What is This Article About? Extreme Summarization with Topic-Aware Convolutional Neural Networks

Shashi Narayan   shashinarayan@google.com
Google Research
6 Pancras Square, Kings Cross, London, UK

Shay B. Cohen   scohen@inf.ed.ac.uk
Mirella Lapata   mlap@inf.ed.ac.uk
Institute for Language, Cognition and Computation
School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh, UK

Abstract

We introduce *extreme summarization*, a new single-document summarization task which aims at creating a short, one-sentence news summary answering the question "What is the article about?". We argue that extreme summarization, by nature, is not amenable to extractive strategies and requires an abstractive modeling approach. In the hope of driving research on this task further: (a) we collect a real-world, large scale dataset by harvesting online articles from the British Broadcasting Corporation (BBC); and (b) 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 on the extreme summarization dataset.

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 & McKeown, 2011 for comprehensive overviews) single-document summarization has consistently garnered attention.

Modern approaches to single document summarization are data-driven, taking advantage of the success of neural network architectures and their ability to learn continuous features without recourse to preprocessing tools or linguistic annotations (Cheng & Lapata, 2016; Nallapati et al., 2016; See et al., 2017; Fan et al., 2017; Paulus et al., 2018; Pasunuru & Bansal, 2018; Celikyilmaz et al., 2018; Narayan et al., 2018b). The application of neural networks to the summarization task has motivated the development of large-scale datasets containing hundreds of thousands of (news) document-summary pairs (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 & Lapata, 2016; Nallapati et al., 2017; Narayan et al., 2018b). Abstractive approaches, despite being more faithful to the actual

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.

The plane's only two occupants were unharmed, it added.

The Diario de Noticias newspaper quoted an eyewitness who said the plane had been flying at a low altitude over the beach, although he did not realise anything was wrong until other beachgoers began running.

One young witness told Reuters news agency: "I was near the water when I saw the plane. I called my parents, the plane fell on the sand and ran over two people, fatally hurting them and another was injured, I think, but I'm not sure, people were running away."

"The plane is still there, but the ambulances arrived quickly. I think maybe the fuel ran out because I find it weird that it landed on the beach."

Other reports said the victims had been sunbathing when the plane made its emergency landing. The Associated Press news agency said the girl who died had been with her parents, who were unhurt. The agency quoted witnesses from local television broadcasts.

Joao Quadros, who was on the beach, tweeted photos of the aftermath, saying the plane had passed by his son by a matter of metres. There had been no noise, he said.

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. One wing seemed to be misaligned in those photos.

The cause of the emergency landing remains unclear.

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

summarization task — professional editors employ various rewrite operations to transform article sentences into a summary including compression, aggregation, and paraphrasing (Jing, 2002) aside from writing sentences from scratch — they either lag behind extractive ones or are mostly extractive, exhibiting a small degree of abstraction (See et al., 2017; Paulus et al., 2018; Pasunuru & Bansal, 2018; Celikyilmaz et al., 2018; Gehrmann 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 this article about?". Figure 1 shows an example of a document and its extreme summary. 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.

To drive research on abstractive summarization forward, 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 is well-suited to extreme summarization. Unlike most recent abstractive ap-

proaches (Rush et al., 2015; Chen et al., 2016; Nallapati et al., 2016; See et al., 2017; Tan & Wan, 2017; Paulus et al., 2018; Pasunuru & 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 capturing whether it is in the theme of the document.

Experimental evaluation on the extreme summarization task shows that our topic-aware convolutional model outperforms an oracle extractive system (in terms of ROUGE) as well as state-of-the-art RNN-based abstractive systems, a vanilla convolutional model (Gehring et al., 2017b) and a convolutional model augmented with the pointer-generator mechanism (See et al., 2017). 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. To further illustrate that the proposed model is generally applicable, we evaluate its performance on the Newsroom Abstractive dataset (Grusky et al., 2018). Our experiments set a new state of the art and highlight interesting differences between our extreme summarization dataset and the Newsroom dataset.

Our contributions in this work are three-fold:[1] we propose a new single-document summarization dataset which encourages the development of abstractive systems; we demonstrate through analysis and empirical results that extractive approaches are not well-suited to the extreme summarization task; and propose a novel topic-aware convolutional sequence-to-sequence model for abstractive summarization. In the remainder, we present an overview of related work (Section 2) and the describe our extreme summarization dataset in more detail (Section 3). Section 4 presents our model while Section 6 discusses our results.

2. Related Work

Summarization Datasets The summarization of news articles has enjoyed wide popularity in natural language processing due to its potential for various information access applications which allow readers to spot emerging trends, person mentions, the evolution of storylines, and so on. The news domain has been the main focus of several Document Understanding (DUC) and Text Analysis conferences (TAC) leading to the creation of various high-quality summarization datasets (Harman & Over, 2004; Over et al., 2007). More recently, the training requirements of neural systems have led to the compilation of larger datasets based on New York Times (Sandhaus, 2008), the Gigaword corpus (Napoles et al., 2012), the CNN and DailyMail news outlets (Hermann et al., 2015), or a combination of several major news publications (Grusky et al., 2018). There has been some interest in summarizing texts from other domains, such as longer scientific articles (Qazvinian et al., 2013; Cohan et al., 2018; Yasunaga et al., 2019), Wikipedia articles (Liu et al., 2018), live sport text commentary scripts (Zhang et al., 2016), movie reviews (Wang & Ling, 2016;

---

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

Angelidis & Lapata, 2018) or online discussion forums and blogs (Hu et al., 2015; Kim et al., 2018; Koupaee & Wang, 2018). In this paper, we focus on generating extreme (single line abstractive) summaries for BBC news articles.

The nature and quality of reference summaries vary for different datasets. DUC datasets contain multi-reference summaries that are manually written especially to evaluate summarization systems. Due to the effort and cost involved in creating multiple reference summaries, DUC datasets are rather small (few hundreds of articles) and fall short of training neural summarization systems. Gigaword summaries are short headlines (Rush et al., 2015). Systems trained on New York Times and CNN/DailyMail learn to generate multi-line abstracts or highlights, however these summaries are mostly extractive and systems trained on them unavoidably learn to perform mainly copying operations even when capable of performing abstraction (See et al., 2017; Grusky et al., 2018). Newsroom (Grusky et al., 2018) summaries are manual descriptions of news articles written by authors and editors in newsrooms of 38 major news publications. Coming from a variety of sources, these summaries exhibit different degrees of abstraction, they are not visible to readers but are often used to index the article (Grusky et al., 2018). In contrast, our summaries are read together with the article, they are the first sentence readers see (often highlighted in boldface) prior to digesting the full article. The Newsroom dataset is fairly large containing 1.3 million articles and their summaries and goes some way towards addressing the concerns relating to biases towards extractive strategies in earlier datasets. We discuss the differences between Newsroom and our dataset in more detail in Section 3 and also present experimental results with our model in Section 6.2.

Summarization Approaches Approaches to document summarization fall under two major paradigms: extractive systems select sentences from the document and assemble them together to generate a summary, while abstractive systems create a summary from scratch, possibly generating new words or phrases which are not in the document.

A great deal of previous work has focused on extractive summarization which is usually modeled as a sentence ranking or binary classification problem (i.e., sentences which are top ranked or predicted as `True` are selected as summaries). Early attempts mostly leverage human-engineered features including sentence position and length (Radev et al., 2004), keywords and the presence of proper nouns (Kupiec et al., 1995; Mani, 2001; Spärck Jones, 2007), information based on frequency (Nenkova et al., 2006) or events (Filatova & Hatzivassiloglou, 2004). These methods often learn to score each sentence independently (Barzilay & Elhadad, 1997; Teufel & Moens, 1997; Barzilay et al., 2002; Erkan & Radev, 2004; Mihalcea & Tarau, 2004; Shen et al., 2007; Schilder & Kondadadi, 2008; Wan, 2010), however summary quality can be improved heuristically (Yih et al., 2007; Dorr et al., 2003), via max-margin methods (Carbonell & Goldstein, 1998; Li et al., 2009), or integer-linear programming (Woodsend & Lapata, 2010; Berg-Kirkpatrick et al., 2011; Woodsend & Lapata, 2012; Almeida & Martins, 2013; Parveen et al., 2015; Martins & Smith, 2009).

Modern extractive summarization models (Kågebäck et al., 2014; Yin & Pei, 2015; Cheng & Lapata, 2016; Nallapati et al., 2017) are data-driven and learn continuous features using neural network architectures without any linguistic preprocessing or reliance on expert feature design. The majority of them conceptualize extractive summarization as a sequence labeling task in which each label specifies whether each document sentence should

be included in the summary (Cheng & Lapata, 2016; Nallapati et al., 2017; Narayan et al., 2017; Yasunaga et al., 2017; Narayan et al., 2018a, 2018b; Zhang et al., 2018; Mendes et al., 2019). These models often rely on recurrent neural networks to derive a meaning representation of the document which is then used to label each sentence, taking the previously labeled sentences into account.

There has also been a surge of interest in neural network models for abstractive summarization which is viewed as a sequence-to-sequence problem (Sutskever et al., 2014; Cho et al., 2014; Bahdanau et al., 2015). Central in most approaches (Rush et al., 2015; Chen et al., 2016; Nallapati et al., 2016; Tan & Wan, 2017; See et al., 2017) is an encoder-decoder architecture modeled by recurrent neural networks. The encoder reads the source sequence into a list of continuous-space representations from which the decoder generates the target sequence. See et al. (2017) refine this sequence-to-sequence architecture with a copy mechanism (Vinyals et al., 2015; Gu et al., 2016) which allows to reuse sequences from the source document and with a coverage mechanism (Tu et al., 2016) which allows to keep track of what has been summarized, discouraging repetition. A few extractive (Narayan et al., 2018b; Wu & Hu, 2018; Dong et al., 2018) and abstractive (Paulus et al., 2018; Li et al., 2018a; Celikyilmaz et al., 2018; Pasunuru & Bansal, 2018; Chen & Bansal, 2018; Kryściński et al., 2018) approaches obtain performance improvements by combining the maximum-likelihood cross-entropy loss with rewards from policy gradient reinforcement learning (Williams, 1992) to directly optimize the evaluation metric relevant for the summarization task.

