On Extractive and Abstractive Neural Document Summarization with Transformer Language Models

Sandeep Subramanian[1,2,3,*] **, Raymond Li**[1,*]**, Jonathan Pilault**[1,2,4,*]**, Christopher Pal**[1,2,4,5]
[1]Element AI, [2]Montréal Institute for Learning Algorithms, [3]Université de Montréal,
[4]École Polytechnique de Montréal, [5]Canada CIFAR AI Chair
[1]{jonathan.pilault}@elementai.com

Abstract

We present a method to produce abstractive summaries of long documents that exceed several thousand words via neural abstractive summarization. We perform a simple extractive step before generating a summary, which is then used to condition the transformer language model on relevant information before being tasked with generating a summary. We show that this extractive step significantly improves summarization results. We also show that this approach produces more abstractive summaries compared to prior work that employs a copy mechanism while still achieving higher rouge scores. *Note: The abstract above was not written by the authors, it was generated by one of the models presented in this paper.*

Introduction

Language models (LMs) are trained to estimate the joint probability of an arbitrary sequence of words or characters using a large corpus of text. They typically factorize the joint distribution of tokens $p(x_1, x_2 \dots x_n)$ into a product of conditional probabilities $\prod_i^n p(x_i|x_{<i})$. It is possible to use n-gram based models to estimate these conditional probabilities via counts, relying on Markovian assumptions. However, Markovian assumptions and the curse of dimensionality make it harder for n-gram LMs to model long range dependencies and learn smooth functions that can learn similarities between words in the vocabulary. This has led to a preference for recurrent or feed-forward neural language models (Bengio et al. 2003; Mikolov et al. 2010) in recent years due to to their ability to learn expressive conditional probability distributions (Radford et al. 2019).

The sequence-to-sequence (seq2seq) paradigm (Sutskever, Vinyals, and Le 2014) uses language models that learn the conditional probability of one sequence given another. Here, a language model serves as a "decoder" that is typically conditioned on a representation of an input sequence produced by an encoder neural network. These types of encoder-decoder architectures have been particularly successful when applied to problems such as machine translation (Bahdanau, Cho, and Bengio 2014) and abstractive summarization (Rush, Chopra, and Weston 2015). The encoder and conditional decoder language models are often paramaterized as recurrent neural networks (RNNs). Attention mechanisms (Bahdanau, Cho, and Bengio 2014) are used in the decoder to provide more informative conditioning on the representations produced by the encoder and to ease gradient flow into the encoder. RNNs however, are limited by their sequential nature, making them 1) difficult to optimize and learn for long sequences with long range dependencies (Hochreiter 1998; Pascanu, Mikolov, and Bengio 2013), and 2) hard to parallelize on modern hardware like GPUs, limiting their scalability.

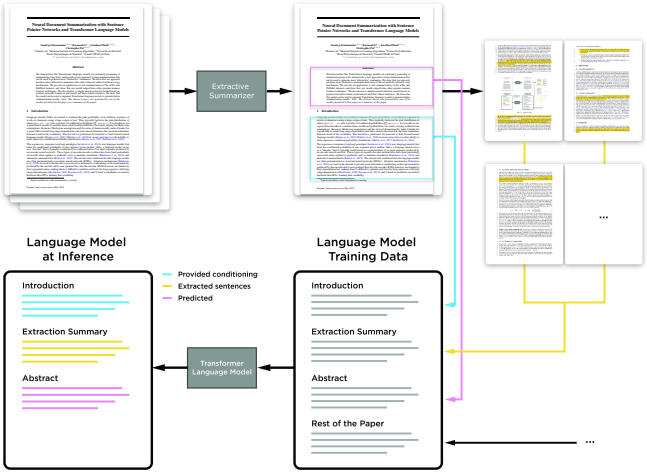

Figure 1: Proposed model for abstractive summarization of a scientific article. An older version of this paper is shown as the reference document. First, a sentence pointer network extracts important sentences from the paper. Next, these sentences are provided along with the whole scientific article to be arranged in the following order: Introduction, extracted Sentences, abstract & the rest of the paper. A transformer language model is trained on articles organized in this format. During inference, the introduction and the extracted sentences are given to the language model as context to generate a summary. In domains like news and patent documents, the introduction is replaced by the entire document.

*Equal contribution, order determined by coin flip

There has therefore been a recent shift towards feedforward architectures for sequential data, such as convolutional models (Kalchbrenner et al. 2016; Van Den Oord et al. 2016; Gehring et al. 2017) or fully attentive models popularized by architectures known as transformers (Vaswani et al. 2017). These techniques have a logarithmic or constant path length (as opposed to linear in RNNs) between a network's output and any of its inputs, making gradient flow much easier, thereby opening up the possibility of learning very long term dependencies.

The abstractive summarization of news or scientific papers typically requires encoding and generating hundreds or thousands of words. Recent work by (Radford et al. 2019) (GPT-2) has demonstrated that transformers with a large receptive field and trained on a lot of data yield language models that are capable of capturing long range dependencies.

If one is interested in creating a coherent, high quality summary of long documents, such GPT-like architectures possess many desirable properties. Their results also show that unconditional language models can implicitly learn to perform summarization or machine translation as a consequence of the data on which it is trained. If the data is formatted sequentially into different aspects of a document (introduction, body, summary), each divided by "tl;dr", the model can be coaxed to generate one of these aspects. For example, the model can be made to solve a summarization task by presenting it similarly formatted data at test time; i.e. a document's introduction and body followed by "tl;dr" that will generate an abstract from a language model conditioned on this context.

