



THE UNIVERSITY *of* EDINBURGH

## Edinburgh Research Explorer

### **Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization**

**Citation for published version:**

Narayan, S, Cohen, S & Lapata, M 2018, Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, pp. 1797-1807, 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31/10/18. <<http://aclweb.org/anthology/D18-1206> >

**Link:**

[Link to publication record in Edinburgh Research Explorer](#)

**Document Version:**

Peer reviewed version

**Published In:**

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing

**General rights**

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

**Take down policy**

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact [openaccess@ed.ac.uk](mailto:openaccess@ed.ac.uk) providing details, and we will remove access to the work immediately and investigate your claim.



# Don't Give Me the Details, Just the Summary!

## Topic-Aware Convolutional Neural Networks for Extreme Summarization

Shashi Narayan Shay B. Cohen Mirella Lapata

Institute for Language, Cognition and Computation

School of Informatics, University of Edinburgh

10 Crichton Street, Edinburgh, EH8 9AB

shashi.narayan@ed.ac.uk, {scohen, mlap}@inf.ed.ac.uk

### Abstract

We introduce *extreme summarization*, a new single-document summarization task which does not favor extractive strategies and calls for an abstractive modeling approach. The idea is to create a short, one-sentence news summary answering the question “What is the article about?”. We collect a real-world, large scale dataset for this task by harvesting online articles from the British Broadcasting Corporation (BBC). We propose a novel abstractive model which is conditioned on the article’s topics and based entirely on convolutional neural networks. We demonstrate experimentally that this architecture captures long-range dependencies in a document and recognizes pertinent content, outperforming an oracle extractive system and state-of-the-art abstractive approaches when evaluated automatically and by humans.<sup>1</sup>

### 1 Introduction

Automatic summarization is one of the central problems in Natural Language Processing (NLP) posing several challenges relating to *understanding* (i.e., identifying important content) and *generation* (i.e., aggregating and rewording the identified content into a summary). Of the many summarization paradigms that have been identified over the years (see Mani, 2001 and Nenkova and McKeown, 2011 for a comprehensive overview), single-document summarization has consistently attracted attention (Cheng and Lapata, 2016; Durrett et al., 2016; Nallapati et al., 2016, 2017; See et al., 2017; Tan and Wan, 2017; Narayan et al., 2017; Fan et al., 2017; Paulus et al., 2018; Pasunuru and Bansal, 2018; Celikyilmaz et al., 2018; Narayan et al., 2018a,b).

<sup>1</sup>Our dataset, code, and demo are available at: <https://github.com/shashiongithub/XSum>.

---

**SUMMARY:** *A man and a child have been killed after a light aircraft made an emergency landing on a beach in Portugal.*

---

**DOCUMENT:** Authorities said the incident took place on Sao Joao beach in Caparica, south-west of Lisbon.

The National Maritime Authority said a middle-aged man and a young girl died after they were unable to avoid the plane.

[6 sentences with 139 words are abbreviated from here.]

Other reports said the victims had been sunbathing when the plane made its emergency landing.

[Another 4 sentences with 67 words are abbreviated from here.]

Video footage from the scene carried by local broadcasters showed a small recreational plane parked on the sand, apparently intact and surrounded by beachgoers and emergency workers.

[Last 2 sentences with 19 words are abbreviated.]

---

Figure 1: An abridged example from our extreme summarization dataset showing the document and its one-line summary. Document content present in the summary is color-coded.

Neural approaches to NLP and their ability to learn continuous features without recourse to pre-processing tools or linguistic annotations have driven the development of large-scale document summarization datasets (Sandhaus, 2008; Hermann et al., 2015; Grusky et al., 2018). However, these datasets often favor extractive models which create a summary by identifying (and subsequently concatenating) the most important sentences in a document (Cheng and Lapata, 2016; Nallapati et al., 2017; Narayan et al., 2018b). Abstractive approaches, despite being more faithful to the actual summarization task, either lag behind extractive ones or are mostly extractive, exhibiting a small degree of abstraction (See et al., 2017; Tan and Wan, 2017;

Paulus et al., 2018; Pasunuru and Bansal, 2018; Celikyilmaz et al., 2018).

In this paper we introduce *extreme summarization*, a new single-document summarization task which is not amenable to extractive strategies and requires an abstractive modeling approach. The idea is to create a short, one-sentence news summary answering the question “What is the article about?”. An example of a document and its extreme summary are shown in Figure 1. As can be seen, the summary is very different from a headline whose aim is to encourage readers to read the story; it draws on information interspersed in various parts of the document (not only the beginning) and displays multiple levels of abstraction including paraphrasing, fusion, synthesis, and inference. We build a dataset for the proposed task by harvesting online articles from the British Broadcasting Corporation (BBC) that often include a first-sentence summary.

We further propose a novel deep learning model which we argue is well-suited to the extreme summarization task. Unlike most existing abstractive approaches (Rush et al., 2015; Chen et al., 2016; Nallapati et al., 2016; See et al., 2017; Tan and Wan, 2017; Paulus et al., 2018; Pasunuru and Bansal, 2018; Celikyilmaz et al., 2018) which rely on an encoder-decoder architecture modeled by recurrent neural networks (RNNs), we present a *topic-conditioned* neural model which is based entirely on convolutional neural networks (Gehring et al., 2017b). Convolution layers capture long-range dependencies between words in the document more effectively compared to RNNs, allowing to perform document-level inference, abstraction, and paraphrasing. Our convolutional encoder associates each word with a topic vector capturing whether it is representative of the document’s content, while our convolutional decoder conditions each word prediction on a document topic vector.

