Abstractive Summarization of *Reddit* Posts with Multi-level Memory Networks

Byeongchang Kim **Hyunwoo Kim** **Gunhee Kim**
Department of Computer Science and Engineering & Center for Superintelligence
Seoul National University, Seoul, Korea
{byeongchang.kim,hyunwoo.kim}@vision.snu.ac.kr gunhee@snu.ac.kr

Abstract

We address the problem of abstractive summarization in two directions: proposing a novel dataset and a new model. First, we collect *Reddit TIFU* dataset, consisting of 120K posts from the online discussion forum Reddit. We use such informal crowd-generated posts as text source, in contrast with existing datasets that mostly use formal documents as source such as news articles. Thus, our dataset could less suffer from some biases that key sentences usually locate at the beginning of the text and favorable summary candidates are already inside the text in similar forms. Second, we propose a novel abstractive summarization model named *multi-level memory networks* (MMN), equipped with multi-level memory to store the information of text from different levels of abstraction. With quantitative evaluation and user studies via Amazon Mechanical Turk, we show the *Reddit TIFU* dataset is highly abstractive and the MMN outperforms the state-of-the-art summarization models. The code and dataset are available at `http://vision.snu.ac.kr/projects/reddit-tifu`.

1 Introduction

Abstractive summarization methods have been under intensive study, yet they often suffer from inferior performance compared to extractive methods (Allahyari et al., 2017; Nallapati et al., 2017; See et al., 2017). Admittedly, by task definition, abstractive summarization is more challenging than extractive summarization. However, we argue that such inferior performance is partly due to some biases of existing summarization datasets. The source text of most datasets (Over et al., 2007; Hermann et al., 2015; Cohan et al., 2018; Grusky et al., 2018; Narayan et al., 2018a) originates from formal documents such as news articles, which have some structural patterns of which extractive methods better take advantage.

In formal documents, there could be a strong tendency that key sentences locate at the beginning of the text and favorable summary candidates are already inside the text in similar forms. Hence, summarization methods could generate good summaries by simply memorizing keywords or phrases from particular locations of the text. Moreover, if abstractive methods are trained on these datasets, they may not show much abstraction (See et al., 2017), because they are implicitly forced to learn structural patterns (Kedzie et al., 2018). Grusky et al. (2018) and Narayan et al. (2018a) recently report similar extractive bias in existing datasets. They alleviate this bias by collecting articles from diverse news publications or regarding intro sentences as gold summary.

Different from previous approaches, we propose to alleviate such bias issue by changing the source of summarization dataset. We exploit user-generated posts from the online discussion forum `Reddit`, especially `TIFU` subreddit, which are more casual and conversational than news articles. We observe that the source text in `Reddit` does not follow strict formatting and disallows models to simply rely on locational biases for summarization. Moreover, the passages rarely contain sentences that are nearly identical to the gold summary. Our new large-scale dataset for abstractive summarization named as *Reddit TIFU* contains 122,933 pairs of an online post as source text and its corresponding long or short summary sentence. These posts are written by many different users, but each pair of post and summary is created by the same user.

Another key contribution of this work is to propose a novel memory network model named *multi-level memory networks* (MMN). Our model is equipped with multi-level memory networks, storing the information of source text from different

Proceedings of NAACL-HLT 2019, pages 2519–2531
Minneapolis, Minnesota, June 2 - June 7, 2019.

levels of abstraction (*i.e.* word-level, sentence-level, paragraph-level and document-level). This design is motivated by that abstractive summarization is highly challenging and requires not only to understand the whole document, but also to find salient words, phrases and sentences. Our model can sequentially read such multiple levels of information to generate a good summary sentence.

Most abstractive summarization methods (See et al., 2017; Li et al., 2017; Zhou et al., 2017; Liu et al., 2018; Cohan et al., 2018; Paulus et al., 2018) employ sequence-to-sequence (seq2seq) models (Sutskever et al., 2014) where an RNN encoder embeds an input document and another RNN decodes a summary sentence. Our MMN has two major advantages over seq2seq-based models. First, RNNs accumulate information in a few fixed-length memories at every step regardless of the length of an input sequence, and thus may fail to utilize far-distant information due to vanishing gradient. It is more critical in summarization tasks, since input text is usually very long (>300 words). On the other hand, our convolutional memory explicitly captures long-term information. Second, RNNs cannot build representations of different ranges, since hidden states are sequentially connected over the whole sequence. This still holds even with hierarchical RNNs that can learn multiple levels of representation. In contrast, our model exploits a set of convolution operations with different receptive fields; hence, it can build representations of not only multiple levels but also multiple ranges (*e.g.* sentences, paragraphs, and the whole document).

Our experimental results show that the proposed MMN model improves abstractive summarization performance on both our new Reddit TIFU and existing Newsroom-Abs (Grusky et al., 2018) and XSum (Narayan et al., 2018a) datasets. It outperforms several state-of-the-art abstractive models with seq2seq architecture such as (See et al., 2017; Zhou et al., 2017; Li et al., 2017). We evaluate with quantitative language metrics (*e.g.* perplexity and ROUGE (Lin, 2004)) and user studies via Amazon Mechanical Turk (AMT).

The contributions of this work are as follows.

1. We newly collect a large-scale abstractive summarization dataset named *Reddit TIFU*. As far as we know, our work is the first to use non-formal text for abstractive summarization.

2. We propose a novel model named *multi-level memory networks* (MMN). To the best of our knowledge, our model is the first attempt to leverage memory networks for the abstractive summarization. We discuss the unique updates of the MMN over existing memory networks in Section 2.

3. With quantitative evaluation and user studies via AMT, we show that our model outperforms state-of-the-art abstractive summarization methods on both Reddit TIFU, Newsroom abstractive subset and XSum dataset.

2 Related Work

Our work can be uniquely positioned in the context of the following three topics.

Neural Abstractive Summarization. Many deep neural network models have been proposed for abstractive summarization. One of the most dominant architectures is to employ RNN-based seq2seq models with attention mechanism such as (Rush et al., 2015; Chopra et al., 2016; Nallapati et al., 2016; Cohan et al., 2018; Hsu et al., 2018; Gehrmann et al., 2018).In addition, recent advances in deep network research have been promptly adopted for improving abstractive summarization. Some notable examples include the use of variational autoencoders (VAEs) (Miao and Blunsom, 2016; Li et al., 2017), graph-based attention (Tan et al., 2017), pointer-generator models (See et al., 2017), self-attention networks (Liu et al., 2018), reinforcement learning (Paulus et al., 2018; Pasunuru and Bansal, 2018), contextual agent attention (Celikyilmaz et al., 2018) and integration with extractive models (Hsu et al., 2018; Gehrmann et al., 2018).

Compared to existing neural methods of abstractive summarization, our approach is novel to replace an RNN-based encoder with explicit multi-level convolutional memory. While RNN-based encoders always consider the whole sequence to represent each hidden state, our multi-level memory network exploits convolutions to control the extent of representation in multiple levels of sentences, paragraphs, and the whole text.

