., 2016; Paulus et al., 2017; Pasunuru and Bansal, 2017), which is defined as $L_{\text{Mixed}} = \gamma L_{\text{RL}} + (1-\gamma)L_{\text{XE}}$, where γ is a tunable hyperparameter.

3.3 Multi-reward Optimization

Optimizing multiple rewards at the same time is important and desired for many language generation tasks. One approach would be to use a weighted combination of these rewards, but this has the issue of finding the complex scaling and weight balance among these reward combinations. To address this issue, we instead introduce a simple multi-reward optimization approach inspired from multi-task learning, where we have different tasks, and all of them share all the model parameters while having their own optimization function (different reward functions in this case). If r_1 and r_2 are two reward functions that we want to optimize simultaneously, then we train the two loss functions of Eqn. 2 in alternate mini-batches.

$$\begin{aligned} L_{\text{RL}_1} &= -(r_1(w^s) - r_1(w^a))\nabla_\theta \log p_\theta(w^s) \\ L_{\text{RL}_2} &= -(r_2(w^s) - r_2(w^a))\nabla_\theta \log p_\theta(w^s) \end{aligned} \quad (2)$$

4 Rewards

ROUGE Reward The first basic reward is based on the primary summarization metric of ROUGE package (Lin, 2004). Similar to Paulus et al. (2017), we found that ROUGE-L metric as a reward works better compared to ROUGE-1 and ROUGE-2 in terms of improving all the metric scores.[1] Since these metrics are based on simple phrase matching/n-gram overlap, they do not focus on important summarization factors such as salient phrase inclusion and directed logical entailment. Addressing these issues, we next introduce two new reward functions.

[1]For the rest of the paper, we mean ROUGE-L whenever we mention ROUGE-reward models.

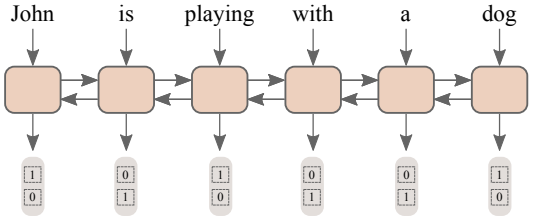

Figure 2: Overview of our saliency predictor model.

Saliency Rewards ROUGE-based rewards have no knowledge about what information is salient in the summary, and hence we introduce a novel reward function called 'ROUGESal' which gives higher weight to the important, salient words/phrases when calculating the ROUGE score (which by default assumes all words are equally weighted). To learn these saliency weights, we train our saliency predictor on sentence and answer spans pairs from the popular SQuAD reading comprehension dataset (Rajpurkar et al., 2016)) (Wikipedia domain), where we treat the human-annotated answer spans (avg. span length 3.2) for important questions as representative salient information in the document. As shown in Fig. 2, given a sentence as input, the predictor assigns a saliency probability to every token, using a simple bidirectional encoder with a *softmax* layer at every time step of the encoder hidden states to classify the token as salient or not. Finally, we use the probabilities given by this saliency prediction model as weights in the ROUGE matching formulation to achieve the final ROUGESal score (see appendix for details about our ROUGESal weighted precision, recall, and F-1 formulations).

Entailment Rewards A good summary should also be logically entailed by the given source document, i.e., contain no contradictory or unrelated information. Pasunuru and Bansal (2017) used entailment-corrected phrase-matching metrics (CIDEnt) to improve the task of video captioning; we instead directly use the entailment knowledge from an entailment scorer and its multi-sentence, length-normalized extension as our 'Entail' reward, to improve the task of abstractive text summarization. We train the entailment classifier (Parikh et al., 2016) on the SNLI (Bowman et al., 2015) and Multi-NLI (Williams et al., 2017) datasets and calculate the entailment probability score between the ground-truth (GT) summary (as premise) and each sentence of the generated summary (as hypothesis), and use avg. score as our

Entail reward.[2] Finally, we add a length normalization constraint to avoid very short sentences achieving misleadingly high entailment scores:

$$\text{Entail} = \text{Entail} \times \frac{\text{\#tokens in generated summary}}{\text{\#tokens in reference summary}} \quad (3)$$

5 Experimental Setup

5.1 Datasets and Training Details

CNN/Daily Mail dataset (Hermann et al., 2015; Nallapati et al., 2016) is a collection of online news articles and their summaries. We use the non-anonymous version of the dataset as described in See et al. (2017). For test-only generalization experiments, we use the DUC-2002 single document summarization dataset[3]. For entailment reward classifier, we use a combination of the full Stanford Natural Language Inference (SNLI) corpus (Bowman et al., 2015) and the recent Multi-NLI corpus (Williams et al., 2017) training datasets. For our saliency prediction model, we use the Stanford Question Answering (SQuAD) dataset (Rajpurkar et al., 2016). All dataset splits and other training details (dimension sizes, learning rates, etc.) for reproducibility are in appendix.

