Supplementary: Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting

Yen-Chun Chen and **Mohit Bansal**
UNC Chapel Hill
{yenchun, mbansal}@cs.unc.edu

A Model Details

A.1 Convolutional Encoder

Here we describe the convolutional sentence representation used in Sec. 2.1.1. We use the temporal convolutional model proposed by Kim (2014) to compute the representation of every individual sentence in the document. First, the words are converted to the distributed vector representation by a learned word embedding matrix W_{emb}. The sequence of the word vectors from each sentence is then fed through 1-D single-layer convolution filters with various window sizes (3, 4, 5) to capture the temporal dependencies of nearby words and then followed by $relu$ non-linear activation and max-over-time pooling. The convolutional representation r_j for the jth sentence is then obtained by concatenating the outputs from the activations of all filter window sizes.

A.2 Abstractor

In this section we discuss the architecture choices for our abstractor network in Sec. 2.2. At a high-level, it is a sequence-to-sequence model with attention and copy mechanism (but no coverage). Note that the abstractor network is a separate neural network from the extractor agent without any form of parameter sharing.

Sequence-Attention-Sequence Model We use a standard encoder-aligner-decoder model (Bahdanau et al., 2015; Luong et al., 2015) with the bilinear multiplicative attention function (Luong et al., 2015), $f_{att}(h_i, z_j) = h_i^\top W_{attn} z_j$, for the context vector e_j. We share the source and target embedding matrix W_{emb} as well as output projection matrix as in Inan et al. (2017); Press and Wolf (2017); Paulus et al. (2018).

Copy Mechanism We add the copying mechanism as in See et al. (2017) to extend the decoder to predict over the extended vocabulary of words in the input document. A copy probability $p_{copy} = \sigma(v_{\hat{z}}^\top \hat{z}_j + v_s^\top z_j + v_w^\top w_j + b)$ is calculated by learnable parameters v's and b, and then is used to further compute a weighted sum of the probability of source vocabulary and the predefined vocabulary. At test time, an OOV prediction is replaced by the document word with the highest attention score.

A.3 Actor-Critic Policy Gradient

Here we discuss the details of the actor-critic policy gradient training. Given the MDP formulation described in Sec. 3.2, the return (total discounted future reward) is

$$R_t = \sum_{t=1}^{N_s} \gamma^t r(t+1) \tag{1}$$

for each recurrent step t. To learn a optimal policy π^* that maximize the state-value function:

$$V^{\pi^*}(c) = \mathbb{E}_{\pi^*}[R_t | c_t = c]$$

we will make use of the action-value function

$$Q^{\pi_\theta}(c, j) = \mathbb{E}_{\pi_\theta}[R_t | c_t = c, j_t = j]$$

We then take the policy gradient theorem and then substitute the action-value function with the Monte-Carlo sample:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(c, j) Q^{\pi_\theta}(c, j)] \tag{2}$$

$$= \frac{1}{N_s} \sum_{t=1}^{N_s} \nabla_\theta \log \pi_\theta(c_t, j_t) R_t \tag{3}$$

which runs a single episode and gets the return (estimate of action-value function) by sampling from the policy π_θ, where N_s is the total number of sentences the agent extracts. This gradient update is also known as the REINFORCE algorithm (Williams, 1992).

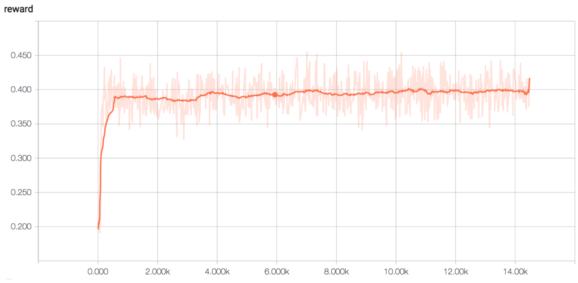

Figure 1: RL learning curve.

