Abstractive Sentence Summarization with Attentive Recurrent Neural Networks

Sumit Chopra
Facebook AI Research
spchopra@fb.com

Michael Auli
Facebook AI Research
michaelauli@fb.com

Alexander M. Rush
Harvard SEAS
srush@seas.harvard.edu

Abstract

Abstractive Sentence Summarization generates a shorter version of a given sentence while attempting to preserve its meaning. We introduce a conditional recurrent neural network (RNN) which generates a summary of an input sentence. The conditioning is provided by a novel convolutional attention-based encoder which ensures that the decoder focuses on the appropriate input words at each step of generation. Our model relies only on learned features and is easy to train in an end-to-end fashion on large data sets. Our experiments show that the model significantly outperforms the recently proposed state-of-the-art method on the Gigaword corpus while performing competitively on the DUC-2004 shared task.

1 Introduction

Generating a condensed version of a passage while preserving its meaning is known as text summarization. Tackling this task is an important step towards natural language understanding. Summarization systems can be broadly classified into two categories. *Extractive models* generate summaries by cropping important segments from the original text and putting them together to form a coherent summary. *Abstractive models* generate summaries from scratch without being constrained to reuse phrases from the original text.

In this paper we propose a novel recurrent neural network for the problem of *abstractive sentence summarization*. Inspired by the recently proposed architectures for machine translation (Bahdanau et al., 2014), our model consists of a conditional recurrent neural network, which acts as a decoder to generate the summary of an input sentence, much like a standard recurrent language model. In addition, at every time step the decoder also takes a conditioning input which is the output of an encoder module. Depending on the current state of the RNN, the encoder computes scores over the words in the input sentence. These scores can be interpreted as a soft alignment over the input text, informing the decoder which part of the input sentence it should focus on to generate the next word. Both the decoder and encoder are jointly trained on a data set consisting of sentence-summary pairs. Our model can be seen as an extension of the recently proposed model for the same problem by Rush et al. (2015). While they use a feed-forward neural language model for generation, we use a recurrent neural network. Furthermore, our encoder is more sophisticated, in that it explicitly encodes the position information of the input words. Lastly, our encoder uses a convolutional network to encode input words. These extensions result in improved performance.

The main contribution of this paper is a novel convolutional attention-based conditional recurrent neural network model for the problem of abstractive sentence summarization. Empirically we show that our model beats the state-of-the-art systems of Rush et al. (2015) on multiple data sets. Particularly notable is the fact that even with a simple generation module, which does not use any extractive feature tuning, our model manages to significantly outperform their ABS+ system on the Gigaword data set and is comparable on the DUC-2004 task.

2 Previous Work

While there is a large body of work for generating extractive summaries of sentences (Jing, 2000; Knight and Marcu, 2002; McDonald, 2006; Clarke and Lapata, 2008; Filippova and Altun, 2013; Filippova et al., 2015), there has been much less research on abstractive summarization. A count-based noisy-channel machine translation model was proposed for the problem in Banko et al. (2000). The task of abstractive sentence summarization was later formalized around the DUC-2003 and DUC-2004 competitions (Over et al., 2007), where the TOPIARY system (Zajic et al., 2004) was the state-of-the-art. More recently Cohn and Lapata (2008) and later Woodsend et al. (2010) proposed systems which made heavy use of the syntactic features of the sentence-summary pairs. Later, along the lines of Banko et al. (2000), MOSES was used directly as a method for text simplification by Wubben et al. (2012). Other works which have recently been proposed for the problem of sentence summarization include (Galanis and Androutsopoulos, 2010; Napoles et al., 2011; Cohn and Lapata, 2013). Very recently Rush et al. (2015) proposed a neural attention model for this problem using a new data set for training and showing state-of-the-art performance on the DUC tasks. Our model can be seen as an extension of their model.

3 Attentive Recurrent Architecture

Let $\mathbf{x}$ denote the input sentence consisting of a sequence of M words $\mathbf{x} = [x_1, \ldots, x_M]$, where each word x_i is part of vocabulary $\mathcal{V}$, of size $|\mathcal{V}| = V$. Our task is to generate a target sequence $\mathbf{y} = [y_1, \ldots, y_N]$, of N words, where $N < M$, such that the meaning of $\mathbf{x}$ is preserved: $\mathbf{y} = \text{argmax}_y\, P(y|\mathbf{x})$, where y is a random variable denoting a sequence of N words.