5.2 Evaluation Metrics

We use the standard ROUGE package (Lin, 2004) and Meteor package (Denkowski and Lavie, 2014) for reporting the results on all of our summarization models. Following previous work (Chopra et al., 2016; Nallapati et al., 2016; See et al., 2017), we use the ROUGE full-length F1 variant.

6 Results

Baseline Cross-entropy Model Our abstractive summarization model has attention, pointer-copy, and coverage mechanism. First, we apply cross-entropy optimization and achieve comparable results on CNN/Daily Mail w.r.t. previous work (See et al., 2017).[4]

Models	R-1	R-2	R-L	M
PREVIOUS WORK				
Nallapati (2016)*	35.46	13.30	32.65	-
See et al. (2017)	39.53	17.28	36.38	18.72
Paulus (2017) (XE)*	38.30	14.81	35.49	-
Paulus (2017) (RL)*	39.87	15.82	36.90	-
OUR MODELS				
Baseline (XE)	39.41	17.33	36.07	18.27
ROUGE (RL)	39.99	17.72	36.66	18.93
Entail (RL)	39.53	17.51	36.44	20.15
ROUGESal (RL)	40.36	17.97	37.00	19.84
ROUGE+Ent (RL)	40.37	17.89	37.13	19.94
ROUGESal+Ent (RL)	40.43	18.00	37.10	20.02

Table 1: Results on CNN/Daily Mail (non-anonymous). * represents previous work on anonymous version. 'XE': cross-entropy loss, 'RL': reinforce mixed loss (XE+RL). Columns 'R': ROUGE, 'M': METEOR.

ROUGE Rewards First, using ROUGE-L as RL reward (shown as ROUGE in Table 1) improves the performance on CNN/Daily Mail in all metrics with stat. significant scores ($p < 0.001$) as compared to the cross-entropy baseline (and also stat. signif. w.r.t. See et al. (2017)). Similar to Paulus et al. (2017), we use mixed loss function (XE+RL) for all our reinforcement experiments, to ensure good readability of generated summaries.

ROUGESal and Entail Rewards With our novel ROUGESal reward, we achieve stat. signif. improvements in all metrics w.r.t. the baseline as well as w.r.t. ROUGE-reward results ($p < 0.001$), showing that saliency knowledge is strongly improving the summarization model. For our Entail reward, we achieve stat. signif. improvements in ROUGE-L ($p < 0.001$) w.r.t. baseline and achieve the best METEOR score by a large margin. See Sec. 7 for analysis of the saliency/entailment skills learned by our models.

Multi-Reward Results Similar to ROUGESal, Entail is a better reward when combined with the complementary phrase-matching metric information in ROUGE; Table 1 shows that the ROUGE+Entail multi-reward combination performs stat. signif. better than ROUGE-reward in ROUGE-1, ROUGE-L, and METEOR ($p < 0.001$), and better than Entail-reward in all

[2] Since the GT summary is correctly entailed by the source document, we directly (by transitivity) use this GT as premise for easier (shorter) encoding. We also tried using the full input document as premise but this didn't perform as well (most likely because the entailment classifiers are not trained on such long premises; and the problem with the sentence-to-sentence avg. scoring approach is discussed below).
We also tried summary-to-summary entailment scoring (similar to ROUGE-L) as well as pairwise sentence-to-sentence avg. scoring, but we found that avg. scoring of ground-truth summary (as premise) w.r.t. each generated summary's sentence (as hypothesis) works better (intuitive because each sentence in generated summary might be a compression of multiple sentences of GT summary or source document).

[3] http://www-nlpir.nist.gov/projects/duc/guidelines/2002.html

[4] Our baseline is statistically equal to the paper-reported scores of See et al. (2017) (see Table 1) on ROUGE-1, ROUGE-2, based on the bootstrap test (Efron and Tibshirani, 1994). Our baseline is stat. significantly better ($p < 0.001$) in all ROUGE metrics w.r.t. the github scores (R-1: 38.82, R-2: 16.81, R-3: 35.71, M: 18.14) of See et al. (2017).

Models	R-1	R-2	R-L	M
Baseline (XE)	35.50	14.57	32.19	14.36
ROUGE (RL)	35.97	15.45	32.72	14.50
ROUGESal+Ent (RL)	38.95	17.05	35.52	16.47

Table 2: ROUGE F1 full length scores of our models on test-only DUC-2002 generalizability setup.