Summarization Datasets. Most existing summarization datasets use formal documents as source text. News articles are exploited the most, including in DUC (Over et al., 2007), Gigaword (Napoles et al., 2012), CNN/DailyMail (Nallapati et al., 2016; Hermann et al., 2015), News-

[Short Summary] (16 words)

TIFU by forgetting my chemistry textbook and all of my notes in a city five hours away

[Long Summary] (29 words)

TL;DR I forgot my chemistry textbook and binder full of notes in Windsor, which is five hour drive away and I am now screwed for the rest of the semester.

[Source Text] (282 words)

(…) So the past three days I was at a sporting event in Windsor. I live pretty far from Windsor, around a 5 hour drive. (…)
A five hour drive later, I finally got back home. I was ready to start catching up on some homework when I realized I left my binder (which has all of my assignments, homework etc.) in it, and my chemistry textbook back in Windsor.
I also have a math and chem test next week which I am now so completely screwed for. (…)

Figure 1: An example post of the TIFU subreddit.

room (Grusky et al., 2018) and XSum (Narayan et al., 2018a) datasets. Cohan et al. (2018) introduce datasets of academic papers from arXiv and PubMed. Hu et al. (2015) propose the LCSTS dataset as a collection of Chinese microblog's short text each paired with a summary. However, it selects only formal text posted by verified organizations such as news agencies or government institutions. Compared to previous summarization datasets, our dataset is novel in that it consists of posts from the online forum Reddit.

Rotten Tomatoes and Idebate dataset (Wang and Ling, 2016) use online text as source, but they are relatively small in scale: 3.7K posts of RottenTomatoes compared to 80K posts of TIFU-short as shown in Table 1. Moreover, Rotten Tomatoes use multiple movie reviews written by different users as single source text, and one-sentence consensus made by another professional editor as summary. Thus, each pair of this dataset could be less coherent than that of our TIFU, which is written by the same user. The Idebate dataset is collected from short arguments of debates on controversial topics, and thus the text is rather formal. On the other hand, our dataset contains the posts of interesting stories happened in daily life, and thus the text is more unstructured and informal.

Neural Memory Networks. Many memory network models have been proposed to improve memorization capability of neural networks (Kaiser et al., 2017; Na et al., 2017; Yoo et al., 2019). Weston et al. (2014) propose one of early memory networks for language question answering (QA); since then, many memory networks have been proposed for QA tasks (Sukhbaatar et al., 2015; Kumar et al., 2016; Miller et al., 2016). Park et al. (2017) propose a convolutional read memory network for personalized image captioning. One of the closest works to ours may be Singh et al. (2017), which use a memory network for text summarization. However, they only deal with extractive summarization by storing embeddings of individual sentences into memory.

Dataset	# posts	# words/post	# words/summ
RottenTomatoes	3,731	2124.7 (1747)	22.2 (22)
Idebate	2,259	178.3 (160)	11.4 (10)
TIFU-short	79,949	342.4 (269)	9.33 (8)
TIFU-long	42,984	432.6 (351)	23.0 (21)

Table 1: Statistics of the Reddit TIFU dataset compared to existing opinion summarization corpora, *RottenTomatoes* and *Idebate* (Wang and Ling, 2016). We show average and median (in parentheses) values.

Compared to previous memory networks, our MMN has four novel features: (i) building a multi-level memory network that better abstracts multi-level representation of a long document, (ii) employing a dilated convolutional memory write mechanism to correlate adjacent memory cells, (iii) proposing normalized gated tanh units to avoid covariate shift within the network, and (iv) generating an output sequence without RNNs.

3 Reddit TIFU Dataset

We introduce the Reddit TIFU dataset whose key statistics are outlined in Table 1. We collect data from Reddit, which is a discussion forum platform with a large number of subreddits on diverse topics and interests. Specifically, we crawl all the posts from 2013-Jan to 2018-Mar in the TIFU subreddit, where every post should strictly follow the posting rules, otherwise they are removed. Thanks to the following rules[1], the posts in this subreddit can be an excellent corpus for abstractive summarization: *Rule 3: Posts and titles without context will be removed. Your title must make an attempt to encapsulate the nature of your f***up. Rule 11: All posts must end with a TL;DR summary that is descriptive of your f***up and its consequences.* Thus, we regard the body text as source, the title as short summary, and the TL;DR summary as long summary. As a result, we make two sets of datasets: *TIFU-short* and *TIFU-long*. Figure 1 shows an example post of the TIFU subreddit.

[1] https://reddit.com/r/tifu/wiki/rules.

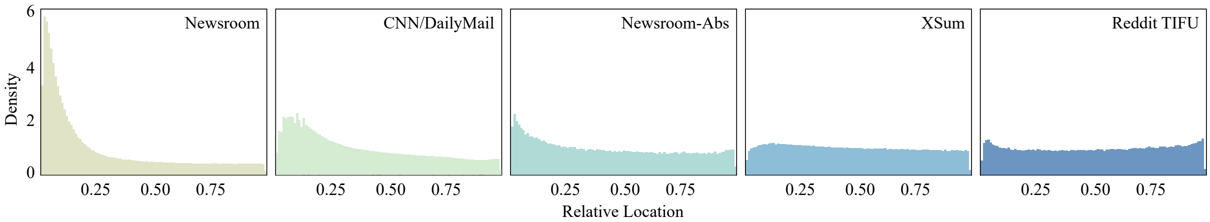


Figure 2: Relative locations of bigrams of gold summary in the source text across different datasets.

	`PG`			`Lead`			`Ext-Oracle`			`PG/Lead`	`PG/Oracle`
Dataset	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	Ratio (R-L)	Ratio (R-L)
`CNN/DM` (Nallapati et al., 2016)	36.4	15.7	33.4	39.6	17.7	36.2	54.7	30.4	50.8	0.92x	0.66x
`NY Times` (Sandhaus, 2008)	44.3	27.4	40.4	31.9	15.9	23.8	52.1	31.6	46.7	1.70x	0.87x
`Newsroom` (Grusky et al., 2018)	26.0	13.3	22.4	30.5	21.3	28.4	41.4	24.2	39.4	0.79x	0.57x
`Newsroom-Abs` (Grusky et al., 2018)	**14.7**	**2.2**	**10.3**	13.7	2.4	11.2	29.7	10.5	27.2	0.92x	0.38x
`XSum` (Narayan et al., 2018a)	29.7	9.2	23.2	16.3	1.6	12.0	29.8	8.8	22.7	1.93x	1.02x
`TIFU-short`	18.3	6.5	17.9	**3.4**	**0.0**	**3.3**	**8.0**	**0.0**	**7.7**	**5.42x**	**2.32x**
`TIFU-long`	19.0	3.7	15.1	**2.8**	**0.0**	**2.7**	**6.8**	**0.0**	**6.6**	**5.59x**	**2.29x**

Table 2: Comparison of F1 ROUGE scores between different datasets (row) and methods (column). `PG` is a state-of-the-art abstractive summarization method, and `Lead` and `Ext-Oracle` are extractive ones. `PG/Lead` and `PG/Oracle` are the ROUGE-L ratios of `PG` with `Lead` and `Ext-Oracle`, respectively. We report the numbers for each dataset (row) from the corresponding cited papers.

