On Extractive and Abstractive Neural Document Summarization with Transformer Language Models

Jonathan Pilault[*1,2,3], **Raymond Li**[*1], **Sandeep Subramanian**[*1,2,4] **and Christopher Pal**[1,2,3,4,5]
[1]Element AI
[2]Mila, [3]Polytechnique Montreal, [4]University of Montreal, [5]Canada CIFAR AI Chair
firstname.lastname@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 also show that this approach produces more abstractive summaries compared to prior work that employs a copy mechanism while still achieving higher ROUGE scores. We provide extensive comparisons with strong baseline methods, prior state of the art work as well as multiple variants of our approach including those using only transformers, only extractive techniques and combinations of the two. We examine these models using four different summarization tasks and datasets: arXiv papers, PubMed papers, the Newsroom and BigPatent datasets. We find that transformer based methods produce summaries with fewer n-gram copies, leading to n-gram copying statistics that are more similar to human generated abstracts. We include a human evaluation, finding that transformers are ranked highly for coherence and fluency, but purely extractive methods score higher for informativeness and relevance. We hope that these architectures and experiments may serve as strong points of comparison for future work.[1]

1 Introduction

Automatic text summarization is the process of compressing a document while preserving key information content and meaning. This process is often achieved through extractive or abstractive techniques. Extractive summarization is the strategy of selecting a subset of words, phrases or sentences from the input document to form a summary. Abstractive summarization consists of creating sentences summarizing content and capturing key ideas and elements of the source text, usually involving significant changes and paraphrases of text from the original source sentences. While extractive summarization is able to preserve saliency, the broader flow or coherency of the multiple sentences forming the summary can be less natural compared to a human generated summary. On the other hand, abstractive methods should produce coherent summaries without copying sentences verbatim while remaining faithful to statements asserted in the input document.

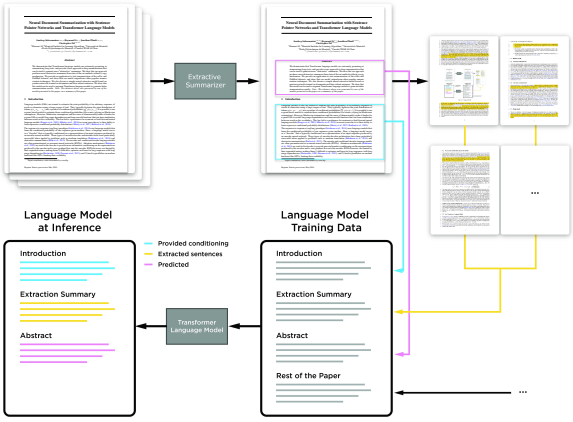

Figure 1: Our approach for abstractive summarization of a scientific article. An older version of this paper is shown as the reference document. First, a sentence pointer network extracts important sentences from the paper. Next, these sentences are provided along with the whole scientific article to 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 can be replaced by the entire document.

Recent work by (Radford et al., 2019) (GPT-2) has demonstrated that Transformer Language

[*]Authors contributed equally to this work
[1]Note: The abstract above was collaboratively written by the authors and one of the models presented in this paper based on an earlier draft of this paper.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 9308–9319,
November 16–20, 2020.

Models (TLMs) trained on web text can inadvertently learn to perform abstractive summarization, since a large crawl of web documents may contain some documents which have a "tl;dr" token followed by a summary. We are interested here in explicitly configuring autoregressive transformer models to generate summaries in an intentional and focused manner. Since summaries or abstracts typically appear at the beginning of a document, a model trained from such web-crawl data does not enforce strong conditioning on the text to be summarized. Our tests using models naively trained on web-crawl data yielded summarization quality far below baseline methods. However, in this paper we explore what can be achieved through simply ordering the passages of an input text, correctly structuring the task definition and training procedure. We also examine the impact of combining this approach with simple but high quality extractive techniques.