Typically the conditional probability is modeled by a parametric function with parameters θ: $P(y|\mathbf{x}) = P(y|\mathbf{x};\theta)$. Training involves finding the θ which maximizes the conditional probability of sentence-summary pairs in the training corpus. If the model is trained to generate the next word of the summary, given the previous words, then the above conditional can be factorized into a product of individual conditional probabilities:

$$P(\mathbf{y}|\mathbf{x};\theta) = \prod_{t=1}^{N} p(y_t|\{y_1, \ldots, y_{t-1}\}, \mathbf{x}; \theta). \quad (1)$$

In this work we model this conditional probability using an RNN Encoder-Decoder architecture, inspired by Cho et al. (2014) and subsequently extended in Bahdanau et al. (2014). We call our model RAS (Recurrent Attentive Summarizer).

3.1 Recurrent Decoder

The above conditional is modeled using an RNN:

$$P(y_t|\{y_1, \ldots, y_{t-1}\}, \mathbf{x}; \theta) = P_t = g_{\theta_1}(h_t, c_t),$$

where h_t is the hidden state of the RNN:

$$h_t = g_{\theta_1}(y_{t-1}, h_{t-1}, c_t).$$

Here c_t is the output of the encoder module (detailed in §3.2). It can be seen as a context vector which is computed as a function of the current state h_{t-1} and the input sequence $\mathbf{x}$.

Our Elman RNN takes the following form (Elman, 1990):

$$\begin{aligned} h_t &= \sigma(W_1 y_{t-1} + W_2 h_{t-1} + W_3 c_t) \\ P_t &= \rho(W_4 h_t + W_5 c_t), \end{aligned}$$

where σ is the sigmoid function and ρ is the softmax, defined as: $\rho(o_t) = e^{o_t} / \sum_j e^{o_j}$ and W_i $(i = 1, \ldots, 5)$ are matrices of learnable parameters of sizes $W_{\{1,2,3\}} \in \mathcal{R}^{d \times d}$ and $W_{\{4,5\}} \in \mathcal{R}^{d \times V}$.

The LSTM decoder is defined as (Hochreiter and Schmidhuber, 1997):

$$\begin{aligned} i_t &= \sigma(W_1 y_{t-1} + W_2 h_{t-1} + W_3 c_t) \\ i'_t &= \tanh(W_4 y_{t-1} + W_5 h_{t-1} + W_6 c_t) \\ f_t &= \sigma(W_7 y_{t-1} + W_8 h_{t-1} + W_9 c_t) \\ o_t &= \sigma(W_{10} y_{t-1} + W_{11} h_{t-1} + W_{12} c_t) \\ m_t &= m_{t-1} \odot f_t + i_t \odot i'_t \\ h_t &= m_t \odot o_t \\ P_t &= \rho(W_{13} h_t + W_{14} c_t). \end{aligned}$$

Operator $\odot$ refers to component-wise multiplication, and W_i $(i = 1, \ldots, 14)$ are matrices of learnable parameters of sizes $W_{\{1,\ldots,12\}} \in \mathcal{R}^{d \times d}$, and $W_{\{13,14\}} \in \mathcal{R}^{d \times V}$.

3.2 Attentive Encoder

We now give the details of the encoder which computes the context vector c_t for every time step t of the decoder above. With a slight overload of notation, for an input sentence $\mathbf{x}$ we denote by x_i the d dimensional learnable embedding of the i-th word ($x_i \in \mathcal{R}^d$). In addition the position i of the word x_i is also associated with a learnable embedding l_i of size d ($l_i \in \mathcal{R}^d$). Then the full embedding for i-th word in $\mathbf{x}$ is given by $a_i = x_i + l_i$. Let us denote by $B^k \in \mathcal{R}^{q \times d}$ a learnable weight matrix which is used to convolve over the full embeddings of consecutive words. Let there be d such matrices ($k \in \{1, \ldots, d\}$). The output of convolution is given by:

$$z_{ik} = \sum_{h=-q/2}^{q/2} a_{i+h} \cdot b^k_{q/2+h}, \tag{2}$$

where b^k_j is the j-th column of the matrix B^k. Thus the d dimensional aggregate embedding vector z_i is defined as $z_i = [z_{i1}, \ldots, z_{id}]$. Note that each word x_i in the input sequence is associated with one aggregate embedding vector z_i. The vectors z_i can be seen as a representation of the word which captures the position in which it occurs in the sentence and also the context in which it appears in the sentence. In our experiments the width q of the convolution matrix B^k was set to 5. To account for words at the boundaries of $\mathbf{x}$ we first pad the sequence on both sides with dummy words before computing the aggregate vectors z_i's.

