Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond

Ramesh Nallapati
IBM Watson
nallapati@us.ibm.com

Bowen Zhou
IBM Watson
zhou@us.ibm.com

Cicero dos Santos
IBM Watson
cicerons@us.ibm.com

Çağlar Gülçehre
Université de Montréal
gulcehrc@iro.umontreal.ca

Bing Xiang
IBM Watson
bingxia@us.ibm.com

Abstract

In this work, we model abstractive text summarization using Attentional Encoder-Decoder Recurrent Neural Networks, and show that they achieve state-of-the-art performance on two different corpora. We propose several novel models that address critical problems in summarization that are not adequately modeled by the basic architecture, such as modeling key-words, capturing the hierarchy of sentence-to-word structure, and emitting words that are rare or unseen at training time. Our work shows that many of our proposed models contribute to further improvement in performance. We also propose a new dataset consisting of multi-sentence summaries, and establish performance benchmarks for further research.

1 Introduction

Abstractive text summarization is the task of generating a headline or a short summary consisting of a few sentences that captures the salient ideas of an article or a passage. We use the adjective 'abstractive' to denote a summary that is not a mere selection of a few existing passages or sentences extracted from the source, but a compressed paraphrasing of the main contents of the document, potentially using vocabulary unseen in the source document.

This task can also be naturally cast as mapping an input sequence of words in a source document to a target sequence of words called summary. In the recent past, deep-learning based models that map an input sequence into another output sequence, called sequence-to-sequence models, have been successful in many problems such as machine translation (Bahdanau et al., 2014),

speech recognition (Bahdanau et al., 2015) and video captioning (Venugopalan et al., 2015). In the framework of sequence-to-sequence models, a very relevant model to our task is the attentional Recurrent Neural Network (RNN) encoder-decoder model proposed in Bahdanau et al. (2014), which has produced state-of-the-art performance in machine translation (MT), which is also a natural language task.

Despite the similarities, abstractive summarization is a very different problem from MT. Unlike in MT, the target (summary) is typically very short and does not depend very much on the length of the source (document) in summarization. Additionally, a key challenge in summarization is to optimally compress the original document in a *lossy manner* such that the key concepts in the original document are preserved, whereas in MT, the translation is expected to be loss-less. In translation, there is a strong notion of almost one-to-one word-level alignment between source and target, but in summarization, it is less obvious.

We make the following main contributions in this work: (i) We apply the off-the-shelf attentional encoder-decoder RNN that was originally developed for machine translation to summarization, and show that it already outperforms state-of-the-art systems on two different English corpora. (ii) Motivated by concrete problems in summarization that are not sufficiently addressed by the machine translation based model, we propose novel models and show that they provide additional improvement in performance. (iii) We propose a new dataset for the task of abstractive summarization of a document into multiple sentences and establish benchmarks.

The rest of the paper is organized as follows. In Section 2, we describe each specific problem in abstractive summarization that we aim to solve, and present a novel model that addresses it. Sec-

tion 3 contextualizes our models with respect to closely related work on the topic of abstractive text summarization. We present the results of our experiments on three different data sets in Section 4. We also present some qualitative analysis of the output from our models in Section 5 before concluding the paper with remarks on our future direction in Section 6.

2 Models

In this section, we first describe the basic encoder-decoder RNN that serves as our baseline and then propose several novel models for summarization, each addressing a specific weakness in the baseline.

2.1 Encoder-Decoder RNN with Attention and Large Vocabulary Trick

Our baseline model corresponds to the neural machine translation model used in Bahdanau et al. (2014). The encoder consists of a bidirectional GRU-RNN (Chung et al., 2014), while the decoder consists of a uni-directional GRU-RNN with the same hidden-state size as that of the encoder, and an attention mechanism over the source-hidden states and a soft-max layer over target vocabulary to generate words. In the interest of space, we refer the reader to the original paper for a detailed treatment of this model. In addition to the basic model, we also adapted to the summarization problem, the large vocabulary 'trick' (LVT) described in Jean et al. (2014). In our approach, the decoder-vocabulary of each mini-batch is restricted to words in the source documents of that batch. In addition, the most frequent words in the target dictionary are added until the vocabulary reaches a fixed size. The aim of this technique is to reduce the size of the soft-max layer of the decoder which is the main computational bottleneck. In addition, this technique also speeds up convergence by focusing the modeling effort only on the words that are essential to a given example. This technique is particularly well suited to summarization since a large proportion of the words in the summary come from the source document in any case.

