Beyond Paragraphs: Language Modeling of Long Sequences

Iz Beltagy

Long sequences are important for NLP

- Many NLP tasks require processing full documents
  - e.g: QA, summarization, document-classification, Coref resolution
- Semantic scholar: 90% percentile paper length = 7,000 tokens
- Wikipedia: about the same as Semantic Scholar

- It is a simple input format for tasks with many short inputs: concatenate in one long string
  - Multi-document summarization
  - Dialogue

But long sequences + Transformers are hard

- Self-attention is expensive
  - Every token attends to every other token
  - Memory and compute are $O(n^2)$

- and as the model get larger, even the O(n) components (feed forwarded) become expensive when n is large

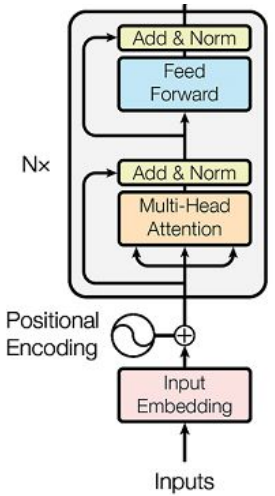


Photo credit: Attention Is All You Need, Vaswani et. al., 2017

Previously,

- Avoid working with long documents
  - e.g. MRQA: skip the question if the answer is not in the first 800 tokens

- or Chunk, extract, combine
  - task and dataset specific
  - loses global context

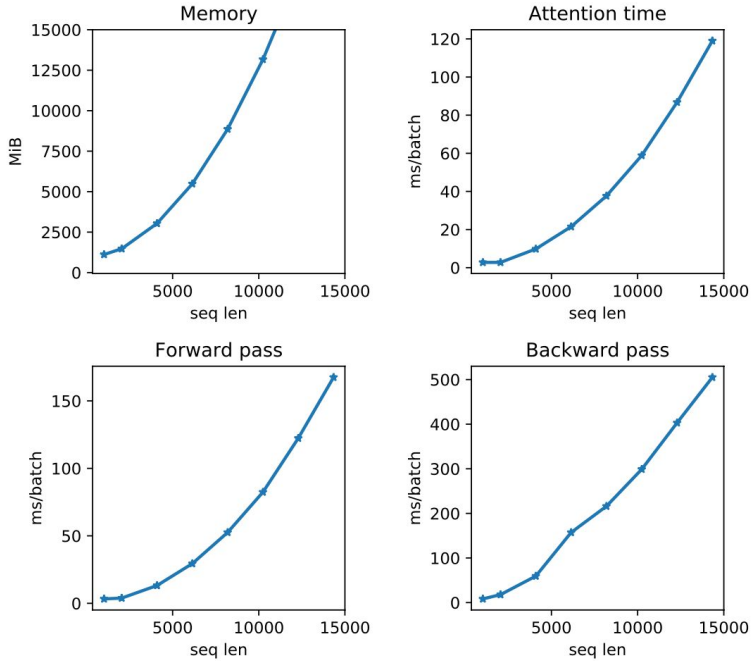
Agenda

- Transformer models that scale to long sequences
  - Dealing with $O(n^2)$ self-attention
  - Longformer and LongformerEncoderDecoder (LED)
- How to train these models
  - Cross document language modeling (CDLM)
  - Zero and fewshot multi-document summarization (PRIMERA)
- Thoughts about scaling to large models and applying to dialogue tasks

Extending Transformers to longer inputs

Recall the computational cost of self-attention O(n^2)

Quickly becomes a bottleneck for long sequences



Single-head single-layer transformer
Titan RTX8000 GPU

Extending Transformers to longer inputs

Many methods to reduce memory and compute cost of Self-attention

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

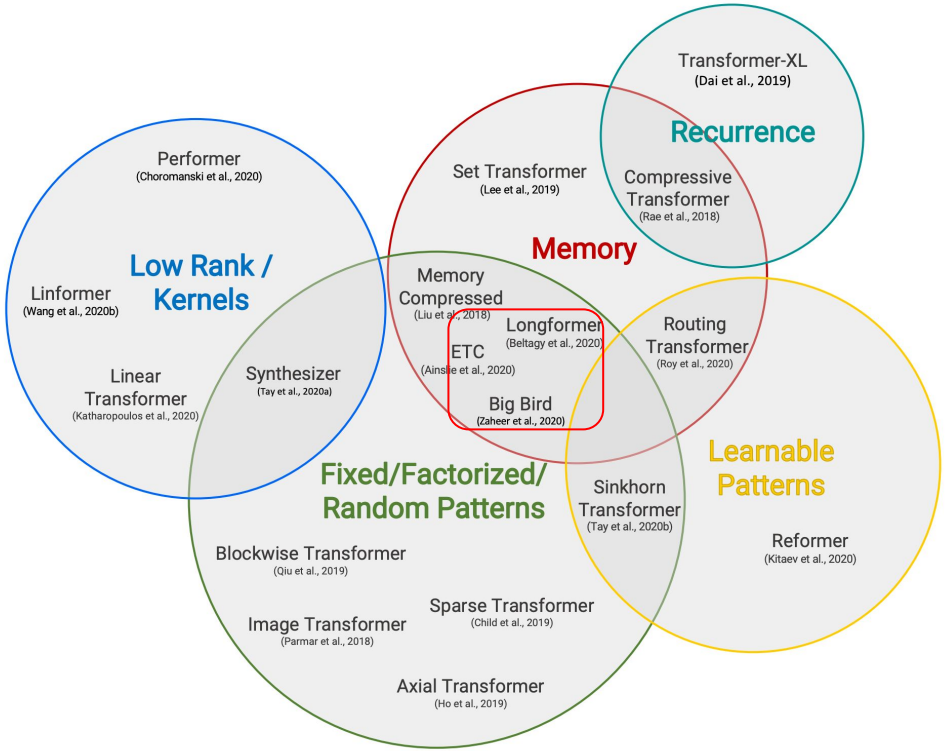
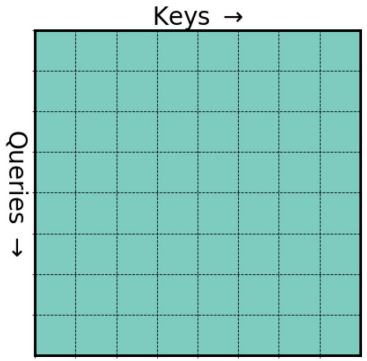


Photo credit: Efficient Transformers: A Survey, Tay et. al., 2020

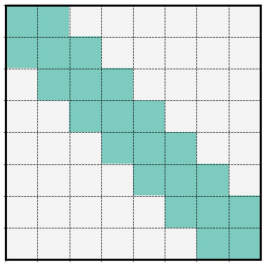
Pre-specified sparsity patterns



Sparsify the attention matrix (QK^T)

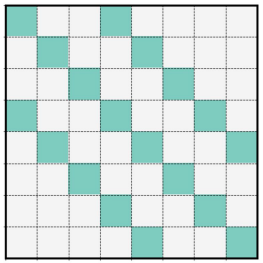
A variety of patterns has been explored in past work

Sliding window



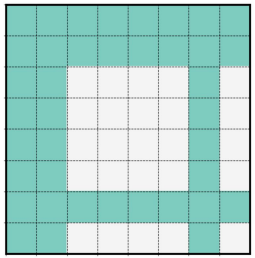
Sparse Transformer
Longformer

Dilated



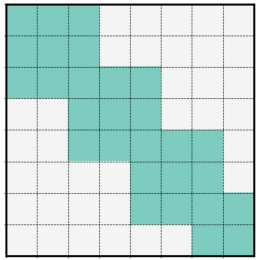
Longformer

Global



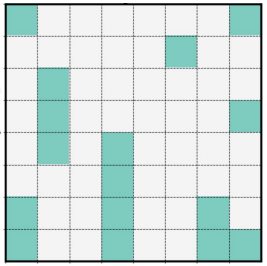
Longformer
ETC
Big Bird
GMAT

Blocked



Longformer
Block-wise
ETC
Big Bird
Sinkhorn

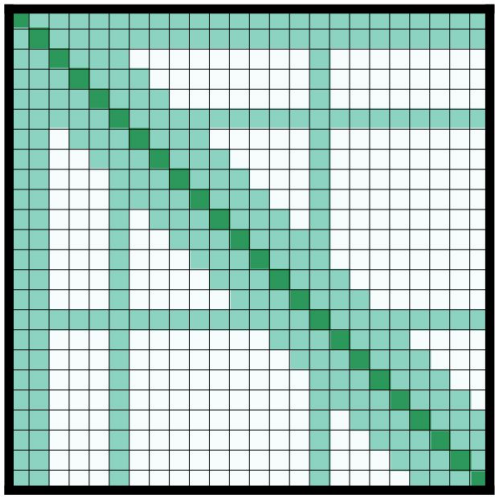
Random



Big Bird

Longformer - local and global attention

- Local attention: sliding window
  - every token attends to a fixed width window around it
  - Intuition: local context is enough to learn good token representation

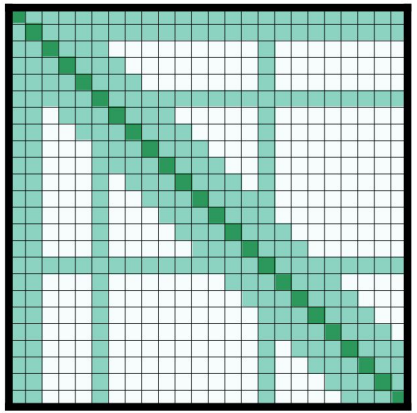


- Global attention: user defined based on the task
  - tokens that attend to every other token and every token attends back to them