While pure language models can be applied to short input documents, memory considerations make it difficult to scale to long documents. Further, as high quality extractive summarization methods illustrate, much of the content of a long document is not needed to create a summary. For these reasons we also explore a hybrid approach which combines an extractive and abstractive approach. We achieve this by stepping away from the classical end-to-end sequence-to-sequence paradigm, using an initial extractive step that reduces the amount of context for a subsequent abstractive step (see figure 1). Such an approach could be thought of as a form of hard attention. Moreover, we show that such a paradigm works even for datasets where the entire input can fit in memory, i.e. see Table 4 and 5. We take an approach whereby we restructure the input to a TLM by reordering the document and inserting standardized delimiters to identify the introduction, our extracted sentences, the abstract or summary and the rest-of-the-article. With our method, the resulting TLM can focus its attention on the relevant content and its model complexity on the summarization task.

In general, as we shall detail in our experiments below, we find that TLMs are surprisingly effective at summarizing *long documents*, outperforming typical seq2seq approaches, even without using copying/pointing mechanisms, an encoder or additional losses. Our contribution consists of an extensive set of large scale experiments comparing our hybrid extractive and abstractive approach to long document summarization with different variants of our model, strong and simple baselines as well as with state-of-the-art summarization models (see section 3.2 for a complete description of comparisons). We examine these models through ROUGE scores, through a study of the amount of n-gram copying performed by different models, as well as through a human evaluation using a standard protocol. We find that our hybrid approach yields results that surpass current state-of-the-art results on several metrics of these evaluations.

We see our extensive experimentation and the wide variety of evaluation protocols provided here as being a key part of the contribution provided by this work and we hope that the analysis, insights and models here will serve as strong yet simple baselines for future comparison and research.

2 Related Work

The earliest attempts at automatic summarization focused on extractive techniques, which find words or sentences in a document that capture its most salient content. 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 et al. (2016b); 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 et al. (2016b) 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. Chen and Bansal (2018) use a pointer network (Vinyals et al., 2015) to sequentially pick sentences from the document that comprise its extractive summary. Such techniques however heavily rely on the span of words from the input document.

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 et al., 2015; Nallapati et al., 2016a; Chopra et al., 2016). The advantages of extractive, abstractive and attention-based models were first combined in (Gu et al., 2016; Gulcehre et al., 2016) with a copy mechanism for out-of-vocabulary words present in the source document. Similarly, (See et al., 2017) used the attention scores to calculate the probability of generating vs copying a word.

The most similar approach to our hybrid extractive and abstractive technique is that of Chen and Bansal (2018); Gehrmann et al. (2018); Hsu et al. (2018); Liu et al. (2018). In such set-ups, an *extractor* first selects salient sentences from the input. Then, an abstractive summarizer rewrites extracted sentences into a final summary. Our framework has a few advantages over previous methods. 1), we explore high capacity transformer LMs akin to Radford et al. (2019) as our abstractive summarizer, which results in grammatical and fluent generations 2), our language modeling formulation of the problem allows us to easily "recycle" the input document and use it additional in-domain data for LM training. 3) We improve over previous approaches without the use of a copy mechanism, which results in fewer n-gram copies from the input document. Liu et al. (2018) generate Wikipedia articles given references to source material and extracted sentences. They rank the importance of paragraphs found in the reference material based on techniques such as TextRank (Mihalcea and Tarau, 2004), a graph based ranking technique. In contrast, the extractive methods we use here are trained discriminatively using an extractive abstract as the target that is generated using an oracle. Wikipedia article synthesis also necessarily combines potentially redundant information from multiple documents that is relatively specific and less abstractive compared to the task of writing the abstract of a scientific paper. As seen in Figure 2, human generated (ground-truth) abstractive summaries in our datasets actually have very little word overlap with the source document.

3 Framework

Our model comprises two distinct trainable components: 1) an extractive model, comprising a hierarchical encoder that outputs sentence representations, used to either point to or classify sentences in the input, and 2) a transformer language model, conditioned on the extracted sentences as well as a part of or the entire input document.

3.1 Extractive Models

We describe the two neural extractive models used in this section. We used different types of extraction techniques to demonstrate the TLM model sensitivity to the extracted sentences. For instance, the Sentence Pointer performs much better on the arxiv dataset (see table 2) but the classifier is stronger on the Pubmed dataset (see table 3).