2.2 Capturing Keywords using Feature-rich Encoder

In summarization, one of the key challenges is to identify the key concepts and key entities in the document, around which the story revolves. In order to accomplish this goal, we may need to go beyond the word-embeddings-based representation of the input document and capture additional linguistic features such as parts-of-speech tags, named-entity tags, and TF and IDF statistics of the words. We therefore create additional look-up based embedding matrices for the vocabulary of each tag-type, similar to the embeddings for words. For continuous features such as TF and IDF, we convert them into categorical values by discretizing them into a fixed number of bins, and use one-hot representations to indicate the bin number they fall into. This allows us to map them into an embeddings matrix like any other tag-type. Finally, for each word in the source document, we simply look-up its embeddings from all of its associated tags and concatenate them into a single long vector, as shown in Fig. 1. On the target side, we continue to use only word-based embeddings as the representation.
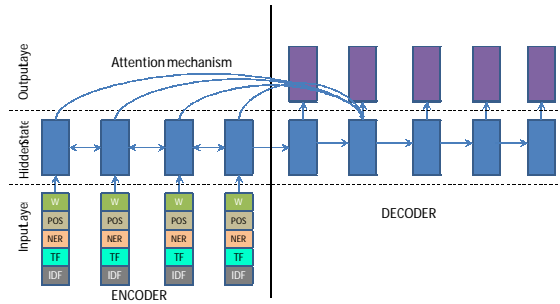

Figure 1: Feature-rich-encoder: We use one embedding vector each for POS, NER tags and discretized TF and IDF values, which are concatenated together with word-based embeddings as input to the encoder.

2.3 Modeling Rare/Unseen Words using Switching Generator-Pointer

Often-times in summarization, the keywords or named-entities in a test document that are central to the summary may actually be unseen or rare with respect to training data. Since the vocabulary of the decoder is fixed at training time, it cannot emit these unseen words. Instead, a most common way of handling these out-of-vocabulary (OOV) words is to emit an 'UNK' token as a placeholder. However this does not result in legible summaries. In summarization, an intuitive way to handle such OOV words is to simply point to their location in the source document instead. We model this no-

tion using our novel switching decoder/pointer architecture which is graphically represented in Figure 2. In this model, the decoder is equipped with a 'switch' that decides between using the generator or a pointer at every time-step. If the switch is turned on, the decoder produces a word from its target vocabulary in the normal fashion. However, if the switch is turned off, the decoder instead generates a pointer to one of the word-positions in the source. The word at the pointer-location is then copied into the summary. The switch is modeled as a sigmoid activation function over a linear layer based on the entire available context at each time-step as shown below.

$$
\begin{aligned}
P(s_i = 1) &= \sigma(\mathbf{v}^s \cdot (\mathbf{W}_h^s \mathbf{h}_i + \mathbf{W}_e^s \mathbf{E}[o_{i-1}] \\
&+ \mathbf{W}_c^s \mathbf{c}_i + \mathbf{b}^s)),
\end{aligned}
$$

where $P(s_i = 1)$ is the probability of the switch turning on at the ith time-step of the decoder, $\mathbf{h}_i$ is the hidden state, $\mathbf{E}[o_{i-1}]$ is the embedding vector of the emission from the previous time step, $\mathbf{c_i}$ is the attention-weighted context vector, and $\mathbf{W}_h^s, \mathbf{W}_e^s, \mathbf{W}_c^s, \mathbf{b}^s$ and $\mathbf{v}^s$ are the switch parameters. We use attention distribution over word positions in the document as the distribution to sample the pointer from.

$$
\begin{aligned}
P_i^a(j) &\propto \exp(\mathbf{v}^a \cdot (\mathbf{W}_h^a \mathbf{h}_{i-1} + \mathbf{W}_e^a \mathbf{E}[o_{i-1}] \\
&+ \mathbf{W}_c^a \mathbf{h}_j^d + \mathbf{b}^a)), \\
p_i &= \arg\max_j (P_i^a(j)) \text{ for } j \in \{1, \ldots, N_d\}.
\end{aligned}
$$

In the above equation, p_i is the pointer value at i^{th} word-position in the summary, sampled from the attention distribution $\mathbf{P}_i^a$ over the document word-positions $j \in \{1, \ldots, N_d\}$, where $P_i^a(j)$ is the probability of the ith time-step in the decoder pointing to the jth position in the document, and $\mathbf{h}_j^d$ is the encoder's hidden state at position j.

At training time, we provide the model with explicit pointer information whenever the summary word does not exist in the target vocabulary. When the OOV word in summary occurs in multiple document positions, we break the tie in favor of its first occurrence. At training time, we optimize the conditional log-likelihood shown below, with additional regularization penalties.

$$
\begin{aligned}
\log P(\mathbf{y}|\mathbf{x}) &= \sum_i (g_i \log\{P(y_i|\mathbf{y}_{-i}, \mathbf{x})P(s_i)\} \\
&+ (1 - g_i) \log\{P(p(i)|\mathbf{y}_{-i}, \mathbf{x})(1 - P(s_i))\})
\end{aligned}
$$

where $\mathbf{y}$ and $\mathbf{x}$ are the summary and document words respectively, g_i is an indicator function that

is set to 0 whenever the word at position i in the summary is OOV with respect to the decoder vocabulary. At test time, the model decides automatically at each time-step whether to generate or to point, based on the estimated switch probability $P(s_i)$. We simply use the $\arg\max$ of the posterior probability of generation or pointing to generate the best output at each time step.