Our topic-conditioned convolutional model differs from earlier approaches both in application and formulation. Unlike abstractive models based on recurrent neural networks, we adopt a fully convolutional endcoder-decoder architecture (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 documents content, while our convolutional decoder conditions each word prediction on a document topic vector. Convolutional alternatives to sequence modeling have been proposed for machine translation (Gehring et al., 2017a), headline generation (Gehring et al., 2017b), and story generation (Fan et al., 2018), however we are not aware of any prior work targetting summarization. The Transformer architecture (Vaswani et al., 2017) presents an alternative to convolutions, also aiming at eliminating the fundamental constraint of sequential computation, and has been successfully applied to sentence and document summarization (Liu et al., 2018; Liu, 2019; Song et al., 2019; Dong et al., 2019).[2]

Our convolutional model uses topic vectors to foreground salient words in the document. The idea is inspired from traditional summarization methods for *content selection* (Mani, 2001; Radev et al., 2004; Nenkova et al., 2006; Spärck Jones, 2007), however, our topics are not manually crafted, they are automatically learned using an LDA model (Blei et al., 2003). Several recent summarization models have explored architectures dedicated to content selection; Li et al. (2018) extract a set of keywords from the document to guide

---

2. Experiments with Transformer architectures are outside the scope of this paper. Recent work (Perez-Beltrachini et al., 2019) on multiodocument summarization shows that Transformer-based models perform on par with their convolutional alternatives.

the summarization process. Zhou et al. (2017) and Li et al. (2018b) use dedicated gates to filter the representation of the source document; while others modulate the attention based on how likely it is for a word or a sentence to be included in a summary (Cohan et al., 2018; Hsu et al., 2018; Gehrmann et al., 2018) or use reinforcement learning to optimize content selection objectives (Pasunuru & Bansal, 2018; Chen & Bansal, 2018). Document-level semantic information (as expressed via latent topics) has been previously integrated with recurrent neural networks (Mikolov & Zweig, 2012; Ghosh et al., 2016; Dieng et al., 2017), however, we are not aware of any existing convolutional models.

3. The XSum Dataset

In this section we present, XSum, our extreme summarization dataset which consists of BBC articles and accompanying single sentence summaries. We describe how XSum was obtained, provide comparisons with popular summarization benchmarks, and analyze how it differs from them both quantitatively and qualitatively.

3.1 Data Collection

Each BBC 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).

To create a large-scale dataset for extreme summarization, we followed the methodology proposed in Hermann et al. (2015). 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.

Tables 1 and 2 compare XSum with the CNN, DailyMail, NY Times, and Newsroom benchmarks. 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 Newsroom dataset, we used the splits and pre-processing steps of Grusky et al. (2018). We present comparisons with the full Newsroom dataset (Newsroom) and its three subsets: mostly extractive (Newsroom-Ext), mostly abstractive (Newsroom-Abs), and mixed (Newsroom-Mixed). As can be seen in Table 1, XSum contains a substantial number of training instances, similar to DailyMail; documents and summaries in XSum are shorter in relation to most datasets (see Table 2) but the vocabulary size is sufficiently large, comparable to CNN.

3.2 How Abstractive is XSum?

To support the claim that XSum summaries are fairly abstractive and as a result systems trained on them could not resort to extractive strategies, we record the percentage of novel n-grams in the gold summaries that do not appear in their source documents. As

Datasets	Corpus Size (# docs)		
	training	validation	test
CNN	90,266	1,220	1,093
DailyMail	196,961	12,148	10,397
NY Times	589,284	32,736	32,739
Newsroom	992,966	108,591	108,650
Newsroom-Mixed	328,634	35,879	36,006
Newsroom-Ext	331,778	36,332	36,122
Newsroom-Abs	332,554	36,380	36,522
XSum	204,045	11,332	11,334

Table 1: Comparison of XSum with benchmark summarization datasets: CNN and Daily-Mail datasets (Hermann et al., 2015), NY Times (Sandhaus, 2008), and Newsroom (Grusky et al., 2018). We present the full Newsroom dataset (Newsroom) and its three subsets: mostly extractive (Newsroom-Ext), mostly abstractive (Newsroom-Abs), and mixed (Newsroom-Mixed). We report corpus size, i.e., the number of documents in, training, validation, and test sets.

Datasets	avg. document length		avg. summary length		vocabulary size	
	words	sentences	words	sentences	document	summary
CNN	760.50	33.98	45.70	3.59	343,516	89,051
DailyMail	653.33	29.33	54.65	3.86	563,663	179,966
NY Times	800.04	35.55	45.54	2.44	1,399,358	294,011
Newsroom	770.09	34.73	30.36	1.43	2,646,681	360,290
Newsroom-Mixed	830.58	36.63	23.78	1.17	1,271,435	169,875
Newsroom-Ext	706.06	31.65	45.78	1.88	1,214,748	243,062
Newsroom-Abs	774.17	35.92	21.49	1.25	1,385,205	157,939
XSum	431.07	19.77	23.26	1.00	399,147	81,092

Table 2: We compare datasets with respect to average document (source) and summary (target) length (in terms of words and sentences), and vocabulary size on both on source and target. See main text for steps taken to split and pre-process these datasets. For the vocabulary, we lower case tokens.

shown in Table 3, there are 36% novel unigrams in the XSum reference summaries compared to 17% in CNN, 17% in DailyMail, 23% in NY Times, and 18% in Newsroom. This indicates that XSum summaries are more abstractive. The proportion of novel constructions grows for larger n-grams 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).

Datasets	% of novel n-grams in gold summary			
	unigrams	bigrams	trigrams	4-grams
CNN	16.75	54.33	72.42	80.37
DailyMail	17.03	53.78	72.14	80.28
NY Times	22.64	55.59	71.93	80.16
Newsroom	18.31	46.80	58.06	62.72
Newsroom-Mixed	13.78	48.37	67.15	77.11
Newsroom-Ext	2.65	7.25	10.25	12.42
Newsroom-Abs	38.25	84.36	96.39	98.28
XSum	35.76	83.45	95.50	98.49

Table 3: Proportion of novel n-grams in gold summaries for CNN, DailyMail, NY Times, Newsroom, and XSum datasets. All results are computed on the test set.

We further evaluate two extractive methods, LEAD and EXT-ORACLE, 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 Newsroom, we extracted the first 2 sentences to form the LEAD summaries. 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 & Hovy, 2003) with respect to the gold summary. For XSum, we simply selected the single-best sentence in the document as summary.

Table 4 reports the performance of the two extractive methods using ROUGE-1 (R1), ROUGE-2 (R2), and ROUGE-L (RL) with the gold summaries as reference. The LEAD baseline performs extremely well on CNN, DailyMail, NY Times and Newsroom confirming that they contain fairly extractive summaries. 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 & Wan, 2017; Paulus et al., 2018; Pasunuru & Bansal, 2018; Celikyilmaz et al., 2018; Gehrmann et al., 2018). Interestingly, LEAD and EXT-ORACLE perform poorly on XSum underlying the fact that it contains genuinely abstractive summaries.

Grusky et al. (2018) also find that CNN / Daily Mail and New York Times are skewed towards extractive summaries (albeit following different analysis metrics). The abstractive subset of their Newsroom dataset (Newsroom-Abs) demonstrates similar patterns to XSum in terms of the percentage of novel n-grams in the gold summary and the performance of extractive methods (LEAD and EXT-ORACLE). However, XSum differs from Newsroom in two key respects. Firstly, Newsroom is a fairly diverse dataset, it contains documents and

Datasets	LEAD			EXT-ORACLE		
	R1	R2	RL	R1	R2	RL
CNN	29.15	11.13	25.95	50.38	28.55	46.58
DailyMail	40.68	18.36	37.25	55.12	30.55	51.24
NY Times	31.85	15.86	23.75	52.08	31.59	46.72
Newsroom	33.04	22.35	30.31	57.09	42.94	53.65
Newsroom-Mixed	27.95	13.87	23.97	51.98	34.04	46.96
Newsroom-Ext	55.87	50.60	54.76	89.63	87.20	89.32
Newsroom-Abs	15.44	2.72	12.32	29.85	7.82	24.86
XSum	16.30	1.61	11.95	29.79	8.81	22.65

Table 4: Performance of extractive baselines on CNN, DailyMail, NY Times, Newsroom, and XSum datasets. We report ROUGE scores for the LEAD baseline and EXT-ORACLE, the extractive oracle system. All results are computed on the test set.

summaries from multiple news outlets representing a large range of summarization styles from highly abstractive to highly extractive, while XSum is not; it covers a single news outlet (i.e., BBC) and a uniform summarization style (i.e., a single sentence). Another difference comes from the way the reference summaries are extracted in these two datasets. Newsroom summaries are extracted using the HTML meta-tag "description," and constitute descriptions of the document's content which are often used for indexing but are not shown to the readers. In comparison, XSum summaries are aimed at the reader and meant to be read together with the article. Newsroom summaries are often indicative – they provide merely an indication of the subject matter of the document without giving away detail on its content. In contrast, XSum summaries are more informative, they contain pertinent information necessary to convey the gist of the document. We further explore these differences in our experimental evaluation (see Section 6.3).

4. Topic-Aware Convolutional Model for Summarization

Unlike tasks like machine translation and paraphrase generation where there is often a one-to-one semantic correspondence between source and target words, document summarization must distill the content of a 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.

Our model builds on the work of Gehring et al. (2017b) who develop an encoder-decoder architecture with an attention mechanism (Sukhbaatar et al., 2015) based exclusively on deep convolutional networks. Their convolutional alternative to sequence modeling has shown promise for machine translation (Gehring et al., 2017a, 2017b) and story generation (Fan et al., 2018). We believe that convolutional architectures are attractive for extreme summarization 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 con-

text sizes. Secondly, hierarchical features can be extracted over larger and larger contents, allowing to represent long-range dependencies efficiently through shorter paths.

We adapt this model to our task by allowing 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. Our model aims to generate informative summaries that are grounded in the input document and its content.