3.1 Preprocessing

We build a vocabulary dictionary $\mathcal{V}$ by choosing the most frequent V(=15K) words in the dataset. We exclude any urls, unicodes and special characters. We lowercase words, and normalize digits to 0. Subreddit names and user ids are replaced with @subreddit and @userid token, respectively. We use `markdown`[2] package to strip markdown format, and `spacy`[3] to tokenize words. Common prefixes of summary sentences (*e.g.* tifu by, tifu-, tl;dr, etc) are trimmed. We do not take OOV words into consideration, since our vocabulary with size 15K covers about 98% of word frequencies in our dataset. We set the maximum length of a document as 500. We exclude the gold summaries whose lengths are more than 20 and 50 for *TIFU-short* and *TIFU-long*, respectively. They amount to about 0.6K posts in both datasets (*i.e.* less than 1% and 3%). We use these maximum lengths, based on previous datasets (*e.g.* 8, 31, 56 words on average per summary in Gigaword, DUC, and CNN/DailyMail datasets, respectively). We randomly split the dataset into 95% for training, 5% for test.

3.2 Abstractive Properties of Reddit TIFU

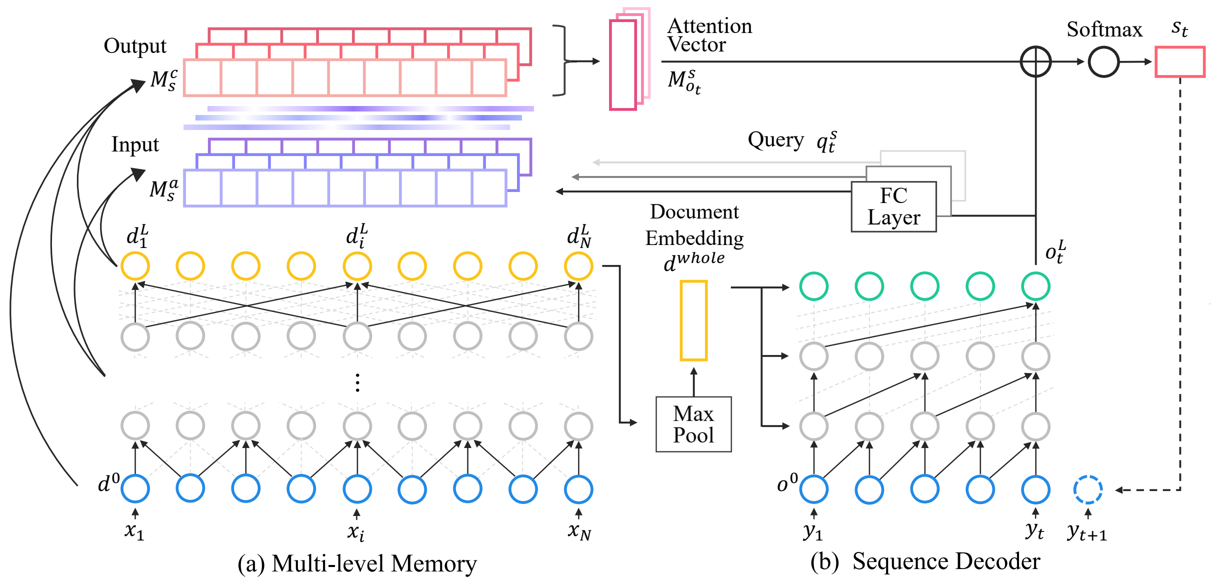
We discuss some abstractive characteristics found in Reddit TIFU dataset, compared to existing summarization datasets based on news articles.

Weak Lead Bias. Formal documents including news articles tend to be structured to emphasize key information at the beginning of the text. On the other hand, key information in informal online text data are more spread across the text. Figure 2 plots the density histogram of the relative locations of bigrams of gold summary in the source text. In the CNN/DailyMail and Newsroom, the bigrams are highly concentrated on the front parts of documents. Contrarily, our Reddit TIFU dataset shows rather uniform distribution across the text.

This characteristic can be also seen from the ROUGE score comparison in Table 2. The `Lead` baseline simply creates a summary by selecting the first few sentences or words in the document. Thus, a high score of the Lead baseline implicates a strong lead bias. The `Lead` scores are the lowest in our TIFU dataset, in which it is more difficult for models to simply take advantage of locational bias for the summary.

Strong Abstractness. Besides the locational bias, news articles tend to contain wrap-up sentences that cover the whole article, and they often have resemblance to its gold summary. Its existence can be measured by the score of the `Ext-Oracle` baseline, which creates a summary by selecting the sentences with the highest average score of F1 ROUGE-1/2/L. Thus, it can be viewed as an upper bound for extractive models (Narayan et al., 2018a,b; Nallapati et al., 2017).

[2] `https://python-markdown.github.io/`.
[3] `https://spacy.io`.



(a) Multi-level Memory

(b) Sequence Decoder

Figure 3: Illustration of the proposed *multi-level memory network* (MMN) model.

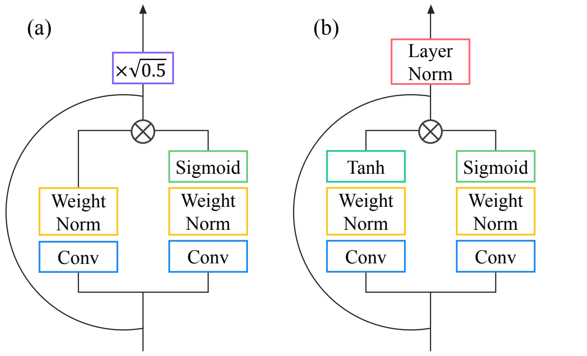


Figure 4: Comparison between (a) the gated linear unit (Gehring et al., 2017) and (b) the proposed normalized gated tanh unit.

In Table 2, the ROUGE scores of the `Ext-Oracle` are the lowest in our TIFU dataset. It means that the sentences that are similar to gold summary scarcely exist inside the source text in our dataset. This property forces the model to be trained to focus on comprehending the entire text instead of simply finding wrap-up sentences.

Finally, `PG/Lead` and `PG/Oracle` in Table 2 are the ROUGE-L ratios of `PG` with `Lead` and `Ext-Oracle`, respectively. These metrics can quantify the dataset according to the degree of difficulty for extractive methods and the suitability for abstractive methods, respectively. High scores of the TIFU dataset in both metrics show that it is potentially an excellent benchmark for evaluation of abstractive summarization systems.

4 Multi-level Memory Networks (MMN)

Figure 3 shows the proposed *multi-level memory network* (MMN) model. The MMN memorizes the source text with a proper representation in the memory and generates a summary sentence one word at a time by extracting relevant information from memory cells in response to previously generated words. The input of the model is a source text $\{x_i\} = x_1, ..., x_N$, and the output is a sequence of summary words $\{y_t\} = y_1, ..., y_T$, each of which is a symbol from the dictionary $\mathcal{V}$.