The pointer mechanism may be more robust in handling rare words because it uses the encoder's hidden-state representation of rare words to decide which word from the document to point to. Since the hidden state depends on the entire context of the word, the model is able to accurately point to unseen words although they do not appear in the target vocabulary.[1]
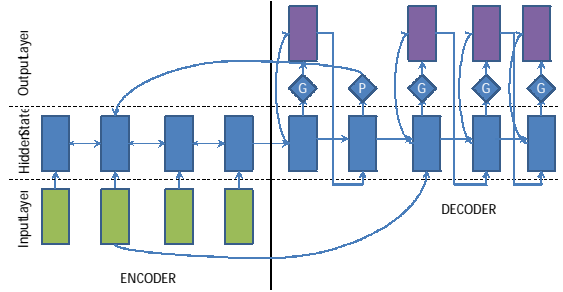

Figure 2: Switching generator/pointer model: When the switch shows 'G', the traditional generator consisting of the softmax layer is used to produce a word, and when it shows 'P', the pointer network is activated to copy the word from one of the source document positions. When the pointer is activated, the embedding from the source is used as input for the next time-step as shown by the arrow from the encoder to the decoder at the bottom.

2.4 Capturing Hierarchical Document Structure with Hierarchical Attention

In datasets where the source document is very long, in addition to identifying the keywords in the document, it is also important to identify the key sentences from which the summary can be drawn. This model aims to capture this notion of two levels of importance using two bi-directional

---

[1] Even when the word does not exist in the source vocabulary, the pointer model may still be able to identify the correct position of the word in the source since it takes into account the contextual representation of the corresponding 'UNK' token encoded by the RNN. Once the position is known, the corresponding token from the source document can be displayed in the summary even when it is not part of the training vocabulary either on the source side or the target side.

RNNs on the source side, one at the word level and the other at the sentence level. The attention mechanism operates at both levels simultaneously. The word-level attention is further re-weighted by the corresponding sentence-level attention and re-normalized as shown below:

$$P^a(j) = \frac{P^a_w(j)P^a_s(s(j))}{\sum_{k=1}^{N_d} P^a_w(k)P^a_s(s(k))},$$

where $P^a_w(j)$ is the word-level attention weight at j^{th} position of the source document, and $s(j)$ is the ID of the sentence at j^{th} word position, $P^a_s(l)$ is the sentence-level attention weight for the lth sentence in the source, N_d is the number of words in the source document, and $P^a(j)$ is the re-scaled attention at the jth word position. The re-scaled attention is then used to compute the attention-weighted context vector that goes as input to the hidden state of the decoder. Further, we also concatenate additional positional embeddings to the hidden state of the sentence-level RNN to model positional importance of sentences in the document. This architecture therefore models key sentences as well as keywords within those sentences jointly. A graphical representation of this model is displayed in Figure 3.
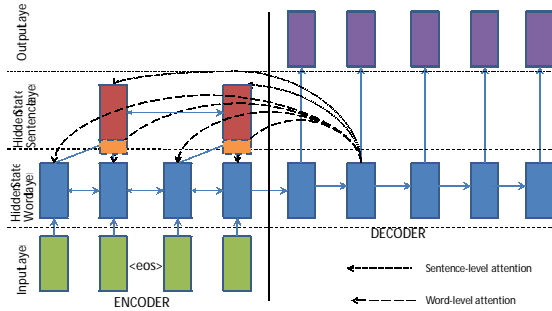

Figure 3: Hierarchical encoder with hierarchical attention: the attention weights at the word level, represented by the dashed arrows are re-scaled by the corresponding sentence-level attention weights, represented by the dotted arrows. The dashed boxes at the bottom of the top layer RNN represent sentence-level positional embeddings concatenated to the corresponding hidden states.

3 Related Work

A vast majority of past work in summarization has been extractive, which consists of identifying key sentences or passages in the source document and reproducing them as summary (Neto et

al., 2002; Erkan and Radev, 2004; Wong et al., 2008a; Filippova and Altun, 2013; Colmenares et al., 2015; Litvak and Last, 2008; K. Riedhammer and Hakkani-Tur, 2010; Ricardo Ribeiro, 2013).

Humans on the other hand, tend to paraphrase the original story in their own words. As such, human summaries are abstractive in nature and seldom consist of reproduction of original sentences from the document. The task of abstractive summarization has been standardized using the DUC-2003 and DUC-2004 competitions.[2] The data for these tasks consists of news stories from various topics with multiple reference summaries per story generated by humans. The best performing system on the DUC-2004 task, called TOPIARY (Zajic et al., 2004), used a combination of linguistically motivated compression techniques, and an unsupervised topic detection algorithm that appends keywords extracted from the article onto the compressed output. Some of the other notable work in the task of abstractive summarization includes using traditional phrase-table based machine translation approaches (Banko et al., 2000), compression using weighted tree-transformation rules (Cohn and Lapata, 2008) and quasi-synchronous grammar approaches (Woodsend et al., 2010).