4.1 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 in 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.

4.2 Topic Sensitive Embeddings

Let D denote a document consisting of a sequence of words $(w_1, \ldots, w_m)$; we embed D into a distributional space $\mathbf{x} = (x_1, \ldots, x_m)$ 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 $\mathbf{p} = (p_1, \ldots, p_m)$ 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; p_i is the position embedding of word w_i at position i in the input sequence. 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 $\mathbf{t}' = (t'_1, \ldots, t'_m)$ the topic distributions of words in the document (where $t'_i \in \mathbb{R}^{f'}$). During encoding, we represent document D via $\mathbf{e} = (e_1, \ldots, e_m)$, where e_i is:

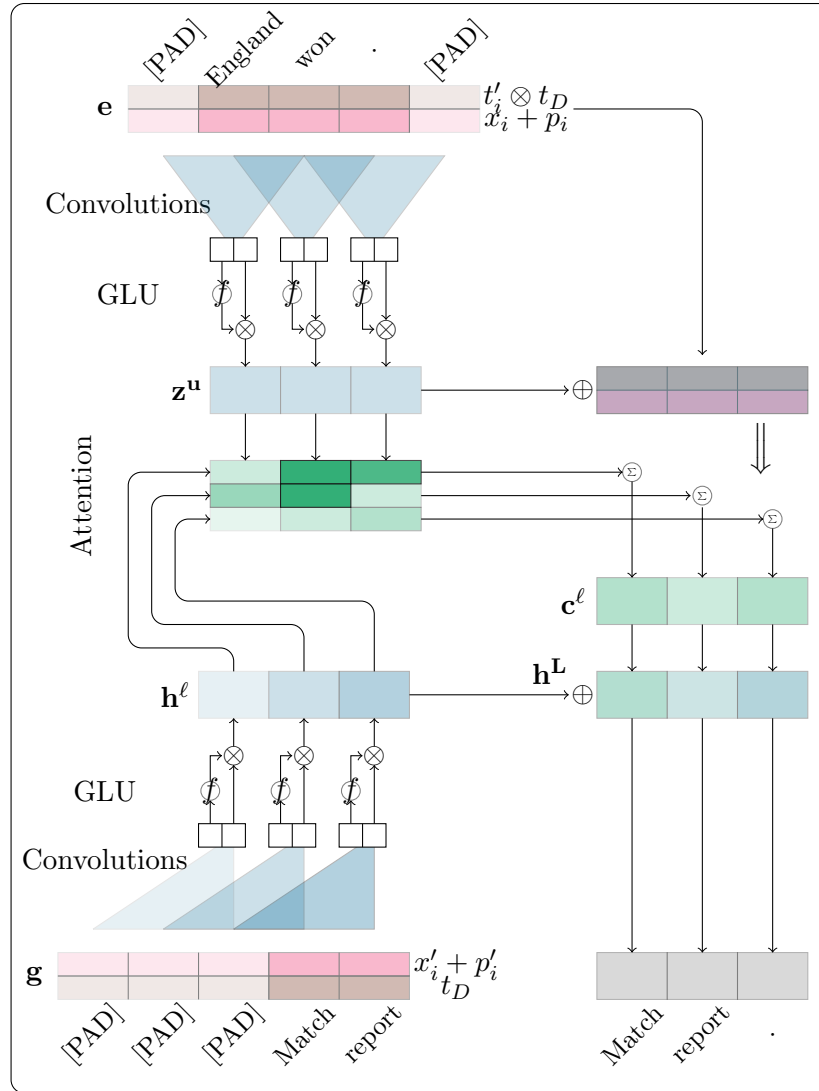$$e_i = [(x_i + p_i); (t'_i \otimes t_D)] \in \mathbb{R}^{f+f'}, \tag{1}$$

Figure 2: Topic-conditioned convolutional model for extreme summarization. The input document "England won." is encoded (top) using word (x_i) and position (p_i) embeddings, and word (t'_i) and document (t_D) topic vectors. At each time step of the decoding, a context representation (bottom left) is learned using word (x'_i) and position (p'_i) embeddings and the document (t_D) topic vector. Attention (center) is the dot product between decoder context representations and encoder representations. The conditional inputs c^l computed by the attention are added to the decoder states h^l which then predict the target words (bottom right). GLU stands for Gated Linear Units. Both encoder and decoder pad input and output sequences respectively to match the receptive width of the convolutional filters. Here, the decoder generates summary "Match report.".

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 $\mathbf{g} = (g_1, \ldots, g_n)$ for previously predicted words $(w'_1, \ldots, w'_n)$ where g_i is:

$$g_i = [(x'_i + p'_i); t_D] \in \mathbb{R}^{f+f'}, \tag{2}$$

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.

4.3 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, $v : \mathbb{R}^{2d} \to \mathbb{R}^d$, Dauphin et al., 2017) on the output of 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 ℓth layer as $\mathbf{h}^\ell = (h_1^\ell, \ldots, h_n^\ell)$ for the decoder network, and $\mathbf{z}^\ell = (z_1^\ell, \ldots, z_m^\ell)$ for the encoder network.

4.4 Multi-hop Attention

Our encoder and decoder are tied via a multi-hop attention mechanism. For each decoder layer ℓ, we compute the attention a_{ij}^ℓ of state i and source element j as:

$$a_{ij}^\ell = \frac{\exp(d_i^\ell \cdot z_j^u)}{\sum_{t=1}^m \exp(d_i^\ell \cdot z_t^u)}, \tag{3}$$

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^ℓ and the previous output element embedding g_i. Vector $\mathbf{z}^\mathbf{u}$ is the output from the last encoder layer u. The conditional input c_i^ℓ to the current decoder layer is a weighted sum of the encoder outputs as well as the input element embeddings e_j:

$$c_i^\ell = \sum_{j=1}^m a_{ij}^\ell (z_j^u + e_j). \tag{4}$$

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 $\mathbf{z}^\mathbf{u}$ only.

Our network uses multiple linear layers to project between the embedding size $(f + f')$ and the convolution output size $2d$. These 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 sides 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 as:

$$p(y_{i+1}|y_1, \ldots, y_i, D, t_D, \mathbf{t}') = \text{softmax}(W_o h_i^L + b_o) \in \mathbb{R}^T \tag{5}$$

where, W_o and b_o are the parameters of the softmax layer and T is the size of the target vocabulary. We also use layer normalization and weight initialization to stabilize learning. We use cross-entropy loss to maximize the likelihood of the ground-truth sequence $(y_1^*, \ldots, y_n^*)$:

$$L(\theta) = \sum_{i=0}^{n-1} \log p(y_{i+1}^*|y_1^*, \ldots, y_i^*, D, t_D, \mathbf{t}', \theta) \tag{6}$$

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 and ease in learning due to the fixed number of non-linearities and transformations for words in the input sequence.

5. Experimental Setup

In this section we present our experimental setup for assessing the performance of our **T**opic-aware **Conv**olutional **S**equence **to S**equence model which we abbreviate to T-ConvS2S. We evaluate our model on our newly collected XSum dataset and show that it is suitable for extreme summarization. We also report experiments on the abstractive subset of the Newsroom dataset (Newsroom-Abs; Grusky et al., 2018). In the following, we discuss implementation details, present the systems used for comparison with our approach, and explain how system output was evaluated.

5.1 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 in See et al. (2017).[3] In particular, we experimented with an attention-based sequence-to-sequence model (SEQ2SEQ), a pointer-generator model which allows us 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 two convolutional abstractive systems: the vanilla convolution sequence-to-sequence model of Gehring et al. (2017b) (CONVS2S) and a variant thereof augmented a copy mechanism (CONVS2S+COPY) which copies words from the input document via pointing (Vinyals et al., 2015; See et al., 2017) while retaining the ability to produce novel words from a fixed vocabulary.[4]

For our experiments on Newsroom-Abs, we again compared T-CONVS2S against the extractive systems RANDOM, LEAD and EXT-ORACLE, the recurrent abstractive systems SEQ2SEQ, PTGEN, and PTGEN+COVG, and the convolutional systems CONVS2S and CONVS2S+COPY. All systems were retrained on the Newsroom-Abs training set. LEAD selects the first 2 sentences to form the summary while RANDOM selects random 2 sentences from the input document to form the summary. EXT-ORACLE creates an oracle summary by selecting the best possible set of sentences in the document that gives the highest ROUGE (Lin & Hovy, 2003) with respect to the gold summary.

5.2 Model Parameters and Optimization

We did not anonymize entities but worked with a lowercased version of the XSum and Newsroom-Abs datasets. During training and at test time input documents were truncated to 400 tokens and the length of the summary was limited to 90 tokens.

We trained two separate LDA models (Blei et al., 2003) on XSum and Newsroom documents (training portion). We therefore obtained for each word a probability distribution

---

3. State-of-the-art abstractive systems on the CNN/Daily mail and New York Times datasets (Celikyilmaz et al., 2018; Pasunuru & Bansal, 2018; Kryściński et al., 2018) use reinforcement learning to directly optimize the evaluation metric relevant for the summarization task. Although our model could be optimized with reinforcement learning objectives, we leave this to future work and present comparisons with related models which are all trained with the maximum-likelihood objective.

4. CONVS2S+COPY estimates the generation probability $p_{gen} \in [0,1]$ at each decoding step i as:

$$p_{gen} = \sigma(w_h h_i^L + w_c c_i^L + w_g g_i + b_{gen})$$

using the final decoder state h_i^L, the final context vector c_i^L and the decoder input g_i. L is the final layer of the decoder. w_h, w_c, w_g and b_{gen} are model parameters and σ is the non-linear sigmoid function. We estimate the final probability distribution $p'(w) \in R^{T'}$ over the extended vocabulary T' denoting the union of the target vocabulary T and the words of the input document as

$$p'(w) = p_{gen}p(w)^{\alpha} + (1 - p_{gen}) \sum_{j:w_j=w} (a_{ij}^L)^{\beta},$$

where $p(w)$ is the target vocabulary distribution (estimated using Equation (5)) and a_{ij}^L is the attention for the final decoder layer L (estimated using Equation (3)). α and β are scaling parameters (used to stabilize convolutional learning) estimated as $\frac{\log(|T'|)}{\log(|T|)}$ and $\frac{\log(|T'|)}{\log(\text{EncLen})}$, respectively (where EncLen is the encoder length). CONVS2S+COPY uses p_{gen} to switch between generating a novel word from a fixed vocabulary by sampling from $p(w)$ or copying a word from the source text by sampling from the attention distribution a_{ij}^L.