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 procedure to determine ground-truth extraction targets is similar to previous work (Nallapati et al., 2017): the ground truth is determined by computing the average $\text{ROUGE}_{1,2,L}$ score of each document sentence against each summary sentence. Considering the input document as a list of N sentences $D = (S_1, \ldots, S_N)$ and the target summary as a list of M sentences $T = (S'_1, \ldots, S'_M)$, our heuristic provides $N \times M$ scores, such that: $\text{SCORES}_{\text{extraction}} = \{\frac{1}{3}\sum_{r \in 1,2,L} \text{ROUGE}_r(S_i, S'_j) | S_i \in D; S'_j \in T\}$.

Since single sentence extraction may not always contain the same information content as a target summary, we extended the number ground-truth extraction sentences per output summary sentence to two. This is done by choosing the top 2 sentences in D that have the highest $\text{SCORES}_{\text{extraction}}$ with respect to a given sentence in T. The resulting $2M$ ordered sentences are used as context in the TLM. The TLM benefits from a more structured and larger context from the extractive summarization model during training.

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 pointer 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 $\mathbf{s}_{i_t}$.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 et al., 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 $\tilde{\mathbf{h}}_t$.

The attention weights $\mathbf{a}_t$ are used as output probability distribution over the document sentences, of the choice for the next extracted sentence. 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:

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

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:

$$o_i = \sigma\left(\mathbf{W}_o \begin{bmatrix}\mathbf{d}_i \\ \mathbf{d}\end{bmatrix} + \mathbf{b}_o\right) \quad (2)$$

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 and training parameters are included in the appendix.

3.2 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. In other words, they typically factorize the joint distribution of tokens $p(x_1, x_2 \ldots x_n)$ into a product of conditional probabilities $\prod_i^n p(x_i|x_{<i})$. We therefore organize the training data for our models such that the ground-truth summary *follows* the information used by the model to generate a summary. As such, we can model the joint distribution of the document and the summary during training, and sample from the conditional distribution of the summary given document when we wish to perform 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) the paper introduction, 2) extracted sentences from the sentence pointer model, 3) the abstract, and 4) the rest of the paper. This ensures that at inference time, we can provide the language model the paper introduction and the extracted sentences as conditioning to gen-

erate 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. On other datasets, the paper introduction would be the entire document. In such case, the rest of the paper does not exist and is therefore not included.

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 (Fan et al., 2018; Radford et al., 2019), with $k = 30$ and a softmax temperature of 0.7 to generate summaries.

4 Results and Analysis

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

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

Table 1: Statistics from Sharma et al. (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.

Data preprocessing Both our extractive and abstractive models use sub-word units computed using *byte pair encoding* (Sennrich et al., 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, abstractive and mixed summarization models on ROUGE scores. ROUGE scores tend to measure lexical overlap (Ng and Abrecht, 2015) which favors extractive methods of summarization. Since ROUGE scores do not capture system summary fluency and readability (which typically does not favor abstractive summarization), we also include a human evaluation. For this reason, Tables 2, 3, 4, 5 have a "Type" column to inform the reader on the type model evaluated (Ext=extractive, Mix=mixed and Abs=abstractive). All prior results reported on the arXiv and Pubmed benchmark are obtained from Cohan et al. (2018), except for the *Bottom-up* model[2] (Gehrmann et al., 2018). Similarly, prior results for the BigPatent dataset are obtained from (Sharma et al., 2019) and Newsroom from (Grusky et al., 2018a) 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., 2016a; Chopra et al., 2016), *Pointer-Generator Seq2Seq* (See et al., 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).

We present our main results on summarizing arXiv and PubMed papers in tables 2, 3. TLM+I+E (G,M) sets a new state-of-the-art on Arxiv, Pubmed and bigPatent datasets on abstractive summarization ROUGE scores. Our extractive models are able to outperform previous extractive baselines on both the arXiv and Pubmed datasets. Our extractive techniques also score higher than our abstractive techniques on arXiv and Pubmed. Again, ROUGE does not capture all aspects of a summary's quality such as fluency and coherence. For instance, previous work that have used RL to maximize ROUGE scores have concluded that "RL has the highest ROUGE-1 and ROUGE-L scores, it produces the least readable summaries" (Paulus et al., 2017). 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, our TLM model performs close to 7 times better than the other purely abstractive model (Seq2Seq with attention). We achieve better performance than the pointer generator even on the

[2]We used the code from https://github.com/sebastianGehrmann/bottom-up-summary with the same parameters.