Longformer - local and global attention

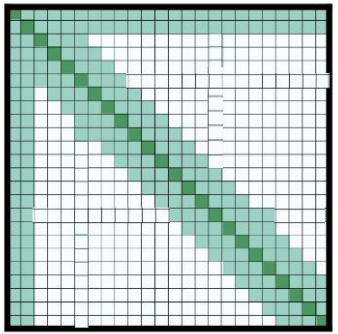
Global attention - flexibility to encode task-specific inductive bias

- LM: local representation
- Classification: aggregate seq into CLS
- QA: compare context and question tokens
- Coreference resolution: share information across mentions



Note: indirect information flow is not enough. Direct attention is needed

- Local attention only is not enough
- Block of global attention as a prefix is not enough



.0

Implementation

Arbitrary sparse matrix multiplication is not supported in DL libraries

Solutions:

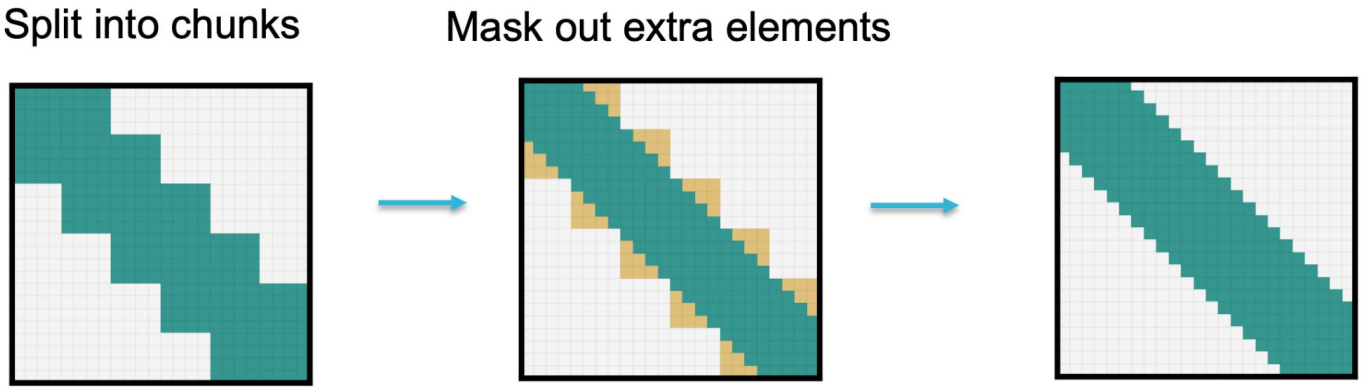
- Perform computations in blocks (Big Bird/ETC, Longformer, Block-wise)
- Customized Kernels (Sparse Transformers, Longformer)

Implementation

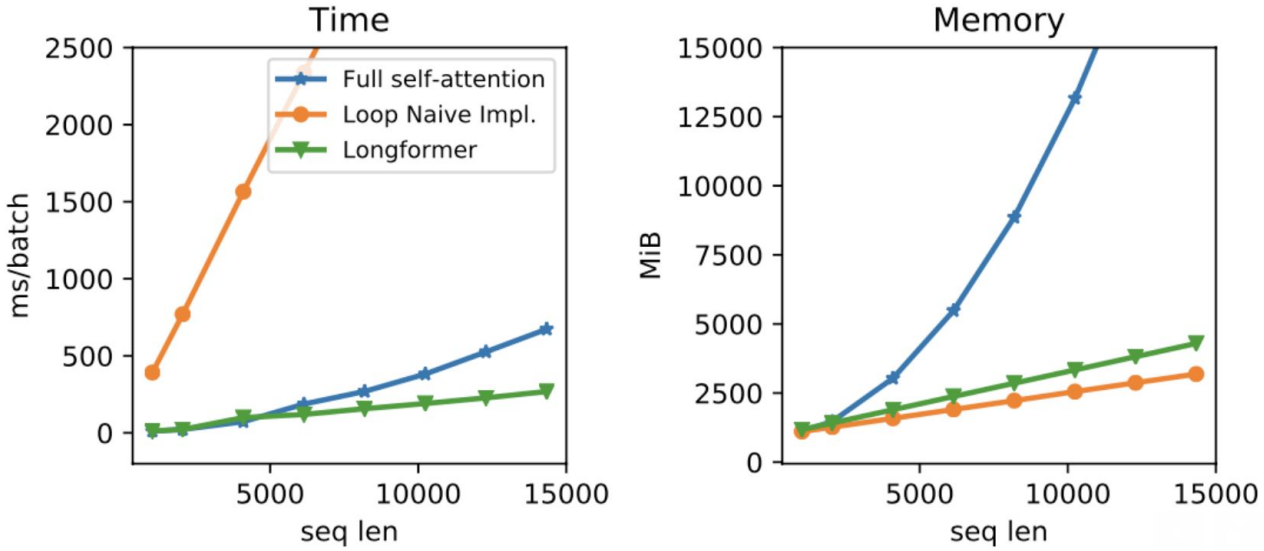
Longformer

Efficient implementation of sparsity using blocks

More computation than needed, but ensures symmetric local attention windows



Sparsifying the attention



Evaluation (1) - Character Level Language Modeling

dataset - Metric (bpc, lower is better)	Ours	SOTA (small model)
enwik8	0.999	1.02
text8	1.103	1.11

Large improvement

- Match SOTA result on the large model
- Largest train seqlen is 23k, evaluate on seqlen 32k
- fp16 and gradient checkpointing are important

Character Level Language Modeling - Details

- Autoregressive language model training from scratch
- Many different ways to configure window sizes
  - increasing window size from 512 to 8192
  - seqlen 32k
- Training in 5 phases, with every phase, 2x window size and seqlen and 0.5x LR
  - starting window sizes: 32 to 512
  - starting seqlen 2048
- Doubling window size has the most impact on bpc
  - At first, loss increases a lot then drops quickly
- Keep selfattention in fp32 because the model learns to use large attention scores

Pretraining and finetuning

Most prior work only focus on language modeling - no pretraining and downstream finetuning.

Models that work well for language modeling don’t necessarily work for downstream tasks.

Pretraining and finetuning

Pretraining procedure for Longformer starting from the RoBERTA checkpoint

1. Replace full self-attention with longformer sliding-window attention
2. Increase size of position embedding matrix. Initialize it by copying the first 512
3. Continue MLM pretraining on a corpus of long docs
4. Copy q, k, v linear projections to get separate projections for global attention

This procedure can be easily applied to other models to get them to work for long documents

Pretraining - MLM results

Model	base	large
RoBERTa (seqlen: 512)	1.846	1.496
Longformer (seqlen: 4,096)	10.299	8.738
+ copy position embeddings	1.957	1.597
+ 2K gradient updates	1.753	1.414
+ 65K gradient updates	1.705	1.358

Finetuning on downstream tasks

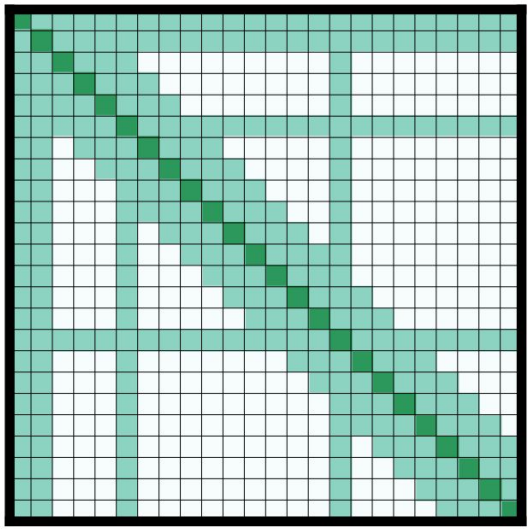
- HotpotQA -- Multihop reasoning and evidence extraction
- TriviaQA -- QA dataset from long Wikipedia articles
- WikiHop -- QA requiring combining facts spread across multiple paragraphs
- Coref -- Coreference chains across the document
- IMDB -- Document classification
- Hyperpartisan news -- Document classification

Finetuning on downstream tasks - Global attention

Classification: on CLS token

QA: on all question tokens

Coref: local attention only



Finetuning on downstream tasks - Results

	QA			Coref.	Classification	
Model	WikiHop	TriviaQA	HotpotQA	OntoNotes	IMDB	Hyperpartisan
RoBERTa-base	72.4	74.3	63.5	78.4	95.3	87.4
Longformer-base	**75.0**	**75.2**	**64.4**	**78.6**	**95.7**	**94.8**

Finetuning on downstream tasks - large model

SOTA on WikiHop and TriviaQA

Model	WikiHop	TriviaQA	HotpotQA
Current SOTA	78.3	73.3	**74.2**
Longformer-large	**81.9**	**77.3**	73.2

Competitive HotpotQA result

- Ours is simpler
- Better models use GNN of entities
  - Can we simulate it with global attention?