XSum documents
T1: murder, charge, court, police, arrest, guilty, sentence, boy, bail, space, crown, trial
T2: abuse, church, bishop, child, catholic, gay, pope, school, christian, priest, cardinal
T3: council, people, government, local, housing, home, house, property, city, plan, authority
T4: party, clinton, 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
Newsroom Abstractive documents
T1: fund, investment, firm, asset, capital, financial, corporate, management, return, profit, equity
T2: building, design, build, square, space, office, architect, center, architecture, interior, project
T3: award, parade, beverly, actress, annual, hills, star, red, hollywood, carpet, premiere, pose
T4: company, business, customer, industry, consumer, service, product, revenue, fortune, startup
T5: military, force, afghanistan, government, security, troops, war, country, taliban, attack, army
T6: party, government, minister, leader, political, prime, election, vote, power, country, parliament

Table 5: Example topics learned by an LDA model on XSum and Newsroom documents (training portion).

over topics which we used to estimate $\mathbf{t}'$; 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 for XSum and 256 topics for Newsroom. LDA models were trained with α[5] set to a fixed normalized asymmetric prior of 1/number of topics; we let the model learn an asymmetric prior η[6] from the data. Table 5 shows some of the topics learned by the LDA models.[7]

For all RNN-based models[8] (Seq2Seq, PtGen, and PtGen+Covg) we used the best settings reported on the CNN and DailyMail data (See et al., 2017) All models had 256 dimensional hidden states and 128 dimensional word embeddings. They were trained using Adagrad (Duchi et al., 2011) with learning rate set to 0.15 and an initial accumulator value of 0.1. We used gradient clipping with a maximum gradient norm of 2, without any regularization and the loss on the validation set to implement early stopping. All models trained on the XSum and Newsroom datasets have the same settings.

For all convolutional models[9] (ConvS2S, ConvS2S+Copy, and T-ConvS2S) we used 512 dimensional hidden states, word embeddings and position embeddings for XSum and 256 dimensional hidden states, word embeddings and position embeddings for Newsroom. All models were trained 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 for ConvS2S and T-ConvS2S, and 0.02 for ConvS2S+Copy.[10] Once the validation perplexity stopped improving, we 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

---

5. α controls the prior distribution over topics for individual documents.

6. η controls the prior distribution over words for individual topics.

7. We used a multi-core implementation of LDA made available by `gensim` at `https://radimrehurek.com/gensim/models/ldamulticore.html`.

8. We used the code available at `https://github.com/abisee/pointer-generator`.

9. We used the code available at `https://github.com/facebookresearch/fairseq-py`.

10. ConvS2S+Copy failed to converge with learning rate greater than 0.02.

the convolutional blocks. Gradients were normalized by the number of non-padding tokens per mini-batch. We also used layer normalization and weight normalization for all layers except for lookup tables to stabilize learning.

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) in all cases.

5.3 Evaluation

We evaluated summarization quality automatically using F_1 ROUGE (Lin & 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.[11] In addition to ROUGE which can be misleading when used as the only means to assess the informativeness of summaries (Schluter, 2017; Hardy et al., 2019), we also evaluated system output by eliciting human judgments in two ways.

In our first experiment, participants were asked to compare summaries produced by different systems. The study was conducted on the Amazon Mechanical Turk platform using *Best-Worst Scaling* (BWS; Louviere & 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 & Mohammad, 2017). Participants were presented with a document and summaries generated from three systems and were asked to decide which summary was the *best* and which one was the *worst* in order of informativeness (does the summary capture important information in the document?) and fluency (is the summary written in well-formed English?). In two separate studies, we randomly selected 50 documents from the XSum and Newsroom-Abs test set. We compared all possible system pairs for each document and 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). Figures 3 and 4 show example summaries from the XSum and Newsroom datasets used for this study.

For our second experiment we used a question-answering (QA) paradigm (Clarke & Lapata, 2010; Narayan et al., 2018b) to assess the degree to which the models retain key information from the document. We wrote fact-based questions for each document, just by reading the reference 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. 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 questions for each summary. 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. We followed the scoring mechanism

---

11. 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
CONVS2S+COPY	29.80	10.10	24.10
T-CONVS2S ($\text{enc}_{t'}$)	31.71	11.38	25.56
T-CONVS2S ($\text{enc}_{(t', t_D)}$)	31.61	11.30	25.51
T-CONVS2S ($\text{enc}_{t'}, \text{dec}_{t_D}$)	31.71	11.34	25.61
T-CONVS2S ($\text{enc}_{(t', t_D)}, \text{dec}_{t_D}$)	31.89	11.54	25.75

Table 6: 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 models are in the middle block, and convolutional systems are in the bottom block.

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.

We used the same 100 documents (50 documents for XSum and 50 documents for Newsroom) as in our first elicitation study. For XSum, we created 100 questions in total; we wrote two fact-based questions per document. For Newsroom summaries, we were often not able to write more than one fact-based questions per document. Consequently, we only have 61 questions in total. Figures 3 and 4 show example summaries and their corresponding questions for XSum and Newsroom, respectively.

6. Results

In this section we present results for our model and comparison systems on the XSum dataset; we also discuss experiments on Newsroom (Grusky et al., 2018) and analyze quantitative and qualitative differences between the two datasets.

6.1 Results on the XSum Dataset

Automatic Evaluation Table 6 summarizes our ROUGE-based results. As can be seen, 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 the 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.[12] We hypothesize that its superior performance stems from the ability to better represent document content (i.e., by capturing long-range dependencies). Surprisingly, ConvS2S+Copy, a ConvS2S enhanced with a "copying" mechanism obtains performance inferior to ConvS2S. Our analysis revealed that the multi-hop attention mechanism of ConvS2S is very effective in resolving the unknown (UNK) words, by simply copying the most attended word w_j (estimated using the average attention scores as $\operatorname{argmax}_{w_j} \sum_{l=1}^{L} a_{ij}^{L}/L$)) from the source text to replace an UNK word. The copy mechanism unnecessarily over-parametrizes the ConvS2S model leading to a drop in performance. For example, ConvS2S correctly resolves two subsequent UNK words to "Dick Advocaat" whereas ConvS2S+Copy incorrectly resolves them to "Dick Dick" as shown at the bottom of Figure 3. The Seq2Seq model without the copy mechanism is prone to generating random rare words (e.g., "Andre Mccormack" for the same example in Figure 3) or unresolved UNK words (see the top example in Figure 4).[13] while PtGen guides the model towards sampling words from the source text.

Table 6 also shows several variants of T-ConvS2S including an encoder network enriched with information about how topical a word is on its own ($\mathrm{enc}_{t'}$) or in the document ($\mathrm{enc}_{(t',t_D)}$). 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 and ConvS2S+Copy. T-ConvS2S performs best when both encoder and decoder are constrained by the document topic ($\mathrm{enc}_{(t',t_D)}$,dec_{t_D}). In the remainder of the paper, we refer to this variant as T-ConvS2S.

How Abstractive are the Generated Summaries? We further assessed the extent to which various models are able to perform rewriting by generating genuinely abstractive summaries. Table 7 shows the proportion of novel n-grams for abstractive systems based on RNNs (Seq2Seq, PtGen and PtGen+Covg) and our convolutional models (ConvS2S, ConvS2S+Copy and T-ConvS2S). We omit extractive systems (lead and ext-oracle) as they are not capable of generating summaries from scratch with novel n-grams.

Overall, we observe that all abstractive models generate a fair amount of novel constructions that go beyond what is said in the source document. This result further supports our claim that XSum is an appropriate testbed for abstractive summarization. The three convolutional models show comparable proportions of novel n-grams, while RNN-based models show greater variance with Seq2Seq generating the highest proportion of novel n-grams. PtGen+Covg performs the least rewriting, followed by PtGen. Interestingly, PtGen

---

12. Statistical significance at the 95% confidence level is estimated using bootstrap resampling (Davison & Hinkley, 1997) with the official ROUGE script.

13. None of the models of See et al. (2017) resolves UNK by simply copying the most attended word from the source text. PtGen and PtGen+Covg rely on the copy mechanism to sample words from the extended target vocabulary, including source words.

LEAD	The Richmond Park and North Kingston MP said he was "honoured" after winning 70% of the 9,227 votes cast using an online primary system.	[9.8, 0.0, 9.8]
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]
SEQ2SEQ	Zac Goldsmith has been re-elected as the Conservative candidate for the general election in London, the party has announced.	[61.1, 23.5, 38.9]
PTGEN	UKIP leader Nigel Goldsmith has been elected as the new mayor of London to elect a new Conservative MP.	[45.7, 6.1, 28.6]
PTGEN+COVG	Zac Goldsmith has been re-elected as the Liberal Democrat candidate for the Labour party conference in Richmond, North Wales.	[44.4, 17.7, 27.8]
CONVS2S	London mayoral candidate Zac Goldsmith has been elected as the new mayor of London.	[53.3, 21.4, 26.7]
CONVS2S+COPY	Zac Goldsmith has been elected mayor of London's mayoral election.	[51.9, 24.0, 37.0]
T-CONVS2S	Former London mayoral candidate Zac Goldsmith has been chosen to stand in the London mayoral election.	[50.0, 26.7, 37.5]
GOLD	Zac Goldsmith will contest the 2016 London mayoral election for the Conservatives, it has been announced.	
Questions	(1) Who will contest for the Conservatives? (Zac Goldsmith) (2) For what election will he/she contest? (The London mayoral election)	
LEAD	The 68-year-old Dutchman was appointed in March, when the black cats were one point above the relegation zone.	[15.4, 0.0, 15.4]
EXT-ORACLE	North-east rivals Newcastle are the only team below them in the Premier League table.	[35.3, 18.8, 35.3]
SEQ2SEQ	Sunderland ladies have confirmed that former Sunderland striker Andre Mccormack has been sacked as the new manager of Championship Club Sunderland.	[20.0, 0.0, 15.0]
PTGEN	Sunderland have appointed former Sunderland boss Dick Advocaat as manager at the end of the season to sign a new deal.	[45.0, 10.5, 30.0]
PTGEN+COVG	Sunderland have sacked manager Dick Advocaat after eight games in charge of the Club's Premier League rivals Sunderland.	[36.8, 11.1, 31.6]
CONVS2S	Sunderland have sacked manager Dick Advocaat after less than three months in charge.	[25.0, 6.7, 18.8]
CONVS2S+COPY	Sunderland have appointed Dick Dick as their new manager on a three-year deal.	[18.2, 0.00, 12.1]
T-CONVS2S	Dick Advocaat has resigned as Sunderland manager until the end of the season.	[56.3, 33.3, 56.3]
GOLD	Dick Advocaat has resigned as Sunderland boss, with the team yet to win in the Premier League this season.	
Questions	(1) Who has resigned? (Dick Advocaat) (2) From what post has he/she resigned? (Sunderland boss)	