Model	Type	ROUGE 1	2	3	L
Previous Work					
Lead-10	Ext	35.52	10.33	3.74	31.44
SumBasic	Ext	29.47	6.95	2.36	26.3
LexRank	Ext	33.85	10.73	4.54	28.99
Seq2Seq	Abs	29.3	6.00	1.77	25.56
Pointer-gen	Mix	32.06	9.04	2.15	25.16
Discourse-aware	Mix	35.80	11.05	3.62	31.80
Bottom-up	Mix	39.96	13.16	5.04	36.28
Our Models					
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	39.65	12.15	4.40	35.76
TLM-I+E (G,M)	Mix	41.62	14.69	6.16	38.03
Oracle					
Gold Ext	Oracle	44.25	18.17	9.14	35.33
TLM-I+E (G,G)	Oracle	46.40	18.15	8.71	42.27

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					
Lead-10	Ext	37.45	14.19	8.26	34.07
SumBasic	Ext	37.15	11.36	5.42	33.43
LexRank	Ext	39.19	13.89	7.27	34.59
Seq2seq	Abs	31.55	8.52	7.05	27.38
Pointer-gen	Mix	35.86	10.22	7.60	29.69
Discourse-aware	Mix	38.93	15.37	9.97	35.21
Bottom-up	Mix	40.02	15.82	8.71	37.28
Our Models					
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	37.06	11.69	5.31	34.27
TLM-I+E (G,M)	Mix	42.13	16.27	8.82	39.21
Oracle					
Gold Ext	Oracle	47.76	20.36	11.52	39.19
TLM-I+E (G,G)	Oracle	46.32	20.15	11.75	43.23

Table 3: 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.

abstractive and mixed which their model should be better suited for 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 8 and table 9, 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 7 shows similar qualitative examples on the Newsroom dataset. Tables 2, 3 and 4 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. It is important to note that, across datasets, introducing extracted sentences with TLM+I+E or TLM+E has consistently performed better over TLM+I or TLM. For bigPatent in table 4 and newsroom in table 5 TLM and TLM+E models have access to the same text since the whole article can fit in the transformer window size. This is particularly interesting since our results show that explicitly delimiting the extracted sentences has large positive affects on summary performance. As anticipated, introducing extracted sentences allows the TLM model to focus less on information retrieval and more on language generation.

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
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
Our Models				
Sent-CLF	Ext	36.20	10.99	31.83
Sent-PTR	Ext	34.21	10.78	30.07
TLM	Abs	36.41	11.38	30.88
TLM+E (G,M)	Mix	**38.65**	**12.31**	**34.09**
Oracle				
Gold Ext	Oracle	43.56	16.91	36.52
OracleFrag	Oracle	91.85	78.66	91.85
TLM+E (G,G)	Oracle	39.99	13.79	35.33

Table 4: Summarization results on the bigPatent dataset. Previous work results from Sharma et al. (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. Note that OracleFrag (Grusky et al., 2018b) (Extractive Oracle Fragments) is an an extraction heuristic that has access to the reference summary".

4.1 Abstractiveness of generated abstracts

Weber et al. (2018) argued that state-of-the-art ab-

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										
Sent-CLF	Ext	53.0	47.0	52.1	26.8	12.6	23.6	15.4	2.7	12.8
Sent-PTR	Ext	60.7	55.2	59.7	28.9	14.1	25.1	15.9	2.8	13.0
TLM	Abs	49.8	39.7	47.4	27.1	11.6	22.8	**20.4**	**6.9**	**17.1**
TLM+E (G,M)	Mix	63.3	57.3	61.8	**31.9**	**16.6**	**27.4**	20.1	6.5	16.6
Oracle										
Gold Ext	Oracle	68.1	64.5	67.3	40.8	24.6	34.2	21.9	5.2	16.3
TLM+E (G,G)	Oracle	78.8	74.0	77.8	38.6	22.0	33.6	24.5	9.6	20.8

Table 5: Summarization results on the Newsroom dataset. Previous work results from Grusky et al. (2018a) and Mendes et al. (2019). Note that extractive/mixed/abstractive columns denote the type of ground-truth summary. The Newsroom dataset has targets that are extracted from the input (extractive), that are created with heuristics (mixed) and that are created by humans (abstractive). Also note that the "Type" column refers to the model type for each row.