With the emergence of deep learning as a viable alternative for many NLP tasks (Collobert et al., 2011), researchers have started considering this framework as an attractive, fully data-driven alternative to abstractive summarization. In Rush et al. (2015), the authors use convolutional models to encode the source, and a context-sensitive attentional feed-forward neural network to generate the summary, producing state-of-the-art results on Gigaword and DUC datasets. In an extension to this work, Chopra et al. (2016) used a similar convolutional model for the encoder, but replaced the decoder with an RNN, producing further improvement in performance on both datasets. In another paper that is closely related to our work, Hu et al. (2015) introduce a large dataset for Chinese short text summarization. They show promising results on their Chinese dataset using an encoder-decoder RNN, but do not report experiments on English corpora. In another very recent work, Cheng and Lapata (2016) used RNN based encoder-decoder for extractive summarization of documents.

Our work starts with the same framework as (Hu et al., 2015), but we go beyond the stan-

---

[2]http://duc.nist.gov/

dard architecture and propose novel models that address critical problems in summarization. We analyze the similarities and differences of our proposed models with related work on abstractive summarization below.

Feature-rich encoder (Sec. 2.2): Linguistic features such as POS tags, and named-entities as well as TF and IDF information were used in many extractive approaches to summarization (Wong et al., 2008b), but they are novel in the context of deep learning approaches for abstractive summarization, to the best of our knowledge.

Switching generator-pointer model (Sec. 2.3): This model combines extractive and abstractive approaches to summarization in a single end-to-end framework. Rush et al. (2015) also used a combination of extractive and abstractive approaches, but their extractive model is a separate log-linear classifier with handcrafted features. Pointer networks (Vinyals et al., 2015) have also been used earlier for the problem of rare words in the context of machine translation (Luong et al., 2015), but the novel addition of switch in our model allows it to strike a balance between when to be faithful to the original source (e.g., for named entities and OOV) and when it is allowed to be creative. We believe such a process arguably mimics how human produces summaries. For a more detailed treatment of this model, and experiments on multiple tasks, please refer to the parallel work published by some of the authors of this work (Gulcehre et al., 2016).

Hierarchical attention model (Sec. 2.4): Previously proposed hierarchical encoder-decoder models use attention only at sentence-level (Li et al., 2015). The novelty of our approach lies in joint modeling of attention at both sentence and word levels, where the word-level attention is further influenced by sentence-level attention, thus capturing the notion of important sentences and important words within those sentences. Concatenation of positional embeddings with the hidden state at sentence-level is also new.

4 Experiments and Results

4.1 Gigaword Corpus

In this series of experiments[3], we used the annotated Gigaword corpus as described in Rush et al.

(2015). We used the scripts made available by the authors of this work[4] to preprocess the data, which resulted in about 3.8M training examples. The script also produces about 400K validation and test examples, but we created a randomly sampled subset of 2000 examples each for validation and testing purposes, on which we report our performance. Further, we also acquired the exact test sample used in Rush et al. (2015) to make precise comparison of our models with theirs. We also made small modifications to the script to extract not only the tokenized words, but also system-generated parts-of-speech and named-entity tags.

Training: For all the models we discuss below, we used 200 dimensional word2vec vectors (Mikolov et al., 2013) trained on the same corpus to initialize the model embeddings, but we allowed them to be updated during training. The hidden state dimension of the encoder and decoder was fixed at 400 in all our experiments. When we used only the first sentence of the document as the source, as done in Rush et al. (2015), the encoder vocabulary size was 119,505 and that of the decoder stood at 68,885. We used Adadelta (Zeiler, 2012) for training, with an initial learning rate of 0.001. We used a batch-size of 50 and randomly shuffled the training data at every epoch, while sorting every 10 batches according to their lengths to speed up training. We did not use any dropout or regularization, but applied gradient clipping. We used early stopping based on the validation set and used the best model on the validation set to report all test performance numbers. For all our models, we employ the large-vocabulary trick, where we restrict the decoder vocabulary size to 2,000[5], because it cuts down the training time per epoch by nearly three times, and helps this and all subsequent models converge in only 50%-75% of the epochs needed for the model based on full vocabulary.

Decoding: At decode-time, we used beam search of size 5 to generate the summary, and limited the size of summary to a maximum of 30 words, since this is the maximum size we noticed in the sampled validation set. We found that the average system summary length from all our models (7.8 to 8.3) agrees very closely with that of the ground truth on the validation set (about 8.7 words), without any specific tuning.

---

[3]We used Kyunghyun Cho's code (`https://github.com/kyunghyuncho/dl4mt-material`) as the starting point.