Experimental results show that when evaluated automatically (in terms of ROUGE) our topic-aware convolutional model outperforms an oracle extractive system and state-of-the-art RNN-based abstractive systems. We also conduct two human evaluations in order to assess (a) which type of summary participants prefer and (b) how much key information from the document is preserved in the summary. Both evaluations overwhelmingly show that human subjects find our summaries

more informative and complete. Our contributions in this work are three-fold: a new single document summarization dataset that encourages the development of abstractive systems; corroborated by analysis and empirical results showing that extractive approaches are not well-suited to the extreme summarization task; and a novel topic-aware convolutional sequence-to-sequence model for abstractive summarization.

## 2 The XSum Dataset

Our extreme summarization dataset (which we call XSum) consists of BBC articles and accompanying single sentence summaries. Specifically, each article is prefaced with an introductory sentence (aka summary) which is professionally written, typically by the author of the article. The summary bears the HTML class “story-body\_introduction,” and can be easily identified and extracted from the main text body (see Figure 1 for an example summary-article pair).

We followed the methodology proposed in Hermann et al. (2015) to create a large-scale dataset for extreme summarization. Specifically, we collected 226,711 Wayback archived BBC articles ranging over almost a decade (2010 to 2017) and covering a wide variety of domains (e.g., News, Politics, Sports, Weather, Business, Technology, Science, Health, Family, Education, Entertainment and Arts). Each article comes with a unique identifier in its URL, which we used to randomly split the dataset into training (90%, 204,045), validation (5%, 11,332), and test (5%, 11,334) set. Table 1 compares XSum with the CNN, DailyMail, and NY Times benchmarks. As can be seen, XSum contains a substantial number of training instances, similar to DailyMail; documents and summaries in XSum are shorter in relation to other datasets but the vocabulary size is sufficiently large, comparable to CNN.

Table 2 provides empirical analysis supporting our claim that XSum is less biased toward extractive methods compared to other summarization datasets. We report the percentage of novel  $n$ -grams in the target gold summaries that do not appear in their source documents. There are 36% novel unigrams in the XSum reference summaries compared to 17% in CNN, 17% in DailyMail, and 23% in NY Times. This indicates that XSum summaries are more abstractive. The proportion of novel constructions grows for larger  $n$ -grams

Datasets	# docs (train/val/test)	avg. document length		avg. summary length		vocabulary size	
		words	sentences	words	sentences	document	summary
CNN	90,266/1,220/1,093	760.50	33.98	45.70	3.59	343,516	89,051
DailyMail	196,961/12,148/10,397	653.33	29.33	54.65	3.86	563,663	179,966
NY Times	589,284/32,736/32,739	800.04	35.55	45.54	2.44	1,399,358	294,011
XSum	204,045/11,332/11,334	431.07	19.77	23.26	1.00	399,147	81,092

Table 1: Comparison of summarization datasets with respect to overall corpus size, size of training, validation, and test set, average document (source) and summary (target) length (in terms of words and sentences), and vocabulary size on both on source and target. For CNN and DailyMail, we used the original splits of [Hermann et al. \(2015\)](#) and followed [Narayan et al. \(2018b\)](#) to preprocess them. For NY Times ([Sandhaus, 2008](#)), we used the splits and pre-processing steps of [Paulus et al. \(2018\)](#). For the vocabulary, we lowercase tokens.

Datasets	% of novel n-grams in gold summary				LEAD			EXT-ORACLE		
	unigrams	bigrams	trigrams	4-grams	R1	R2	RL	R1	R2	RL
CNN	16.75	54.33	72.42	80.37	29.15	11.13	25.95	50.38	28.55	46.58
DailyMail	17.03	53.78	72.14	80.28	40.68	18.36	37.25	55.12	30.55	51.24
NY Times	22.64	55.59	71.93	80.16	31.85	15.86	23.75	52.08	31.59	46.72
XSum	<b>35.76</b>	<b>83.45</b>	<b>95.50</b>	<b>98.49</b>	<b>16.30</b>	<b>1.61</b>	<b>11.95</b>	<b>29.79</b>	<b>8.81</b>	<b>22.65</b>

Table 2: Corpus bias towards extractive methods in the CNN, DailyMail, NY Times, and XSum datasets. We show the proportion of novel  $n$ -grams in gold summaries. We also report ROUGE scores for the LEAD baseline and the extractive oracle system EXT-ORACLE. Results are computed on the test set.

across datasets, however, it is much steeper in XSum whose summaries exhibit approximately 83% novel bigrams, 96% novel trigrams, and 98% novel 4-grams (comparison datasets display around 47–55% new bigrams, 58–72% new trigrams, and 63–80% novel 4-grams).

We further evaluated two extractive methods on these datasets. LEAD is often used as a strong lower bound for news summarization ([Nenkova, 2005](#)) and creates a summary by selecting the first few sentences or words in the document. We extracted the first 3 sentences for CNN documents and the first 4 sentences for DailyMail ([Narayan et al., 2018b](#)). Following previous work ([Durrett et al., 2016](#); [Paulus et al., 2018](#)), we obtained LEAD summaries based on the first 100 words for NY Times documents. For XSum, we selected the first sentence in the document (excluding the one-line summary) to generate the LEAD. Our second method, EXT-ORACLE, can be viewed as an upper bound for extractive models ([Nallapati et al., 2017](#); [Narayan et al., 2018b](#)). It creates an oracle summary by selecting the best possible set of sentences in the document that gives the highest ROUGE ([Lin and Hovy, 2003](#)) with respect to the gold summary. For XSum, we simply selected the single-best sentence in the document as summary.