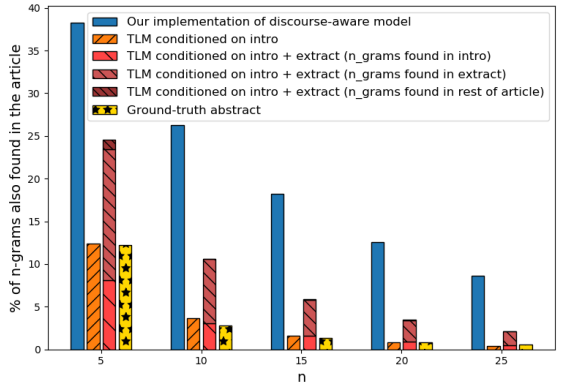

Figure 2: n-gram overlaps between the abstracts generated by different models and the input article on the arXiv dataset. We show in detail which part of the input was copied for our TLM conditioned on intro + extract.

stractive summarization systems that use a copy mechanism effectively generate the summary by copying over large chunks 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))[3] 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.

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

4.2 Human Evaluation

We performed a human evaluation using the same experimental setup as in (Grusky et al., 2018a) in Table 6. For the same 60 Newsroom test articles, we obtain the summaries for 5 different models (ground truth, sentence classifier, sentence pointer, TLM conditioned on article, TLM conditioned on article + pointer extracts).

Model	Type	Evaluation criteria			
		COH	FLU	INF	REL
Ground truth summaries	Orac	3.73	3.98	3.19	3.59
TLM - Intro + Extract	Mix	**3.78**	3.75	3.09	3.59
TLM - Intro	Mix	3.77	**3.90**	3.11	3.50
Sentence pointer	Ext	3.67	3.66	3.24	3.78
Sentence classifier	Ext	3.62	3.79	**3.47**	**3.89**

Table 6: Human evaluation on Newsroom abstractive summarization test data. Each pair of (article, summary) is presented to three unique crowd workers, who are asked to judge the summaries along four criteria: Coherence (COH: does the summary make sense as a whole), Fluency (FLU: is it well written), Informativeness (INF: does the summary catch the most important points of the article), and Relevance (REL: are the facts in the summary consistent with the article).

As expected, Transformers are quite good making coherent and fluent summaries but not necessarily on informativeness and relevance. Transformers have a logarithmic or constant path length (as opposed to linear in RNNs) between a networks output and any of its inputs, making gradient flow much easier. This is a clear advantage over RNNs that tend to repeat sentences. Transformers are also known to hallucinate (Lee et al., 2019) but we notice that including extracted sentences, TLM

+ Intro + Extract, improve relevance by 3% over TLM + Intro, bringing relevance closer to extractive methods. Interestingly, on Coherence, both our TLM variants also score better than the ground truth. Over the four categories, TLM + Intro + Extract performs best on average over TLM + Intro, despite the former having higher ROUGE scores on the abstractive test set in table 5. Somewhat counter-intuitively we observe that human written summaries are often rated lower than model summaries. However, other work has also found that human written ground truth summaries consistently receive lower scores when compared to model written summaries when evaluated by turkers (see for example Table 3 in the PEGASUS paper of (Zhang et al., 2020)). We believe that this could be because Newsroom summaries are sometimes noisy, ungrammatical and incoherent.

Document — A new plan from the government of the Philippines would offer free wireless internet to people across the country while also likely eating into the annual revenue of the nations telecoms. Bloomberg reports that the Philippines government plans to roll-out its free Wi-Fi services to roughly half of the countrys municipalities over the next few months and the country has its sights set on nationwide coverage by the end of 2016. The free wireless internet service will be made available in public areas such as schools, hospitals, airports and parks, and is expected to cost the government roughly $32 million per year. [...]
Abstractive — : The government is reportedly considering a nationwide service plan to give free Wi-Fi access to rural areas.
Mixed — The government of the Philippines is considering a new plan to provide free wireless internet to the nation's largest cities and towns.
Extractive — The new plan will include free wireless internet to residents across the country while also probably eating into the annual revenue of the country's telecoms.
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 7: Qualitative Results - News articles and our model generated summaries on the NewsRoom dataset