In this work, we take this idea a step further by doing away with the sequence-to-sequence paradigm and formatting data for abstractive summarization in a manner that transformer language models can make use all of the available data present in the documents and their summaries (akin to language model pre-training on mono-lingual data in machine translation (Gulcehre et al. 2015)). Specifically, we use a single GPT-like Transformer LM (TLM) trained on documents followed by their summaries. During inference, we generate from the LM, conditioned on the document (see figure 1). Unlike most previous approaches to neural abstractive summarization, we do not use a seq2seq formulation with an explicit encoder and decoder for word generation. We split the task in two: an extractive step and an abstractive step (Chen and Bansal 2018; Gehrmann, Deng, and Rush 2018). To deal with extremely long documents that exceed several thousand words, we first perform sentence extraction using two different hierarchical document models - one based on pointer networks (Vinyals, Fortunato, and Jaitly 2015), similar to the variant proposed in (Chen and Bansal 2018) and the other based on a sentence classifier (Nallapati, Zhai, and Zhou 2017). This extracts important sentences from the document (described in section) that can be used to better condition the transformer LM on relevant information before being tasked with generating a summary. We show that this extractive step significantly improves summarization results.

The contributions of this work are two fold:

- We demonstrate that transformer language models are surprisingly effective at summarizing long scientific articles and outperform typical seq2seq approaches, even without a copy mechanism.
- We show that that our approach produces more "abstractive" summaries compared to prior work that employs a copy mechanism (See, Liu, and Manning 2017) while still achieving higher ROUGE scores.

Related Work

Automatic summarization systems seek to condense the size of a piece of text while preserving most of its important information content and meaning. The earliest attempts at automatic summarization focused on extractive techniques, which finds words or sentences in a document that capture its most salient content. In the past, various similarity scores based on specific sentence features (keywords, position, length, frequency, linguistic) and metrics (structure-based, vector-based and graph-based) were employed to estimate salience (Steinberger and Jezek 2004; Erkan and Radev 2004) between a sentence in a document and its reference summary. More recently, with advances in distributed representations of words, phrases and sentences, researchers have proposed to use these to compute similarity scores. Such techniques were further refined by (Nallapati, Zhou, and Ma 2016; Cheng and Lapata 2016; Chen and Bansal 2018) with encoder-decoder architectures - the representations learned by the encoder are used to choose the most salient sentences. (Cheng and Lapata 2016) and (Nallapati, Zhou, and Ma 2016) trained encoder-decoder neural networks as a binary classifier to determine if each sentence in a document should belong to the extractive summary or not. (Nallapati et al. 2016) also present an alternative that can pick an unordered set of sentences from the source document to assemble an extractive summary. (Chen and Bansal 2018) use a pointer network (Vinyals, Fortunato, and Jaitly 2015) to sequentially pick sentences from the document that comprise its extractive summary.

Human summarizers have four common characteristics. They are able to (1) interpret a source document, (2) prioritize the most important parts of the input text, (3) paraphrase key concepts into coherent paragraphs and (4) generate diverse output summaries. While extractive methods are arguably well suited for identifying the most relevant information, such techniques may lack the fluency and coherency of human generated summaries. Abstractive summarization has shown the most promise towards addressing points (3) and (4) above. Abstractive generation may produce sentences not seen in the original input document. Motivated by neural network success in machine translation experiments, the attention-based encoder decoder paradigm has recently been widely studied in abstractive summarization (Rush, Chopra, and Weston 2015; Nallapati et al. 2016; Chopra, Auli, and Rush 2016). By dynamically accessing the relevant pieces of information based on the hidden states of the decoder during generation of the output sequence, the model revisits the input and attends to important information. The advantages of extractive, abstractive and attention-

based models were first combined in (Gu et al. 2016) with a copy mechanism for out-of-vocabulary words present in the source document. Similarly, (See, Liu, and Manning 2017) used the attention scores to calculate the probability of generating vs copying a word. A coverage mechanism was also added to penalize the attention score of previously attended words, diminishing the model's tendency to repeat itself.

Framework

Our model comprises two distinct and independently trainable components 1) a hierarchical document representation model that either points to or classifies sentences in a document to build an extractive summary 2) a transformer language model that conditions on the extracted sentences as well as a part of or the entire document.

Extractive Models

We describe the two neural extractive models used in this work in this section.

Hierarchical Seq2seq Sentence Pointer Our extractive model is similar to the sentence pointer architecture developed by (Chen and Bansal 2018) with the main difference being the choice of encoder. We use a hierarchical bidirectional LSTM encoder with word and sentence level LSTMs while (Chen and Bansal 2018) use a convolutional word level encoder for faster training and inference. The decoder is in both cases is an LSTM.

The extractive model considers the document as a list of N sentences $D = (S_1, \ldots, S_N)$, and each sentence as a list of tokens. We are given a ground-truth extracted summary of M sentences $(S_{i_1}, \ldots, S_{i_M})$, where the $i_1 < \ldots < i_M$ are the indices of the extracted sentences. The procedure to determine ground-truth extraction targets are identical to previous work - finding two sentences in the document that have the highest ROUGE score with each sentence in the summary.

We use an encoder-decoder architecture for this extractor. The encoder has a hierarchical structure that combines a token and sentence-level RNN. First, the "sentence-encoder" or token-level RNN is a bi-directional LSTM (Hochreiter and Schmidhuber 1997) encoding each sentence. The last hidden state of the last layer from the two directions produces sentence embeddings: $(\mathbf{s}_1, \ldots, \mathbf{s}_N)$, where N is the number of sentences in the document. The sentence-level LSTM or the "document encoder", another bi-directional LSTM, encodes this sequence of sentence embeddings to produce document representations: $(\mathbf{d}_1, \ldots, \mathbf{d}_N)$.