Figure 3: 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.

trained on XSum only copies 4% of 4-grams from the source document, 10% of trigrams, 27% of bigrams, and 73% of unigrams. This is in sharp contrast to PTGEN trained on CNN/DailyMail which copies more than 85% of 4-grams in the source document, 90% of

Models	% of novel n-grams in generated summaries			
	unigrams	bigrams	trigrams	4-grams
SEQ2SEQ	36.66	82.17	95.58	98.63
PTGEN	27.40	73.33	90.43	96.04
PTGEN+COVG	25.71	70.76	88.87	95.24
CONVS2S	31.28	79.50	94.28	98.10
CONVS2S+COPY	32.30	79.56	94.46	98.21
T-CONVS2S	30.73	79.18	94.10	98.03
GOLD	35.76	83.45	95.50	98.49

Table 7: Proportion of novel n-grams in summaries generated by various models on the XSum test set.

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

Table 8: System ranking according to human judgments and QA-based evaluation for the XSum dataset.

trigrams, 95% of bigrams, and 99% of unigrams (See et al., 2017). We should point out that the summaries being evaluated have on average comparable lengths: summaries generated by SEQ2SEQ, PTGEN, and PTGEN+COVG contain 23.02, 22.86, and 22.52 words, respectively; those generated by CONVS2S, CONVS2S+COPY, and T-CONVS2S have 20.07, 19.82 and 20.22 words, respectively, while GOLD summaries are the longest with 23.26 words.

Human Evaluation Recall that system generated summaries were evaluated in two studies one aimed at eliciting judgments of summary quality and the other following a question-answering paradigm. In both studies, participants were asked to evaluate summaries produced from the EXT-ORACLE baseline, PTGEN, the best performing RNN-based system according to ROUGE (see Table 6), CONVS2S, our topic-aware model T-CONVS2S, and the human-authored gold summary (GOLD). We did not include summaries from the LEAD or CONVS2S+COPY as they were significantly inferior to other models. Table 8 presents our results.

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

Models	R1	R2	RL
RANDOM	13.02	1.51	10.46
LEAD (Grusky et al., 2018)	13.76	2.42	11.29
LEAD	15.44	2.72	12.32
EXT-ORACLE	29.85	7.82	24.86
PTGEN (Grusky et al., 2018)	14.71	2.27	11.48
SEQ2SEQ	15.23	4.21	12.88
PTGEN	17.61	5.15	14.73
PTGEN+COVG	16.13	4.33	13.47
CONVS2S	16.77	5.57	14.54
CONVS2S+COPY	16.31	5.34	14.24
T-CONVS2S	16.97	5.56	14.70

Table 9: Results on NewsRoom-Abs 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 second block, and our convolutional abstractive systems are in the third block.

and PTGEN (using a one-way ANOVA with posthoc Tukey HSD tests; $p < 0.01$). All other differences are not statistically significant.

The rightmost column in Table 8 shows the results of the QA evaluation. Based on the 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 all systems are 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 Figure 3 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 Figure 3 second block) despite containing the correct answer "Dick Advocaat" due to the wrong context. T-CONVS2S is able to extract important entities from the document with the right theme.

6.2 Results on the Newsroom-Abs Dataset

We next examine whether our approach extends to other datasets with similar characteristics. Specifically, we examine the performance of the proposed model and related models on the abstractive portion of the Newsroom dataset (Newsroom-Abs; Grusky et al., 2018).

Automatic Evaluation Table 9 summarizes the results of our ROUGE-based evaluation. In addition to earlier discussed systems (see Section 5.1), we also include Grusky

et al.'s (2018) versions of lead and PtGen. Our lead selects the first 2 sentences to form the summary compared to the lead reported in Grusky et al. (2018) which selects the first 3 sentences. Our lead is more in line with the average number of sentences observed in the reference summaries which is 1.25 (see Table 2), and as result obtains better performance. We also found that PtGen (Grusky et al., 2018) was trained on the whole Newsroom dataset. To make a fair comparison, we report results with models trained on the NewsRoom-Abs portion of the dataset only. The discrepancy in the results between our PtGen and the PtGen model reported in Grusky et al. (2018) can be explained by the usage of different training sets.

Our convolutional models, ConvS2S, ConvS2S+Copy, and T-ConvS2S, significantly outperform the lead baselines.[14] ConvS2S significantly outperforms Seq2Seq, its RNN counterpart but lags behind PtGen. T-ConvS2S performs competitively against PtGen on R1 and RL scores and better on R2 (5.56 vs 5.15). The superior performance of T-ConvS2S over ConvS2S confirms our hypothesis that T-ConvS2S enhanced with topic information is better at identifying pertinent content and generating informative summaries. The worse performance of ConvS2S+Copy against ConvS2S further supports our claim that the multi-hop attention mechanism already in place in ConvS2S is very effective at resolving UNKs simply by copying the most attended words from the source. However, this is not case with the RNN-based models (See et al., 2017). As previously discussed, Seq2Seq is prone to generating UNKs (see the top block in Figure 4), while PtGen corrects for this with the copy mechanism. The summaries generated by Seq2Seq have a total of 11,418 UNK words, whil PtGen only generates 4,467 UNK words on the Newsroom-Abs test set.[15]

Interestingly, all abstractive summaries fall behind the extractive oracle (ext-oracle) on this dataset. In contrast, most abstractive models (PtGen, ConvS2S, ConvS2S+Copy and T-ConvS2S) were able to outperform ext-oracle on Xsum. This suggests that Newsroom-Abs still has some bias towards extractive methods and improved sentence selection would bring further performance gains for extractive approaches on this dataset. Models trained on XSum are better at generating good quality abstracts, e.g., T-ConvS2S achieves ROUGE scores of 31.89/11.54/25.75 compared to 16.97/5.56/14.70 on Newsroom-Abs. There are two reasons for this. Firstly, Newsroom has a great variety of summarization styles due to its collection from multiple news outlets; it is hard for abstractive methods to effectively model this. Secondly, XSum reference summaries are more representative of document content, whereas, Newsroom-Abs summaries are more indicative (see Section 6.3 for examples of reference summaries from XSum and Newsroom). It is probably harder for abstractive models to generate indicative summaries describing the source text rather directly presenting its content.

Human Evaluation For both evaluation protocols, participants were asked to assess summaries produced from the ext-oracle baseline, PtGen, ConvS2S, our topic-aware model T-ConvS2S, and the human-authored gold summary (gold). We did not include summaries from the lead, or ConvS2S+Copy as they were significantly inferior to other models. Table 10 presents our results.

---

14. Again, we use the `pyrouge` script to estimate statistical significance.

15. Surprisingly, none of the RNN-based abstractive models generates UNK words on XSum. We believe this is due to the smaller vocabulary size (81,092 for XSum vs. 157,939 for Newsroom-Abs; Table 2).

LEAD	Recently, I had to clear out all our kitchen cabinets to prep for a renovation. This taught me a few things:	[6.1, 0.0, 5.1]
EXT-ORACLE	Once I started trying to avoid chemical cleaners, I think my stash doubled. Here's how to make them: you can use either vinegar or lemon juice in this recipe.	[14.3, 0.0, 10.4]
SEQ2SEQ	A new recipe for [UNK] [UNK] is a recipe for [UNK] [UNK], [UNK], [UNK], [UNK], [UNK], [UNK], [UNK] and [UNK].	[10.0, 0.0, 9.8]
PTGEN	DIY recipes can help you find a way to enjoy the perfect lemon. Here's how to get it.	[13.3, 0.0, 11.8]
PTGEN+COVG	DIY simple recipes from your kitchen, and your kitchen cabinets are n't just a few things to do.	[6.9, 0.0, 6.2]
CONVS2S	From DIY furniture to DIY kitchen products, here's how to get rid of your kitchen.	[7.7, 0.0, 7.4]
CONVS2S+COPY	How to get rid of your kitchen.	[10.5, 0.0, 10.0]
T-CONVS2S	DIY DIY products for cleaning up your kitchen.	[10.5, 0.0, 10.0]
GOLD	These effective natural cleaners can be made at reasonable costs	
Questions	(1) What type of cleaners are being discussed in this article? (Natural cleaners)	
LEAD	During a shaky performance in the witness box, Pistorius changed his story and his defence from previous statements read at his bail hearing and the start of his trial; adding the last words he exchanged with Steenkamp and that he heard the lavatory door slam, and first saying he fired at an intruder, then that he fired by mistake. Oscar Pistorius sits in the dock during his trial in Pretoria (Reuters)	[26.3, 4.8, 17.7]
EXT-ORACLE	On his account of challenging an intruder, Pistorius admitted he didn't check the sound he heard with his girlfriend lying awake next to him. Witness Dr Johan Stipp told the court he saw a light on in Pistorius' bathroom when he heard shots fired; the athlete said he was "too scared" to switch the lights on until after the shooting. A psychologist said his reaction might seem "extraordinary" for an able-bodied person, but could be explained by his disability.	[33.3, 5.6, 19.2]
SEQ2SEQ	Oscar Pistorius's decision to remove his girlfriend Reeva Steenkamp in the face of the murder of Reeva Steenkamp has been charged with murder.	[31.2, 13.7, 20.1]
PTGEN	Five couples who have been killed in a shooting at a gun in Pretoria after the shooting death of the gun.	[16.4, 0.0, 9.9]
PTGEN+COVG	Five couples have been arrested in the dock trial of an intruder in the dock on the dock during a dock trial.	[23.3, 2.9, 12.3]
CONVS2S	A day in the life of armed America.	[11.3, 0.0, 7.0]
CONVS2S+COPY	Oscar Pistorius, who was shot in the shooting of Reeva Steenkamp, has been accused of killing his girlfriend, Reeva Steenkamp.	[34.9, 11.1, 22.5]
T-CONVS2S	A day after the shooting of Reeva Steenkamp, Oscar Pistorius has been accused of murdering his girlfriend Reeva Steenkamp in the shooting of Reeva Steenkamp.	[34.9, 14.1, 22.5]
GOLD	Oscar Pistorius is on trial for the premeditated murder of his girlfriend Reeva Steenkamp. The state say he killed her deliberately after an argument; the athlete says he believed she was an intruder. The following are 10 reasons why the judge might find him guilty, or innocent	
Questions	Who is on trial for the premeditated murder of his girlfriend? (Oscar Pistorius) Who was his girlfriend? (Reeva Steenkamp) What is the state's point of view? (he killed her deliberately after an argument)	