[4]https://github.com/facebook/NAMAS

[5]Larger values improved performance only marginally, but at the cost of much slower training.

Computational costs: We trained all our models on a single *Tesla K40* GPU. Most models took about 10 hours per epoch on an average except the hierarchical attention model, which took 12 hours per epoch. All models typically converged within 15 epochs using our early stopping criterion based on the validation cost. The wall-clock training time until convergence therefore varies between 6-8 days depending on the model. Generating summaries at test time is reasonably fast with a throughput of about 20 summaries per second on a single GPU, using a batch size of 1.

Evaluation metrics: In Rush et al. (2015), the authors used full-length version of Rouge recall[6] to evaluate their systems on the Gigaword corpus[7]. However, full-length recall favors longer summaries, so it may not be fair to use this metric to compare two systems that differ in summary lengths. Full-length F1 solves this problem since it can penalize longer summaries. Therefore, we use full-length F1 scores from 1, 2 and L variants of Rouge using the official script to evaluate our systems. However, in the interest of fair comparison with previous work, we also report full-length recall scores where necessary. In addition, we also report the percentage of tokens in the system summary that occur in the source (which we call 'src. copy rate' in Table 1).

We describe all our experiments and results on the Gigaword corpus below.

words-lvt2k-1sent: This is the baseline attentional encoder-decoder model with the large vocabulary trick. This model is trained only on the first sentence from the source document, as done in Rush et al. (2015).

words-lvt2k-2sent: This model is identical to the model above except for the fact that it is trained on the first two sentences from the source. On this corpus, adding the additional sentence in the source does seem to aid performance, as shown in Table 1. We also tried adding more sentences, but the performance dropped, which is probably because the latter sentences in this corpus are not pertinent to the summary.

words-lvt2k-2sent-hieratt: Since we used two sentences from source document, we trained the hierarchical attention model proposed in Sec 2.4. As shown in Table 1, this model improves perfor-

mance compared to its flatter counterpart by learning the relative importance of the first two sentences automatically.

feats-lvt2k-2sent: Here, we still train on the first two sentences, but we exploit the parts-of-speech and named-entity tags in the annotated gigaword corpus as well as TF, IDF values, to augment the input embeddings on the source side as described in Sec 2.2. In total, our embedding vector grew from the original 100 to 155, and produced incremental gains compared to its counterpart *words-lvt2k-2sent* as shown in Table 1, demonstrating the utility of syntax based features in this task.

feats-lvt2k-2sent-ptr: This is the switching generator/pointer model described in Sec. 2.3, but in addition, we also use feature-rich embeddings on the document side as in the above model. Our experiments indicate that the new model is able to achieve the best performance on our test set by all three Rouge variants as shown in Table 1.

Comparison with state-of-the-art: (Rush et al., 2015) reported *recall-only* from *full-length* version of Rouge, but the authors kindly provided us with their F1 numbers, as well as their test sample. We compared the performance of our model *words-lvt2k-1sent* with their best system on their sample, on both Recall as well as F1, as displayed in Table 1. The reason we did not evaluate our best models here is that this test set consisted of only 1 sentence from the source document, and did not include NLP annotations, which are needed in our best models. The table shows that, despite this fact, our model outperforms the state of the art model of Rush et al. (2015), on both recall and F1, with statistical significance. In addition, our models exhibit better abstractive ability as shown by the *src. copy rate* metric in the last column of the table.

We believe the bidirectional RNN we used to model the source captures richer contextual information of every word than the bag-of-embeddings representation used by Rush et al. (2015) in their convolutional and attentional encoders, which might explain our superior performance. Further, explicit modeling of important information such as multiple source sentences, word-level linguistic features, using the switch mechanism to point to source words when needed, and hierarchical attention, solve specific problems in summarization, each boosting performance incrementally.

---
[6] http://www.beroure.com/Pages/default.aspx

[7] confirmed from personal communication with the first-author of the paper.

#	Model name	Rouge-1	Rouge-2	Rouge-L	Src. copy rate (%)
	Full length F1 on our internal test set				
1	words-lvt2k-1sent	34.97	17.17	32.70	75.85
2	words-lvt2k-2sent	35.73	17.38	33.25	79.54
3	words-lvt2k-2sent-hieratt	36.05	18.17	33.52	78.52
4	feats-lvt2k-2sent	35.90	17.57	33.38	78.92
5	feats-lvt2k-2sent-ptr	*36.40	17.77	*33.71	78.70
	Full length Recall on the test set used by (Rush et al., 2015)				
6	ABS+ (Rush et al., 2015)	31.47	12.73	28.54	91.50
7	words-lvt2k-1sent	*34.19	*16.29	*32.13	74.57
	Full length F1 on the test set used by (Rush et al., 2015)				
8	ABS+ (Rush et al., 2015)	29.78	11.89	26.97	91.50
9	words-lvt2k-1sent	*32.67	*15.59	*30.64	74.57

Table 1: Performance comparison of various models. '*' indicates statistical significance of the corresponding model with respect to the baseline model on its dataset as given by the 95% confidence interval in the official Rouge script. We report statistical significance only for the best performing models. 'src. copy rate' for the reference data on our validation sample is 45%. Please refer to Section 4 for explanation of notation.