Model	ans.	supp.	joint
TAP 2 (ensemble) (Glaß et al., 2019)	79.8	86.7	70.7
SAE (Tu et al., 2019)	79.6	86.7	71.4
Quark (dev) (Groeneveld et al., 2020)	81.2	87.0	72.3
C2F Reader (Shao et al., 2020)	81.2	87.6	72.8
Longformer-large	81.3	88.3	73.2
ETC-large†	81.2	89.1	73.6
GSAN-large†	81.6	88.7	73.9
HGN-large (Fang et al., 2019)	82.2	88.5	74.2

Finetuning on downstream tasks - ablation

Model	Accuracy / Δ
Longformer (seqlen: 4,096)	73.8
RoBERTa-base (seqlen: 512)	72.4 / -1.4
Longformer (seqlen: 4,096, 15 epochs)	75.0 / +1.2
Longformer (seqlen: 512, attention: n^2)	71.7 / -2.1
Longformer (seqlen: 512, attention: window)	68.8 / -5.0
Longformer (seqlen: 2,048)	73.1 / -0.7
Longformer (no MLM pretraining)	73.2 / -0.6
Longformer (no linear proj.)	72.2 / -1.6
Longformer (no linear proj. no global atten.)	65.5 / -8.3

Table 11: WikiHop development set ablations

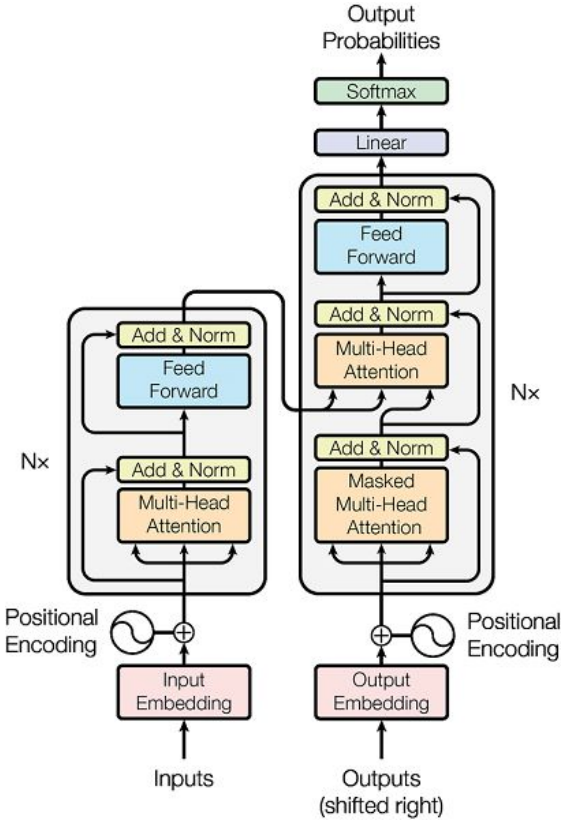
LongformerEncoderDecoder (LED)

So far, we explored

- Decoder-only model: auto-regressive LM
- Encoder-only model: MLM pretraining

Next - EncoderDecoder model

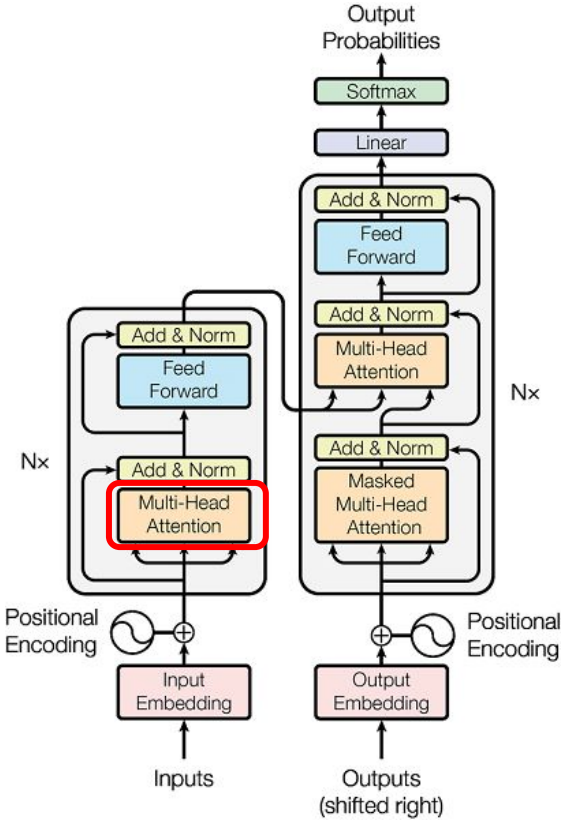
- Important for seq2seq NLP tasks (e.g., summarization, translation)



LongformerEncoderDecoder (LED)

Assuming that the output sequence length is short

- Encoder self-attention
  - Size: input x input
  - Replace with sparse attention
- Decoder self-attention
  - Size: output x output
  - Keep full self-attention
- Decoder cross-attention
  - Size: input x output
  - Keep full self-attention

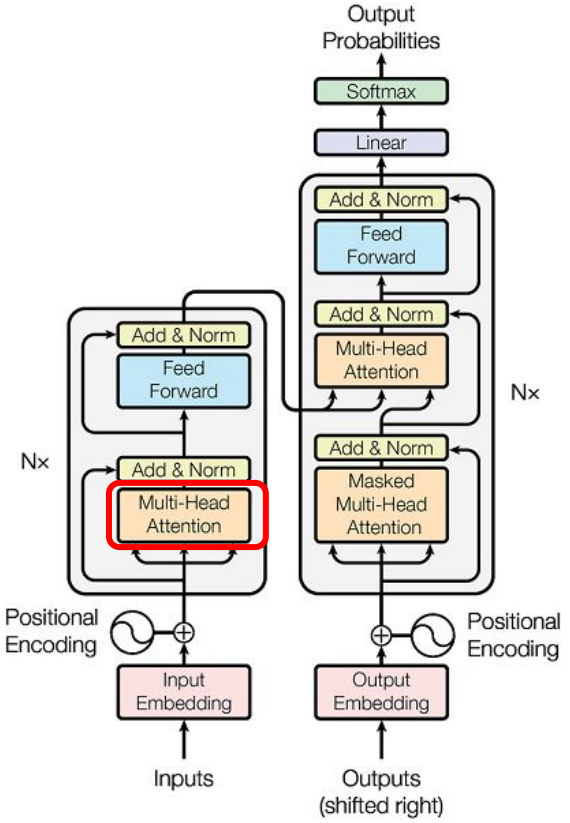


LongformerEncoderDecoder (LED)

Building model

- Starting from BART
- Replace self-attention of encoder with sparse-attention
- Extends position embeddings by copying the first 512 positions repeatedly

- No additional training



LongformerEncoderDecoder (LED) - Evaluation

Summarization results - arXiv dataset

Strong results by processing 16K token long documents without further pretraining

	R-1	R-2	R-L
Discourse-aware (2018)	35.80	11.05	31.80
Extr-Abst-TLM (2020)	41.62	14.69	38.03
Dancer (2020)	42.70	16.54	38.44
Pegasus (2020)	44.21	16.95	38.83
LED-large (seqlen: 4,096) (ours)	44.40	17.94	39.76
BigBird (seqlen: 4,096) (2020)	**46.63**	19.02	41.77
LED-large (seqlen: 16,384) (ours)	**46.63**	**19.62**	**41.83**

Conclusion

We should consider tasks beyond LM to develop and evaluate long-sequence models

Local + global spare attention pattern is a good inductive bias for a wide range of tasks

Global attention is task specific

```
model = transformers.AutoModel('allenai/longformer-base-4096')
```

```
model = transformers.AutoModel('allenai/led-base-16384')
```

Procedure to add long-sequence support to existing pretrained checkpoints

Agenda

- Transformer models that scale to long sequences
  - Dealing with $O(n^2)$ self-attention
  - Longformer and LongformerEncoderDecoder (LED)
- How to train these models
  - **Cross document language modeling (CDLM)**
  - Zero and fewshot multi-document summarization (PRIMERA)
- Thoughts about scaling to large models and applying to dialogue tasks

Bar-Ilan University
אוניברסיטת בר־אילן

BIU NLP

CDLM: Cross-Document Language Modeling

Findings of EMNLP 2021 🏝️

Avi Caciularu

Arman Cohan

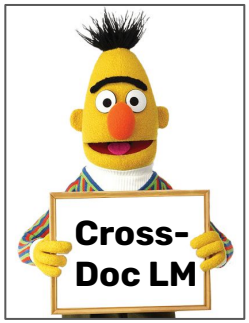
Iz Beltagy

Matt Peters

Arie Cattan

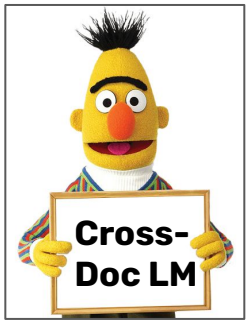
Ido Dagan

LMs - Single text Vs. Multi-text



LMs - Single text Vs. Multi-text

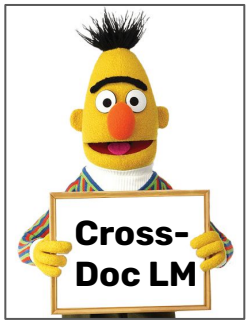
- Pretrained on **single texts**



- Pretrained on **multiple texts** in parallel

LMs - Single text Vs. Multi-text

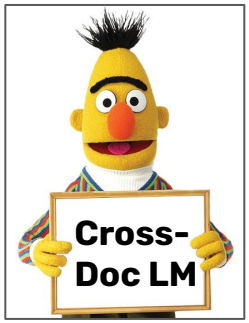
- Pretrained on **single texts**
- Powerful representations for **internal text structure** (syntax, semantics, ...)



- Pretrained on **multiple texts** in parallel
- Powerful **cross-text relationships**, particularly mapping and alignment

LMs - Single text Vs. Multi-text

- Pretrained on **single texts**
- Powerful representations for **internal text structure** (syntax, semantics, ...)
- Great for **single text** tasks (QA, POS, ...)