Figure 4: Example output summaries on the Newsroom Abstractive test set with [ROUGE-1, ROUGE-2 and ROUGE-L] scores, gold standard 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.

To our surprise, EXT-ORACLE summaries were considered best, whereas the human-authored summaries were ranked 2nd followed by PTGEN and T-CONVS2S. CONVS2S was ranked worst with the lowest score of −0.397. In line with our findings in Section 6.3, participants found EXT-ORACLE summaries to be more informative than human-authored summaries which are often indicative in nature. We carried out pairwise comparisons be-

Models	Score	QA
EXT-ORACLE	0.473	41.31
PTGEN	-0.047	20.98
CONVS2S	-0.397	11.97
T-CONVS2S	-0.160	23.28
GOLD	0.130	90.98

Table 10: System ranking according to human judgments and QA-based evaluation for the Newsroom-Abs dataset.

tween all models to assess whether system differences are statistically significant. The difference between T-CONVS2S and PTGEN is not statistically significant (using a one-way ANOVA with posthoc Tukey HSD tests; $p < 0.01$), while all other differences are.

The rightmost column in Table 10 shows the results of the QA evaluation. Based on the oracle extracts, participants can answer 41.31% of the questions correctly. Summaries generated by T-CONVS2S, PTGEN, and CONVS2S provide answers to 23.28%, 20.98%, and 11.97% of the questions, respectively. EXT-ORACLE performs best on both QA and judgment elicitation evaluations. However, we should point out that EXT-ORACLE has the advantage of selecting the best set of sentences (as determined by ROUGE) without any length constraints. Consequently, summaries generated by EXT-ORACLE tend to be longer with 38.84 words on average (see Figure 4 second block). Summaries generated by PTGEN, CONVS2S, and T-CONVS2S contain 23.72, 19.41 and 20.66 words, respectively, while GOLD summaries contain 23.26 words. Perhaps unsurprisingly T-CONVS2S, which was slightly lagging behind PTGEN on R1 and RL scores, performs better than both PTGEN and CONVS2S in terms of correctly answering questions. The QA evaluation shows that T-CONVS2S is able to generate informative summaries with pertinent information embedded in the appropriate context. Pairwise differences between systems are all statistically significant ($p < 0.01$) with the exception of T-CONVS2S and PTGEN, and, PTGEN and CONVS2S.

6.3 Informative vs. Indicative Summaries

Our experiments have so far revealed differences in the nature of XSum and Newsroom summaries. XSum summaries are often informative of the document content while Newsroom summaries are indicative, i.e., they describe the source text rather than directly presenting the information it contains. In this section, we provide further empirical support for this claim.

We conducted a human evaluation where participants were asked to rate reference summaries from both datasets. Specifically, they were presented with a document and its gold summary and asked to decide whether it was informative (i.e., it relayed pertinent content from the document), partially informative, or uninformative. The study was conducted on Amazon Mechanical Turk with the same 100 test documents used of our judgment elicitation and QA studies on XSum and Newsroom-Abs. We collected judgments from three different participants for each document. The order of documents and systems were randomized.

	Reference Summaries	Score
XSum	Match reports from Saturday's Scottish Premiership and Championship games.	1.67
	The case for raising US interest rates has "strengthened", the head of the Federal Reserve has said.	2.00
	Internet giant Amazon's owner Jeff Bezos has made an amazing underwater discovery.	2.00
	Deputy First Minister Martin Mcguinness pulled out of a trip to China this week due to medical advice, it has been revealed.	2.00
	The Conservatives have accused Labour and Plaid Cymru of being involved in an "unedifying squabble" over who to support if there is a hung parliament.	2.00
	Police have released CCTV images of a man they want to speak to following a racist assault on a Glasgow bus.	2.33
	A yellow 'be aware' weather warning was re-introduced in parts of Wales on Sunday.	2.33
	Ross County's Jim Mcintyre is the only Premiership boss in the running for PFA Scotland's manager of the year award.	2.33
	A council plans to employ its own staff to help young people with mental health problems.	2.33
	UK-born Oliver Hart and Bengt Holmstrom of Finland have won the Nobel Economics prize for work on Contract Theory.	2.33
	Leinster resisted a spirited Cardiff Blues revival to strengthen their position at the top of the Pro12 table.	2.33
	DNA analysis of a 45,000-year-old human has helped scientists pinpoint when our ancestors interbred with Neanderthals.	2.33
	Tributes have been paid to a former Glamorgan cricketer who was found dead at his Swansea flat on friday.	2.33
	A new visitor centre at the battle of Britain memorial – designed in the shape of a spitfire wing – is beginning to take shape.	2.67
	The first of the missing Nigerian schoolgirls to be rescued since her capture two years ago has had an emotional reunion with her mother.	2.67
Newsroom-Abs	The pair costarred in a Star Wars-themed sketch for an unaired TV show in 2008	1.33
	Things were beginning to get ugly in the Democratic race	1.33
	Installations will take place in 100 stores.	1.33
	Sean Penn, actor, director, hothead, misanthrope. So what's he still doing in haiti?	1.33
	Collection of all usatoday.com coverage of the Exorcist, including articles, videos, photos, and quotes.	1.33
	Bain wasn't just about private equity.	1.33
	An expert's advice for college students and parents	1.33
	The 62nd instalment in a weekly series that debunks the web's hoaxes, rumors and exaggerations.	1.67
	The ability to cope with risk and work insane hours comes in handy. Take it from these former financiers.	1.67
	The FED needs to stop this cat-and-mouse game and just say it isn't raising rates anytime soon, says Carol Roth.	2.00
	The government hopes everyone has exchanged their old notes by now.	2.00
	Washington Post food critic Tom Sietsema takes a fresh look at 13 restaurants he has previously reviewed. How do they fare the second time around?	2.00
	Biden's son had mild stroke – the Oval : tracking the Obama presidency	2.00
	The cost of home shopping could soar as top delivery company ups its charges	2.00
	Probiotic bacteria can affect behavior.	2.33

Figure 5: Summaries ranked from least to most informative from the XSum and Newsroom-Abs test sets. We also show their informativeness scores (higher is better).

Table 11 present the results of this study. We measure the informativeness of each dataset as the average score assigned by crowdworkers across summaries; a summary re-

Dataset	Informative	Part. Informative	Uninformative	Score
XSum	68.00	26.00	6.00	2.62
Newsroom-Abs	48.67	33.33	18.00	2.30

Table 11: XSum and Newsroom-Abs summaries and their informativeness. The middle column proportionately shows the number times a summary was judged "Informative", "Partially Informative", or "Uninformative." The last column shows the informativeness score for each dataset (higher is better).
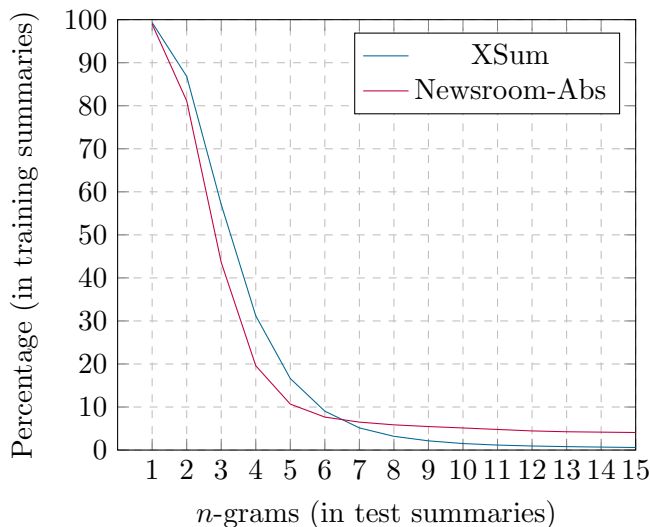

Figure 6: Percentage of n-grams in test summaries seen in training summaries.

ceives a score of 3 if it is deemed informative, 2 if it is partially informative, and 1 if it is uninformative. Therefore, the informativeness score for a dataset as a whole may vary from 1 to 3, with 1 being least informative and 3 being most informative. We also report the proportion of times a dataset was considered informative, partially informative, and uninformative. XSum reference summaries were mostly considered informative (68%), around a quarter of them (26%) were deemed partially informative, and only 6% were found uninformative. In comparison, less than half (48.67%) of Newsroom-Abs summaries were found informative, the remaining being either partially informative or uninformative. XSum achieved an informativeness score of 2.62 compared to 2.30 for the Newsroom dataset. We carried out pairwise comparisons to assess whether informativeness differences are statistically significant. We found that XSum is significantly more informative than Newsroom (using a one-way ANOVA with posthoc Tukey HSD tests; $p < 0.01$). Figure 5 shows the 15 best summaries from each dataset ranked from least to most informative.