Given these aggregate vectors of words, we compute the context vector c_t (the encoder output) as:

$$c_t = \sum_{j=1}^{M} \alpha_{j,t-1} x_j, \tag{3}$$

where the weights $\alpha_{j,t-1}$ are computed as

$$\alpha_{j,t-1} = \frac{\exp(z_j \cdot h_{t-1})}{\sum_{i=1}^{M} \exp(z_i \cdot h_{t-1})}. \tag{4}$$

3.3 Training and Generation

Given a training corpus $\mathcal{S} = \{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^{S}$ of S sentence-summary pairs, the above model can be trained end-to-end using stochastic gradient descent by minimizing the negative conditional log likelihood of the training data with respect to θ:

$$\mathcal{L} = -\sum_{i=1}^{S} \sum_{t=1}^{N} \log P(y^i_t | \{y^i_1, \ldots, y^i_{t-1}\}, \mathbf{x}^i; \theta), \tag{5}$$

where the parameters θ constitute the parameters of the decoder and the encoder.

Once the parametric model is trained we generate a summary for a new sentence $\mathbf{x}$ through a word-based beam search such that $P(y|\mathbf{x})$ is maximized, $\operatorname{argmax} P(y_t | \{y_1, \ldots, y_{t-1}\}, \mathbf{x})$. The search is parameterized by the number of paths k that are pursued at each time step.

4 Experimental Setup

4.1 Datasets and Evaluation

Our models are trained on the annotated version of the Gigaword corpus (Graff et al., 2003; Napoles et al., 2012) and we use only the annotations for tokenization and sentence separation while discarding other annotations such as tags and parses. We pair the first sentence of each article with its headline to form sentence-summary pairs. The data is pre-processed in the same way as Rush et al. (2015) and we use the same splits for training, validation, and testing. For Gigaword we report results on the same randomly held-out test set of 2000 sentence-summary pairs as (Rush et al., 2015).[1] We also evaluate our models on the DUC-2004 evaluation data set comprising 500 pairs (Over et al., 2007). Our evaluation is based on three variants of ROUGE (Lin, 2004), namely, ROUGE-1 (unigrams), ROUGE-2 (bigrams), and ROUGE-L (longest-common substring).

4.2 Architectural Choices

We implemented our models in the Torch library (http://torch.ch/)[2]. To optimize our loss (Equation 5) we used stochastic gradient descent with mini-batches of size 32. During training we measure the perplexity of the summaries in the validation set and adjust our hyper-parameters, such as the learning rate, based on this number.

[1] We remove pairs with empty titles resulting in slightly different accuracy compared to Rush et al. (2015) for their systems.

[2] Our code can found at www://github.com/facebook/namas

Model	Perplexity
Bag-of-Words	43.6
Convolutional (TDNN)	35.9
Attention-based (ABS)	27.1
RAS-Elman	**18.9**
RAS-LSTM	20.3

Table 1: Perplexity on the Gigaword validation set. Bag-of-words, Convolutional (TDNN) and ABS are the different encoders of Rush et. al., 2015.

For the decoder we experimented with both the Elman RNN and the Long-Short Term Memory (LSTM) architecture (as discussed in § 3.1). We chose hyper-parameters based on a grid search and picked the one which gave the best perplexity on the validation set. In particular we searched over the number of hidden units H of the recurrent layer, the learning rate η, the learning rate annealing schedule γ (the factor by which to decrease η if the validation perplexity increases), and the gradient clipping threshold κ. Our final Elman architecture (RAS-Elman) uses a single layer with $H = 512$, $\eta = 0.5$, $\gamma = 2$, and $\kappa = 10$. The LSTM model (RAS-LSTM) also has a single layer with $H = 512$, $\eta = 0.1$, $\gamma = 2$, and $\kappa = 10$.

5 Results

On the Gigaword corpus we evaluate our models in terms of perplexity on a held-out set. We then pick the model with best perplexity on the held-out set and use it to compute the F1-score of ROUGE-1, ROUGE-2, and ROUGE-L on the test sets, all of which we report. For the DUC corpus however, inline with the standard, we report the recall-only ROUGE. As baseline we use the state-of-the-art attention-based system (ABS) of Rush et al. (2015) which relies on a feed-forward network decoder. Additionally, we compare to an enhanced version of their system (ABS+), which relies on a range of separate extractive summarization features that are added as log-linear features in a secondary learning step with minimum error rate training (Och, 2003).