The vanilla REINFORCE algorithm is known for high variance. To mitigate this problem we add a critic network with trainable parameters θ_c having the same structure as the pointer-network's decoder (described in Sec. 2.1.2) but change the final output layer to regress the state-value function $V^{\pi_{\theta_a,\omega}}(c)$. The predicted value $b_{\theta_c,\omega}(c)$ is called the baseline and is subtracted from the action-value function to estimate the *advantage*

$$A^{\pi_\theta}(c,j) = Q^{\pi_{\theta_a,\omega}}(c,j) - b_{\theta_c,\omega}(c)$$

where $\theta = \{\theta_a, \theta_c, \omega\}$ denotes the set of all trainable parameters. The new policy gradient for our extractor can be estimated by substituting the action-value function in Eqn. 2 by the advantage and then use Monte-Carlo samples (use R_t to estimate Q):[1]

$$\nabla_{\theta_a,\omega} J(\theta_a,\omega) \approx \frac{1}{N_s}\sum_{t=1}^{N_s}[\nabla_{\theta_a,\omega}\log\pi_\theta(c,j)A^{\pi_\theta}(c,j)] \quad (4)$$

Here we also show an interesting finding of the effect adding the EOE action. In Fig. 1, we can see that the average reward is low in the beginning but quickly goes up after the agent picks up the EOE action. The low beginning reward is because the agent does not choose the EOE action hence keep getting zero rewards when extracting extra sentences, which lowers the average.

A.4 Sentence Selection Baseline ff-ext

In this subsection, we describe the detailed network structure of the feed-forward extractor baseline (ff-ext). Following the hierarchical sentence representation described in Sec. 2.1.1, if we add another assumption that there exists a sequence $j_{i1}, j_{i2}, \ldots, j_{iN_s}$ where $j_{i1} < j_{i2} < \cdots < j_{iN_s}$ such that

$$[d_{i1}, d_{i2}, \cdots, d_{iN_d}] = x_i \quad \text{and}$$
$$[g(d_{j_{i1}}), g(d_{j_{i2}}), \cdots, g(d_{j_{iN_s}})] = y_i \quad (5)$$

i.e., the extracted document are summarized in the order as is, we could apply the following feed-forward structure for sentence selection. We first learn a document representation by

$$\hat{x} = \tanh(W_d \frac{1}{N_d}\sum_{j=1}^{N_d} h_j + b_d) \quad (6)$$

where N_d, N_s each denotes the number of sentences in the document x and the summary y respectively. And then we compute the extraction probability:

$$P(d_j = 1|h_j, \hat{x}) = \sigma(W_c h_j + h_j^\top W_s \hat{x} + b)$$

for each sentence in the document. Assuming we have the groundtruth extraction labels $j_1, \ldots, j_{N_s}$, the above formulation treats sentence selection as a sequence of binary classification problems, where Ws and bs are trainable parameters. We can therefore train the sentence selection network end-to-end by cross-entropy loss, where Ws and bs are trainable parameters.

At test time, the feed-forward extractor chooses the top-k sentences and then concatenates them as the original order in the document. Note that we still refer to this network as feed-forward extractor (ff-ext) to distinguish from the pointer network extractor (rnn-ext) though it contains recurrent structure.

B Training Details

B.1 Dataset Details

We use the CNN/Daily Mail dataset first proposed by Hermann et al. (2015) for reading comprehension task. This dataset has been modified for summarization by Nallapati et al. (2017). This dataset differs from previous Gigaword dataset (Rush et al., 2015) in the length of the text: both documents and summaries for CNN/Daily Mail is much longer. The standard split of the dataset contains 287,227 documents for training, 13,368 documents for validation, and 11,490 for testing. Note that the original release of this dataset by Hermann et al. (2015) is an anonymized version, where the named entities are anonymized and treated as a

[1] We found that updating with mini-batch of episodes and standardizing R_t over all time steps and all episodes within the batch helps converging.

single word in the evaluation n-gram matching. On the other hand, See et al. (2017) proposed to use the non-anonymized, original-text version of the dataset. For a fair comparison to prior works, we show results on both versions of the dataset. The experiment runs training and evaluation for each version separately (but we transfer the same tuned hyperparameters from original to anonymized version).

The DUC-2002 dataset contains 567 document-summary pairs for single-document summarization. Due to its small size, we utilize it in a test-only setup: we directly use the CNN/Daily Mail (original text) trained model to summarize the DUC documents for testing generalization/transfer our models. The results of See et al. (2017) on DUC is obtained by running their publicly available pretrained model. We evaluate the results using the official ROUGE F1 script.