Figure 6 shows the percentage of n-grams in test summaries which have been already seen in training summaries. For both datasets, as the size of n-grams increases, their chance of having been seen in the training summaries decreases rapidly. For XSum, this percentage drops to almost 0% for any n-grams larger than size 10. Interestingly, this is not the case for
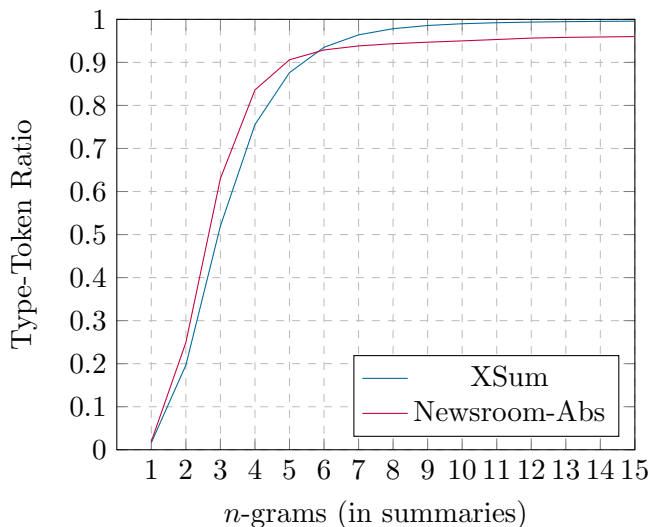
Figure 7: Type-Token Ratio for summary n-grams in the entire dataset.

Newsroom-Abs, where more than 4% of n-grams (sizes 10 to 15) in test summaries have been already seen in training summaries. This result suggests that Newsroom-Abs summaries are somewhat formulaic displaying a certain degree of repetition that goes beyond simple phrases. As an example consider the summary from Figure 5 "Collection of all usatoday.com coverage of the Exorcist, including articles, videos, photos, and quotes." which is rather generic and would apply to any movie, not just the Exorcist. In fact two crowdworkers labeled this summary as uninformative and one as partially informative (informativeness score is 1.33). Figure 7 shows the type-token ratio of different n-grams in gold summaries as a measure of how often constructions are being reused; a higher type-token ratio represents larger variation in terms of n-grams. XSum summaries exhibit more variation for n-grams of size larger than 5 corraborating our claim that Newsroom-Abs summaries are more repetitive.

7. Conclusions

In this paper we introduced the task of "extreme summarization" together with a large-scale dataset which pushes the boundaries of abstractive methods. We further proposed a novel "topic-aware" fully-convolutional deep learning model which is well-suited to extreme summarization. And designed a question-answering paradigm to assess the degree to which abstractive models retain key information from the document. 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. Finally, experimental results support our claim that extreme summarization is a good testbed for abstractive summarization; the task, as operationalized via our dataset, encourages models to create informative summaries which promote novel constructions and are less skewed towards extractive mechanisms.

Extreme summarization revisits interesting problems in abstractive summarization with the relatively simpler objective of generating single sentence summaries rather than multi-line summaries (Cohan et al., 2018; Liu et al., 2018; Yasunaga et al., 2019). Models trained for extreme summarization require document-level inference, abstraction, and paraphrasing to generate summaries which are informative and consistent with the input document. Throughout this paper we have argued that our model is better suited for this task than recurrent abstractive models due to its ability to foreground pertinent content using topic vectors and model long-range dependencies using a multi-layer convolutional architecture. In the future, we would like to create more linguistically-aware encoders and decoders incorporating co-reference and entity linking. It would also be interesting to use contextualised word representations (Peters et al., 2018; Devlin et al., 2018; Yang et al., 2019) to enhance modeling of long-range dependencies within our model.

Beyond generating single sentences, we would like to adapt our method to create multi-sentence summaries. For instance, this would allow us to assess whether our model's ability to capture long-range dependencies translates to more readable and coherent summaries. Finally, our method might be relevant to summarizing texts from other domains, e.g., generating multi-line abstracts of scientific articles (Cohan et al., 2018; Yasunaga et al., 2019), creating Wikipedia pages in a multi-document summarization setting (Liu et al., 2018), and aggregating product or movie reviews (Wang & Ling, 2016; Angelidis & Lapata, 2018).

Acknowledgments

References

Almeida, M. B., & Martins, A. F. T. (2013). Fast and robust compressive summarization with dual decomposition and multi-task learning. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pp. 196–206, Sofia, Bulgaria.

Angelidis, S., & Lapata, M. (2018). Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3675–3686.

Bahdanau, D., Cho, K., & Bengio, Y. (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.

Barzilay, R., & Elhadad, M. (1997). Using lexical chains for text summarization. In *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, pp. 10–17, Madrid, Spain.

Barzilay, R., Elhadad, N., & McKeown, K. R. (2002). Inferring strategies for sentence ordering in multidocument news summarization. *Journal of Artificial Intelligence Research*, *17*(1), 35–55.

Berg-Kirkpatrick, T., Gillick, D., & Klein, D. (2011). Jointly learning to extract and compress. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 481–490, Portland, Oregon, USA.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *The Journal of Machine Learning Research*, *3*, 993–1022.

Carbonell, J., & Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 335–336, Melbourne, Australia.

Celikyilmaz, A., Bosselut, A., He, X., & Choi, Y. (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.

Chen, Q., Zhu, X., Ling, Z., Wei, S., & Jiang, H. (2016). Distraction-based neural networks for modeling documents. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pp. 2754–2760, New York, USA.

Chen, Y.-C., & Bansal, M. (2018). Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 675–686, Melbourne, Australia.

Cheng, J., & Lapata, M. (2016). Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 484–494, Berlin, Germany.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods on Natural Language Processing*, pp. 1724–1734, Doha, Qatar.

Clarke, J., & Lapata, M. (2010). Discourse constraints for document compression. *Computational Linguistics*, *36*(3), 411–441.

Cohan, A., Dernoncourt, F., Kim, D. S., Bui, T., Kim, S., Chang, W., & Goharian, N. (2018). A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 615–621, New Orleans, Louisiana.

Dauphin, Y. N., Fan, A., Auli, M., & Grangier, D. (2017). Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 933–941, Sydney, Australia.

Davison, A. C., & Hinkley, D. V. (1997). *Bootstrap Methods and Their Application*. Cambridge University Press.

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *CoRR, abs/1810.04805.*

Dieng, A. B., Wang, C., Gao, J., & Paisley, J. (2017). TopicRNN: A recurrent neural network with long-range semantic dependency. In *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France.

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M., & Hon, H. (2019). Unified language model pre-training for natural language understanding and generation. *CoRR, abs/1905.03197.*

Dong, Y., Shen, Y., Crawford, E., van Hoof, H., & Cheung, J. C. K. (2018). BanditSum: Extractive summarization as a contextual bandit. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3739–3748, Brussels, Belgium.

Dorr, B., Zajic, D., & Schwartz, R. (2003). Hedge trimmer: A parse-and-trim approach to headline generation. In *Proceedings of the Text Summarization Workshop at NAACL*, pp. 1–8, Edmonton, Canada.

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research, 12*, 2121–2159.

Durrett, G., Berg-Kirkpatrick, T., & Klein, D. (2016). Learning-based single document summarization with compression and anaphoricity constraints. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1998–2008, Berlin, Germany.

Erkan, G., & Radev, D. R. (2004). LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research, 22*(1), 457–479.

Fan, A., Grangier, D., & Auli, M. (2017). Controllable abstractive summarization. *CoRR, abs/1711.05217.*

Fan, A., Lewis, M., & Dauphin, Y. (2018). Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, Melbourne, Australia.

Filatova, E., & Hatzivassiloglou, V. (2004). Event-based extractive summarization. In *Proceedings of ACL Workshop on Text Summarization Branches Out*, pp. 104–111, Barcelona, Spain.

Gehring, J., Auli, M., Grangier, D., & Dauphin, Y. (2017a). A convolutional encoder model for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 123–135, Vancouver, Canada.

Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y. N. (2017b). Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70, pp. 1243–1252, Sydney, Australia.

Gehrmann, S., Deng, Y., & Rush, A. (2018). Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4098–4109.

Ghosh, S., Vinyals, O., Strope, B., Roy, S., Dean, T., & Heck, L. (2016). Contextual LSTM (CLSTM) models for large scale NLP tasks. *CoRR, abs/1602.06291*.

Grusky, M., Naaman, M., & Artzi, Y. (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.

Gu, J., Lu, Z., Li, H., & Li, V. O. (2016). Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1631–1640, Berlin, Germany.

Hardy, H., Narayan, S., & Vlachos, A. (2019). HighRES: Highlight-based reference-less evaluation of summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy.

Harman, D., & Over, P. (2004). The effects of human variation in duc summarization evaluation. In *Text Summarization Branches Out*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, USA.

Hermann, K. M., Kočiský, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., & Blunsom, P. (2015). Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28*, pp. 1693–1701. Morgan, Kaufmann.

Hsu, W.-T., Lin, C.-K., Lee, M.-Y., Min, K., Tang, J., & Sun, M. (2018). A unified model for extractive and abstractive summarization using inconsistency loss. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 132–141, Melbourne, Australia.

Hu, B., Chen, Q., & Zhu, F. (2015). LCSTS: A large scale chinese short text summarization dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1967–1972, Lisbon, Portugal.

Jing, H. (2002). Using hidden Markov modeling to decompose human-written summaries. *Computational Linguistics, 28*(4), 527–544.

Kim, B., Kim, H., & Kim, G. (2018). Abstractive summarization of reddit posts with multi-level memory networks. *CoRR, abs/1811.00783*.

Kiritchenko, S., & Mohammad, S. (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*, pp. 465–470, Vancouver, Canada.

Koupaee, M., & Wang, W. Y. (2018). WikiHow: A large scale text summarization dataset. *CoRR, abs/1810.09305*.

Kågebäck, M., Mogren, O., Tahmasebi, N., & Dubhashi, D. (2014). Extractive summarization using continuous vector space models. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pp. 31–39, Gothenburg, Sweden.