Table 1 shows that both our RAS-Elman and RAS-LSTM models achieve lower perplexity than

	RG-1	RG-2	RG-L
ABS	29.55	11.32	26.42
ABS+	29.76	11.88	26.96
RAS-Elman ($k = 1$)	33.10	14.45	30.25
RAS-Elman ($k = 10$)	**33.78**	**15.97**	**31.15**
RAS-LSTM ($k = 1$)	31.71	13.63	29.31
RAS-LSTM ($k = 10$)	32.55	14.70	30.03
Luong-NMT	33.10	14.45	30.71

Table 2: F1 ROUGE scores on the Gigaword test set. ABS and ABS+ are the systems of Rush et al. 2015. k refers to the size of the beam for generation; $k = 1$ implies greedy generation. RG refers to ROUGE. Rush et al. (2015) previously reported ROUGE recall, while as we use the more balanced F-measure.

	RG-1	RG-2	RG-L
ABS	26.55	7.06	22.05
ABS+	28.18	8.49	23.81
RAS-Elman ($k = 1$)	29.13	7.62	23.92
RAS-Elman ($k = 10$)	**28.97**	**8.26**	**24.06**
RAS-LSTM ($k = 1$)	26.90	6.57	22.12
RAS-LSTM ($k = 10$)	27.41	7.69	23.06
Luong-NMT	28.55	8.79	24.43

Table 3: ROUGE results (recall-only) on the DUC-2004 test sets. ABS and ABS+ are the systems of Rush et al. 2015. k refers to the size of the beam for generation; $k = 1$ implies greedy generation. RG refers to ROUGE.

ABS as well as other models reported in Rush et al. (2015). The RAS-LSTM performs slightly worse than RAS-Elman, most likely due to over-fitting. We attribute this to the relatively simple nature of this task which can be framed as English-to-English translation with few long-term dependencies. The ROUGE results (Table 2) show that our models comfortably outperform both ABS and ABS+ by a wide margin on all metrics. This is even the case when we rely only on very fast greedy search ($k = 1$), while as ABS uses a much wider beam of size $k = 50$; the stronger ABS+ system also uses additional extractive features which our model does not. These features cause ABS+ to copy 92% of words from the input into the summary, whereas our model copies only 74% of the words leading to more abstractive summaries. On DUC-2004 we report recall ROUGE as is customary on this dataset. The results (Table 3) show that our models are better than ABS+. However the improvements are smaller than for Gi-

gaword which is likely due to two reasons: First, tokenization of DUC-2004 differs slightly from our training corpus. Second, headlines in Gigaword are much shorter than in DUC-2004.

For the sake of completeness we also compare our models to the recently proposed standard Neural Machine Translation (NMT) systems. In particular, we compare to a smaller re-implementation of the attentive stacked LSTM encoder-decoder of Luong et al. (2015). Our implementation uses two-layer LSTMs for the encoder-decoder with 500 hidden units in each layer. Tables 2 and 3 report ROUGE scores on the two data sets. From the tables we observe that the proposed RAS-Elman model is able to match the performance of the NMT model of Luong at al. (2015). This is noteworthy because RAS-Elman is significantly simpler than the NMT model at multiple levels. First, the encoder used by RAS-Elman is extremely light-weight (attention over the convolutional representation of the input words), compared to Luong's (a 2 hidden layer LSTM). Second, the decoder used by RAS-Elman is a single layer standard (Elman) RNN as opposed to a multi-layer LSTM. In an independent work, Nallapati et. al (2016) also trained a collection of standard NMT models and report numbers in the same ballpark as RAS-Elman on both datasets.

In order to better understand which component of the proposed architecture is responsible for the improvements, we trained the recurrent model with Rush et. al., (2015)'s ABS encoder on a subset of the Gigaword dataset. The ABS encoder, which does not have the position features, achieves a final validation perplexity of 38 compared to 29 for the proposed encoder, which uses position features as well as context information. This clearly shows the benefits of using the position feature in the proposed encoder.