The decoder is an autoregressive LSTM taking the sentence-level LSTM hidden state of the previously extracted sentence as input and predicting the next extracted sentence. Let i_t the index of the previous extracted sentence at time step t. The input to the decoder is $\mathbf{s}_{i_t}$, or a zero vector at time-step $t = 0$. The decoder's output is computed by an attention mechanism from the decoder's hidden state $\mathbf{h}_t$ over the document representations $(\mathbf{d}_1, \ldots, \mathbf{d}_N)$. We used the dot product attention method from (Luong, Pham, and Manning 2015). The attention weights $\mathbf{a}_t$ produce a context vector $\mathbf{c}_t$, which is then used to compute an attention aware hidden state $\tilde{\mathbf{h}}_t$. Following the input-feeding approach from (Luong, Pham, and Manning 2015), the attention aware hidden state $\tilde{\mathbf{h}}_t$ is concatenated to the input in the next time step, giving the following recurrence $\mathbf{h}_t = \text{LSTM}\left([\mathbf{s}_{i_t}^T \; \tilde{\mathbf{h}}_{t-1}^T]^T, \mathbf{h}_{t-1}\right)$, with

$$\tilde{\mathbf{h}}_t = \mathbf{W}_{\tilde{h}} \begin{bmatrix} \mathbf{c}_t \\ \mathbf{h}_t \end{bmatrix}, \quad \mathbf{c}_t = \sum_{i=1}^{N} a_t(i)\mathbf{d}_i, \quad \alpha_t(i) = \mathbf{d}_i^T \mathbf{h}_t, \quad (1)$$

$$a_t(i) = \frac{\exp\left(\alpha_t(i)\right)}{\sum_{i'} \exp\left(\alpha_t(i')\right)}, \quad \text{for } i = 1..N. \quad (2)$$

The attention weights $\mathbf{a}_t$ are used as output probability distribution over the document sentences, of the choice for the next extracted sentence. We choose the convention to signal the end of the extraction by putting the same index twice in a row. Thus, the input to the decoder is the following sequence: $\mathbf{0}, \mathbf{s}_{i_1}, ..., \mathbf{s}_{i_M}$, and the target: $i_1, ..., i_M, i_M$, where M is the length of the ground-truth extracted summary and both sequences have $M + 1$ elements. The model is trained to minimize the cross-entropy of picking the correct sentence at each decoder time step. At inference, we use beam-search to generate the extracted summary.

Sentence Classifier As with the pointer network, we use a hierarchical LSTM to encode the document and produce a sequence of sentence representations $\mathbf{d}_1, ..., \mathbf{d}_N$ where N is the number of sentences in the document. We compute a final document representation as follows:

$$\mathbf{d} = \tanh\left(\mathbf{b}_d + \mathbf{W}_d \frac{1}{N} \sum_{i=1}^{N} \mathbf{d}_i\right) \quad (3)$$

where $\mathbf{b}_d$ and $\mathbf{W}_d$ are learnable parameters. Finally, the probability of each sentence belonging to the extractive summary is given by:

$$o_i = \sigma\left(\mathbf{W}_o \begin{bmatrix} \mathbf{d}_i \\ \mathbf{d} \end{bmatrix} + \mathbf{b}_o\right) \quad (4)$$

where σ is the sigmoid activation function. The model is trained to minimize the binary cross-entropy loss with respect to the sentences in the gold-extracted summary.

Model Details The model uses word embeddings of size 300. The token-level LSTM (sentence encoder), sentence-level LSTM (document encoder) and decoder each have 2 layers of 512 units and a dropout of 0.5 is applied at the output of each intermediate layer. We trained it with Adam, a learning rate 0.001, a weight decay of 10^{-5}, and using batch sizes of 32. We evaluate the model every 200 updates, using a patience of 50. At inference, we decode using beam search with a beam size of 4 for the pointer model and pick the k most likely sentences from the sentence classifier, where k is the average number of sentences in the summary across the training dataset.

Transformer Language Models (TLM)

Instead of formulating abstractive summarization as a seq2seq problem using an encoder-decoder architecture, we only use a single transformer language model that is trained *from scratch*, with appropriately "formatted" data (see figure 1, we also describe the formatting later in this section).

We use a transformer (Vaswani et al. 2017) language model (TLM) architecture identical to (Radford et al. 2019). Our model has 220M parameters with 20 layers, 768 dimensional embeddings, 3072 dimensional position-wise MLPs and 12 attention heads. The only difference in our architectures (to our knowledge) is that we do not scale weights at initialization. We trained the language model for 5 days on 16 V100 GPUs on a single Nvidia DGX-2 box. We used a linear ramp-up learning rate schedule for the first $40,000$ updates, to maximum learning rate of $2.5 \times e^{-4}$ followed by a cosine annealing schedule to 0 over the next $200,000$ steps with the Adam optimizer. We used mixed-precision training (Micikevicius et al. 2017) with a batch size of 256 sequences of 1024 tokens each.

In order to get an unconditional language model to do abstractive summarization, we can use the fact that LMs are trained by factorizing the joint distribution over words autoregressively. We organized the training data for the LM such that the ground-truth summary *follows* the information used by the model to generate a system summary. This way, we model the joint distribution of document and summary during training, and sample from the conditional distribution of summary given document at inference.