Table 2 reports the performance of the two extractive methods using ROUGE-1 (R1), ROUGE-2 (R2), and ROUGE-L (RL) with the gold sum-

maries as reference. The LEAD baseline performs extremely well on CNN, DailyMail and NY Times confirming that they are biased towards extractive methods. EXT-ORACLE further shows that improved sentence selection would bring further performance gains to extractive approaches. Abstractive systems trained on these datasets often have a hard time beating the LEAD, let alone EXT-ORACLE, or display a low degree of novelty in their summaries ([See et al., 2017](#); [Tan and Wan, 2017](#); [Paulus et al., 2018](#); [Pasunuru and Bansal, 2018](#); [Celikyilmaz et al., 2018](#)). Interestingly, LEAD and EXT-ORACLE perform poorly on XSum underlying the fact that it is less biased towards extractive methods.

In line with our findings, [Grusky et al. \(2018\)](#) have recently reported similar extractive biases in existing datasets. They constructed a new dataset called “Newsroom” which demonstrates a high diversity of summarization styles. XSum is not diverse, it focuses on a single news outlet (i.e., BBC) and a uniform summarization style (i.e., a single sentence). However, it is sufficiently large for neural network training and we hope it will spur further research towards the development of abstractive summarization models.

### 3 Convolutional Sequence-to-Sequence Learning for Summarization

Unlike tasks like machine translation and paraphrase generation where there is often a one-to-

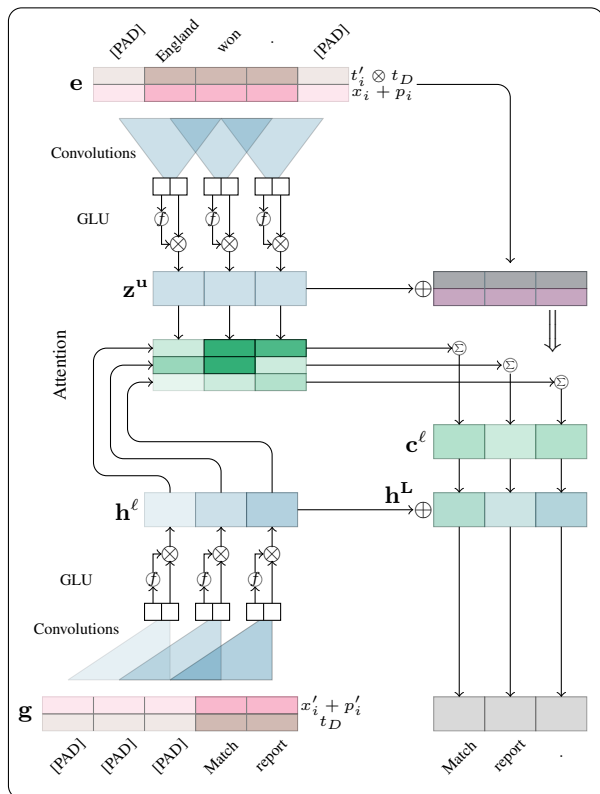


Figure 2: Topic-conditioned convolutional model for extreme summarization.

one semantic correspondence between source and target words, document summarization must distill the content of the document into a few important facts. This is even more challenging for our task, where the compression ratio is extremely high, and pertinent content can be easily missed.

Recently, a convolutional alternative to sequence modeling has been proposed showing promise for machine translation (Gehring et al., 2017a,b) and story generation (Fan et al., 2018). We believe that convolutional architectures are attractive for our summarization task for at least two reasons. Firstly, contrary to recurrent networks which view the input as a chain structure, convolutional networks can be stacked to represent large context sizes. Secondly, hierarchical features can be extracted over larger and larger contents, allowing to represent long-range dependencies efficiently through shorter paths.

Our model builds on the work of Gehring et al. (2017b) who develop an encoder-decoder architecture for machine translation with an attention mechanism (Sukhbaatar et al., 2015) based exclusively on deep convolutional networks. We adapt this model to our summarization task by allow-

ing it to recognize pertinent content (i.e., by foregrounding salient words in the document). In particular, we improve the convolutional encoder by associating each word with a vector representing topic salience, and the convolutional decoder by conditioning each word prediction on the document topic vector.

**Model Overview** At the core of our model is a simple convolutional block structure that computes intermediate states based on a fixed number of input elements. Our convolutional encoder (shown at the top of Figure 2) applies this unit across the document. We repeat these operations in a stacked fashion to get a multi-layer hierarchical representation over the input document where words at closer distances interact at lower layers while distant words interact at higher layers. The interaction between words through hierarchical layers effectively captures long-range dependencies.

Analogously, our convolutional decoder (shown at the bottom of Figure 2) uses the multi-layer convolutional structure to build a hierarchical representation over what has been predicted so far. Each layer on the decoder side determines useful source context by attending to the encoder representation before it passes its output to the next layer. This way the model remembers which words it previously attended to and applies multi-hop attention (shown at the middle of Figure 2) per time step. The output of the top layer is passed to a softmax classifier to predict a distribution over the target vocabulary.

Our model assumes access to word and document topic distributions. These can be obtained by any topic model, however we use Latent Dirichlet Allocation (LDA; Blei et al. 2003) in our experiments; we pass the distributions obtained from LDA directly to the network as additional input. This allows us to take advantage of topic modeling without interfering with the computational advantages of the convolutional architecture. The idea of capturing document-level semantic information has been previously explored for recurrent neural networks (Mikolov and Zweig, 2012; Ghosh et al., 2016; Dieng et al., 2017), however, we are not aware of any existing convolutional models.