4.1 Text Embedding

Online posts include lots of morphologically similar words, which should be closely embedded. Thus, we use the `fastText` (Bojanowski et al., 2016) trained on the Common Crawl corpus, to initialize the word embedding matrix $\mathbf{W}_{emb}$. We use the same embedding matrix $\mathbf{W}_{emb}$ for both source text and output sentences. That is, we represent a source text $\{x_i\}_{i=1}^N$ in a distributional space as $\{\mathbf{d}_i^0\}_{i=1}^N$ by $\mathbf{d}_i^0 = \mathbf{W}_{emb}\mathbf{x}_i$ where $\mathbf{x}_i$ is a one-hot vector for i-th word in the source text. Likewise, output words $\{y_t\}_{t=1}^T$ is embedded as $\{\mathbf{o}_t^0\}_{t=1}^T$, and $\mathbf{d}_i^0$ and $\mathbf{o}_t^0 \in \mathbb{R}^{300}$.

4.2 Construction of Multi-level Memory

As shown in Figure 3(a), the multi-level memory network takes the source text embedding $\{\mathbf{d}_i^0\}_{i=1}^N$ as an input, and generates S number of memory tensors $\{\mathbf{M}_s^{a/c}\}_{s=1}^S$ as output, where superscript a and c denote input and output memory representation, respectively. The multi-level memory network is motivated by that when human understand a document, she does not remember it as a single whole document but ties together several levels of abstraction (*e.g.* word-level, sentence-level, paragraph-level and document-level). That is, we generate S sets of memory tensors, each of which associates each cell with different number of neighboring word embeddings based on the level of abstraction. To build memory slots of such multi-level memory, we exploit a multi-layer CNN as the write network, where each layer is chosen based on the size of its receptive field.

However, one issue of convolution is that large receptive fields require many layers or large filter sizes. For example, stacking 6 layers with a filter size of 3 results in a receptive field size of 13, *i.e.* each output depends on 13 input words. In order to grow the receptive field without increasing the computational cost, we exploit the *dilated* convolution (Yu and Koltun, 2016; Oord et al., 2016a) for the write network.

Memory Writing with Dilated Convolution. In dilated convolution, the filter is applied over an area larger than its length by skipping input values with a certain gap. Formally, for a 1-D n-length input $\mathbf{x} \in \mathbb{R}^{n\times 300}$ and a filter $\mathbf{w} : \{1, ..., k\} \to \mathbb{R}^{300}$, the dilated convolution operation $\mathcal{F}$ on s elements of a sequence is defined as

$$\mathcal{F}(\mathbf{x}, s) = \sum_{i=1}^{k} \mathbf{w}(i) * \mathbf{x}_{s+d\cdot(i-\lfloor k/2 \rfloor)} + \mathbf{b}, \quad (1)$$

where d is the dilation rate, k is the filter size, $s - d \cdot (i - \lfloor k/2 \rfloor)$ accounts for the direction of dilation and $\mathbf{w} \in \mathbb{R}^{k\times 300\times 300}$ and $\mathbf{b} \in \mathbb{R}^{300}$ are the parameters of the filter. With $d = 1$, the dilated convolution reduces to a regular convolution. Using a larger dilation enables a single output at the top level to represent a wider range of input, thus effectively expanding the receptive field.

To the embedding of a source text $\{\mathbf{d}_i^0\}_{i=1}^N$, we recursively apply a series of dilated convolutions $F(\mathbf{d}^0) \in \mathbb{R}^{N\times 300}$. We denote the output of the l-th convolution layer as $\{\mathbf{d}_i^l\}_{i=1}^N$.

Normalized Gated Tanh Units. Each convolution is followed by our new activation of *normalized gated tanh unit* (NGTU), which is illustrated in Figure 4(b):

$$\mathrm{GTU}(\mathbf{d}^l) = \tanh(\mathcal{F}_f^l(\mathbf{d}^l)) \circ \sigma(\mathcal{F}_g^l(\mathbf{d}^l)), \quad (2)$$

$$\mathbf{d}^{l+1} = \mathrm{LayerNorm}(\mathbf{d}^l + \mathrm{GTU}(\mathbf{d}^l)), \quad (3)$$

where σ is a sigmoid, $\circ$ is the element-wise multiplication and F_f^l and F_g^l denote the filter and gate for l-th layer dilated convolution, respectively.

The NGTU is an extension of the existing gated tanh units (GTU) (Oord et al., 2016a,b) by applying weight normalization (Salimans and Kingma, 2016) and layer normalization (Ba et al., 2016). This mixed normalization improves earlier work of Gehring et al. (2017), where only weight normalization is applied to the GLU. As in Figure 4(a), it tries to preserve the variance of activations throughout the whole network by scaling the output of residual blocks by $\sqrt{0.5}$. However, we observe that this heuristic does not always preserve the variance and does not empirically work well in our dataset. Contrarily, the proposed NGTU not only guarantees preservation of activation variances but also significantly improves the performance.

Multi-level Memory. Instead of using only the last layer output of CNNs, we exploit the outputs of multiple layers of CNNs to construct S sets of memories. For example, memory constructed from the 4-th layer, whose receptive field is 31, may have sentence-level embeddings, while memory from the 8-th layer, whose receptive field is 511, may have document-level embeddings. We obtain each s-th level memory $\mathbf{M}_s^{a/c}$ by resembling key-value memory networks (Miller et al., 2016):

$$\mathbf{M}_s^a = \mathbf{d}^{\mathbf{m}(s)}, \ \mathbf{M}_s^c = \mathbf{d}^{\mathbf{m}(s)} + \mathbf{d}^0. \quad (4)$$

Recall that $\mathbf{M}_s^a$ and $\mathbf{M}_s^c \in \mathbb{R}^{N\times 300}$ are input and output memory matrix, respectively. $\mathbf{m}(s)$ indicates an index of convolutional layer used for the s-th level memory. For example, if we set $S = 3$ and $\mathbf{m} = \{3, 6, 9\}$, we make three-level memories, each of which uses the output of the 3-rd, 6-th, and 9-th convolution layer, respectively. To output memory representation $\mathbf{M}_s^c$, we add the document embedding $\mathbf{d}^0$ as a skip connection.

4.3 State-Based Sequence Generation

We discuss how to predict the next word y_{t+1} at time step t based on the memory state and previously generated words $y_{1:t}$. Figure 3(b) visualizes the overall procedure of decoding.