ROUGE metrics. Finally, we combined our two rewards ROUGESal+Entail to incorporate both saliency and entailment knowledge, and it gives the best results overall ($p < 0.001$ in all metrics w.r.t. both baseline and ROUGE-reward models), setting the new state-of-the-art.[5]

Test-Only Transfer (DUC-2002) Results Finally, we also tested our model's generalizability/transfer skills, where we take the models trained on CNN/Daily Mail and directly test them on DUC-2002 in a test-only setup. As shown in Table 2, our final ROUGESal+Entail multi-reward RL model is statistically significantly better than both the cross-entropy (pointer-generator + coverage) baseline as well as ROUGE reward RL model, in terms of all 4 metrics with a large margin (with $p < 0.001$). This demonstrates that our ROUGESal+Entail model learned better transferable and generalizable skills of saliency and logical entailment.

7 Output Analysis

Saliency Analysis We analyzed the output summaries generated by See et al. (2017), and our baseline, ROUGE-reward and ROUGESal-reward models, using our saliency prediction model (Sec. 4), and the scores are 27.95%, 28.00%, 28.80%, and 30.86%. We also used the original CNN/Daily Mail Cloze Q&A setup (Hermann et al., 2015) with the fill-in-the-blank answers treated as salient information, and the results are 60.66%, 59.36%, 60.67%, and 64.66% for the four models. Both these experiments illustrate that our ROUGESal reward model is stat. signif. better in saliency than the See et al. (2017), our baseline, and ROUGE-reward models ($p < 0.001$).

Entailment Analysis We also analyzed the entailment scores of the generated summaries from See et al. (2017), and our baseline, ROUGE-reward, and Entail-reward models, and the results are 27.33%, 27.21%, 28.23%, and 28.98%.[6]

[5]Our last three rows in Table 1 are all stat. signif. better in all metrics with $p < 0.001$ compared to See et al. (2017).

[6]Based on our ground-truth summary to output summary sentences' average entailment score (see Sec. 4); similar trends hold for document-to-summary entailment scores.

Models	2-gram	3-gram	4-gram
See et al. (2017)	2.24	6.03	9.72
Baseline (XE)	2.23	5.58	8.81
ROUGE (RL)	2.69	6.57	10.23
ROUGESal (RL)	2.37	6.00	9.50
Entail (RL)	2.63	6.56	10.26

Table 3: Abstractiveness: novel n-gram percentage.

We observe that our Entail-reward model achieves stat. significant entailment scores ($p < 0.001$) w.r.t. all the other three models.

Abstractiveness Analysis In order to measure the abstractiveness of our models, we followed the 'novel n-gram overlap' approach suggested in See et al. (2017). First, we found that all our reward-based RL models have significantly ($p < 0.01$) more novel n-grams than our cross-entropy baseline (see Table 3). Next, the Entail-reward model 'maintains' stat. equal abstractiveness as the ROUGE-reward model, likely because it encourages rewriting to create logical subsets of information, while the ROUGESal-reward model does a bit worse, probably because it focuses on copying more salient information (e.g., names). Compared to previous work (See et al., 2017), our Entail-reward and ROUGE-reward models achieve statistically significant improvement ($p < 0.01$) while ROUGESal is comparable.

8 Conclusion

We presented a summarization model trained with novel RL reward functions to improve the saliency and directed logical entailment aspects of a good summary. Further, we introduced the novel and effective multi-reward approach of optimizing multiple rewards simultaneously in alternate mini-batches. We achieve the new state-of-the-art on CNN/Daily Mail and also strong test-only improvements on a DUC-2002 transfer setup.

Acknowledgments

We thank the reviewers for their helpful comments. This work was supported by DARPA (YFA17-D17AP00022), Google Faculty Research Award, Bloomberg Data Science Research Grant, and NVidia GPU awards. The views, opinions, and/or findings contained in this article are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied, of the funding agency.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, and Hui Jiang. 2016. Distraction-based neural networks for modeling documents. In *IJCAI*.

Jackie Chi Kit Cheung and Gerald Penn. 2014. Unsupervised sentence enhancement for automatic summarization. In *EMNLP*. pages 775–786.

Sumit Chopra, Michael Auli, and Alexander M Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *HLT-NAACL*.

James Clarke and Mirella Lapata. 2008. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research* 31:399–429.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*, Springer, pages 177–190.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *EACL*.

Shibhansh Dohare and Harish Karnick. 2017. Text summarization using abstract meaning representation. *arXiv preprint arXiv:1706.01678* .