**Topic Sensitive Embeddings** Let  $D$  denote a document consisting of a sequence of words



$(w_1, \dots, w_m)$ ; we embed  $D$  into a distributional space  $\mathbf{x} = (x_1, \dots, x_m)$  where  $x_i \in \mathbb{R}^f$  is a column in embedding matrix  $M \in \mathbb{R}^{V \times f}$  (where  $V$  is the vocabulary size). We also embed the absolute word positions in the document  $\mathbf{p} = (p_1, \dots, p_m)$  where  $p_i \in \mathbb{R}^f$  is a column in position matrix  $P \in \mathbb{R}^{N \times f}$ , and  $N$  is the maximum number of positions. Position embeddings have proved useful for convolutional sequence modeling (Gehring et al., 2017b), because, in contrast to RNNs, they do not observe the temporal positions of words (Shi et al., 2016). Let  $t_D \in \mathbb{R}^{f'}$  be the topic distribution of document  $D$  and  $\mathbf{t}' = (t'_1, \dots, t'_m)$  the topic distributions of words in the document (where  $t'_i \in \mathbb{R}^{f'}$ ). During encoding, we represent document  $D$  via  $\mathbf{e} = (e_1, \dots, e_m)$ , where  $e_i$  is:

$$e_i = [(x_i + p_i); (t'_i \otimes t_D)] \in \mathbb{R}^{f+f'},$$

and  $\otimes$  denotes point-wise multiplication. The topic distribution  $t'_i$  of word  $w_i$  essentially captures how topical the word is in itself (local context), whereas the topic distribution  $t_D$  represents the overall theme of the document (global context). The encoder essentially enriches the context of the word with its topical relevance to the document.

For every output prediction, the decoder estimates representation  $\mathbf{g} = (g_1, \dots, g_n)$  for previously predicted words  $(w'_1, \dots, w'_n)$  where  $g_i$  is:

$$g_i = [(x'_i + p'_i); t_D] \in \mathbb{R}^{f+f'},$$

$x'_i$  and  $p'_i$  are word and position embeddings of previously predicted word  $w'_i$ , and  $t_D$  is the topic distribution of the input document. Note that the decoder does not use the topic distribution of  $w'_i$  as computing it on the fly would be expensive. However, every word prediction is conditioned on the topic of the document, enforcing the summary to have the same theme as the document.

**Multi-layer Convolutional Structure** Each convolution block, parametrized by  $W \in \mathbb{R}^{2d \times kd}$  and  $b_w \in \mathbb{R}^{2d}$ , takes as input  $X \in \mathbb{R}^{k \times d}$  which is the concatenation of  $k$  adjacent elements embedded in a  $d$  dimensional space, applies one dimensional convolution and returns an output element  $Y \in \mathbb{R}^{2d}$ . We apply Gated Linear Units (GLU,  $v : \mathbb{R}^{2d} \rightarrow \mathbb{R}^d$ , Dauphin et al. 2017) on the output of the convolution  $Y$ . Subsequent layers operate over the  $k$  output elements of the previous layer and are connected through residual

connections (He et al., 2016) to allow for deeper hierarchical representation. We denote the output of the  $\ell$ th layer as  $\mathbf{h}^\ell = (h_1^\ell, \dots, h_n^\ell)$  for the decoder network, and  $\mathbf{z}^\ell = (z_1^\ell, \dots, z_m^\ell)$  for the encoder network.

**Multi-hop Attention** Our encoder and decoder are tied to each other through a multi-hop attention mechanism. For each decoder layer  $\ell$ , we compute the attention  $a_{ij}^\ell$  of state  $i$  and source element  $j$  as:

$$a_{ij}^\ell = \frac{\exp(d_i^\ell \cdot z_j^u)}{\sum_{t=1}^m \exp(d_i^\ell \cdot z_t^u)},$$

where  $d_i^\ell = W_d^\ell h_i^\ell + b_i^\ell + g_i$  is the decoder state summary combining the current decoder state  $h_i^\ell$  and the previous output element embedding  $g_i$ . The vector  $\mathbf{z}^u$  is the output from the last encoder layer  $u$ . The conditional input  $c_i^\ell$  to the current decoder layer is a weighted sum of the encoder outputs as well as the input element embeddings  $e_j$ :

$$c_i^\ell = \sum_{j=1}^m a_{ij}^\ell (z_j^u + e_j).$$

The attention mechanism described here performs multiple attention ‘‘hops’’ per time step and considers which words have been previously attended to. It is therefore different from single-step attention in recurrent neural networks (Bahdanau et al., 2015), where the attention and weighted sum are computed over  $\mathbf{z}^u$  only.

Our network uses multiple linear layers to project between the embedding size  $(f + f')$  and the convolution output size  $2d$ . They are applied to  $\mathbf{e}$  before feeding it to the encoder, to the final encoder output  $\mathbf{z}^u$ , to all decoder layers  $\mathbf{h}^\ell$  for the attention score computation, and to the final decoder output  $\mathbf{h}^L$  before the softmax. We pad the input with  $k - 1$  zero vectors on both left and right side to ensure that the output of the convolutional layers matches the input length. During decoding, we ensure that the decoder does not have access to future information; we start with  $k$  zero vectors and shift the convolutional block to the right after every prediction. The final decoder output  $\mathbf{h}^L$  is used to compute the distribution over the target vocabulary  $T$  as:

$$p(y_{i+1} | y_1, \dots, y_i, D, t_D, \mathbf{t}') = \text{softmax}(W_o h_i^L + b_o) \in \mathbb{R}^T.$$

We use layer normalization and weight initialization to stabilize learning.