B.2 Hyperparameter Details

All hyper-parameters are tuned on the validation set of the original text version of CNN/DM. We use mini-batches of 32 samples for all the training. Adam optimizer (Kingma and Ba, 2014) is used with learning rate 0.001 for ML and 0.0001 for RL training (other hyper-parameters at their default). We apply gradient clipping (Pascanu et al., 2013) using 2-norm of 2.0. We do not use any regularization technique except early-stopping. We also found that halving the learning rate whenever validation loss stops decreasing speeds up convergence. For RL training, we use $\gamma = 0.95$ for the discount factor in Eqn. 1. We first train the abstractor and extractors separately until convergence with maximum-likelihood objectives, then apply RL training on the trained sub-modules. For all LSTM-RNNs we use 256 hidden units. We use single layer LSTM-RNN with 256 hidden units for all models. The initial states of RNN are learned for our extractor agent. For the abstractor network, we learn a linear mapping to transform the encoder final states to the decoder initial states. We also train a word2vec (Mikolov et al., 2013) of 128 dimension on the same corpus to initialize the embedding matrix for all maximum-likelihood trained models and the embedding matrix is updated during training. We set a vocabulary size of 30000 most common words in the training set. For saving the memory space in training, we truncate the input article sentences to a maximum length of 100 tokens and summary sentences to 30 tokens (note that this is counted at the sentence-level for our abstractor training). We use all possible sentence pairs within every summary without limit. At test time, the length of input is not limited and the generation limit remains 30 maximum tokens for the abstractor. For all non-RL models, the number of sentences to extract is tuned on the validation set. For the reranking (see Sec 3.3), we set $N = 2$ (bi-gram) and $k = 5$ (beam size).[2] The diversity ratio of the diverse beam-search (Li et al., 2016) is set to 1.0.

B.3 Training Speed

It took a total of 19.71 hours[3] to train our model. On the other hand, See et al. (2017) reported more than 78 hours of training. The training speed gain is mainly from the shortened input/target pairs of our abstractor model. Since our encoder-decoder-aligner structure operates on sentence pair, it trains much faster the the document-summary pair used in the pointer-generator model (See et al., 2017). We also report here the speed of training our abstractor as time per training update.[4] Our abstractor only requires 0.54 seconds per updates while See et al. (2017) needs 3.42. For all our speed experiments we use K40 GPUs (similar to See et al. (2017). The reduced sequence length gives us an advantage of 6x. Also, the model proposed by See et al. (2017) needs careful scheduling of the sentence lengths.

C Generation Samples

Please see Fig. 2 and Fig. 3 for the output examples (see the discussion of this example in Sec 7.2).

[2] Due to the fact that the size of the reranking list is exponential to the number of sentences of the generated summary n, we pruned the beam so as to allow completion (of dev-set summarization) in a reasonable amount of time, as following: for $n \leq 5$, we use our standard beam size of $k = 5$, but for larger n values, we use gradually-reduced k values: $(6, 4), (7-8, 3), (9+, 2)$ for (n, k).

[3] 4.15 hours for the abstractor, 15.56 hours for the RL training. Extractor ML training can be run at the same time with abstractor training and is approximately 1.5 hours.

[4] We use their publicly available code and run training (without coverage mechanism) on our machine for a fair comparison. The number of vocabulary, embedding dimension, RNN hidden units are also set to the same as our model. We set their maximum encoder and decoder steps to 400 and 100 respectively, as reported in their paper.

Source document
*[the oxford university women 's boat race team were rescued from the thames by the royal national lifeboat institution (rnli) on wednesday after being overcome by choppy waters .] §[crew members from the chiswick rnli station came to the assistance of the oxford crew and their cox , who were training for the boat race which - along with the men 's race - takes place on saturday , april 11 .] †[after the rowers were returned safely to putney , the sunken eight was recovered and returned to oxford 's base .] ‡[the royal national lifeboat institution come to the assistance of the oxford university women 's team .] the oxford crew were training on the thames for the boat race which takes place on saturday , april 11 . the rnli revealed the conditions were caused by strong wind against the tide creating three successive waves that poured over the boat 's riggers , ' creating an influx of water that could not be managed by the craft 's bilge pump ' . in a statement rnli helmsman ian owen said : ' while we have rescued quite a number of rowers over the years , this is the first time i 've been involved in helping such a prestigious team . ' the weather can be unpredictable on the thames , and the oxford university team dealt with the situation as safely and calmly as possible . we wish them all the best for their upcoming race . ' chiswick and tower stations are the busiest in the country , and the rnli has saved over 3,600 people since the service began in 2002 . the rnli alternative boat race fundraising event on april 10 takes place the day before the bny mellon boat race on the same famous stretch of river . for more information , please visit : rnli.org / boatrace .
Ground truth summary
the crew were training for the boat race which takes place on april 11 . the sunken eight was recovered and returned to oxford 's base . the choppy conditions were caused by strong wind against the tide creating three successive waves that poured over the boat 's riggers .
rnn-ext + abs + RL (ROUGE-1: 48.54, ROUGE-2: 27.72 ROUGE-L: 48.54)
*the oxford university women 's boat race team were rescued from the thames by the royal national lifeboat institution . §crew members were training for the boat race which takes place on saturday . †the rowers were returned to oxford 's base . ‡the royal national lifeboat institution come to the assistance of the oxford university women 's team .
+rerank (ROUGE-1: 60.42, ROUGE-2: 42.55, ROUGE-L: 60.42)
*the oxford university women 's boat race team were rescued from the thames . §crew members were training for the boat race which takes place on saturday . †the sunken eight was recovered and returned to oxford 's base . ‡the royal national lifeboat institution come to the assistance of the team .