Kryściński, W., Paulus, R., Xiong, C., & Socher, R. (2018). Improving abstraction in text summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1808–1817, Brussels, Belgium.

Kupiec, J., Pedersen, J., & Chen, F. (1995). A trainable document summarizer. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 406–407, Seattle, Washington, USA.

Li, C., Xu, W., Li, S., & Gao, S. (2018). Guiding generation for abstractive text summarization based on key information guide network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 55–60, New Orleans, Louisiana.

Li, L., Zhou, K., Xue, G.-R., Zha, H., & Yu, Y. (2009). Enhancing diversity, coverage and balance for summarization through structure learning. In *Proceedings of the 18th international Conference on World Wide Web*, pp. 71–80, Madrid, Spain.

Li, P., Bing, L., & Lam, W. (2018a). Actor-critic based training framework for abstractive summarization. *CoRR, abs/1803.11070*.

Li, W., Xiao, X., Lyu, Y., & Wang, Y. (2018b). Improving neural abstractive document summarization with explicit information selection modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1787–1796, Brussels, Belgium.

Lin, C. Y., & Hovy, E. (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*, pp. 71–78, Edmonton, Canada.

Liu, P. J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., & Shazeer, N. (2018). Generating wikipedia by summarizing long sequences. In *Proceedings of the 6th International Conference on Learning Representations*, Vancouver Canada.

Liu, Y. (2019). Fine-tune BERT for extractive summarization. *CoRR, abs/1903.10318*.

Louviere, J. J., Flynn, T. N., & Marley, A. A. J. (2015). *Best-worst scaling: Theory, methods and applications*. Cambridge University Press.

Louviere, J. J., & Woodworth, G. G. (1991). Best-worst scaling: A model for the largest difference judgments. *University of Alberta: Working Paper*, -.

Mani, I. (2001). *Automatic Summarization*. Natural language processing. John Benjamins Publishing Company.

Martins, A., & Smith, N. A. (2009). Summarization with a joint model for sentence extraction and compression. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*, pp. 1–9, Boulder, Colorado.

Mendes, A., Narayan, S., Miranda, S., Marinho, Z., Martins, A. F. T., & Cohen, S. B. (2019). Jointly extracting and compressing documents with summary state representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3955–3966, Minneapolis, Minnesota.

Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into texts. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 404–411, Barcelona, Spain.

Mikolov, T., & Zweig, G. (2012). Context dependent recurrent neural network language model. In *Proceedings of the Spoken Language Technology Workshop*, pp. 234–239. IEEE.

Nallapati, R., Zhai, F., & Zhou, B. (2017). SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pp. 3075–3081, San Francisco, California USA.

Nallapati, R., Zhou, B., Santos, C. d., Gulcehre, C., & Xiang, B. (2016). Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 280–290, Berlin, Germany.

Napoles, C., Gormley, M., & Van Durme, B. (2012). Annotated gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction at NAACL*, pp. 95–100, Montreal, Canada.

Narayan, S., Cardenas, R., Papasarantopoulos, N., Cohen, S. B., Lapata, M., Yu, J., & Chang, Y. (2018a). Document modeling with external attention for sentence extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, Melbourne, Australia.

Narayan, S., Cohen, S. B., & Lapata, M. (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.

Narayan, S., Papasarantopoulos, N., Cohen, S. B., & Lapata, M. (2017). Neural extractive summarization with side information. *CoRR, abs/1704.04530*.

Nenkova, A. (2005). Automatic text summarization of newswire: Lessons learned from the Document Understanding Conference. In *Proceedings of the 29th National Conference on Artificial Intelligence*, pp. 1436–1441, Pittsburgh, Pennsylvania, USA.

Nenkova, A., & McKeown, K. (2011). Automatic summarization. *Foundations and Trends in Information Retrieval, 5*(2–3), 103–233.

Nenkova, A., Vanderwende, L., & McKeown, K. (2006). A compositional context sensitive multi-document summarizer: Exploring the factors that influence summarization. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 573–580, Seattle, Washington, USA.

Over, P., Dang, H., & Harman, D. (2007). Duc in context. *Information Processing and Management, 43*(6), 1506–1520.

Parveen, D., Ramsl, H.-M., & Strube, M. (2015). Topical coherence for graph-based extractive summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1949–1954, Lisbon, Portugal.

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning*, pp. 1310–1318, Atlanta, GA, USA.

Pasunuru, R., & Bansal, M. (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.

Paulus, R., Xiong, C., & Socher, R. (2018). A deep reinforced model for abstractive summarization. In *Proceedings of the 6th International Conference on Learning Representations*, Vancouver, BC, Canada.

Perez-Beltrachini, L., Liu, Y., & Lapata, M. (2019). Generating summaries with topic templates and structured convolutional decoders. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2227–2237, New Orleans, Louisiana.

Qazvinian, V., Radev, D. R., Mohammad, S., Dorr, B. J., Zajic, D. M., Whidby, M., & Moon, T. (2013). Generating extractive summaries of scientific paradigms. *Journal of Artificial Intelligence Research*, *46*(1), 165–201.

Radev, D., Allison, T., Blair-Goldensohn, S., Blitzer, J., Çelebi, A., Dimitrov, S., Drabek, E., Hakim, A., Lam, W., Liu, D., Otterbacher, J., Qi, H., Saggion, H., Teufel, S., Topper, M., Winkel, A., & Zhang, Z. (2004). MEAD — A platform for multidocument multilingual text summarization. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pp. 699–702, Lisbon, Portugal.

Rush, A. M., Chopra, S., & Weston, J. (2015). A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 379–389, Lisbon, Portugal.

Sandhaus, E. (2008). The New York Times Annotated Corpus. *Linguistic Data Consortium, Philadelphia*, *6*(12).

Schilder, F., & Kondadadi, R. (2008). FastSum: Fast and accurate query-based multi-document summarization. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics and HLT: Short Papers*, pp. 205–208, Columbus, Ohio, USA.

Schluter, N. (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*, pp. 41–45, Valencia, Spain.

See, A., Liu, P. J., & Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 1073–1083, Vancouver, Canada.

Shen, D., Sun, J.-T., Li, H., Yang, Q., & Chen, Z. (2007). Document summarization using conditional random fields. In *Proceedings of the 20th International Joint Conference on Artifical intelligence*, pp. 2862–2867, Hyderabad, India.

Shi, X., Knight, K., & Yuret, D. (2016). Why neural translations are the right length. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2278–2282, Austin, Texas.

Song, K., Tan, X., Qin, T., Lu, J., & Liu, T. (2019). MASS: Masked sequence to sequence pre-training for language generation. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pp. 5926–5936, Long Beach, US.

Spärck Jones, K. (2007). Automatic summarising: The state of the art. *Information Processing and Management*, *43*(6), 1449–1481.

Sukhbaatar, S., szlam, a., Weston, J., & Fergus, R. (2015). End-to-end memory networks. In *Advances in Neural Information Processing Systems 28*, pp. 2440–2448. Morgan, Kaufmann.

Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on International Conference on Machine Learning*, pp. 1139–1147, Atlanta, GA, USA.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, pp. 3104–3112. Morgan, Kaufmann.

Tan, J., & Wan, X. (2017). Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 1171–1181, Vancouver, Canada.

Teufel, S., & Moens, M. (1997). Sentence extraction as a classification task. In *Proceedings of ACL Workshop on Intelligent and Scalable Text Summarization*, pp. 58–65, Madrid, Spain.

Tu, Z., Lu, Z., Liu, Y., Liu, X., & Li, H. (2016). Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 76–85, Berlin, Germany.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. Morgan, Kaufmann.

Vinyals, O., Fortunato, M., & Jaitly, N. (2015). Pointer networks. In *Advances in Neural Information Processing Systems 28*, pp. 2692–2700. Morgan, Kaufmann.

Wan, X. (2010). Towards a unified approach to simultaneous single-document and multi-document summarizations. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pp. 1137–1145, Beijing, China.

Wang, L., & Ling, W. (2016). Neural network-based abstract generation for opinions and arguments. In *Proceedings of the 2016 Conference of the North American Chapter*

of the Association for Computational Linguistics: Human Language Technologies, pp. 47–57, San Diego, California.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning, 8*(3-4), 229–256.

Woodsend, K., & Lapata, M. (2010). Automatic generation of story highlights. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 565–574, Uppsala, Sweden.

Woodsend, K., & Lapata, M. (2012). Multiple aspect summarization using integer linear programming. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 233–243, Jeju Island, Korea.

Wu, Y., & Hu, B. (2018). Learning to extract coherent summary via deep reinforcement learning. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pp. 5602–5609, New Orleans, USA.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J. G., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized autoregressive pretraining for language understanding. *CoRR, abs/1906.08237*.

Yasunaga, M., Kasai, J., Zhang, R., Fabbri, A., Li, I., Friedman, D., & Radev, D. (2019). SciSummNet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, Honolulu, Hawaii, USA.

Yasunaga, M., Zhang, R., Meelu, K., Pareek, A., Srinivasan, K., & Radev, D. (2017). Graph-based neural multi-document summarization. In *Proceedings of the 21st Conference on Computational Natural Language Learning*, pp. 452–462, Vancouver, Canada.

Yih, W. T., Goodman, J., Vanderwende, L., & Suzuki, H. (2007). Multi-document summarization by maximizing informative content-words. In *Proceedings of the 20th International Joint Conference on Artifical intelligence*, pp. 1776–1782, Hyderabad, India.

Yin, W., & Pei, Y. (2015). Optimizing sentence modeling and selection for document summarization. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pp. 1383–1389, Buenos Aires, Argentina.

Zhang, J., Yao, J.-g., & Wan, X. (2016). Towards constructing sports news from live text commentary. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1361–1371, Berlin, Germany.

Zhang, X., Lapata, M., Wei, F., & Zhou, M. (2018). Neural latent extractive document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 779–784, Brussels, Belgium.

Zhou, Q., Yang, N., Wei, F., & Zhou, M. (2017). Selective encoding for abstractive sentence summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 1095–1104, Vancouver, Canada.