4.2 DUC Corpus

The DUC corpus[8] comes in two parts: the 2003 corpus consisting of 624 document, summary pairs and the 2004 corpus consisting of 500 pairs. Since these corpora are too small to train large neural networks on, Rush et al. (2015) trained their models on the Gigaword corpus, but combined it with an additional log-linear extractive summarization model with handcrafted features, that is trained on the DUC 2003 corpus. They call the original neural attention model the ABS model, and the combined model ABS+. The latter model is current state-of-the-art since it outperforms all previously published baselines including non-neural network based extractive and abstractive systems, as measured by the official DUC metric of limited-length recall. In these experiments, we use the same metric to evaluate our models too, but we omit reporting numbers from other systems in the interest of space.

In our work, we simply run the model trained on Gigaword corpus as it is, without tuning it on the DUC validation set. The only change we made to the decoder is to suppress the model from emitting the end-of-summary tag, and force it to emit exactly 30 words for every summary, since the official evaluation on this corpus is based on limited-length Rouge recall. On this corpus too, since we have only a single sentence from source and no NLP annotations, we ran just the model *words-lvt2k-1sent*.

The performance of this model on the test set is compared with ABS and ABS+ models, as well as TOPIARY, the top performing system on DUC-2004 in Table 2. We note that although our model

consistently outperforms ABS+ on all three variants of Rouge, the differences are not statistically significant. However, when the comparison is made with ABS model, which is really the true un-tuned counterpart of our model, the results are indeed statistically significant.

Model	Rouge-1	Rouge-2	Rouge-L
TOPIARY	25.12	6.46	20.12
ABS	26.55	7.06	22.05
ABS+	28.18	8.49	23.81
words-lvt2k-1sent	**28.35**	**9.46**	**24.59**

Table 2: Evaluation of our models using the limited-length Rouge Recall on DUC validation and test sets. Our best model, although trained exclusively on the Gigaword corpus, consistently outperforms the ABS+ model which is tuned on the DUC-2003 validation corpus in addition to being trained on the Gigaword corpus.

We would also like to bring the reader's attention to the concurrently published work of Chopra et al. (2016) where they also used an RNN based decoder for summary generation. While their numbers on Gigaword corpus are slightly better than our best performance on all three Rouge F1 metrics, our performance is marginally higher on DUC-2004 corpus on Rouge-2 and Rouge-L. We believe their work also confirms the effectiveness of RNN-based models for abstractive text summarization.

4.3 CNN/Daily Mail Corpus

The existing abstractive text summarization corpora including Gigaword and DUC consist of only one sentence in each summary. In this section, we present a new corpus that comprises multi-sentence summaries. To produce this corpus, we modify an existing corpus that has been used

---
[8] *http://duc.nist.gov/duc2004/tasks.html*

Model	Rouge-1	Rouge-2	Rouge-L
words-lvt2k	**32.49**	**11.84**	**29.47**
words-lvt2k-ptr	32.12	11.72	29.16
words-lvt2k-hieratt	31.78	11.56	28.73

Table 3: Performance of various models on CNN/Daily Mail test set using full-length Rouge-F1 metric. Bold faced numbers indicate best performing system.

for the task of passage-based question answering (Hermann et al., 2015). In this work, the authors used the human generated abstractive summary bullets from new-stories in *CNN* and *Daily Mail* websites as questions (with one of the entities hidden), and stories as the corresponding passages from which the system is expected to answer the fill-in-the-blank question. The authors released the scripts that crawl, extract and generate pairs of passages and questions from these websites. With a simple modification of the script, we restored all the summary bullets of each story in the original order to obtain a multi-sentence summary, where each bullet is treated as a sentence. In all, this corpus has 286,817 training pairs, 13,368 validation pairs and 11,487 test pairs, as defined by their scripts. The source documents in the training set have 766 words spanning 29.74 sentences on an average while the summaries consist of 53 words and 3.72 sentences. The unique characteristics of this dataset such as long documents, and ordered multi-sentence summaries present interesting challenges, and we hope will attract future researchers to build and test novel models on it.

The dataset is released in two versions: one consisting of actual entity names, and the other, in which entity occurrences are replaced with document-specific integer-ids beginning from 0. Since the vocabulary size is smaller in the anonymized version, we used it in all our experiments below. We limited the source vocabulary size to 150K, and the target vocabulary to 60K, the source and target lengths to at most 800 and 100 words respectively. We used 100-dimensional word2vec embeddings trained on this dataset as input, and we fixed the model hidden state size at 200. We also created explicit pointers in the training data by matching only the anonymized entity-ids between source and target on similar lines as we did for the OOV words in Gigaword corpus.