4.3 Qualitative Results

Here we provide some qualitative results. Running our algorithm on a close to final version of this paper (excluding this section) and selecting the best sample from a set of 10-20 runs we found the following abstract: "we present a hybrid extractive and abstractive approach for generating summaries from long documents. we use an initial extractive step that reduces the amount of context for a subsequent abstractive step (see figure [fig: model]). we show that this approach can produce a good summarization quality on both short and long documents, even without using copying and pointing mechanisms. further, by considering the context in both the text and the discourse, we find that the hybrid approach is effective at capturing the underlying context. we examine these models through rouge scores, through a study of the amount of n-gram copying performed by different models, as well as through a human evaluation using a standard protocol. our results show that our hybrid approach yields results that outperform current state-of-the-art results on several metrics of these evaluations."

5 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. As with other problem domains, we have observed that abstractive summaries generated by transformers can generate imaginary content. We advise 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. While we believe that this work is a step forward towards generating more abstractive summaries, it remains an open challenge to develop abstactive models that respect the underlying facts of the content being summarized while matching the creative ability of humans to coherently and concisely synthesize summaries. However, the TLMs we use here are of moderate size compared to what is now possible. It would be very interesting to see what kind of performance larger models could achieve.

Acknowledgments

We would like to thank Nicolas Chapados and Harm de Vries for their helpful feedback and Minh Dao for helping with the figures. We would also like to thank Guillaume Lample for discussions about TLM training.

Generated abstract for "the attention model paper" (Bahdanau et al., 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).
Generated abstract for "the seq2seq paper" (Sutskever et al., 2014) - We present a novel approach to sequence to sequence learning using deep recurrent neural networks (LSTMs). First , we transform a sequence to a vector representation that contains all the long short term memory (LSTM) features, and then we use the LSTM to generate elements of the output sequence using a simple left to right beam-search decoder. The advantage of this approach is that it learns to map all the words in the input sequence to a fixed dimensional vector representation, while the LSTM does not require any special training to learn the sentence representation. We demonstrate the LSTMs ability to train on the WMT14 English to French translation task using a simple left to right beam-search decoder. The model was found to be highly invariant to the active and passive voice.	**Original abstract** - Deep Neural Networks (DNNs) are powerful models that have achieved excellent performance on difficult learning tasks. Although DNNs work well whenever large labeled training sets are available, they cannot be used to map sequences to sequences. In this paper, we present a general end-to-end approach to sequence learning that makes minimal assumptions on the sequence structure. Our method uses a multilayered Long Short-Term Memory (LSTM) to map the input sequence to a vector of a fixed dimensionality, and then another deep LSTM to decode the target sequence from the vector. Our main result is that on an English to French translation task from the WMT'14 dataset, the translations produced by the LSTM achieve a BLEU score of 34.8 on the entire test set, where the LSTM's BLEU score was penalized on out-of-vocabulary words. Additionally, the LSTM did not have difficulty on long sentences. For comparison, a phrase-based SMT system achieves a BLEU score of 33.3 on the same dataset. When we used the LSTM to rerank the 1000 hypotheses produced by the aforementioned SMT system, its BLEU score increases to 36.5, which is close to the previous best result on this task. The LSTM also learned sensible phrase and sentence representations that are sensitive to word order and are relatively invariant to the active and the passive voice. Finally, we found that reversing the order of the words in all source sentences (but not target sentences) improved the LSTM's performance markedly, because doing so introduced many short term dependencies between the source and the target sentence which made the optimization problem easier.
Generated abstract for "the GAN paper" (Goodfellow et al., 2014) - Adversarial nets are a class of deep generative networks that utilize the advantages of piecewise linear units. Here, we explore a novel model estimation procedure that combines several methods, such as backpropagation and dropout, to learn generative models. The proposed adversarial net framework can be thought of as analog to a team of counterfeiters, who try to produce fake currency and use it without being detected, while the discriminative model is analogous to the police, trying to detect the counterfeit currency. We refer to this special case as adversarial nets. We explore the special case when the generative model generates samples by passing random noise through a multilayer perceptron, and the discriminative model is also a multilayer perceptron.	**Original abstract** - We propose a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D, a unique solution exists, with G recovering the training data distribution and D equal to 1/2 everywhere. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation. There is no need for any Markov chains or unrolled approximate inference networks during either training or generation of samples. Experiments demonstrate the potential of the framework through qualitative and quantitative evaluation of the generated samples.