- Pretrained on **multiple texts** in parallel
- Powerful **cross-text relationships**, particularly mapping and alignment
- The first general LM for **multi-document tasks**

Cross-Document LM

MLM Training - two key ideas

1) Masked tokens should be able to attend to all documents

- “*cross-document masked tokens”*

2) Train on clusters of “related documents”

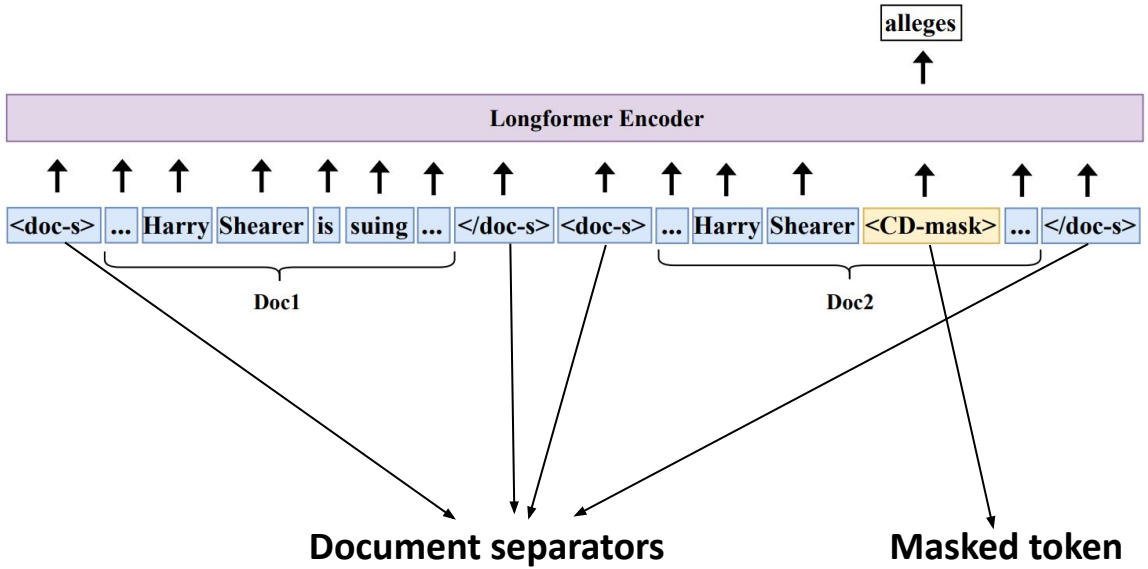
Together, the model

- can look beyond the document boundary
- and can find useful information in other documents to predict the mask
- thus, learning to aggregate information across documents and producing cross-document representations,

Cross-Document Language Modeling (CDLM)

global attention

local attention



- Train on a concatenation of multiple **related documents** in one long sequence
- Use **Longformer with Global Attention** on masked tokens

Outline

- Cross-Document Tasks
- Our approach: Cross-Document Masking
- **Evaluation**
- Conclusion

Ablation Models - Training Data

CDLM

Related documents

Random CDLM

Unrelated documents

Longformer

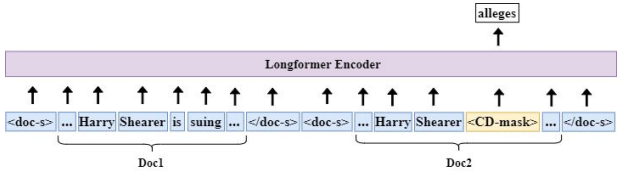
Single documents

Ablation Models - Attention Pattern

global attention

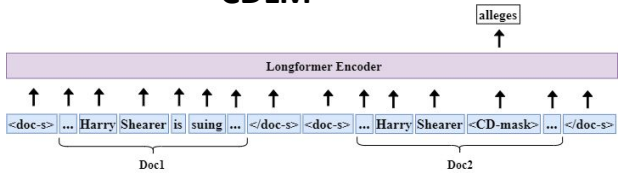
local attention

CDLM



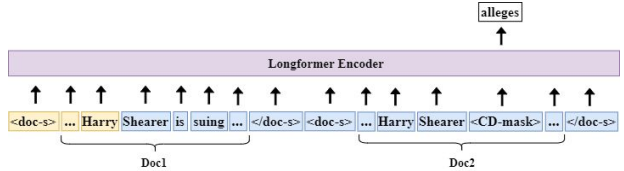
Global attention on masked tokens

Local CDLM



No global attention

Prefix CDLM (as in BigBird)



Global attention on the first tokens

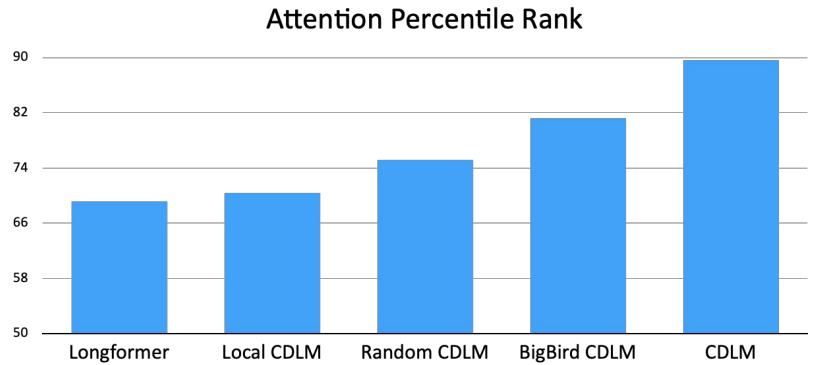
Ablation Results

	F1	Δ
full document CDLM	85.6	
– Longformer	84.6	-1.0
– Local CDLM	84.7	-0.9
– Rand CDLM	84.1	-1.5
– Prefix CDLM	85.1	-0.5

Table 3: Ablation results (CoNLL F1) on our model on the test set of ECB+ event coreference.

Attention Analysis

Doc 1: President Obama will **name** Dr. Regina Benjamin as U.S. Surgeon General in a Rose Garden announcement late this morning. Benjamin, an Alabama family physician, [...]
Doc 2: [...] Obama **nominates** new surgeon general: MacArthur "genius grant "fellow Regina Benjamin. [...]



More analysis in our paper!

Agenda

- Transformer models that scale to long sequences
  - Dealing with $O(n^2)$ self-attention
  - Longformer and LongformerEncoderDecoder (LED)
- How to train these models
  - Cross document language modeling (CDLM)
  - **Zero and fewshot multi-document summarization (PRIMERA)**
- Thoughts about scaling to large models and applying to dialogue tasks

PRIMERA
Pyramid-based Masked Sentence Pre-training for Multi-document Summarization

Wen Xiao[1], Iz Beltagy[2], Giuseppe Carenini[1], Arman Cohan[2]

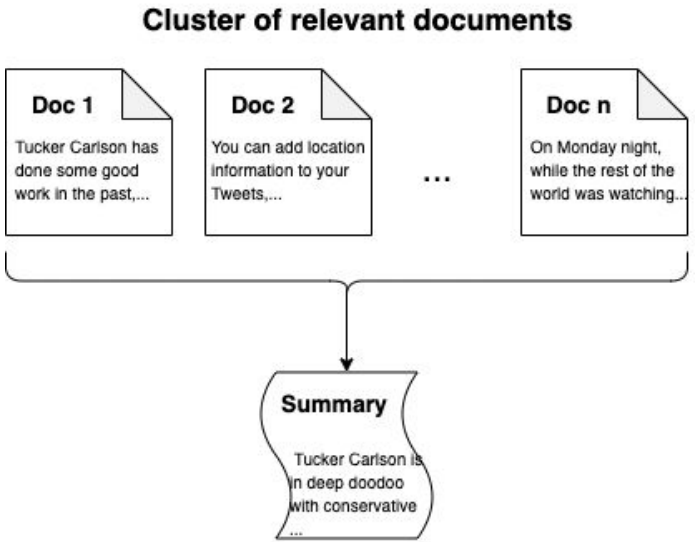
[1] University of British Columbia
[2] Allen Institute of Artificial Intelligence

What is Multi-Document Summarization (MDS)?

Task: Generate a summary given **a cluster of relevant documents,**

e.g.

- News articles
- Scientific papers

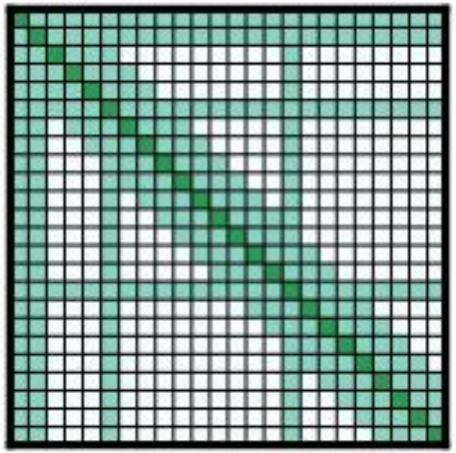


Previous Methods for MDS

- Dataset-specific Models
  - Graph-based — Require domain specific additional information
  - Hierarchical — Use dataset-specific, customized architectures, hard to leverage pre-trained LM
- Pre-trained Generation Models
  - General-Purpose (BART, T5,...) — Require large amount of data to fine-tune
  - Task-specific
    - PEGASUS — Mainly focus on single-document input
    - PRIMERA — Mainly for Multi-document Summarization

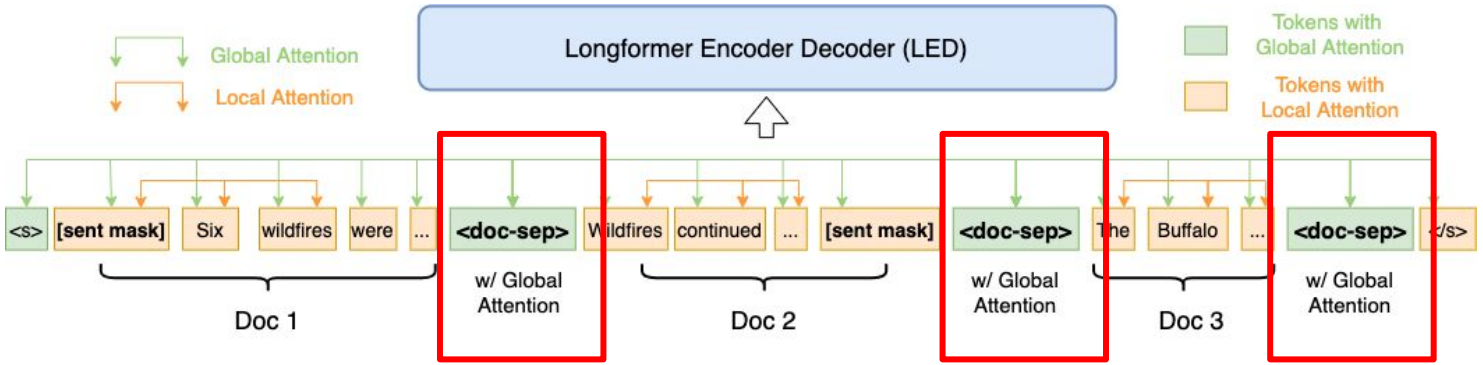
Overview of PRIMERA

- Architecture: Longformer Encoder Decoder (LED)
  - Global + Local Attention
  - Allows for long sequence inputs