When dealing with extremely long documents that may not fit into a single window of tokens seen by a transformer language model, such as an entire scientific article, we use its introduction as a proxy for having enough information to generate an abstract (summary) and use the remainder of the paper as in domain language model training data (Fig 1). In such cases, we organize the arXiv and PubMed datasets as follows: 1) paper introduction 2) extracted sentences from the sentence pointer model 3) abstract 4) rest of the paper. On other datasets, the paper introduction would be the entire document and there would no rest of the paper. This ensures that at inference, we can provide the language model the paper introduction and the extracted sentences as conditioning to generate its abstract. We found that using the ground truth extracted sentences during training and the model extracted sentences at inference performed better than using the model extracted sentences everywhere.

We use a special token to indicate the start of the summary and use it at test time to signal to the model to start generating the summary. The rest of the article is provided as additional in-domain training data for the LM. The entire dataset is segmented into non-overlapping examples of $1,024$ tokens each. We use "topk" sampling at inference (Radford et al. 2019), with $k = 30$ and a softmax temperature of 0.7 to generate summaries.

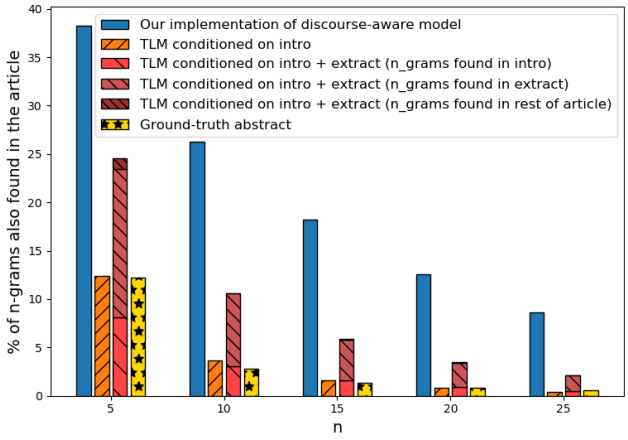

Figure 2: n-gram overlaps between the abstracts generated by different models and the input article on the arXiv dataset. We show in detail which part of the input was copied for our TLM conditioned on intro + extract.

Results and Analysis

Experimental setup

Datasets We experiment with four different large-scale and long document summarization datasets - arXiv, PubMed (Cohan et al. 2018), bigPatent (Sharma, Li, and Wang 2019) and Newsroom (Grusky, Naaman, and Artzi 2018). Statistics are reported in Table 1.

Table 1: Statistics from (Sharma, Li, and Wang 2019) for the datasets used in this work - The number of document/summary pairs, the ratio of the number of words in the document to the abstract and the number of words in the summary and document.

Dataset	#Documents	Comp Ratio	Sum Len	Doc Len
arXiv	215,913	39.8	292.8	6,913.8
PubMed	133,215	16.2	214.4	3,224.4
Newsroom	1,212,726	43.0	30.4	750.9
BigPatent	1,341,362	36.4	116.5	3,572.8

Data preprocessing Both our extractive and abstractive models use sub-word units computed using *byte pair encoding* (Sennrich, Haddow, and Birch 2015) with $40,000$ replacements. To address memory issues in the sentence pointer network, we only keep 300 sentences per article, and 35 tokens per sentence.

Evaluation We evaluate our method using full-length F-1 ROUGE scores (Lin 2004) and re-used the code from (Cohan et al. 2018) for this purpose. All ROUGE numbers reported in this work have a 95% confidence interval of at most 0.24.

Comparison We compare our results to several previously proposed extractive and abstractive models. All prior results reported on the arXiv and Pubmed benchmark are obtained from (Cohan et al. 2018). Similarly, prior results for the BigPatent dataset are obtained from (Sharma, Li, and Wang 2019) and Newsroom from (Grusky, Naaman, and Artzi 2018) and (Mendes et al. 2019). These methods include *LexRank* (Erkan and Radev 2004), *SumBasic* (Vanderwende et al. 2007), *LSA* (Steinberger and Jezek 2004), *Attention-Seq2Seq* (Nallapati et al. 2016; Chopra, Auli, and Rush 2016), *Pointer-Generator Seq2Seq* (See, Liu, and Manning 2017), *Discourse aware*, which is a hierarchical extension to the pointer generator model, (Cohan et al. 2018), *Sent-rewriting* (Chen and Bansal 2018), *RNN-Ext* (Chen and Bansal 2018), *Exconsumm* (Mendes et al. 2019).

Discussion

We present our main results on summarizing arXiv and PubMed papers in tables 2, 4. Our extractive models are able to outperform previous extractive baselines on both the arXiv and Pubmed datasets. Our TLM conditioned on the extractive summary produced by our best extractive model (TLM-I+E (G,M)) outperforms prior abstractive/mixed results on the arXiv, Pubmed and bigPatent datasets, except on ROUGE-L. On Newsroom, we do better than the only other abstractive model (Seq2Seq with attention) by a massive margin and achieve better performance than the pointer generator even on the abstractive and mixed which their model should be better suited to since it has a copy mechanism. The Exconsumm model (Mendes et al. 2019) however, which is primarily an extractive model does better on this dataset. We suspect the poor ROUGE-L result is due to the absence of a copy mechanism that makes it hard to get *exact* large n-gram matches. Figure 2 further supports this hypothesis, it is evident that a model with a copy mechanism is often able to copy even upto 25-grams from the article. Further, (Graham 2015) finds that ROUGE-L is poorly correlated with human judgements when compared to ROUGE-1,2,3. In table 7 and Table 8, we present qualitative results of abstracts of notable papers in our field and of our TLM conditioned on the introductions and extracted summaries of a random example from the arXiv test set. Table 3 shows similar qualitative examples on the Newsroom dataset. Tables 2, 4 and 5 also provide different train / test settings for our TLM conditioned on extracted sentences. We show a performance upper bound conditioning the Transformer LM on oracle / ground-truth extracted sentences at both train and test time (TLM-I+E (G,G)). We also experiment with using either the ground-truth extracted sentences (TLM-I+E (G,M)) or the model extracted sentences (TLM-I+E (M,M)) during training and find that latter slightly impairs performance. Finally, figure 3, presents a visualization of the word embeddings learned by our TLM.