Our topic-enhanced model calibrates long-range dependencies with globally salient content. As a result, it provides a better alternative to vanilla convolutional sequence models (Gehring et al., 2017b) and RNN-based summarization models (See et al., 2017) for capturing cross-document inferences and paraphrasing. At the same time it retains the computational advantages of convolutional models. Each convolution block operates over a fixed-size window of the input sequence, allowing for simultaneous encoding of the input, ease in learning due to the fixed number of non-linearities and transformations for words in the input sequence.

## 4 Experimental Setup

In this section we present our experimental setup for assessing the performance of our **Topic-aware Convolutional Sequence to Sequence** model which we call T-CONVS2S for short. We discuss implementation details and present the systems used for comparison with our approach.

**Comparison Systems** We report results with various systems which were all trained on the XSum dataset to generate a one-line summary given an input news article. We compared T-CONVS2S against three extractive systems: a baseline which randomly selects a sentence from the input document (RANDOM), a baseline which simply selects the leading sentence from the document (LEAD), and an oracle which selects a single-best sentence in each document (EXT-ORACLE). The latter is often used as an upper bound for extractive methods. We also compared our model against the RNN-based abstractive systems introduced by See et al. (2017). In particular, we experimented with an attention-based sequence to sequence model (SEQ2SEQ), a pointer-generator model which allows to copy words from the source text (PTGEN), and a pointer-generator model with a coverage mechanism to keep track of words that have been summarized (PTGEN-COVG). Finally, we compared our model against the vanilla convolution sequence to sequence model (CONVS2S) of Gehring et al. (2017b).

**Model Parameters and Optimization** We did not anonymize entities but worked on a lower-cased version of the XSum dataset. During train-

---

T1:	charge, court, murder, police, arrest, guilty, sentence, boy, bail, space, crown, trial
T2:	church, abuse, bishop, child, catholic, gay, pope, school, christian, priest, cardinal
T3:	council, people, government, local, housing, home, house, property, city, plan, authority
T4:	clinton, party, trump, climate, poll, vote, plaid, election, debate, change, candidate, campaign
T5:	country, growth, report, business, export, fall, bank, security, economy, rise, global, inflation
T6:	hospital, patient, trust, nhs, people, care, health, service, staff, report, review, system, child

---

Table 3: Example topics learned by the LDA model on XSum documents (training portion).

ing and at test time the input document was truncated to 400 tokens and the length of the summary limited to 90 tokens.

The LDA model (Blei et al., 2003) was trained on XSum documents (training portion). We therefore obtained for each word a probability distribution over topics which we used to estimate  $t'$ ; the topic distribution  $t_D$  can be inferred for any new document, at training and test time. We explored several LDA configurations on held-out data, and obtained best results with 512 topics. Table 3 shows some of the topics learned by the LDA model.

For SEQ2SEQ, PTGEN and PTGEN-COVG, we used the best settings reported on the CNN and DailyMail data (See et al., 2017).<sup>2</sup> All three models had 256 dimensional hidden states and 128 dimensional word embeddings. They were trained using Adagrad (Duchi et al., 2011) with learning rate 0.15 and an initial accumulator value of 0.1. We used gradient clipping with a maximum gradient norm of 2, but did not use any form of regularization. We used the loss on the validation set to implement early stopping.

For CONVS2S<sup>3</sup> and T-CONVS2S, we used 512 dimensional hidden states and 512 dimensional word and position embeddings. We trained our convolutional models with Nesterov’s accelerated gradient method (Sutskever et al., 2013) using a momentum value of 0.99 and renormalized gradients if their norm exceeded 0.1 (Pascanu et al., 2013). We used a learning rate of 0.10 and once the validation perplexity stopped improving, we

<sup>2</sup>We used the code available at <https://github.com/abisee/pointer-generator>.

<sup>3</sup>We used the code available at <https://github.com/facebookresearch/fairseq-py>.

reduced the learning rate by an order of magnitude after each epoch until it fell below  $10^{-4}$ . We also applied a dropout of 0.2 to the embeddings, the decoder outputs and the input of the convolutional blocks. Gradients were normalized by the number of non-padding tokens per mini-batch. We also used weight normalization for all layers except for lookup tables.

All neural models, including ours and those based on RNNs (See et al., 2017) had a vocabulary of 50,000 words and were trained on a single Nvidia M40 GPU with a batch size of 32 sentences. Summaries at test time were obtained using beam search (with beam size 10).

## 5 Results

**Automatic Evaluation** We report results using automatic metrics in Table 4. We evaluated summarization quality using  $F_1$  ROUGE (Lin and Hovy, 2003). Unigram and bigram overlap (ROUGE-1 and ROUGE-2) are a proxy for assessing informativeness and the longest common subsequence (ROUGE-L) represents fluency.<sup>4</sup>

On the XSum dataset, SEQ2SEQ outperforms the LEAD and RANDOM baselines by a large margin. PTGEN, a SEQ2SEQ model with a “copying” mechanism outperforms EXT-ORACLE, a “perfect” extractive system on ROUGE-2 and ROUGE-L. This is in sharp contrast to the performance of these models on CNN/DailyMail (See et al., 2017) and Newsroom datasets (Grusky et al., 2018), where they fail to outperform the LEAD. The result provides further evidence that XSum is a good testbed for abstractive summarization. PTGEN-COVG, the best performing abstractive system on the CNN/DailyMail datasets, does not do well. We believe that the coverage mechanism is more useful when generating multi-line summaries and is basically redundant for extreme summarization.