We first apply max-pooling to the output of the last layer of the encoder network to build a whole document embedding $\mathbf{d}^{whole} \in \mathbb{R}^{300}$:

$$\mathbf{d}^{whole} = \mathrm{maxpool}([\mathbf{d}_1^L; ...; \mathbf{d}_N^L]). \quad (5)$$

The decoder is designed based on WaveNet (Oord et al., 2016a) that uses a series of causal dilated convolutions, denoted by $\hat{\mathcal{F}}(\mathbf{o}^l_{1:t}) \in \mathbb{R}^{t\times 300}$. We globally condition $\mathbf{d}^{whole}$ to obtain embeddings of previously generated words $\mathbf{o}^l_{1:t}$ as:

$$\mathbf{h}^l_{f/g} = \hat{\mathcal{F}}^l_{f/g}(\mathbf{o}^l_{1:t} + \mathbf{W}^l_{f/g}\mathbf{d}^{whole}), \quad (6)$$

$$\mathbf{h}^l_a = \tanh(\mathbf{h}^l_f) \circ \sigma(\mathbf{h}^l_g), \quad (7)$$

$$\mathbf{o}^{l+1}_{1:t} = \text{LayerNorm}(\mathbf{o}^l_{1:t} + \mathbf{h}^l_a), \quad (8)$$

where $\mathbf{h}^l_{f/g}$ are the filter and gate hidden state respectively, and learnable parameters are $\mathbf{W}^l_f$ and $\mathbf{W}^l_g \in \mathbb{R}^{300\times 300}$. We initialize $\mathbf{o}^0_t = \mathbf{W}_{emb}\mathbf{y}_t$. We set the level of the decoder network to $L = 3$ for TIFU-short and $L = 5$ for TIFU-long.

Next, we generate S number of query vectors $\{\mathbf{q}^s_t\}^S_{s=1}$ at time t to our memory network as

$$\mathbf{q}^s_t = \tanh(\mathbf{W}^s_q\mathbf{o}^L_t + \mathbf{b}^s_q), \quad (9)$$

where $\mathbf{W}^s_q \in \mathbb{R}^{300\times 300}$ and $\mathbf{b}^s_q \in \mathbb{R}^{300}$.

Each of these query vectors $\{\mathbf{q}^s_t\}^S_{s=1}$ is fed into the attention function of each level of memory. As in (Vaswani et al., 2017), the attention function is

$$\mathbf{M}^s_{o_t} = \text{softmax}(\frac{\mathbf{q}^s_t(\mathbf{M}^a_s)^T}{\sqrt{d^{emb}}})\mathbf{M}^c_s, \quad (10)$$

where we set $d^{emb} = 300$ for the embedding dimension and $\mathbf{M}^s_{o_t} \in \mathbb{R}^{300}$.

Next, we obtain the output word probability:

$$\mathbf{s}_t = \text{softmax}(\mathbf{W}_o[\mathbf{M}^1_{o_t}; ...; \mathbf{M}^S_{o_t}; \mathbf{o}^L_t]), \quad (11)$$

where $\mathbf{W}_o \in \mathbb{R}^{(300\times(S+1))\times V}$. Finally, we select the word with the highest probability $y_{t+1} = \text{argmax}_{\mathbf{s}\in\mathcal{V}}(\mathbf{s}_t)$. Unless y_{t+1} is an EOS token, we repeat generating the next word by feeding y_{t+1} into the output convolution layer of Eq.(8).

4.4 Training

We use the softmax cross-entropy loss from estimated y_t to its target $y_{GT,t}$. However, it forces the model to predict extremes (zero or one) to distinguish among the ground truth and alternatives. The label smoothing alleviates this issue by acting as a regularizer that makes the model less confident in its prediction. We smooth the target distribution with a uniform prior distribution u (Pereyra et al., 2017; Edunov et al., 2017; Vaswani et al., 2017). Thus, the loss over the training set $\mathcal{D}$ is

$$\mathcal{L} = -\sum \log p_\theta(\mathbf{y}|\mathbf{x}) - D_{KL}(u||p_\theta(\mathbf{y}|\mathbf{x})).$$

We implement label smoothing by modifying the ground truth distribution for word $y_{GT,t}$ to be $p(y_{GT,t}) = 1 - \epsilon$ and $p(y') = \epsilon/\mathcal{V}$ for $y' \neq y_{GT,t}$ where ϵ is a smoothing parameter set to 0.1. Further details can be found in the Appendix.

5 Experiments

5.1 Experimental Setting

Evaluation Metrics. We evaluate the summarization performance with two language metrics: perplexity and standard F1 ROUGE scores (Lin, 2004). We remind that lower perplexity and higher ROUGE scores indicate better performance.

Datasets. In addition to Reddit TIFU, we also evaluate on two existing datasets: abstractive subset of `Newsroom` (Grusky et al., 2018) and `XSum` (Narayan et al., 2018a). These are suitable benchmarks for evaluation of our model in two aspects. First, they are specialized for abstractive summarization, which meets well the goal of this work. Second, they have larger vocabulary size (40K, 50K) than Reddit TIFU (15K), and thus we can evaluate the learning capability of our model.

Baselines. We compare with three abstractive summarization methods, one basic seq2seq model, two heuristic extractive methods and variants of our model. We choose `PG` (See et al., 2017), `SEASS` (Zhou et al., 2017), `DRGD` (Li et al., 2017) as the state-of-the-art methods of abstractive summarization. We test the attention based seq2seq model denoted as `s2s-att` (Chopra et al., 2016). As heuristic extractive methods, the `Lead-1` uses the first sentence in the text as summary, and the `Ext-Oracle` takes the sentence with the highest average score of F1 ROUGE-1/2/L with the gold summary in the text. Thus, `Ext-Oracle` can be viewed as an upper-bound for extractive methods.

We also test variants of our method `MMN-*`. To validate the contribution of each component, we exclude one of key components from our model as follows: (i) `-NoDilated` with conventional convolutions instead, (ii) `-NoMulti` with no multi-level memory (iii) `-NoNGTU` with existing gated linear units (Gehring et al., 2017). That is, `-NoDilated` quantifies the improvement by the dilated convolution, `-NoMulti` assesses the effect of multi-level memory, and `-NoNGTU` validates the normalized gated tanh unit.

Please refer to the Appendix for implementation details of our method.

Methods	PPL	R-1	R-2	R-L
TIFU-short				
Lead-1	n/a	3.4	0.0	3.3
Ext-Oracle	n/a	8.0	0.0	7.7
s2s-att (Chopra et al., 2016)	46.2	18.3	6.4	17.8
PG (See et al., 2017)	40.9	18.3	6.5	17.9
SEASS (Zhou et al., 2017)	62.6	18.5	6.4	18.0
DRGD (Li et al., 2017)	69.2	14.6	3.3	14.2
MMN	32.1	**20.2**	**7.4**	**19.8**
MMN-NoDilated	**31.8**	19.5	6.8	19.1
MMN-NoMulti	34.4	19.0	6.1	18.5
MMN-NoNGTU	40.8	18.6	5.6	18.1
TIFU-long				
Lead-1	n/a	2.8	0.0	2.7
Ext-Oracle	n/a	6.8	0.0	6.6
s2s-att (Chopra et al., 2016)	180.6	17.3	3.1	14.0
PG (See et al., 2017)	175.3	16.4	3.0	13.5
SEASS (Zhou et al., 2017)	387.0	17.5	2.9	13.9
DRGD (Li et al., 2017)	176.6	16.8	2.0	13.6
MMN	**114.1**	**19.0**	**3.7**	**15.1**
MMN-NoDilated	124.2	17.6	3.4	14.1
MMN-NoMulti	124.5	14.0	1.5	11.8
MMN-NoNGTU	235.4	14.0	2.6	12.1

Table 3: Summarization results measured by perplexity and ROUGE-1/2/L on the TIFU-short/long dataset.