Abstractiveness of generated abstracts

(Weber et al. 2018) argued that state-of-the-art abstractive summarization systems that use a copy mechanism effectively generate the summary by copying over large chunks

Table 2: Summarization results on the arXiv dataset. Previous work results from (Cohan et al. 2018). The following lines are a simple baseline Lead-10 extractor and the pointer and classifier models. Our transformer LMs (TLM) are conditioned either on the Introduction (I) or along with extracted sentences (E) either from ground-truth (G) or model (M) extracts.

Model	Type	ROUGE 1	2	3	L
Previous Work					
SumBasic	Ext	29.47	6.95	2.36	26.3
LexRank	Ext	33.85	10.73	4.54	28.99
LSA	Ext	29.91	7.42	3.12	25.67
Seq2Seq	Abs	29.3	6.00	1.77	25.56
Pointer-gen	Mix	32.06	9.04	2.15	25.16
Discourse	Mix	35.80	11.05	3.62	**31.80**
Our Models					
Lead-10	Ext	35.52	10.33	3.74	31.44
Sent-CLF	Ext	34.01	8.71	2.99	30.41
Sent-PTR	Ext	42.32	15.63	7.49	38.06
TLM-I	Abs	39.80	12.20	4.42	22.36
TLM-I+E (M,M)	Mix	41.59	14.26	5.94	23.55
TLM-I+E (G,M)	Mix	**42.43**	**15.24**	**6.68**	24.08
Oracle					
Gold Ext	Orac	44.25	18.17	9.14	35.33
TLM-I+E (G,G)	Orac	46.52	18.19	8.73	26.88

Table 3: Qualitative Results - News articles and our model generated summaries on the NewsRoom dataset

Document — The bull from El Pilar breeding ranch in the province of Salamanca, pummeled the man whose white trousers slid down during the ordeal. He was then tossed into the air by the right horn of the bull.He was finally dragged to safety by onlookers after fellow runners grabbed the bull's tail to pull him off.A third runner, a 42-year-old from Catalonia, was gored twice in his right arm just yards away from the bullring at the end of the run but he managed to leap over the heavy wooden barrier to escape further injury.A spokesman from the Navarre Hospital said all three would survive their injuries.Fifteen people have been killed during the running of the bulls since records began a century ago, the last in 2009 when Daniel Jimeno, 27, from Madrid was gored in the neck, heart and lungs.The eight day long fiesta, immortalised in Ernest Hemingway's novel "The Sun Also Rises," draws crowds of more than a million revellers each year who dress in the traditional white with red neckerchiefs and sashes.The six bulls which run the 928 yard dash from a corral to the bullring are killed in the afternoon bullfight.
Abstractive Summary — A Spanish man who was gored in the chest by a bull in the San Fermin bullring earlier this week has died.
Mixed Summary — A Spanish man who was gored during a run by a bullfighter in the western region of Salamanca has escaped with his left arm.
Extractive Summary — The bull from El Pilar breeding ranch in the province of Salamanca, pummelled the man whose white trousers slid down during the ordeal.
Document — (CBS) - Controversy over a new Microsoft patent has people questioning whether or not the intention has racist undertones. CNET reported that Microsoft has been granted a U.S. patent that will steer pedestrians away from areas that are high in crime. [...]
Absractive Summary — The new Microsoft patent claims a device could provide pedestrian navigation directions from a smartphone.
Mixed Summary Microsoft won a U.S. patent for a new way to steer pedestrians out of areas that are high in crime

Table 4: Summarization results on the PubMed dataset. Previous work results from (Cohan et al. 2018). The following lines are a simple baseline Lead-10 extractor and the pointer and classifier models. Our transformer LMs (TLM) are conditioned either on the Introduction (I) or along with extracted sentences (E) either from ground-truth (G) or model (M) extracts.

Model	Type	ROUGE			
		1	2	3	L
Previous Work					
SumBasic	Ext	37.15	11.36	5.42	33.43
LexRank	Ext	39.19	13.89	7.27	34.59
LSA	Ext	33.89	9.93	5.04	29.70
Seq2seq	Abs	31.55	8.52	7.05	27.38
Pointer-gen	Mix	35.86	10.22	7.60	29.69
Discourse	Mix	38.93	15.37	**9.97**	**35.21**
Our Models					
Lead-10	Ext	37.45	14.19	8.26	34.07
Sent-CLF	Ext	45.01	19.91	12.13	41.16
Sent-PTR	Ext	43.30	17.92	10.67	39.47
TLM-I	Abs	36.63	11.67	5.40	21.00
TLM-I+E (M,M)	Mix	40.59	15.59	8.55	23.59
TLM-I+E (G,M)	Mix	**41.43**	**15.89**	8.62	24.32
Oracle					
Gold Ext	Orac	47.76	20.36	11.52	39.19
TLM-I+E (G,G)	Orac	45.63	19.75	11.52	27.67

from the article, essentially doing "extractive" summarization. Following this work, we measure how much a model copies from the article by counting the proportion of n-grams from the generated abstract that are also found in the article. These statistics measured on the arXiv dataset are presented in figure 2. First, the original abstract and our TLM conditioned on the intro have small and very similar overlap fractions with the original article. A model using a pointing mechanism (we used our own implementation of the model developed by (Cohan et al. 2018))[1] copies more than our transformer model, especially for higher n-grams. In particular, more than 10% of the 20-grams from the abstracts generated by the pointing model are also found in the article, showing that it tends to copy long sequences of words. On the other hand, our proposed model produces more "abstractive" summaries, demonstrating its ability to paraphrase. Our model tends to copy longer sequences when conditioned on the introduction and the sentences from the extractor. We hypothesize that providing extracted sentences from the article that already contain a lot of words present in the reference abstract, makes the transformer's task easier, by allowing it to copy words and phrases from the extracted sentences. We find empirical evidence of this in figure 2, showing that the majority of n-gram copies come from the extracted sentences. For 5-grams, close to 2/3rd of the words copied are from the extracted sentences. As the number of grams increases to 25-grams, 4/5th of the words copied are from the extracted sentences.