CONVS2S, the convolutional variant of SEQ2SEQ, significantly outperforms all RNN-based abstractive systems. We hypothesize that its superior performance stems from the ability to better represent document content (i.e., by capturing long-range dependencies). Table 4 shows several variants of T-CONVS2S including an encoder network enriched with information about how topical a word is on its own

<sup>4</sup>We used `pyrouge` to compute all ROUGE scores, with parameters “-a -c 95 -m -n 4 -w 1.2.”

Models	R1	R2	RL
Random	15.16	1.78	11.27
LEAD	16.30	1.60	11.95
EXT-ORACLE	29.79	8.81	22.66
SEQ2SEQ	28.42	8.77	22.48
PTGEN	29.70	9.21	23.24
PTGEN-COVG	28.10	8.02	21.72
CONVS2S	31.27	11.07	25.23
T-CONVS2S ( $enc_{t'}$ )	31.71	11.38	25.56
T-CONVS2S ( $enc_{t'}, dec_{t_D}$ )	31.71	11.34	25.61
T-CONVS2S ( $enc_{(t', t_D)}$ )	31.61	11.30	25.51
T-CONVS2S ( $enc_{(t', t_D)}, dec_{t_D}$ )	<b>31.89</b>	<b>11.54</b>	<b>25.75</b>

Table 4: ROUGE results on XSum test set. We report ROUGE-1 (R1), ROUGE-2 (R2), and ROUGE-L (RL)  $F_1$  scores. Extractive systems are in the upper block, RNN-based abstractive systems are in the middle block, and convolutional abstractive systems are in the bottom block.

Models	% of novel $n$ -grams in generated summaries			
	unigrams	bigrams	trigrams	4-grams
LEAD	0.00	0.00	0.00	0.00
EXT-ORACLE	0.00	0.00	0.00	0.00
PTGEN	27.40	73.33	90.43	96.04
CONVS2S	<b>31.26</b>	<b>79.50</b>	<b>94.28</b>	<b>98.10</b>
T-CONVS2S	30.73	79.18	94.10	98.03
GOLD	35.76	83.45	95.50	98.49

Table 5: Proportion of novel  $n$ -grams in summaries generated by various models on the XSum test set.

( $enc_{t'}$ ) or in the document ( $enc_{(t', t_D)}$ ). We also experimented with various decoders by conditioning every prediction on the topic of the document, basically encouraging the summary to be in the same theme as the document ( $dec_{t_D}$ ) or letting the decoder decide the theme of the summary. Interestingly, all four T-CONVS2S variants outperform CONVS2S. T-CONVS2S performs best when both encoder and decoder are constrained by the document topic ( $enc_{(t', t_D)}, dec_{t_D}$ ). In the remainder of the paper, we refer to this variant as T-CONVS2S.

We further assessed the extent to which various models are able to perform rewriting by generating genuinely abstractive summaries. Table 5 shows the proportion of novel  $n$ -grams for LEAD, EXT-ORACLE, PTGEN, CONVS2S, and T-CONVS2S. As can be seen, the convolutional models exhibit the highest proportion of novel  $n$ -grams. We should also point out that the summaries being evaluated have on average comparable lengths; the summaries generated by PTGEN contain 22.57 words, those generated by CONVS2S and T-



EXT-ORACLE	Caroline Pidgeon is the Lib Dem candidate, Sian Berry will contest the election for the Greens and UKIP has chosen its culture spokesman Peter Whittle.	[34.1, 20.5, 34.1]
PTGEN	UKIP leader <b>Nigel Goldsmith</b> has been <b>elected</b> as the <b>new mayor of London</b> to elect a new conservative MP.	[45.7, 6.1, 28.6]
CONVS2S	London mayoral candidate <b>Zac Goldsmith</b> has been <b>elected</b> as the <b>new mayor of London</b> .	[53.3, 21.4, 26.7]
T-CONVS2S	Former London mayoral candidate <b>Zac Goldsmith</b> has been <b>chosen to stand in the London mayoral election</b> .	[50.0, 26.7, 37.5]
GOLD	<b>Zac Goldsmith</b> will <b>contest</b> the 2016 <b>London mayoral election</b> for the conservatives, it has been announced.	
Questions	(1) Who will <b>contest</b> for the conservatives? ( <b>Zac Goldsmith</b> ) (2) For what election will he/she contest? ( <b>The London mayoral election</b> )	
EXT-ORACLE	North-east rivals Newcastle are the only team below them in the Premier League table.	[35.3, 18.8, 35.3]
PTGEN	Sunderland have <b>appointed</b> former <b>Sunderland boss Dick Advocaat</b> as manager at the end of the season to sign a new deal.	[45.0, 10.5, 30.0]
CONVS2S	Sunderland have <b>sacked manager Dick Advocaat</b> after less than three months in charge.	[25.0, 6.7, 18.8]
T-CONVS2S	<b>Dick Advocaat</b> has <b>resigned</b> as <b>Sunderland manager</b> until the end of the season.	[56.3, 33.3, 56.3]
GOLD	<b>Dick Advocaat</b> has <b>resigned</b> as <b>Sunderland boss</b> , with the team yet to win in the Premier League this season.	
Questions	(1) Who has <b>resigned</b> ? ( <b>Dick Advocaat</b> ) (2) From what post has he/she resigned? ( <b>Sunderland boss</b> )	
EXT-ORACLE	The Greater Ardoyne residents collective (GARC) is protesting against an agreement aimed at resolving a long-running dispute in the area.	[26.7, 9.3, 22.2]
PTGEN	A residents' group has been granted permission for GARC to hold a parade on <b>the outskirts of Crumlin, County Antrim</b> .	[28.6, 5.0, 28.6]
CONVS2S	A protest has been held in the <b>Republic of Ireland</b> calling for <b>an end to parading</b> parading in <b>North Belfast</b> .	[42.9, 20.0, 33.3]
T-CONVS2S	A protest has been held in <b>North Belfast</b> over a protest against <b>the Orange Order</b> in <b>North Belfast</b> .	[45.0, 26.3, 45.0]
GOLD	Church leaders have appealed to a nationalist residents' group to call off a protest against <b>an Orange Order parade</b> in <b>North Belfast</b> .	
Questions	(1) Where is the protest supposed to happen? ( <b>North Belfast</b> ) (2) What are they protesting against? ( <b>An Orange Order parade</b> )	