	Newsroom-Abs			**XSum**		
Methods	R-1	R-2	R-L	R-1	R-2	R-L
s2s-att	6.2	1.1	5.7	28.4	8.8	22.5
PG	14.7	2.2	11.4	29.7	9.2	23.2
ConvS2S	-	-	-	31.3	11.1	25.2
T-ConvS2S	-	-	-	31.9	11.5	25.8
MMN (Ours)	**17.5**	**4.7**	**14.2**	**32.0**	**12.1**	**26.0**

Table 4: Summarization results in terms of ROUGE-1/2/L on Newsroom-Abs (Grusky et al., 2018) and XSum (Narayan et al., 2018a). Except MMN, all scores are referred to the original papers. T-ConvS2S is the topic-aware convolutional seq2seq model.

5.2 Quantitative Results

Table 3 compares the summarization performance of different methods on the TIFU-short/long dataset. Our model outperforms the state-of-the-art abstractive methods in both ROUGE and perplexity scores. PG utilizes a pointer network to copy words from the source text, but it may not be a good strategy in our dataset, which is more abstractive as discussed in Table 2. SEASS shows strong performance in DUC and Gigaword dataset, in which the source text is a single long sentence and the gold summary is its shorter version. Yet, it may not be sufficient to summarize much longer articles of our dataset, even with its second-level representation. DRGD is based on the variational autoencoder with latent variables to capture the structural patterns of gold summaries. This idea can be useful for the similarly structured formal documents but may not go well with diverse online text in the TIFU dataset.

	TIFU-short			**TIFU-long**		
vs. Baselines	Win	Lose	Tie	Win	Lose	Tie
s2s-att	**43.0**	28.3	28.7	**32.0**	24.0	44.0
PG	**38.7**	28.0	33.3	**42.3**	33.3	24.3
SEASS	**35.7**	28.0	36.3	**47.0**	37.3	15.7
DRGD	**46.7**	17.3	15.0	**61.0**	23.0	16.0
Gold	27.0	**58.0**	15.0	22.3	**73.7**	4.0

Table 5: AMT results on the TIFU-short/long between our MMN and four baselines and gold summary. We show percentages of responses that turkers vote for our approach over baselines.

These state-of-the-art abstractive methods are not as good as our model, but still perform better than extractive methods. Although the Ext-Oracle heuristic is an upper-bound for extractive methods, it is not successful in our highly abstractive dataset; it is not effective to simply retrieve existing sentences from the source text. Moreover, the performance gaps between abstractive and extractive methods are much larger in our dataset than in other datasets (See et al., 2017; Paulus et al., 2018; Cohan et al., 2018), which means too that our dataset is highly abstractive.

Table 4 compares the performance of our MMN on Newsroom-Abs and XSum dataset. We report the numbers from the original papers. Our model outperforms not only the RNN-based abstractive methods but also the convolutional-based methods in all ROUGE scores. Especially, even trained on single end-to-end training procedure, our model outperforms T-ConvS2S, which necessitates two training stages of LDA and ConvS2S. These results assure that even on formal documents with large vocabulary sizes, our multi-level memory is effective for abstractive datasets.

5.3 Qualitative Results

We perform two types of qualitative evaluation to complement the limitation of automatic language metrics as summarization evaluation.

User Preferences. We perform Amazon Mechanical Turk (AMT) tests to observe general users' preferences between the summarization of different algorithms. We randomly sample 100 test examples. At test, we show a source text and two summaries generated by our method and one baseline in a random order. We ask turkers to choose the more relevant one for the source text. We obtain answers from three different turkers for each test example. We compare with four abstractive baselines (s2s-att, PG, SEASS and DRGD)

[Source Text]
(…) I decided to go over to my friends house to a small party at 1 in the morning. I knew my parents would say no so I snuck out of the house. (…) I had been talking to my mom about how sad even hearing the theme song made me. Also she had seen me watching a bunch of sad anime theme songs and tearing up a little so she must have thought I was depressed. When I got home today my mom was practically in tears. (…)

[Short Summary]
(GT) sneaking out of my friends house last night
(Ours) sneaking out of my friends house
(PG) not watching my friends
(SEASS) accidentally spoiling my mom song
(s2s-att) sneaking out of town
(DRGD) watching a movie

[Source Text]
(…) Saturday was on my way to a party and this dog was walking in the road. (…) Since it was a holiday I couldn't get her scanned for a chip but she was obviously neglected. Missing fur from flea infestation, (…) Yesterday I was able to go get her scanned for a chip. No chip. So I get ready to take her home and deflea her. (…) Anyway a third party today starts accusing me of stealing (…) and talking about pressing charges. (…)

[Long Summary]
(GT) Saved a dog. Had to give dog back to possible abusers. Being accused of stealing the fucking dog. No good deed goes unpunished.
(Ours) tried to help a dog got a bit and got accused of stealing
(PG) _EOS
(SEASS) called a dog a _UNK might get charged with _UNK
(s2s-att) got accused of being a dog by stealing a _UNK bit the dog and accused of stealing dog to the police
(DRGD) i was a _UNK dog and I wasn't playing attention and got arrested for being a _UNK _UNK

Figure 5: Examples of abstractive summary generated by our model and baselines. In each set, we too show the source text and gold summary.

and the gold summary (Gold).

Table 5 summarizes the results of AMT tests, which validate that human annotators significantly prefer our results to those of baselines. As expected, the gold summary is voted the most.

Summary Examples. Figure 5 shows selected examples of abstractive summarization. Baselines often generate the summary by mostly focusing on some keywords in the text, while our model produces the summary considering both keywords and the whole context thanks to multi-level memory. We present more examples in the Appendix.

6 Conclusions

We introduced a new dataset *Reddit TIFU* for abstractive summarization on informal online text. We also proposed a novel summarization model named *multi-level memory networks* (MMN). Experiments showed that the Reddit TIFU dataset is uniquely abstractive and the MMN model is highly effective. There are several promising future directions. First, ROUGE metrics are limited to correctly capture paraphrased summaries, for which a new automatic metric of abstractive summarization may be required. Second, we can explore the data in other online forums such as Quora, Stackoverflow and other subreddits.

Acknowledgments

We thank Chris Dongjoo Kim, Yunseok Jang and the anonymous reviewers for their helpful comments. This work was supported by Kakao and Kakao Brain corporations and IITP grant funded by the Korea government (MSIT) (No. 2017-0-01772, Development of QA systems for Video Story Understanding to pass the Video Turing Test). Gunhee Kim is the corresponding author.

References

Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. 2017. Text Summarization Techniques: A Brief Survey. In *arXiv:1707.02268*.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer Normalization. In *Stat*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching Word Vectors with Subword Information. In *TACL*.

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep Communicating Agents for Abstractive Summarization. In *NAACL-HLT*.