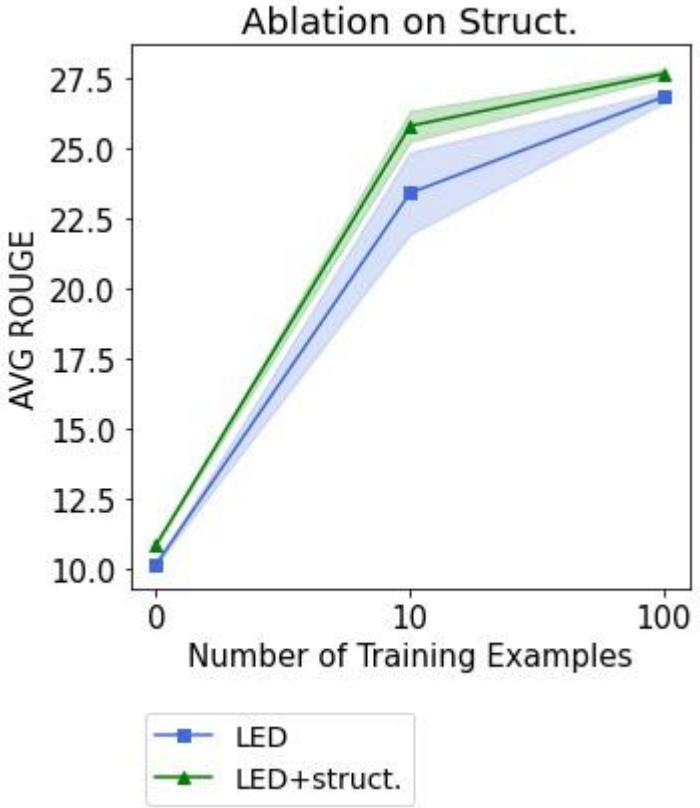


Overview of PRIMERA

- Architecture: Longformer Encoder Decoder (LED)
  - Global + Local Attention
  - Allows for long sequence input
- Input Structure:
  - documents separated with document separator(*<doc-sep>*)



Impact of the Proposed Input Structure



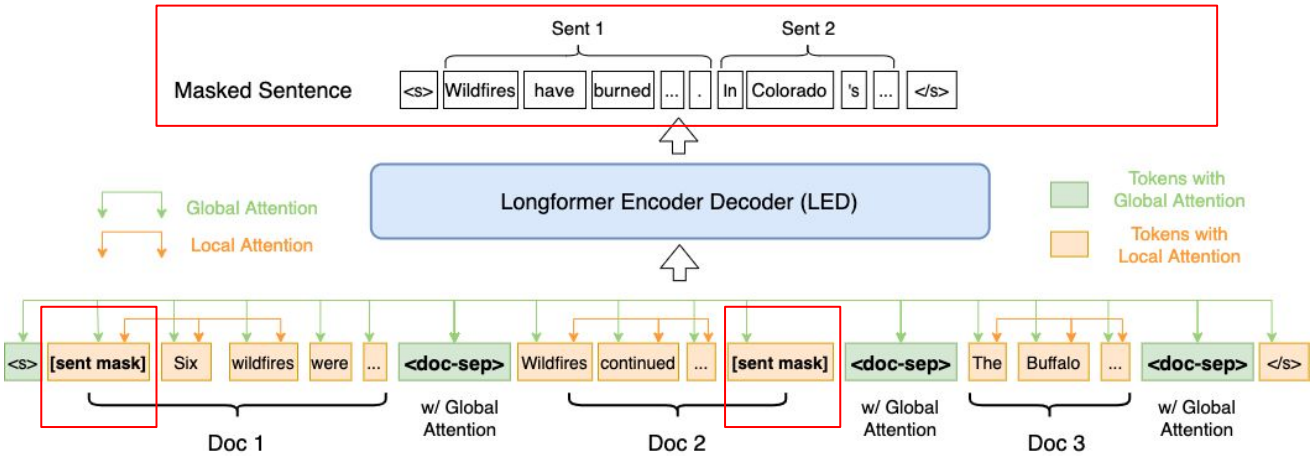
- Input structure improves the results

Overview of PRIMERA

- Architecture: Longformer (with local and global attention)
  - Global + Local Attention
  - Allows for long sequence input
- Input Structure:
  - documents separated with document separator(<*doc-sep*>)
  - Global Attention on <*doc-sep*>
- Pre-training:
  - **Goal:** Teach the model to identify and aggregate salient information across a “cluster” of related documents
  - **Multi-doc corpus:** Newshead (360k clusters, 3.5 doc/cluster on average)
  - **Objective**: Gap Sentence Generation (GSG)
  - **Novel Masking Strategy:** Entity Pyramid

Pre-training : Objective

- **Gap Sentence Generation** [Zhang et al., 2020]:
  - Select several SALIENT sentences from the input documents (as pseudo-summary)
  - Mask out the selected sentences
  - Generate them in order in the decoder



How to select SALIENT sentences?

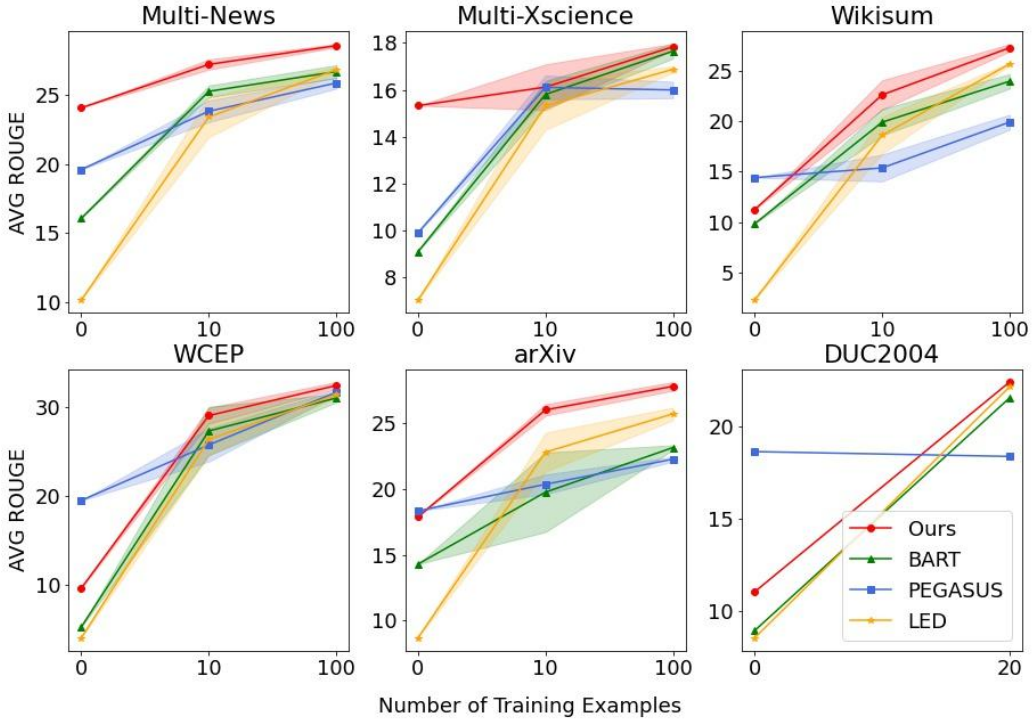
Previous work for single document input (PEGASUS):

- **Intuition**: select the **most central** sentences in the document
- The score is defined as the ROUGE score between **each sentence** and **rest of the document**

Doesn't work for multi-document summarization

- Our proposed method: Entity Pyramid

Zero and Few-shot Evaluation



- Our model outperform all the other pre-trained models on all the datasets.

Fully-supervised Evaluation

DATASETS	Prev. SOTA			PRIMERA		
	R1	R2	RL	R1	R2	RL
Multi-News	49.2	19.6	24.5	**49.9**	**21.1**	**25.9**
Multi-XScience	**34.1**	6.8	18.2	31.9	**7.4**	18.0
WCEP	35.4	15.1	25.6	**46.1**	**25.2**	**37.9**
arXiv	46.6	19.6	41.8	**47.6**	**20.8**	**42.6**

- Our model achieves SOTA on several multi-document summarization datasets, as well a single-document summarization dataset.

Takeaway

- With the right model, concatenating everything in one long string is a powerful input representation
- **CDLM** and **PRIMERA** are examples of training long-document models
- For best results, design the attention pattern and its accompanying pretrining objective
- In addition to local attention, design the global attention to encode task-specific inductive bias

```
model = transformers.AutoModel('biu-nlp/cdlm')

model = transformers.AutoModel('allenai/PRIMERA')
```

Agenda

- Transformer models that scale to long sequences
  - Dealing with $O(n^2)$ self-attention
  - Longformer and LongformerEncoderDecoder (LED)
- How to train these models
  - Cross document language modeling (CDLM)
  - Zero and fewshot multi-document summarization (PRIMERA)
- **Thoughts about scaling to large models and applying to dialogue tasks**

Scaling to larger models

$$\text{compute} \approx 2\, n_{\text{layer}}\, d_{\text{model}}\, (12\, d_{\text{model}} + n_{\text{ctx}})$$

GPT3-125M (small model)

- $12\, d_{\text{model}} = 9{,}216$
- n_{ctx} term is significant

GPT3-175B (huge model)

- $12\, d_{\text{model}} = 147{,}456$
- n_{ctx} term is insignificant

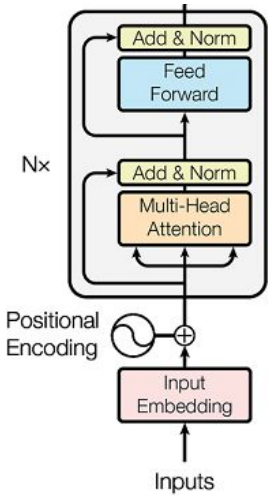


Photo credit: Attention Is All You Need, Vaswani et. al., 2017

Scaling Laws for Neural Language Models - Kaplan et. al., 2020

Scaling to larger models

$$\text{compute} \approx 2 \times n_{\text{layer}}\, d_{\text{model}}\, (12 \times d_{\text{model}} + n_{\text{ctx}})$$

- As the model size increase, the FF layers become the bottleneck, not the $O(n^2)$ self-attention.