Bradley Efron and Robert J Tibshirani. 1994. *An introduction to the bootstrap*. CRC press.

Katja Filippova, Enrique Alfonseca, Carlos A Colmenares, Lukasz Kaiser, and Oriol Vinyals. 2015. Sentence compression by deletion with lstms. In *EMNLP*. pages 360–368.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd international conference on computational linguistics*. ACL, pages 340–348.

Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond T Ng, and Bita Nejat. 2014. Abstractive summarization of product reviews using discourse structure. In *EMNLP*. volume 14, pages 1602–1613.

George Giannakopoulos. 2009. Automatic summarization from multiple documents. *Ph. D. dissertation* .

Anand Gupta, Manpreet Kaur, Adarsh Singh, Aseem Goel, and Shachar Mirkin. 2014. Text summarization through entailment-based minimum vertex cover. *Lexical and Computational Semantics (* SEM 2014)* page 75.

Sanda Harabagiu and Andrew Hickl. 2006. Methods for using textual entailment in open-domain question answering. In *ACL*. pages 905–912.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NIPS*. pages 1693–1701.

Masaru Isonuma, Toru Fujino, Junichiro Mori, Yutaka Matsuo, and Ichiro Sakata. 2017. Extractive summarization using multi-task learning with document classification. In *EMNLP*. pages 2091–2100.

Sergio Jimenez, George Duenas, Julia Baquero, Alexander Gelbukh, Av Juan Dios Bátiz, and Av Mendizábal. 2014. UNAL-NLP: Combining soft cardinality features for semantic textual similarity, relatedness and entailment. In *In SemEval*. pages 732–742.

Hongyan Jing. 2000. Sentence reduction for automatic text summarization. In *ANLP*.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Kevin Knight and Daniel Marcu. 2002. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence* 139(1):91–107.

Alice Lai and Julia Hockenmaier. 2014. Illinois-LH: A denotational and distributional approach to semantics. *Proc. SemEval* 2:5.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 workshop*. volume 8.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A Smith. 2015. Toward abstractive summarization using semantic representations. In *NAACL: HLT*. pages 1077–1086.

Yashar Mehdad, Giuseppe Carenini, Frank W Tompa, and Raymond T Ng. 2013. Abstractive meeting summarization with entailment and fusion. In *Proc. of the 14th European Workshop on Natural Language Generation*. pages 136–146.

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

Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *EMNLP*.

Ramakanth Pasunuru and Mohit Bansal. 2017. Reinforced video captioning with entailment rewards. In *EMNLP*.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304* .

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. In *ICLR*.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. 2016. Self-critical sequence training for image captioning. *arXiv preprint arXiv:1612.00563* .

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *CoRR*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *ACL*.

Sandeep Subramanian, Tong Wang, Xingdi Yuan, and Adam Trischler. 2017. Neural models for key phrase detection and question generation. *arXiv preprint arXiv:1706.04560* .

Jun Suzuki and Masaaki Nagata. 2016. Rnn-based encoder-decoder approach with word frequency estimation. In *EACL*.

Lu Wang, Hema Raghavan, Vittorio Castelli, Radu Florian, and Claire Cardie. 2016. A sentence compression based framework to query-focused multi-document summarization. *arXiv preprint arXiv:1606.07548* .

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426* .

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8(3-4):229–256.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* .

Wojciech Zaremba and Ilya Sutskever. 2015. Reinforcement learning neural turing machines. *arXiv preprint arXiv:1505.00521* 362.

Yongzheng Zhang, Nur Zincir-Heywood, and Evangelos Milios. 2004. World wide web site summarization. *Web Intelligence and Agent Systems: An International Journal* 2(1):39–53.

A Supplementary Material

A.1 Saliency Rewards

Here, we describe the ROUGE-L formulation at summary-level and later describe how we incorporate saliency information into it. Given a reference summary of u sentences containing a total of m tokens ($\{w_{r,k}\}_{k=1}^{m}$) and a generated summary of v sentences with a total of n tokens ($\{w_{c,k}\}_{k=1}^{n}$), let r_i be the reference summary sentence and c_j be the generated summary sentence. Then, the precision (P_{lcs}), recall (R_{lcs}), and F-score (F_{lcs}) for ROUGE-L are defined as follows:

$$P_{lcs} = \frac{\sum_{i=1}^{u} LCS_{\cup}(r_i, C)}{n} \tag{4}$$

$$R_{lcs} = \frac{\sum_{i=1}^{u} LCS_{\cup}(r_i, C)}{m} \tag{5}$$

$$F_{lcs} = \frac{(1+\beta^2) R_{lcs} P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}} \tag{6}$$