[1]This model achieved the following ROUGE-1, 2, 3 and L on the arXiv dataset: $41.33, 14.73, 6.80, 36.34$

Table 5: Summarization results on the bigPatent dataset. Previous work results from (Sharma, Li, and Wang 2019). Our transformer LMs (TLM) are conditioned on the whole document or additionally with extracted sentences (E) either from ground-truth (G) or model (M) extracts.

Model	Type	ROUGE		
		1	2	L
Previous Work				
Lead-3	Ext	31.27	8.75	26.18
TextRank	Ext	35.99	11.14	29.60
SumBasic	Ext	27.44	7.08	23.66
LexRank	Ext	35.57	10.47	29.03
RNN-Ext	Ext	34.63	10.62	29.43
Seq2Seq	Abs	28.74	7.87	24.66
Pointer-gen	Mix	30.59	10.01	25.65
Pointer-gen (Cov)	Mix	33.14	11.63	28.55
Sent-rewriting	Mix	37.12	11.87	**32.45**
Oracle				
Gold Ext	Orac	43.56	16.91	36.52
OracleFrag	Orac	91.85	78.66	91.85
Our Models				
Sent-CLF	Ext	36.20	10.99	31.83
Sent-PTR	Ext	34.21	10.78	30.07
TLM	Abs	37.52	10.63	22.79
TLM+E (G,M)	Mix	**38.65**	**12.31**	24.71
TLM+E (G,G)	Orac	39.99	13.79	25.26

Table 6: Summarization results on the Newsroom dataset. Previous work results from (Grusky, Naaman, and Artzi 2018) and (Mendes et al. 2019).

Model	Type	Extractive			Mixed			Abstractive		
		ROUGE								
		1	2	L	1	2	L	1	2	L
Previous Work										
Seq2Seq	Abs	6.1	0.2	5.4	5.7	0.2	5.1	6.2	1.1	5.7
TextRank	Ext	32.4	19.7	28.7	22.3	7.9	17.7	13.5	1.9	10.5
Pointer-gen	Mix	39.1	27.9	36.2	25.5	11.0	21.1	14.7	2.3	11.4
Lead-3	Ext	53.0	49.0	52.4	25.1	12.9	22.1	13.7	2.4	11.2
Exconsumm	Mix	**68.4**	**62.9**	**67.3**	**31.7**	**16.1**	**27.0**	17.1	3.1	14.1
Our Models										
TLM	Abs	53.3	44.2	50.1	28.1	12.1	23.0	**18.5**	**3.9**	**14.7**

Conclusion

We have demonstrated that Transformer language models can generate high quality summaries of long sequences of text via an extractive step followed by an abstractive step. We quantitatively measure the positive impact of the extractive step, by comparing it to a abstractive model variant that only sees the input text itself. Our approach outperforms previous extractive and abstractive summarization methods on the arXiv, PubMed and bigPatent datasets and is less prone to copying entire phrases or sentences from the input text. The fluency and coherency of the sample summaries suggests that these models are ready for comprehensive human evaluation studies. As with other problem domains, we have observed that abstractive summaries generated by transformers can generate imaginary content. We ad-

Table 7: Qualitative Results — Generated abstracts of select papers using our Intro Only TLM.