Sumit Chopra, Michael Auli, and Alexander M Rush. 2016. Abstractive Sentence Summarization With Attentive Recurrent Neural Networks. In *NAACL-HLT*.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents. In *NAACL-HLT*.

Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2017. Classical Structured Prediction Losses for Sequence to Sequence Learning. In *NAACL-HLT*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional Sequence to Sequence Learning. In *ICML*.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *EMNLP*.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the Difficulty of Training Deep Feedforward Neural Networks. In *AISTATS*.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A Dataset of 1.3 Million Summaries with Diverse Extractive Strategies. In *NAACL-HLT*.

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*.

Wan-Ting Hsu, Chieh-Kai Lin, Ming-Ying Lee, Kerui Min, Jing Tang, and Min Sun. 2018. A Unified Model for Extractive and Abstractive Summarization using Inconsistency Loss. In *ACL*.

Baotian Hu, Qingcai Chen, and Fangze Zhu. 2015. Lcsts: A large scale chinese short text summarization dataset. In *EMNLP*.

Łukasz Kaiser, Ofir Nachum, Aurko Roy, and Samy Bengio. 2017. Learning to Remember Rare Events. In *ICLR*.

Chris Kedzie, Kathleen McKeown, and Hal Daume III. 2018. Content Selection in Deep Learning Models of Summarization. In *EMNLP*.

Diederik Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

Ankit Kumar, Ozan Irsoy, Jonathan Su, James Bradbury, Robert English, Brian Pierce, Peter Ondruska, Ishaan Gulrajani, and Richard Socher. 2016. Ask me Anything: Dynamic Memory Networks for Natural Language Processing. In *ICML*.

Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. 2017. Deep Recurrent Generative Decoder for Abstractive Text Summarization. In *EMNLP*.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *TSBO*.

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating Wikipedia by Summarizing Long Sequences. In *ICLR*.

Yishu Miao and Phil Blunsom. 2016. Language as a Latent Variable: Discrete Generative Models for Sentence Compression. In *EMNLP*.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-Value Memory Networks for Directly Reading Documents. In *EMNLP*.

Seil Na, Sangho Lee, Jisung Kim, and Gunhee Kim. 2017. A Read-Write Memory Network for Movie Story Understanding. In *ICCV*.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents. In *AAAI*.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive Text Summarization Using Sequence-to-sequence RNNs and Beyond. In *CoNLL*.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated Gigaword. In *NAACL-HLT AKBC-WEKEX*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018a. Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. In *EMNLP*.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018b. Ranking Sentences for Extractive Summarization with Reinforcement Learning. In *NAACL-HLT*.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016a. WaveNet: A Generative Model for Raw Audio. In *SSW*.

Aaron van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. 2016b. Conditional Image Generation With Pixelcnn Decoders. In *NIPS*.

Paul Over, Hoa Dang, and Donna Harman. 2007. DUC in Context. In *IPM*.

Cesc Chunseong Park, Byeongchang Kim, and Gunhee Kim. 2017. Attend to You: Personalized Image Captioning with Context Sequence Memory Networks. In *CVPR*.

Ramakanth Pasunuru and Mohit Bansal. 2018. Multi-Reward Reinforced Summarization with Saliency and Entailment. In *NAACL-HLT*.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A Deep Reinforced Model for Abstractive Summarization. In *ICLR*.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing Neural Networks by Penalizing Confident Output Distributions. In *ICLR*.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. In *EMNLP*.

Tim Salimans and Diederik P Kingma. 2016. Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks. In *NIPS*.

Evan Sandhaus. 2008. New York Times Annotated Corpus. In *LDC*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the Point: Summarization with Pointer-Generator Networks. In *ACL*.

Abhishek Kumar Singh, Manish Gupta, and Vasudeva Varma. 2017. Hybrid MemNet for Extractive Summarization. In *CIKM*.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end Memory Networks. In *NIPS*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *NIPS*.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive Document Summarization with a Graph-based Attentional Neural Model. In *ACL*.

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 *NIPS*.

Lu Wang and Wang Ling. 2016. Neural Network-Based Abstract Generation for Opinions and Arguments. In *NAACL-HLT*.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory Networks. In *ICLR*.

Seungjoo Yoo, Hyojin Bahng, Sunghyo Chung, Junsoo Lee, Jaehyuk Chang, and Jaegul Choo. 2019. Coloring with Limited Data: Few-shot Colorization via Memory-Augmented Networks. In *CVPR*.

Fisher Yu and Vladlen Koltun. 2016. Multi-scale Context Aggregation by Dilated Convolutions. In *ICLR*.

Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. 2017. Selective Encoding for Abstractive Sentence Summarization. In *ACL*.

A Implementation Details

All the parameters are initialized with the Xavier method (Glorot and Bengio, 2010). We apply the Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9, \beta_2 = 0.999$ and $\epsilon = 1e - 8$. We apply weight normalization (Salimans and Kingma, 2016) to all layers. We set learning rate to 0.001 and clip gradient at 0.3. At every 4 epochs, we divide learning rate by 10 until it reaches 0.0001. We train our models up to 12 epochs for TIFU-short and 60 epochs for TIFU-long.

Table 6 summarizes the setting of hyperparameters for our model in all experiments on TIFU-short/long dataset, Newsroom abstractive subset and XSum.

Description	**Common Configurations**	
Initial learning rate	0.001	
Embedding dimension (d^{emb})	300	
Kernel size (k)	3	
Dilation rate (d)	2^l	
Description	**TIFU-short**	**TIFU-long**
Grad clip	0.3	0.3
# of encoder layers	9	8
# of decoder layers	3	5
Layers used for memory ($\mathbf{m}$)	{3, 6, 9}	{4, 8}
Smoothing parameter (ϵ)	0.1	0.05
Description	**Newsroom-Abs**	**XSum**
Grad clip	0.3	0.8
# of encoder layers	10	9
# of decoder layers	6	6
Layers used for memory ($\mathbf{m}$)	{3, 6, 10}	{4, 7, 9}
Smoothing parameter (ϵ)	0.05	0.05

Table 6: Model hyperparameters in experiments on TIFU-short/long, Newsroom abstractive subset and XSum.

	Novel N-gram Ratio			
Dataset	1-gram	2-gram	3-gram	4-gram
`CNN/DailyMail`	10.3	49.9	70.5	80.3
`NY Times`	11.0	45.5	67.2	77.9
`Newsroom`	15.6	45.4	57.2	62.2
`Newsroom-Ext`	1.5	5.9	8.9	11.1
`Newsroom-Mix`	11.6	47.0	66.5	76.8
`Newsroom-Abs`	33.9	83.9	97.1	99.5
`XSum`	35.8	83.5	95.5	98.5
`TIFU-short`	29.7	71.5	88.1	93.8
`TIFU-long`	27.4	76.7	92.5	97.0

Table 7: Comparison of novel N-gram ratios between Reddit TIFU and other summarization datasets.