Table 8: Qualitative Results — Generated abstracts of select papers using our Intro Only TLM.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. *arXiv preprint arXiv:1805.11080*.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. *arXiv preprint arXiv:1603.07252*.

Sumit Chopra, Michael Auli, and Alexander M Rush. 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*, pages 93–98.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. *CoRR*, abs/1804.05685.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*.

Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. 2018. Bottom-up abstractive summarization. *arXiv preprint arXiv:1808.10792*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. pages 2672–2680.

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

Max Grusky, Mor Naaman, and Yoav Artzi. 2018a. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. *arXiv preprint arXiv:1804.11283*.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018b. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana. Association for Computational Linguistics.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O. K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *CoRR*, abs/1603.06393.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. *arXiv preprint arXiv:1603.08148*.

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

Wan-Ting Hsu, Chieh-Kai Lin, Ming-Ying Lee, Kerui Min, Jing Tang, and Min Sun. 2018. A unified model for extractive and abstractive summarization using inconsistency loss. *arXiv preprint arXiv:1805.06266*.

Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. 2019. Hallucinations in neural machine translation.

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

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Afonso Mendes, Shashi Narayan, Sebastião Miranda, Zita Marinho, André FT Martins, and Shay B Cohen. 2019. Jointly extracting and compressing documents with summary state representations. *arXiv preprint arXiv:1904.02020*.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. 2017. Mixed precision training. *arXiv preprint arXiv:1710.03740*.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016a. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Ramesh Nallapati, Bowen Zhou, and Mingbo Ma. 2016b. Classify or select: Neural architectures for extractive document summarization. *arXiv preprint arXiv:1611.04244*.

Jun-Ping Ng and Viktoria Abrecht. 2015. Better summarization evaluation with word embeddings for ROUGE. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1925–1930, Lisbon, Portugal. Association for Computational Linguistics.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. *CoRR*, abs/1704.04368.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Eva Sharma, Chen Li, and Lu Wang. 2019. Bigpatent: A large-scale dataset for abstractive and coherent summarization. *arXiv preprint arXiv:1906.03741*.

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

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. pages 3104–3112.

Lucy Vanderwende, Hisami Suzuki, Chris Brockett, and Ani Nenkova. 2007. Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing & Management*, 43(6):1606–1618.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. pages 2692–2700.

Noah Weber, Leena Shekhar, Niranjan Balasubramanian, and Kyunghyun Cho. 2018. Controlling decoding for more abstractive summaries with copy-based networks. *arXiv preprint arXiv:1803.07038*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *ICML*.

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 9: 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.

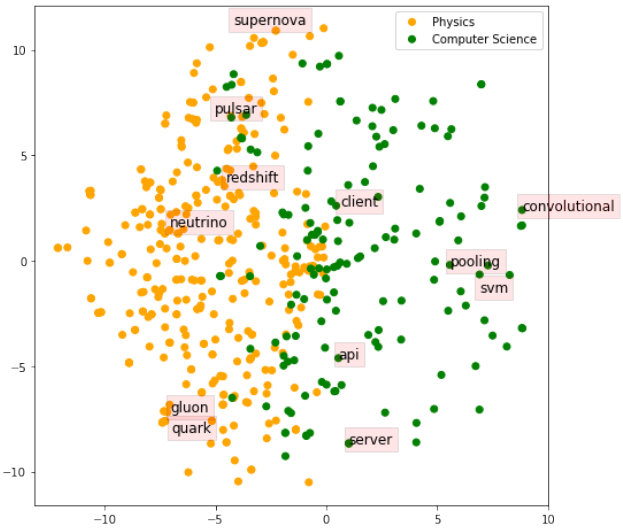

Figure 3: t-sne visualization of the TLM-learned word embeddings. The model appears to partition the space based on the broad paper categoty in which it frequently occurs.

Extractive 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.