Generated abstract for "the attention model paper" (Bahdanau, Cho, and Bengio 2014) - In this paper, we propose a neural machine translation model, which learns to align and translate a source sentence without encoding it into a fixed length vector. The proposed model allows the neural network to align and translate jointly, and in addition, predicts a target word based on the context vectors associated with these source positions. We show that the proposed model achieves significantly improved translation performance than the conventional encoder decoder neural network approach, when the sentences in the training corpus are long . Quantitative analysis reveals that the proposed model finds a linguistically plausible soft alignment between a source sentence and the target sentence. Finally, we compare the performance of the proposed model with the state-of-the-art standard neural network translation models and show that the proposed approach outperforms the conventional approach in terms of both translation performance and the quality of the translations.	**Original abstract** - Neural machine translation is a recently proposed approach to machine translation. Unlike the traditional statistical machine translation, the neural machine translation aims at building a single neural network that can be jointly tuned to maximize the translation performance. The models proposed recently for neural machine translation often belong to a family of encoder-decoders and consists of an encoder that encodes a source sentence into a fixed-length vector from which a decoder generates a translation. In this paper, we conjecture that the use of a fixed-length vector is a bottleneck in improving the performance of this basic encoder-decoder architecture, and propose to extend this by allowing a model to automatically (soft-)search for parts of a source sentence that are relevant to predicting a target word, without having to form these parts as a hard segment explicitly. With this new approach, we achieve a translation performance comparable to the existing state-of-the-art phrase-based system on the task of English-to-French translation. Furthermore, qualitative analysis reveals that the (soft-)alignments found by the model agree well with our intuition.
Generated abstract for "the GPT-2 paper" (Radford et al. 2019) - Language models have been successful in creating rich representations of complex systems, including tasks such as sentiment analysis. In this paper we demonstrate their ability to generalize well across domains and tasks. We demonstrate the utility of language models by demonstrating that a variety of tasks can be modeled using language models. We demonstrate the potential of language models in downstream tasks by highlighting the ability of these models to perform a wide range of tasks in a zero shot setting.	**Original abstract** - Natural language processing tasks, such as question answering, machine translation, reading comprehension, and summarization, are typically approached with supervised learning on task-specific datasets. We demonstrate that language models begin to learn these tasks without any explicit supervision when trained on a new dataset of millions of webpages called WebText. When conditioned on a document plus questions, the answers generated by the language model reach 55 F1 on the CoQA dataset - matching or exceeding the performance of 3 out of 4 baseline systems without using the 127,000+ training examples. The capacity of the language model is essential to the success of zero-shot task transfer and increasing it improves performance in a log-linear fashion across tasks. Our largest model, GPT-2, is a 1.5B parameter Transformer that achieves state of the art results on 7 out of 8 tested language modeling datasets in a zero-shot setting but still underfits WebText. Samples from the model reflect these improvements and contain coherent paragraphs of text. These findings suggest a promising path towards building language processing systems which learn to perform tasks from their naturally occurring demonstrations.
Generated abstract for "the BERT paper" (Devlin et al. 2018) - We develop an architecture for deep bidirectional language models that allows the representation to fuse the left and the right context of a given token. Our framework uses bidirectional transformer encoder representations that enable the model to fuse the context representations of a token. We show that our architecture achieves state-of-the-art performance on a large suite of tasks, outperforming many systems with task-specific architectures. Our results are based on a set of eleven task specific tasks that are benchmarks for a large suite of nlp tasks.	**Original abstract** - We introduce a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers. Unlike recent language representation models, BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications. BERT is conceptually simple and empirically powerful. It obtains new state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE score to 80.5% (7.7% point absolute improvement), MultiNLI accuracy to 86.7% (4.6% absolute improvement), SQuAD v1.1 question answering Test F1 to 93.2 (1.5 point absolute improvement) and SQuAD v2.0 Test F1 to 83.1 (5.1 point absolute improvement).

vise that such evaluations should probe multiple aspects of the summarization results including both factual correctness and coherency. We also note that for evaluating the correctness of the summaries of scientific articles and patents one must have highly trained evaluators who are willing to invest significant amounts of time to read the underlying papers and patents. Such studies could therefore require significant investments of resources. We have also presented an upper bound on extractive + abstractive models, by conditioning the abstractive step on gold-extracted sentences. In future work, we are also interested in exploring the possibility of training the extractive and abstractive steps in an end-to-end manner. While we believe that this work is a step forward towards generating more abstractive summaries, it remains an open challenge to develop models that respect the underlying facts of the content being summarized while matching the creative ability of humans to coherently and concisely synthesize summaries.

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bengio, Y.; Ducharme, R.; Vincent, P.; and Jauvin, C. 2003. A neural probabilistic language model. *Journal of machine learning research* 3(Feb):1137–1155.

Chen, Y.-C., and Bansal, M. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. *arXiv preprint arXiv:1805.11080*.

Cheng, J., and Lapata, M. 2016. Neural summariza-

tion by extracting sentences and words. *arXiv preprint arXiv:1603.07252*.

Chopra, S.; Auli, M.; and Rush, A. M. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 93–98.

Cohan, A.; Dernoncourt, F.; Kim, D. S.; Bui, T.; Kim, S.; Chang, W.; and Goharian, N. 2018. A discourse-aware attention model for abstractive summarization of long documents. *CoRR* abs/1804.05685.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; and Dauphin, Y. N. 2017. Convolutional sequence to sequence learning. 1243–1252.

Gehrmann, S.; Deng, Y.; and Rush, A. M. 2018. Bottom-up abstractive summarization. *arXiv preprint arXiv:1808.10792*.

Graham, Y. 2015. Re-evaluating automatic summarization with bleu and 192 shades of rouge. 128–137.

Grusky, M.; Naaman, M.; and Artzi, Y. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. *arXiv preprint arXiv:1804.11283*.

Gu, J.; Lu, Z.; Li, H.; and Li, V. O. K. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *CoRR* abs/1603.06393.

Gulcehre, C.; Firat, O.; Xu, K.; Cho, K.; Barrault, L.; Lin, H.-C.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2015. On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.

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

Hochreiter, S. 1998. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6(02):107–116.

Kalchbrenner, N.; Espeholt, L.; Simonyan, K.; Oord, A. v. d.; Graves, A.; and Kavukcuoglu, K. 2016. Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099*.

Lin, C.-Y. 2004. Looking for a few good metrics: Automatic summarization evaluation-how many samples are enough? In *NTCIR*.

Luong, M.-T.; Pham, H.; and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Mendes, A.; Narayan, S.; Miranda, S.; Marinho, Z.; Martins, A. F.; and Cohen, S. B. 2019. Jointly extracting and compressing documents with summary state representations. *arXiv preprint arXiv:1904.02020*.

Micikevicius, P.; Narang, S.; Alben, J.; Diamos, G.; Elsen, E.; Garcia, D.; Ginsburg, B.; Houston, M.; Kuchaiev, O.; Venkatesh, G.; et al. 2017. Mixed precision training. *arXiv preprint arXiv:1710.03740*.

Mikolov, T.; Karafiát, M.; Burget, L.; Černockỳ, J.; and Khudanpur, S. 2010. Recurrent neural network based language model.

Nallapati, R.; Zhou, B.; Gulcehre, C.; Xiang, B.; et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Nallapati, R.; Zhai, F.; and Zhou, B. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents.