Table 6: Example output summaries on the XSum test set with [ROUGE-1, ROUGE-2 and ROUGE-L] scores, goldstandard reference, and corresponding questions. Words highlighted in blue are either the right answer or constitute appropriate context for inferring it; words in red lead to the wrong answer.

CONVS2S have 20.07 and 20.22 words, respectively, while GOLD summaries are the longest with 23.26 words. Interestingly, PTGEN trained on XSum only copies 4% of 4-grams in the source document, 10% of trigrams, 27% of bigrams, and 73% of unigrams. This is in sharp contrast to PTGEN trained on CNN/DailyMail exhibiting mostly extractive patterns; it copies more than 85% of 4-grams in the source document, 90% of trigrams, 95% of bigrams, and 99% of unigrams (See et al., 2017). This result further strengthens our hypothesis that XSum is a good testbed for abstractive methods.

**Human Evaluation** In addition to automatic evaluation using ROUGE which can be misleading when used as the only means to assess the informativeness of summaries (Schluter, 2017), we also evaluated system output by eliciting human judgments in two ways.

In our first experiment, participants were asked

to compare summaries produced from the EXT-ORACLE baseline, PTGEN, the best performing system of See et al. (2017), CONVS2S, our topic-aware model T-CONVS2S, and the human-authored gold summary (GOLD). We did not include extracts from the LEAD as they were significantly inferior to other models.

The study was conducted on the Amazon Mechanical Turk platform using *Best-Worst Scaling* (BWS; Louviere and Woodworth 1991; Louviere et al. 2015), a less labor-intensive alternative to paired comparisons that has been shown to produce more reliable results than rating scales (Kiritchenko and Mohammad, 2017). Participants were presented with a document and summaries generated from two out of five systems and were asked to decide which summary was *better* and which one was *worse* in order of informativeness (does the summary capture important information in the document?) and fluency (is the summary written in well-formed English?). Examples of

Models	Score	QA
EXT-ORACLE	-0.121	15.70
PTGEN	-0.218	21.40
CONVS2S	-0.130	30.90
T-CONVS2S	<b>0.037</b>	<b>46.05</b>
GOLD	0.431	97.23

Table 7: System ranking according to human judgments and QA-based evaluation.

system summaries are shown in Table 6. We randomly selected 50 documents from the XSum test set and compared all possible combinations of two out of five systems for each document. We collected judgments from three different participants for each comparison. The order of summaries was randomized per document and the order of documents per participant.

The score of a system was computed as the percentage of times it was chosen as best minus the percentage of times it was selected as worst. The scores range from -1 (worst) to 1 (best) and are shown in Table 7. Perhaps unsurprisingly human-authored summaries were considered best, whereas, T-CONVS2S was ranked 2nd followed by EXT-ORACLE and CONVS2S. PTGEN was ranked worst with the lowest score of -0.218. We carried out pairwise comparisons between all models to assess whether system differences are statistically significant. GOLD is significantly different from all other systems and T-CONVS2S is significantly different from CONVS2S and PTGEN (using a one-way ANOVA with posthoc Tukey HSD tests;  $p < 0.01$ ). All other differences are not statistically significant.

For our second experiment we used a question-answering (QA) paradigm (Clarke and Lapata, 2010; Narayan et al., 2018b) to assess the degree to which the models retain key information from the document. We used the same 50 documents as in our first elicitation study. We wrote two fact-based questions per document, just by reading the summary, under the assumption that it highlights the most important content of the news article. Questions were formulated so as not to reveal answers to subsequent questions. We created 100 questions in total (see Table 6 for examples). Participants read the output summaries and answered the questions as best they could without access to the document or the gold summary. The more questions can be answered, the better the corresponding system is at summarizing the document as a whole. Five participants answered ques-

tions for each summary.

We followed the scoring mechanism introduced in Clarke and Lapata (2010). A correct answer was marked with a score of one, partially correct answers with a score of 0.5, and zero otherwise. The final score for a system is the average of all its question scores. Answers again were elicited using Amazon’s Mechanical Turk crowdsourcing platform. We uploaded the data in batches (one system at a time) to ensure that the same participant does not evaluate summaries from different systems on the same set of questions.

Table 7 shows the results of the QA evaluation. Based on summaries generated by T-CONVS2S, participants can answer 46.05% of the questions correctly. Summaries generated by CONVS2S, PTGEN and EXT-ORACLE provide answers to 30.90%, 21.40%, and 15.70% of the questions, respectively. Pairwise differences between systems are all statistically significant ( $p < 0.01$ ) with the exception of PTGEN and EXT-ORACLE. EXT-ORACLE performs poorly on both QA and rating evaluations. The examples in Table 6 indicate that EXT-ORACLE is often misled by selecting a sentence with the highest ROUGE (against the gold summary), but ROUGE itself does not ensure that the summary retains the most important information from the document. The QA evaluation further emphasizes that in order for the summary to be felicitous, information needs to be embedded in the appropriate context. For example, CONVS2S and PTGEN will fail to answer the question “Who has resigned?” (see Table 6 second block) despite containing the correct answer “Dick Advoaat” due to the wrong context. T-CONVS2S is able to extract important entities from the document with the right theme.