- GPU memory becomes an issue and will need model parallelism

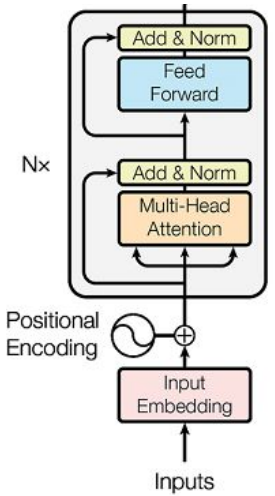


Photo credit: Attention Is All You Need, Vaswani et. al., 2017

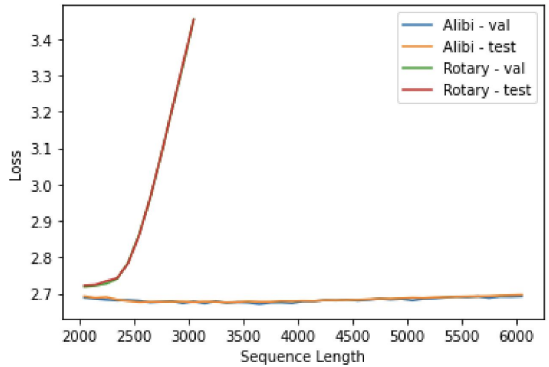
Scaling Laws for Neural Language Models - Kaplan et. al., 2020

Scaling to larger models - Training

Position embedding

- Use ALiBi
  - Better LM performance
  - Inference can extrapolate to lengths haven't seen during training
  - Train on short sequences and test on longer ones

Positional Embedding	Average EAI Results
None	41.23
Learned	41.71
Rotary	41.46
ALiBi	**43.70**

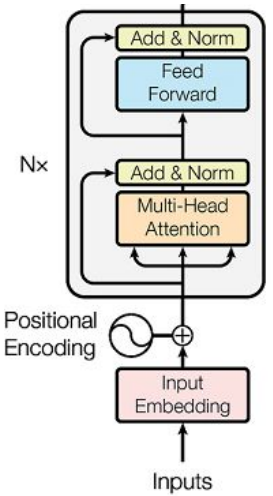


What Language Model to Train if You Have One Million GPU Hours? - Scao et. al., 2022

Scaling to larger models - Training

Pretraining objective

- Use autoregressive LM
  - Enables efficient caching at test time
    - Only cache Q, K, V of previous tokens
  - Use sparse attention with local + global mask to control size of cache



Apply to dialogue and conversations

Represent the dialogue as one long string

Use auto-regressive LM to enable adding responses incrementally

Design your global attention based on the application

- e.g. global attention on the first token of each sentence

Agenda

- Transformer models that scale to long sequences
  - Dealing with $O(n^2)$ self-attention
  - Longformer and LongformerEncoderDecoder (LED)
- How to train these models
  - Cross document language modeling (CDLM)
  - Zero and fewshot multi-document summarization (PRIMERA)
- Thoughts about scaling to large models and applying to dialogue tasks

Conclusion

- Many modeling innovations that enable transformer to process long sequences
- Concatenating everything in one long string is a powerful input representation
- Sparse attention pattern of local + global can be flexible and efficient
- Design a pretraining objective that goes well with the attention pattern
- Scaling to large models is still challenging

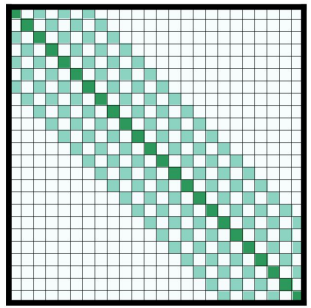
Check our tutorial
Beyond Paragraphs: NLP for Long Sequences
for more details
https://github.com/allenai/naacl2021-longdoc-tutorial

Prior work

Model	attention matrix	char-lm	other tasks	pretrain
Transformer-XL (2019)	ltr	yes	no	no
Adaptive Span (2019)	ltr	yes	no	no
Compressive (2020)	ltr	yes	no	no
Reformer (2020)	sparse	yes	no	no
Sparse (2019)	sparse	yes	no	no
BP-Transformer (2019)	sparse	yes	MT	no
Blockwise (2019)	sparse	no	QA	yes
Our Longformer	sparse	yes	multiple	yes

Implementation

- Required banded-multiplication is not supported in existing DL libraries
- Implementation 2: custom cuda kernel
  - Implemented in TVM
    - Compiles python code into cuda c++
    - Easier to use than writing cuda from scratch
  - Supports dilation
  - Only computes the non-zero elements (memory efficient)
  - A bit harder to deploy and use



Performance

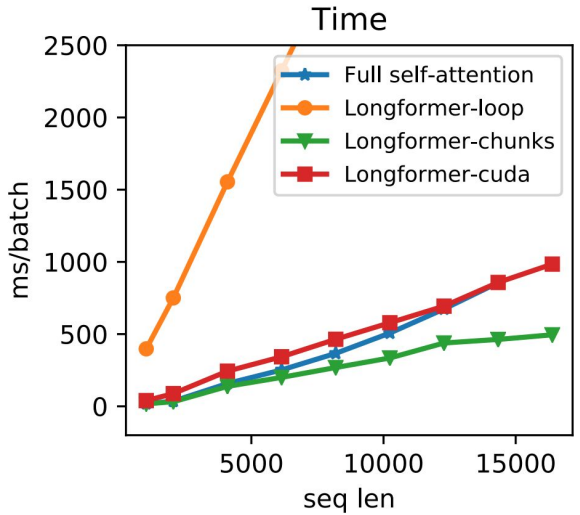
`loop` is a naive implementation using loops

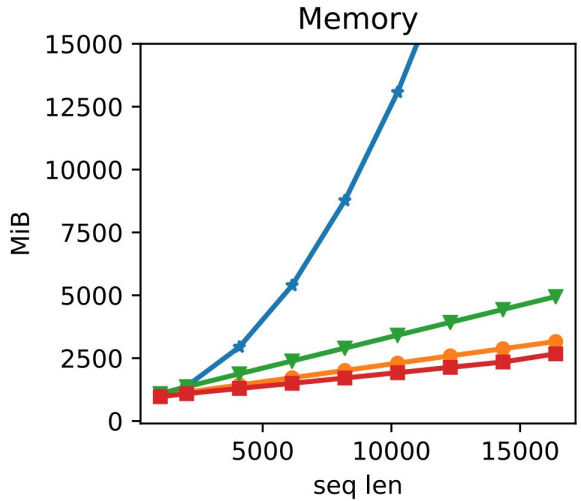
`loop` and `cuda` are the most memory efficient

`chunks` uses 2x more memory than `cuda` (but still linea

`chunks` faster than `cuda` because it uses nvidia MM

Use `chunks` for pretrain/finetune, use `cuda` for char-lm





```python
import torch
import tvm
from tvm.contrib import dlpack

d1 = tvm.var('d1')  # define dimensions as variables
d2 = tvm.var('d2')  # define dimensions as variables
d3 = tvm.var('d3')  # define dimensions as variables
A = tvm.placeholder((d1, d2), name='A', dtype='float32')  # first tensor
B = tvm.placeholder((d2, d3), name='B', dtype='float32')  # second tensor
k = tvm.reduce_axis((0, d2), name='k')  # dimension to sum over
output_shape = (d1, d3)
algorithm = lambda i, j: tvm.sum(A[i, k] * B[k, j], axis=k)  # explain computation
R = tvm.compute(output_shape, algorithm, name='R')

s = tvm.create_schedule(R.op)
s[R].bind(s[R].op.axis[1], tvm.thread_axis("blockIdx.x"))  # map computation to gpu resources
tvm_fn = tvm.build(s, [A, B, R], target='cuda', target_host='llvm')  # generate and compile C++
tvm_fn.export_library('libmm.so')  # save to disk

tvm_fn = tvm.module.load('libmm.so')  # load from disk
my_mm_pytorch_fn = dlpack.to_pytorch_func(tvm_fn)  # package it as a PyTorch function

X = torch.randn(128, 256, device='cuda')  # allocate pytorch input tensors
Y = torch.randn(256, 32, device='cuda')  # allocate pytorch input tensors
Z = X.new_empty(128, 32, device='cuda')  # allocate pytorch output tensor
my_mm_pytorch_fn(X, Y, Z)  # call the tvm kernel

torch.allclose(X.matmul(Y), Z, atol=1e-04)  # compare tvm output with pytorch
```

BBC:

"…with shared-record 20 Grand Slam men's singles titles, Djokovic is at the peak of his career…"

ThePrint:

"…Novak Djokovic tied Roger Federer and Rafael Nadal's men's record of 20 Grand Slam titles.…"