Computational costs: We used a single Tesla K-40 GPU to train our models on this dataset as well. While the flat models (*words-lvt2k* and *words-lvt2k-ptr*) took under 5 hours per epoch, the hier-

archical attention model was very expensive, consuming nearly 12.5 hours per epoch. Convergence of all models is also slower on this dataset compared to Gigaword, taking nearly 35 epochs for all models. Thus, the wall-clock time for training until convergence is about 7 days for the flat models, but nearly 18 days for the hierarchical attention model. Decoding is also slower as well, with a throughput of 2 examples per second for flat models and 1.5 examples per second for the hierarchical attention model, when run on a single GPU with a batch size of 1.

Evaluation: We evaluated our models using the full-length Rouge F1 metric that we employed for the Gigaword corpus, but with one notable difference: in both system and gold summaries, we considered each highlight to be a separate sentence.[9]

Results: Results from three models we ran on this corpus are displayed in Table 3. Although this dataset is smaller and more complex than the Gigaword corpus, it is interesting to note that the Rouge numbers are in the same range. However, our switching pointer/generator model as well as the hierarchical attention model described in Sec. 2.4 fail to outperform the baseline attentional decoder, indicating that further research and experimentation needs to be done on this dataset. These results, although preliminary, should serve as a good baseline for future researchers to compare their models against.

5 Qualitative Analysis

Table 4 presents a few high quality and poor quality output on the validation set from *feats-lvt2k-2sent*, one of our best performing models. Even when the model differs from the target summary, its summaries tend to be very meaningful and relevant, a phenomenon not captured by word/phrase matching evaluation metrics such as Rouge. On the other hand, the model sometimes 'misinterprets' the semantics of the text and generates a summary with a comical interpretation as shown in the poor quality examples in the table. Clearly, capturing the 'meaning' of complex sentences remains a weakness of these models.

Our next example output, presented in Figure 4, displays the sample output from the switching generator/pointer model on the Gigaword corpus.

---

[9]This was done by modifying the pre-processing script such that each highlight gets its own "<a>" tag in the xml file that goes as input to the evaluation script.

Good quality summary output
S: a man charged with the murder last year of a british back-packer confessed to the slaying on the night he was charged with her killing , according to police evidence presented at a court hearing tuesday . ian douglas previte , ## , is charged with murdering caroline stuttle , ## , of yorkshire , england **T**: man charged with british backpacker 's death confessed to crime police officer claims **O**: man charged with murdering british backpacker confessed to murder
S: following are the leading scorers in the english premier league after saturday 's matches : ## - alan shearer -lrb- newcastle united -rrb- , james beattie . **T**: leading scorers in english premier league **O**: english premier league leading scorers
S: volume of transactions at the nigerian stock exchange has continued its decline since last week , a nse official said thursday . the latest statistics showed that a total of ##.### million shares valued at ###.### million naira -lrb- about #.### million us dollars -rrb- were traded on wednesday in , deals . **T**: transactions dip at nigerian stock exchange **O**: transactions at nigerian stock exchange down
Poor quality summary output
S: broccoli and broccoli sprouts contain a chemical that kills the bacteria responsible for most stomach cancer , say researchers , confirming the dietary advice that moms have been handing out for years . in laboratory tests the chemical , <unk> , killed helicobacter pylori , a bacteria that causes stomach ulcers and often fatal stomach cancers . **T**: for release at #### <unk> mom was right broccoli is good for you say cancer researchers **O**: broccoli sprouts contain deadly bacteria
S: norway delivered a diplomatic protest to russia on monday after three norwegian fisheries research expeditions were barred from russian waters . the norwegian research ships were to continue an annual program of charting fish resources shared by the two countries in the barents sea region . **T**: norway protests russia barring fisheries research ships **O**: norway grants diplomatic protest to russia
S: j.p. morgan chase 's ability to recover from a slew of recent losses rests largely in the hands of two men , who are both looking to restore tarnished reputations and may be considered for the top job someday . geoffrey <unk> , now the co-head of j.p. morgan 's investment bank , left goldman , sachs & co. more than a decade ago after executives say he lost out in a bid to lead that firm . **T**: # executives to lead j.p. morgan chase on road to recovery **O**: j.p. morgan chase may be considered for top job

Table 4: Examples of generated summaries from our best model on the validation set of Gigaword corpus. **S**: source document, **T**: target summary, **O**: system output. Although we displayed equal number of good quality and poor quality summaries in the table, the good ones are far more prevalent than the poor ones.

china 's tang **gonghong** set a world record with a clean and jerk lift of ### kilograms to win the women 's over-## kilogram weightlifting title at the asian games on tuesday .

china 's tang gonghong wins women 's weightlifting weightlifting title at asian games

owing to criticism , nbc said on wednesday that it was ending a **three-month-old** experiment that would have brought the first liquor advertisements onto national broadcast network television .

nbc says it is ending a three-month-old experiment

Figure 4: Sample output from switching generator/pointer networks. An arrow indicates that a pointer to the source position was used to generate the corresponding summary word.