Figure 2: Example from the dataset showing the generated summary of our best models. The colored (marked) sentences correspond to our extractor's sentence selection. The listed ROUGE scores are computed for this specific example.

Source document
(cnn) have mercy ! lifetime has its follow-up to its “ unauthorized saved by the bell ” tv movie : the network is now taking on full house . *[the female-skewing cable network has greenlit “ the unauthorized full house story ” (working title) , the hollywood reporter has learned .] §[in the same vein as its “ saved by the bell ” pic , lifetime ’s full house story will look at the rise of the cast – including john stamos , bob saget and the mary-kate and ashley olsen – and explore the pressure they faced to balance idyllic family life on the show with the more complicated reality of their own lives outside the series . additionally , it will look at the warm bond that grew between the cast as the show became one of america ’s most beloved family sitcoms .] †[casting will begin immediately . an air date for the “ full house ” tell-all has yet to be determined .] see more broadcast tv ’s returning shows 2015-16 . ‡[ron mcgee , who penned the “ unauthorized saved by the bell story , ” will write the “ full house ” take . the telepic will be produced by the bell team of front street pictures and ringaling productions , with harvey kahn and stephen bulka also on board to exec produce .] for lifetime , the news comes after its two-hour bell take fizzled on labor day 2014 . despite tons of build-up and excitement from diehard fans of the original comedy series , the bell take drew only 1.6 million total viewers , with 1.1 million viewers among the 18-49 and 25-54 demographics . that pic was based on former star dustin diamond ’s behind the bell 2009 tell-all , with dylan everett starring as mark-paul gosselaar and sam kindseth as diamond . full house aired on abc from 1987 to 1995 . netflix this month revived the beloved family comedy as “ fuller house , ” with original stars candace cameron-bure (d.j.) , her on-screen sister , jodie sweetin (stephanie) , and best friend andrea barber (kimmy) , in a 13-episode follow-up series . from its start as an unassuming family comedy in 1987 to its eventual wildly popular 192-episode run , “ full house ” was “ the little sitcom that could . ” it made huge stars of its cast – from bob saget and dave coulier , who were grinding away on the standup circuit , to john stamos breaking hearts on general hospital , and the olsen twins . see the original story at the hollywood reporter ’s website . 2015 the hollywood reporter . all rights reserved .
Ground truth summary
the network has reportedly greenlit the tell-all . lifetime previously did an unauthorized movie on “ saved by the bell ”
rnn-ext + abs + RL (ROUGE-1: 25.00, ROUGE-2: 7.41 ROUGE-L: 25.00)
*the female-skewing cable network has greenlit “ the unauthorized full house story ” §the cast will look at the warm bond that grew between the cast . †ron mcgee will write the “ full house ” take . ‡casting will begin immediately .
+rerank (ROUGE-1: 37.93, ROUGE-2: 17.86, ROUGE-L: 37.93)
*the female-skewing cable network has greenlit “ the unauthorized full house story ” §lifetime ’s full house story will look at the rise of the cast . †ron mcgee penned the “ unauthorized saved by the bell story ” ‡casting will begin immediately .

Figure 3: Example from the dataset showing the generated summary of our best models. The colored (marked) sentences correspond to our extractor’s sentence selection. The listed ROUGE scores are computed for this specific example.

References

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

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems (NIPS)*.

Hakan Inan, Khashayar Khosravi, and Richard Socher. 2017. Tying word vectors and word classifiers: A loss framework for language modeling. In *ICLR*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *ICLR*.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. A simple, fast diverse decoding algorithm for neural generation. *arXiv preprint*, abs/1611.08562.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *AAAI Conference on Artificial Intelligence*.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1310–1318, Atlanta, Georgia, USA. PMLR.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *ICLR*.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163. Association for Computational Linguistics.

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.

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. Association for Computational Linguistics.

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