B Novel N-gram Ratios

Table 7 compares the ratios of novel N-grams in the reference summary between datasets. Following (See et al., 2017; Narayan et al., 2018a), we compute this ratio as follows; we first count the number of N-grams in the reference summary that do not appear in the source text and divide it with the total number of N-grams. The higher the ratio is, the less the identical N-grams are in the source text. The `CNN/DailyMail`, `New York Times`, `Newsroom` datasets all, for example, exhibit low novel 1-gram ratios as 10.3%, 11.0%, 15.6 % respectively. This means that about 90% of the words in reference summary already exist inside the source text. It is due to that the summaries from formal documents (*e.g.* news and academic papers) tend to have same expressions with the source documents. Therefore, these datasets may be more suitable for extractive summarization than abstractive one; on the other hand, our dataset is more abstractive.

We also compare the novel N-gram ratio for `XSum` and three subsets of `Newsroom`;

(i) `Newsroom-Ext`, a subset favorable for extractive methods, (ii) `Newsroom-Mix`, a subset favorable for mixed methods, and (iii) `Newsroom-Abs`, a subset favorable for abstrac-

[Source Text]
(…) This weekend I went out and bought myself a motorcycle! I've been planning on buying one this summer and I finally went out and did it. (…) I geared up and went on my way for my first ride. (…) I went all the way back to my house and alas, no phone. So I spent the next 4 hours walking the route to my girlfriends house looking along the side of the road for my phone. These are all back roads to my girlfriends house so I had to really get in the undergrowth to look for my phone. It got dark and I headed home feeling dejected. The next morning my friend came over with my phone! The only damage was a few scuffs to my otterbox. I couldn't believe he found it, and as it turns out it fell off my bike and tumbled onto the side of his driveway. I chalked this up as a win and considered myself lucky. Until yesterday. I woke up with poison ivy all over my body. And when I say all over my body, I mean ALL OVER my body. I have some on my arms, legs, face, and most importantly all over my dick. And the worst spot of them all is on my dick. I have never been so uncomfortable in my life. I must have had some of the oil on my hands and scratched an itch down there. Needless to say I haven't been on my new motorcycle since I've had this and I've been doing as little moving as possible at my job and at home. F***.

[Short Summary]

(Gold) taking a ride on my new motorcycle

(Ours) buying my new motorcycle

(s2s-att) trying to be a good samaritan

(PG) not wearing my cargo helmet

(SEASS) going for a ride

(DRGD) getting my phone stuck

[Source Text]
(…) Later that night, the rest of the family proceeded to play cards and become quite intoxicated. Me, being the little shit that I was, and probably still am, took this opportunity to raid the liquor cooler, and made off with a bottle of wine. I, along with a few other of my underage cousins, ran off to consume our loot. Now, in most situations, this really wouldn't be a that big deal, however, getting drunk at 18 wasn't where my f*** up occurred. (…) John tells her it'll be okay, that he will smooth talk it over with them, and that she should bring them a bottle of their favorite wine. Well John and Jane are having dinner outside to meet the parents, and when Jane goes to retrieve her gift of wine to John's parents, she discovers the wine is gone. Jane then begins to panic, and starts tearing up the surrounding area looking for it. John's parents have no idea what she's freaking out about, or why John would bring crazy to the family reunion. (…) Jane now slips into complete hysteria, and runs inside to lock herself in the bathroom. (…) The day after, John convinces his parents to try again, and all goes very well, especially since last night's thief is still recovering from his first wine hangover. (…)

[Short Summary]

(Gold) stealing a bottle of wine

(Ours) getting drunk and stealing a wine bottle

(s2s-att) getting drunk and making a family cry

(PG) ruining my future dinner vacation

(SEASS) accidentally stealing alcohol from my cousin's parent′s party

(DRGD) smoking a cigarette

[Source Text]
(…) We use an internal messaging application software at work which has been great for communicating with other teammates. A lot of us have started using it to complain about things we are not happy about at work. (…) This leads me to today where just as i am about to go home my manager calls me in to a private meeting looking really upset. Then they mentioned the program name and that they had received an email, and suddenly I realized I had fucked up one of the quirks of this program is that when someone is offline it emails them the message. A recently ex co-worker is still active on the chat for quick questions for the next 2 weeks. They came online so we started having a conversation, then another co-worker walked up to me for a chat who has been having a rough week and complained about our boss. When they finished their rant, I then messaged my ex co-worker that my boss wasn't popular with the staff at the moment as that was the second minor complaint I had heard that week. They had gone offline, so an email was sent to their old work email. Past employees emails get sent to the boss, in case important emails are sent to them. So after the meeting I still have my job. I had an awkward conversation with my boss (…)

[Long Summary]

(Gold) message program at work emailed a private message between a past co worker and myself to my boss saying how people where not happy with them

(Ours) sent a message to my boss and now i 'm in a meeting with my boss

(s2s-att) i lied about my boss to get my job and now i 'm in a job with a new job

(PG) i got a program that sacked from work and i got sent to a metting by my boss

[Source Text]
(…) My girlfriend and I have just started to "get a bit more uncomfortable with each other". (…) My curfew for the night had been midnight, however we both got a bit carried away in the beautiful setting (…) I had over extended my curfew by around an hour. This prompted my mother to call me angrily and groggily claiming that she was mad at me. I defended myself by saying that we both fell asleep listening to the docile tones of Steve Harvey on "Family Feud". (…) I sent my girlfriend a text that said something along the lines of "X acts of affection. Comments questions, or complaints?" However upon hitting the send button I noticed that it was heading in the direction of my birth giver. I panicked and luckily managed to put it into airplane mode! (…) I quickly googled "How to prevent texts from sending" and it said to simply delete the text while it was on airplane mode. I did so, and proceeded to turn airplane mode off, however the "always correct Internet" was wrong and I hear my mom receive the text in the next room over. I quickly went into her room to try and crack her phone's code in order to diffuse the bomb (…) I then asked my mother for her phone so that I could "call mine". She reluctantly agreed and I hurriedly rushed to my room. (…)

[Long Summary]

(Gold) did some funky stuff with my date

(Ours) accidentally sent an inappropriate text to a girl and then accidentally sent it to my mother′s phone and now i'm trouble

(PG) tifu by answering an airplane call and accidentally adding my mothers phone and her phone to find out she was sleeping on me

(SEASS) I accidentally sent a text to my mom that I was sending her a text from the _UNK bomb

Figure 6: Examples of abstractive summary generated by our model and baselines. In each set, we too show the source text and reference summary.

tive methods. We summarize two interesting observations as follows. First, as expected, the more favorable for abstractive methods is, the higher novel n-gram ratio is. Second, novel n-gram ratios of `Newsroom-Abs` and `XSum` are higher than those of our dataset, even though their data sources are news publications. Thus, we argue that novel n-gram ratios are pretty good but not a sufficient measure to find extractive bias in the summarization dataset.

C More Examples

Figure 6 illustrates selected examples of summary generation. In each set, we show a source text, a reference summary and generated summaries by our method and baselines. In the examples, while baselines generate summary by mostly focusing on some keywords, our model produces summary considering both keywords and the whole context thanks to the multi-level memory.