## 6 Conclusions

In this paper we introduced the task of “extreme summarization” together with a large-scale dataset which pushes the boundaries of abstractive methods. Experimental evaluation revealed that models which have abstractive capabilities do better on this task and that high-level document knowledge in terms of topics and long-range dependencies is critical for recognizing pertinent content and generating informative summaries. In the future, we would like to create more linguistically-aware encoders and decoders incorporating co-reference and entity linking.

**Acknowledgments** We gratefully acknowledge the support of the European Research Council (Lapata; award number 681760), the European Union under the Horizon 2020 SUMMA project (Narayan, Cohen; grant agreement 688139), and Huawei Technologies (Cohen).

## References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, California, USA.
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022.
- Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, New Orleans, USA.
- Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, and Hui Jiang. 2016. Distraction-based neural networks for modeling documents. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pages 2754–2760, New York, USA.
- 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*, pages 484–494, Berlin, Germany.
- James Clarke and Mirella Lapata. 2010. Discourse constraints for document compression. *Computational Linguistics*, 36(3):411–441.
- Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 933–941, Sydney, Australia.
- Adji B. Dieng, Chong Wang, Jianfeng Gao, and John Paisley. 2017. Topicrnn: A recurrent neural network with long-range semantic dependency. In *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France.
- John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159.
- Greg Durrett, Taylor Berg-Kirkpatrick, and Dan Klein. 2016. Learning-based single-document summarization with compression and anaphoricity constraints. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1998–2008, Berlin, Germany.
- Angela Fan, David Grangier, and Michael Auli. 2017. Controllable abstractive summarization. *CoRR*, abs/1711.05217.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, Melbourne, Australia.
- Jonas Gehring, Michael Auli, David Grangier, and Yann Dauphin. 2017a. A convolutional encoder model for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 123–135, Vancouver, Canada.
- Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017b. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1243–1252, Sydney, Australia.
- Shalini Ghosh, Oriol Vinyals, Brian Strope, Scott Roy, Tom Dean, and Larry Heck. 2016. Contextual LSTM (CLSTM) models for large scale NLP tasks. *CoRR*, abs/1602.06291.
- Max Grusky, Mor Naaman, and Yoav Artzi. 2018. NEWSROOM: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, New Orleans, USA.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, Las Vegas, USA.
- Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28*, pages 1693–1701. Morgan, Kaufmann.
- Svetlana Kiritchenko and Saif Mohammad. 2017. Best-worst scaling more reliable than rating scales: A case study on sentiment intensity annotation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 465–470, Vancouver, Canada.
- Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 71–78, Edmonton, Canada.

- Jordan J Louviere, Terry N Flynn, and Anthony Alfred John Marley. 2015. *Best-worst scaling: Theory, methods and applications*. Cambridge University Press.
- Jordan J Louviere and George G Woodworth. 1991. Best-worst scaling: A model for the largest difference judgments. *University of Alberta: Working Paper*.
- Inderjeet Mani. 2001. *Automatic Summarization*. Natural language processing. John Benjamins Publishing Company.
- Tomas Mikolov and Geoffrey Zweig. 2012. Context dependent recurrent neural network language model. In *Proceedings of the Spoken Language Technology Workshop*, pages 234–239. IEEE.
- Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pages 3075–3081, San Francisco, California USA.
- Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çağlar Gülçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany.
- Shashi Narayan, Ronald Cardenas, Nikos Papasaratopoulos, Shay B. Cohen, Mirella Lapata, Jiangsheng Yu, and Yi Chang. 2018a. Document modeling with external attention for sentence extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, Melbourne, Australia.
- Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018b. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, New Orleans, USA.
- Shashi Narayan, Nikos Papasaratopoulos, Shay B. Cohen, and Mirella Lapata. 2017. Neural extractive summarization with side information. *CoRR*, abs/1704.04530.
- Ani Nenkova. 2005. Automatic text summarization of newswire: Lessons learned from the Document Understanding Conference. In *Proceedings of the 29th National Conference on Artificial Intelligence*, pages 1436–1441, Pittsburgh, Pennsylvania, USA.
- Ani Nenkova and Kathleen McKeown. 2011. Automatic summarization. *Foundations and Trends in Information Retrieval*, 5(2–3):103–233.
- Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning*, pages 1310–1318, Atlanta, GA, USA.
- Ramakanth Pasunuru and Mohit Bansal. 2018. Multi-reward reinforced summarization with saliency and entailment. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, New Orleans, USA.
- Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *Proceedings of the 6th International Conference on Learning Representations*, Vancouver, BC, Canada.
- Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal.
- Evan Sandhaus. 2008. The New York Times Annotated Corpus. *Linguistic Data Consortium, Philadelphia*, 6(12).
- Natalie Schluter. 2017. The limits of automatic summarisation according to rouge. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Short Papers*, pages 41–45, Valencia, Spain.
- Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1073–1083, Vancouver, Canada.
- Xing Shi, Kevin Knight, and Deniz Yuret. 2016. Why neural translations are the right length. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2278–2282, Austin, Texas.
- Sainbayar Sukhbaatar, arthur szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems 28*, pages 2440–2448. Morgan, Kaufmann.
- Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. 2013. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on International Conference on Machine Learning*, pages 1139–1147, Atlanta, GA, USA.
- Jiwei Tan and Xiaojun Wan. 2017. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1171–1181, Vancouver, Canada.