Finally in Figure 1 we highlight anecdotal examples of summaries produced by the RAS-Elman system on the Gigaword dataset. The first two examples highlight typical improvements in the RAS model over ABS+. Generally the model produces more fluent summaries and is better able to capture the main actors of the input. For instance in Sentence 1, RAS-Elman correctly distinguishes the actions of "pepe" from "ferreira", and in Sentence 2 it identifies the correct role of the "think tank". The final two examples highlight typical mistakes of the models. In Sentence 3 both models take literally the figurative use of the idiom "a silence that spoke volumes," and produce fluent but nonsensical summaries. In Sentence 4 the RAS model mistakes the content of a relative clause for the main verb, leading to a summary with the opposite meaning of the input. These difficult cases are somewhat rare in the Gigaword, but they highlight future challenges for obtaining human-level sentence summary.

I(1): brazilian defender pepe is out for the rest of the season with a knee injury , his porto coach jesualdo ferreira said saturday .
G: football : pepe out for season
A+: ferreira out for rest of season with knee injury
R: brazilian defender pepe out for rest of season with knee injury

I(2): economic growth in toronto will suffer this year because of sars , a think tank said friday as health authorities insisted the illness was under control in canada 's largest city .
G: sars toll on toronto economy estimated at c$ # billion
A+: think tank under control in canada 's largest city
R: think tank says economic growth in toronto will suffer this year

I(3): colin l. powell said nothing – a silence that spoke volumes to many in the white house on thursday morning .
G: in meeting with former officials bush defends iraq policy
A+: colin powell speaks volumes about silence in white house
R: powell speaks volumes on the white house

I(4): an international terror suspect who had been under a controversial loose form of house arrest is on the run , british home secretary john reid said tuesday .
G: international terror suspect slips net in britain
A+: reid under house arrest terror suspect on the run
R: international terror suspect under house arrest

Figure 1: Example sentence summaries produced on Gigaword. **I** is the input, **G** is the true headline, **A** is ABS+, and **R** is RAS-ELMAN.

6 Conclusion

We extend the state-of-the-art model for abstractive sentence summarization (Rush et al., 2015) to a recurrent neural network architecture. Our model is a simplified version of the encoder-decoder framework for machine translation (Bahdanau et al., 2014). The model is trained on the Gigaword corpus to generate headlines based on the first line of each news article. We comfortably outperform the previous state-of-the-art on both Gigaword data and the DUC-2004 challenge even though our model does not rely on additional extractive features.

References

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

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*, pages 318–325. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of EMNLP 2014*, pages 1724–1734.

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

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. Association for Computational Linguistics.

Trevor Cohn and Mirella Lapata. 2013. An abstractive approach to sentence compression. *ACM Transactions on Intelligent Systems and Technology (TIST'13)*, 4,3(41).

Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive Science*, 14(2):179–211.

Katja Filippova and Yasemin Altun. 2013. Overcoming the lack of parallel data in sentence compression. In *EMNLP*, pages 1481–1491.

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

Dimitrios Galanis and Ion Androutsopoulos. 2010. An extractive supervised two-stage method for sentence compression. In *Proceedings of NAACL-HLT 2010*.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*.

S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

H Jing. 2000. Sentence reduction for automatic text summarization. In *ANLP-00*, pages 703–711.

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

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81.

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

R McDonald. 2006. Discriminative sentence compression with soft syntactic evidence. In *EACL-06*, pages 297–304.

Ramesh Nallapati, Bing Xiang, and Zhou Bowen. 2016. Sequence-to-sequence rnns for text summarization. In *http://arxiv.org/abs/1602.06023*.

Courtney Napoles, Chris Callison-Burch, Juri Ganitkevitch, and Benjamin Van Durme. 2011. Paraphratic sentence compression with a character-based metric: Tightening without deletion. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation (MTTG'11)*.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 95–100. Association for Computational Linguistics.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics.

Paul Over, Hoa Dang, and Donna Harman. 2007. Duc in context. *Information Processing & Management*, 43(6):1506–1520.

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

Kristian Woodsend, Yansong Feng, and Mirella Lapata. 2010. Generation with quasi-synchronous grammar. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 513–523. Association for Computational Linguistics.

Sander Wubben, Antal Van Den Bosch, and Emiel Krahmer. 2012. Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 1015–1024. Association for Computational Linguistics.

David Zajic, Bonnie Dorr, and Richard Schwartz. 2004. Bbn/umd at duc-2004: Topiary. In *Proceedings of the HLT-NAACL 2004 Document Understanding Workshop, Boston*, pages 112–119.