Nallapati, R.; Zhou, B.; and Ma, M. 2016. Classify or select: Neural architectures for extractive document summarization. *arXiv preprint arXiv:1611.04244*.

Pascanu, R.; Mikolov, T.; and Bengio, Y. 2013. On the difficulty of training recurrent neural networks. 1310–1318.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners.

Rush, A. M.; Chopra, S.; and Weston, J. 2015. A neural attention model for abstractive sentence summarization. *CoRR* abs/1509.00685.

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get to the point: Summarization with pointer-generator networks. *CoRR* abs/1704.04368.

Sennrich, R.; Haddow, B.; and Birch, A. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Sharma, E.; Li, C.; and Wang, L. 2019. Bigpatent: A large-scale dataset for abstractive and coherent summarization. *arXiv preprint arXiv:1906.03741*.

Steinberger, J., and Jezek, K. 2004. Using latent semantic analysis in text summarization and summary evaluation. *Proc. ISIM* 4:93–100.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. 3104–3112.

Van Den Oord, A.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A. W.; and Kavukcuoglu, K. 2016. Wavenet: A generative model for raw audio. *SSW* 125.

Vanderwende, L.; Suzuki, H.; Brockett, C.; and Nenkova, A. 2007. Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing & Management* 43(6):1606–1618.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. *CoRR* abs/1706.03762.

Vinyals, O.; Fortunato, M.; and Jaitly, N. 2015. Pointer networks. 2692–2700.

Weber, N.; Shekhar, L.; Balasubramanian, N.; and Cho, K. 2018. Controlling decoding for more abstractive summaries with copy-based networks. *arXiv preprint arXiv:1803.07038*.

Appendix

Samples from the arXiv test set

Samples from the TLM-I and TLM-I+E (G,M) models on a random example from the arXiv test set.

Table 8: Qualitative Results - Generated abstracts from our models on a random example from the test set of (Cohan et al. 2018)

TLM conditioned on intro — deep convolutional neural networks (cnns) have created new benchmarks in the object recognition challenge . most of these benchmark datasets contain thousands of images of interest . in addition to these datasets , several standard data - sets , such as pascal voc 2012 and the ms coco , have been used in conjunction with the same network in order to improve the object detection task . in this paper , we present a method for object localization using the deep convolutional neural networks trained for the simpler task of image - level classification . we propose localization candidates (or bounding boxes) which maximize classification scores for a class of interest . we explore the beam search strategy where the number of candidate bounding boxes are restricted as we traverse from the root node towards the leaf nodes . we then gradually explore smaller and smaller regions of interest till a point is reached where the classifier is not able to discriminate among the classes of interest . the detection candidates are organized in a search tree , the root node being represented by the entire test image . we then gradually explore smaller and smaller regions of interest till a point is reached where the classifier is not able to distinguish among the classes of interest .
TLM conditioned on intro + pointer network extracted sentences — deep convolutional neural networks (cnns) are widely used in image recognition applications . however , they face a number of challenges . first , cnn models are computationally expensive to train and to perform . second , they are not easy to implement and can not exploit global features . to alleviate these issues , we present a method that leverages the correlation between semantic information present in feature maps and localization of an object of interest . the method leverages feature map activations for object localization . we demonstrate that this method is significantly faster than state - of - the - art in localization accuracy by a significant margin of up to 8 map on two standard data - sets with complex scenes , pascal voc 2012 and the much larger ms coco .
Ground truth abstract — object localization is an important computer vision problem with a variety of applications . the lack of large scale object - level annotations and the relative abundance of image - level labels makes a compelling case for weak supervision in the object localization task . deep convolutional neural networks are a class of state-of-the-art methods for the related problem of object recognition . in this paper , we describe a novel object localization algorithm which uses classification networks trained on only image labels . this weakly supervised method leverages local spatial and semantic patterns captured in the convolutional layers of classification networks . we propose an efficient beam search based approach to detect and localize multiple objects in images . the proposed method significantly outperforms the state-of-the-art in standard object localization data - sets with a 8 point increase in map scores .

T-SNE of learned word embeddings

We visualize the word embeddings learned by our TLM model using t-sne. We find that words that are often associated with computer science are clustered in a different part of space when compared to words associated with physics. We use the arXiv REST API to find the submission category of each paper in the training set and then find the $\sim$300 most representative words for each category, using TF-IDF scores and plot them.

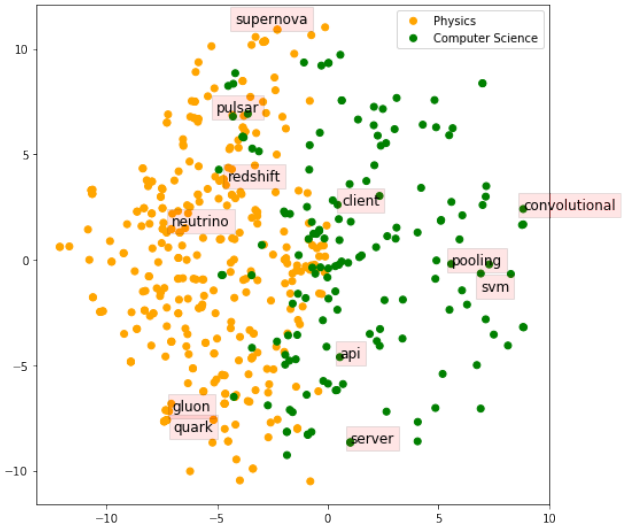

Figure 3: t-sne visualization of the TLM-learned word embeddings. The model appears to partition the space based on the broad paper categoty in which it frequently occurs.