where $LCS_{\cup}$ takes the *union* Longest Common Subsequence (LCS) between a reference summary sentence r_i and every generated summary sentence c_j ($c_j \in C$), and β is defined in Lin (2004). In the above ROUGE-L scores, we assume that every token has equal weight, i.e, 1. However, every summary has salient tokens which should be rewarded with more weight. Hence, we use the weights obtained from our novel saliency predictor to modify the ROUGE-L scores with salient information as follows:

$$P^s_{lcs} = \frac{\sum_{i=1}^{u} LCS^*_{\cup}(r_i, C)}{\sum_{k=1}^{n} \eta(w_{c,k})} \tag{7}$$

$$R^s_{lcs} = \frac{\sum_{i=1}^{u} LCS^*_{\cup}(r_i, C)}{\sum_{k=1}^{m} \eta(w_{r,k})} \tag{8}$$

$$F^s_{lcs} = \frac{(1+\beta^2) R^s_{lcs} P^s_{lcs}}{R^s_{lcs} + \beta^2 P^s_{lcs}} \tag{9}$$

where $\eta(w)$ is the weight assigned by the saliency predictor for token w, and β is defined in Lin (2004).[7] Let $\{w_k\}_{k=1}^{p}$ be the union LCS set, then $LCS^*_{\cup}(r_i, C)$ is defined as follows:

$$LCS^*_{\cup}(r_i, C) = \sum_{k=1}^{p} \eta(w_k) \tag{10}$$

[7] If a token is repeated at multiple times in the input sentence, we average the probabilities of those instances.

A.2 Experimental Setup

A.2.1 Datasets

CNN/Daily Mail Dataset CNN/Daily Mail dataset (Hermann et al., 2015; Nallapati et al., 2016) is a collection of online articles and their summaries. The summaries are based on the human written highlights of these articles. The dataset has $287,226$ training pairs, $13,368$ validation pairs, and $11,490$ test pairs. We use the non-anonymous version of the dataset as described in See et al. (2017).

DUC Test Corpus We use the DUC-2002 single document summarization dataset[8] as a test-only setup where we directly take the pretrained models trained on CNN/Daily Mail dataset and test them on DUC-2002, in order to check for our model's domain transfer capabilities. This corpus consists of 567 documents with one or two human annotated reference summaries.

SNLI and MultiNLI corpus We use the full Stanford Natural Language Inference (SNLI) corpus (Bowman et al., 2015) and the recent MultiNLI corpus (Williams et al., 2017) data for building our entailment classifier. We use the standard splits following previous work.

SQuAD Dataset We use Stanford Question Answering Dataset (SQuAD) for our saliency prediction model. We process the SQuAD dataset to collect the sentence and their corresponding salient phrases pairs. Here again, we use the standard split following previous work.

A.2.2 Training Details

During training, all our LSTM-RNNs are set with hidden state size of 256. We use a vocabulary size of 50k, where word embeddings are represented in 128 dimension, and both the encoder and decoder share the same embedding for each word. We encode the source document using a 400 time-step unrolled LSTM-RNN and 100 time-step unrolled LSTM-RNN for decoder. We clip the gradients to a maximum gradient norm value of 2.0 and use Adam optimizer (Kingma and Ba, 2015) with a learning rate of 1×10^{-3} for pointer baseline and 1×10^{-4} while training along with coverage loss, and 1×10^{-6} for reinforcement learning. Following See et al. (2017), we add coverage mechanism to a converged pointer model. For mixed-loss (XE+RL) optimization, we use the following γ values for various rewards: 0.9985 for ROUGE, 0.9999 for Entail and ROUGE+Entail, and 0.9995 for ROUGESal and ROUGESal+Entail. For reinforcement learning, we only use 5000 training samples ($< 2\%$ of the actual data) to speed up convergence, but we found it to work well in practice. During inference time, we use a beam search of size 4.

Models	Accuracy
Entailment Classifier	74.50%
Saliency Predictor	16.87%

Table 4: Performance of our entailment classifier and saliency predictor.

A.3 Results

A.3.1 Saliency and Entailment Scorer

Table 4 presents the performance of our saliency predictor (on the SQuAD-based dev set for answer span classification accuracy) and entailment classifier (on the Multi-NLI dev set accuracy). Our entailment classifier is comparable to the state-of-the-art models.[9]

[8] http://www-nlpir.nist.gov/projects/duc/guidelines/2002.html

[9] RepEval leaderboard: https://repeval2017.github.io/shared/