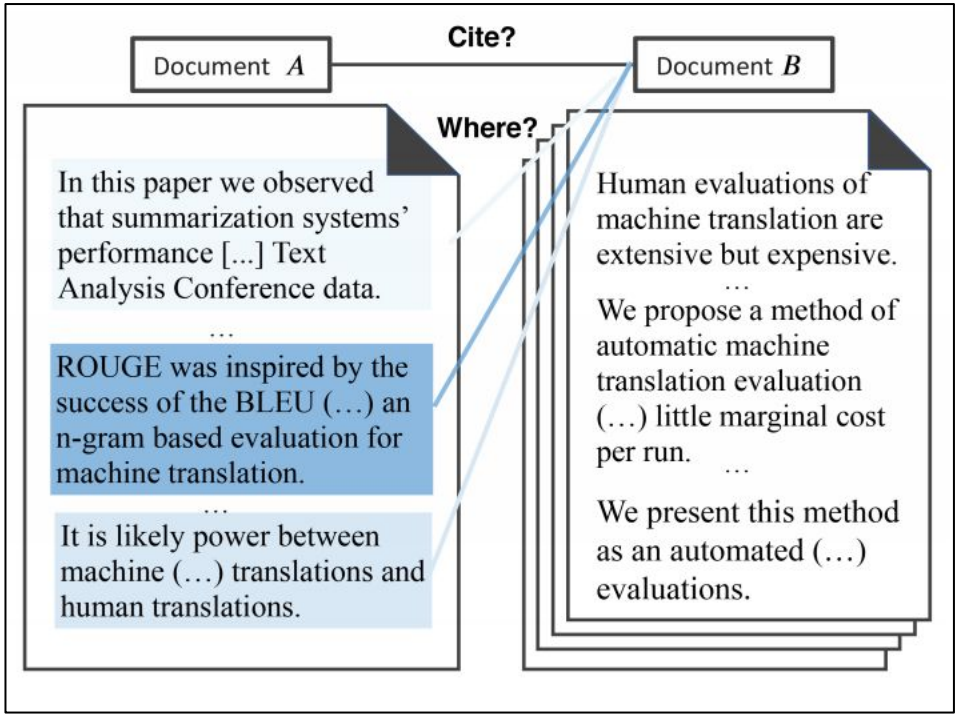
Cross-Document Coreference Resolution

Doc 1: "*Harry Shearer* is *suing* *Vivendi's* **Universal Music** for $125 million for allegedly fraudulent ..."

Doc 2:"...*Harry Shearer* *alleges* **parent company of Universal Music and StudioCanal** withheld millions..."

Doc 3: "*Shearer* was then *joined in the lawsuit* with *StudioCanal* and its French parent *Vivendi* by his co-stars"

Paper Citation Recommendation



[Zhou et al. 2020]

Cross-Document Tasks

- Cross-Document Coreference Resolution
- Multi-Hop Question Answering
- Citation Recommendation
- etc...

Outline

- Cross-Document Tasks
- **Our approach: Cross-Document Masking** 
- Analysis & Extrinsic Evaluation
- Conclusion

Results

- CD event coreference (+1.2 CoNLL F1)
- CD entity coreference (+7.6 CoNLL F1!)
- Paper citation recommendation (+4.4 F1!)
- CDLM has the best CD perplexity among the ablations

Outline

- Cross-Document Tasks
- Our approach: Cross-Document Masking
- Analysis & Extrinsic Evaluation
- **Conclusion**

Conclusions

- CDLM is a **general LM** for CD applications and downstream tasks
- CDLM models **CD relations, mapping and alignment**
- Uses fewer parameters (not a task specific model)

- Future work
  - Encoder-decoder architecture
  - Adding document retrieval component

https://huggingface.co/biu-nlp/cdlm

How to select SALIENT sentences?

Previous work for single document input (PEGASUS):

- Random
- Lead-K
- Principle Best
  - **Intuition**: select the **most central** sentences in the document
  - The score is defined as the ROUGE score between **each sentence** and **rest of the document**

$$\text{Score}(s_i) = \text{Rouge}(s_i, D/\{s_i\})$$

How to select SALIENT sentences?

- However, multi-document input tends to be more **redundant** than single document input.
- And such strategy would prefer **exact match between sentences**, resulting in selection of less representative information.

Example of the problem with vanilla SGS

Doc #1	Wildfires have burned across tens of thousands of acres of parched terrain in Colorado, spurring thousands of evacuations ..., residents have sought shelter in middle schools, and local officials fear tourists usually drawn to the region for the summer may not come.
Doc #2	... In Colorado's southwest, authorities have shuttered the San Juan National Forest in southwestern Colorado and residents of more than 2,000 homes were forced to evacuate. No homes had been destroyed ... "Under current conditions, one abandoned campfire or spark could cause a catastrophic wildfire, ..., with human life and property," said San Juan National Forest Fire Staff Officer Richard Bustamante...
Doc #3	The Buffalo Fire west of Denver is ... Several wildfires in Colorado have prompted thousands of home evacuations ... Nearly 1,400 homes have been evacuated in Summit County, Colorado, ... "Under current conditions, one abandoned campfire or spark could cause a catastrophic wildfire, ... , with human life and property," said Richard Bustamante, SJNF forest fire staff officer ...

New Masking Strategy: Entity Pyramid

Goal:

Select sentences that best represent the entire cluster of input documents

Inspired by Pyramid Evaluation

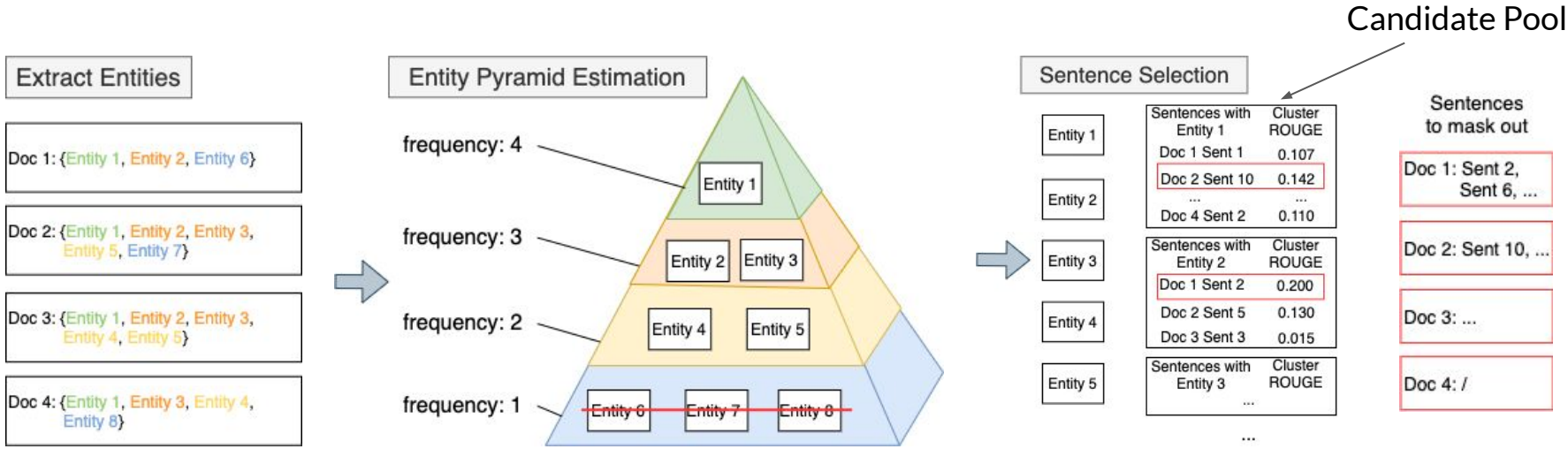
Pyramid Evaluation
with multiple refs

- The importance of information is quantified by the frequency of the gold references that include it.
- The more gold references include a fact, the more important it is.
- The facts are identified by human-labeled Summary Content Units (SCUs)

Entity Pyramid
with multiple docs

- The importance of information is quantified by the frequency of the documents that include it.
- The more documents include a fact, the more important it is.
- The facts are identified by Entities

New Masking Strategy: Entity Pyramid

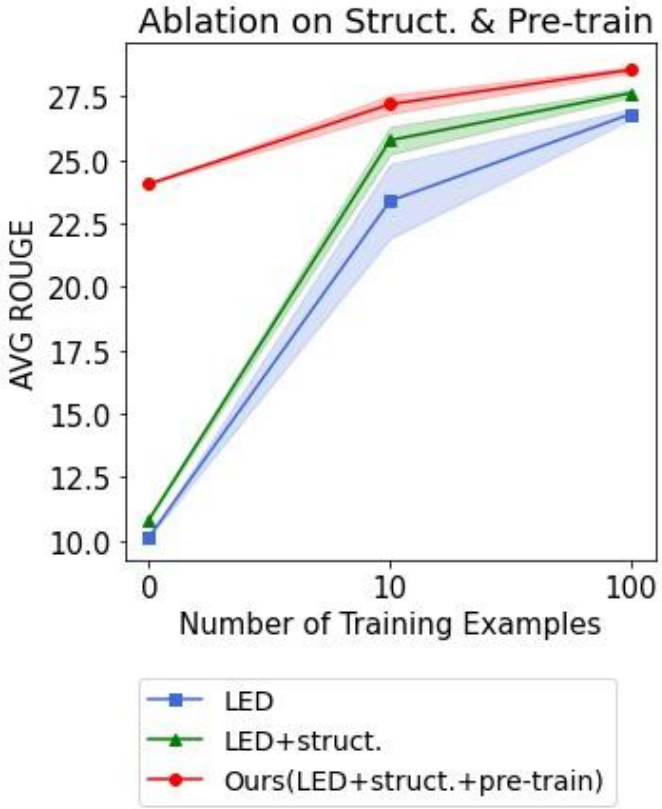


Cluster ROUGE: $$Score(s_i) = \sum_{\{doc_j \in C, s_i \notin doc_j\}} \text{ROUGE}(s_i, doc_j)$$