It is apparent from the examples that the model learns to use pointers very accurately not only for named entities, but also for multi-word phrases. Despite its accuracy, the performance improvement of the overall model is not significant. We believe the impact of this model may be more pronounced in other settings with a heavier tail distribution of rare words. We intend to carry out more experiments with this model in the future.

On CNN/Daily Mail data, although our models are able to produce good quality multi-sentence summaries, we notice that the same sentence or phrase often gets repeated in the summary. We believe models that incorporate intra-attention such as Cheng et al. (2016) can fix this problem by encouraging the model to 'remember' the words it has already produced in the past.

6 Conclusion

In this work, we apply the attentional encoder-decoder for the task of abstractive summarization with very promising results, outperforming state-of-the-art results significantly on two different datasets. Each of our proposed novel models addresses a specific problem in abstractive summarization, yielding further improvement in performance. We also propose a new dataset for multi-sentence summarization and establish benchmark numbers on it. As part of our future work, we plan to focus our efforts on this data and build more robust models for summaries consisting of multiple sentences.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. 2015. End-to-end attention-based large vocabulary speech recognition. *CoRR*, abs/1508.04395.

Michele Banko, Vibhu O. Mittal, and Michael J Witbrock. 2000. Headline generation based on statistical translation. *In Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, 22:318–325.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. *CoRR*, abs/1601.06733.

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

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555.

Trevor Cohn and Mirella Lapata. 2008. Sentence compression beyond word deletion. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, pages 137–144.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *CoRR*, abs/1103.0398.

Carlos A. Colmenares, Marina Litvak, Amin Mantrach, and Fabrizio Silvestri. 2015. Heads: Headline generation as sequence prediction using an abstract feature-rich space. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 133–142.

G. Erkan and D. R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Katja Filippova and Yasemin Altun. 2013. Overcoming the lack of parallel data in sentence compression. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1481–1491.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *CoRR*, abs/1506.03340.

Baotian Hu, Qingcai Chen, and Fangze Zhu. 2015. Lcsts: A large scale chinese short text summarization dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1967–1972, Lisbon, Portugal, September. Association for Computational Linguistics.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2014. On using very large target vocabulary for neural machine translation. *CoRR*, abs/1412.2007.

B. Favre K. Riedhammer and D. Hakkani-Tur. 2010. Long story short âĂŞ global unsupervised models for keyphrase based meeting summarization. In *Speech Communication*, pages 801–815.

Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. 2015. A hierarchical neural autoencoder for paragraphs and documents. *CoRR*, abs/1506.01057.

M. Litvak and M. Last. 2008. Graph-based keyword extraction for single-document summarization. In *Coling 2008*, pages 17–24.

Thang Luong, Ilya Sutskever, Quoc V. Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, pages 11–19.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546.

Joel Larocca Neto, Alex Alves Freitas, and Celso A. A. Kaestner. 2002. Automatic text summarization using a machine learning approach. In *Proceedings of the 16th Brazilian Symposium on Artificial Intelligence: Advances in Artificial Intelligence*, pages 205–215.

David Martins de Matos JoÃčo P. Neto Anatole Gershman Jaime Carbonell Ricardo Ribeiro, LuÃ■s Marujo. 2013. Self reinforcement for important passage retrieval. In *36th international ACM SIGIR conference on Research and development in information retrieval*, pages 845–848.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *CoRR*, abs/1509.00685.

Subhashini Venugopalan, Marcus Rohrbach, Jeff Donahue, Raymond J. Mooney, Trevor Darrell, and Kate Saenko. 2015. Sequence to sequence - video to text. *CoRR*, abs/1505.00487.

O. Vinyals, M. Fortunato, and N. Jaitly. 2015. Pointer Networks. *ArXiv e-prints*, June.

Kam-Fai Wong, Mingli Wu, and Wenjie Li. 2008a. Extractive summarization using supervised and semi-supervised learning. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, pages 985–992.

Kam-Fai Wong, Mingli Wu, and Wenjie Li. 2008b. Extractive summarization using supervised and semi-supervised learning. In *Proceedings of the 22nd Annual Meeting of the Association for Computational Linguistics*, pages 985–992.

Kristian Woodsend, Yansong Feng, and Mirella Lapata. 2010. Title generation with quasi-synchronous grammar. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 513–523, Stroudsburg, PA, USA. Association for Computational Linguistics.

David Zajic, Bonnie J. Dorr, and Richard Schwartz. 2004. Bbn/umd at duc-2004: Topiary. In *Proceedings of the North American Chapter of the Association for Computational Linguistics Workshop on Document Understanding*, pages 112–119.

Matthew D. Zeiler. 2012. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701.