Example

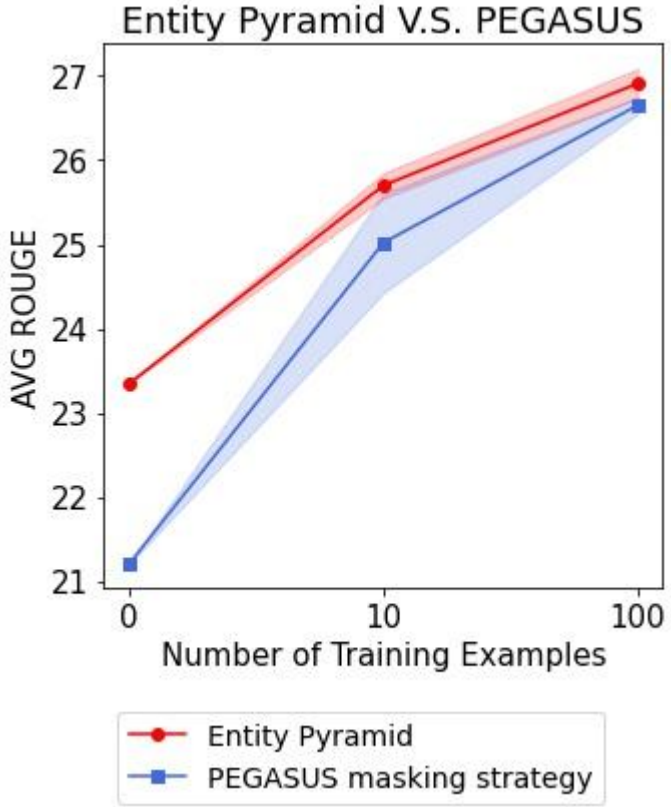
Doc #1	Wildfires have burned across tens of thousands of acres of parched terrain in Colorado, spurring thousands of evacuations (0.107) ..., residents have sought shelter in middle schools, and local officials fear tourists usually drawn to the region for the summer may not come.
Doc #2	... ***In Colorado's southwest, authorities have shuttered the San Juan National Forest in southwestern Colorado and residents of more than 2,000 homes were forced to evacuate. (0.187)*** No homes had been destroyed ... "Under current conditions, one abandoned campfire or spark could cause a catastrophic wildfire, ..., with human life and property," said San Juan National Forest Fire Staff Officer Richard Bustamante...
Doc #3	The Buffalo Fire west of Denver is ... Several wildfires in Colorado have prompted thousands of home evacuations (0.172)... Nearly 1,400 homes have been evacuated in Summit County, Colorado, (0.179)..... "Under current conditions, one abandoned campfire or spark could cause a catastrophic wildfire, ... , with human life and property," said Richard Bustamante, SJNF forest fire staff officer ...
Entity List	Colorado(3), Wildfires(3), 416(2), Tuesday(2), San Juan National Forest(2),....

Impact of Proposed Pre-training



- Pre-training helps improving the model, especially for the zero-shot setting.

Impact of Pre-training Strategies

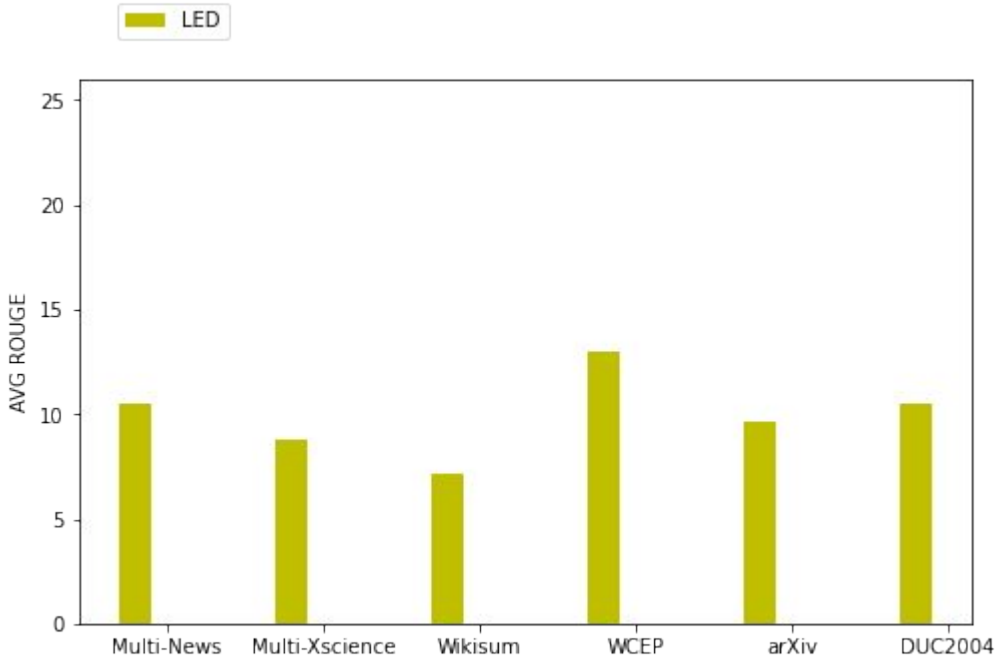


- Same architecture (LED-Base)
- Same input structure
- Same pre-training objective
- Same pre-training dataset
- Zero/Few-shot setting
- **The Entity Pyramid strategy works better than the Principle strategy used in PEGASUS.**

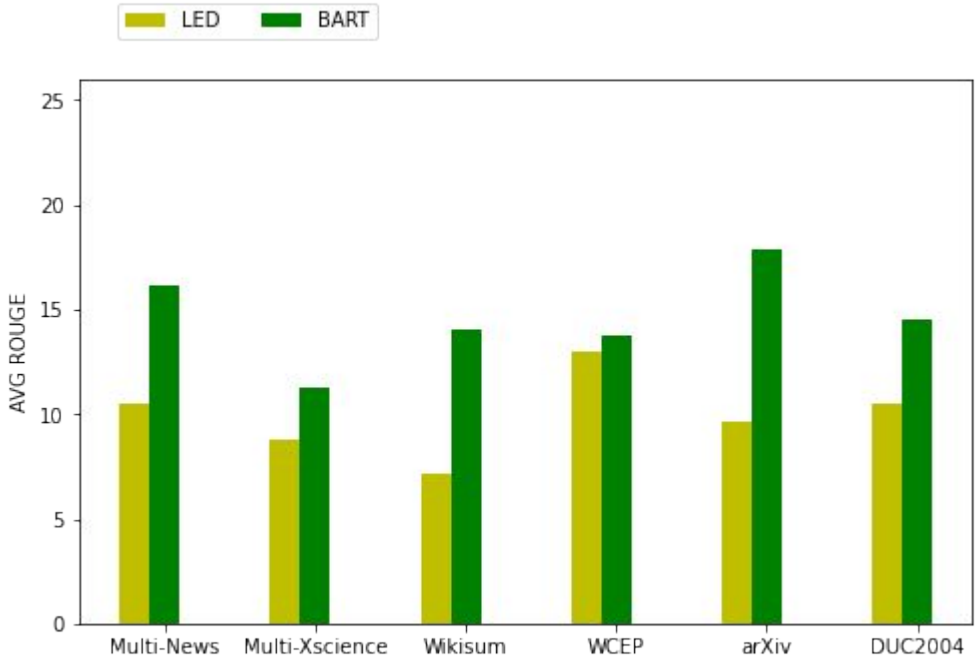
Experiments - Automatic Evaluation

- Evaluation Datasets:
  - Multi-Doc.: Multi-News, Multi-XScience, WCEP, Wikisum, DUC2004
  - Single Doc. arXiv
- Settings:
  - Zero-shot (with length limit)
  - Few-shot: 10/100 training examples, 5 runs for each model
  - Fully supervised
- Compared Models:
  - BART
  - PEGASUS

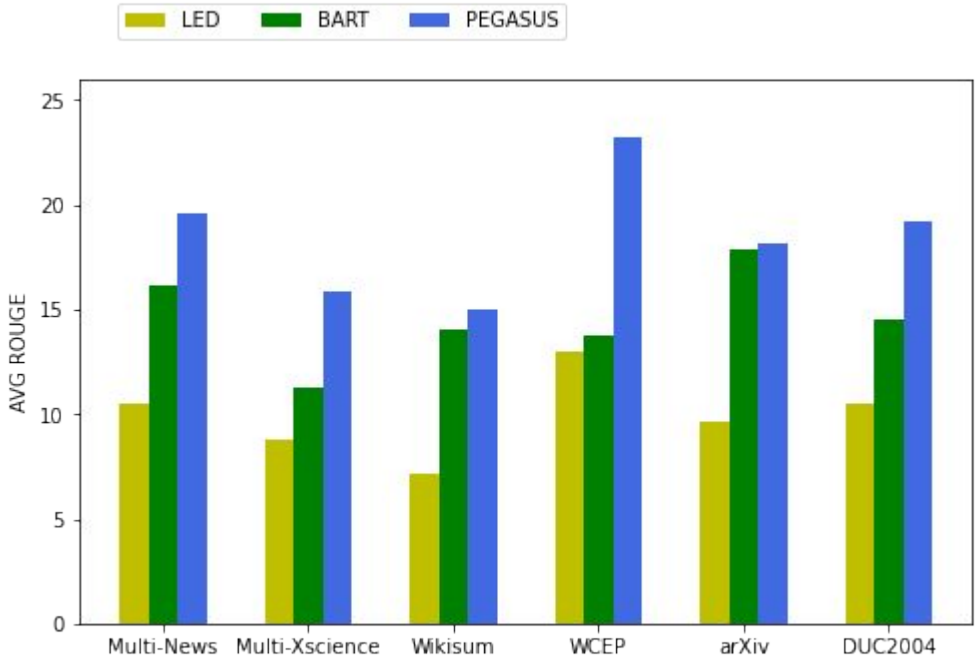
Results on Zero-shot



Results on Zero-shot

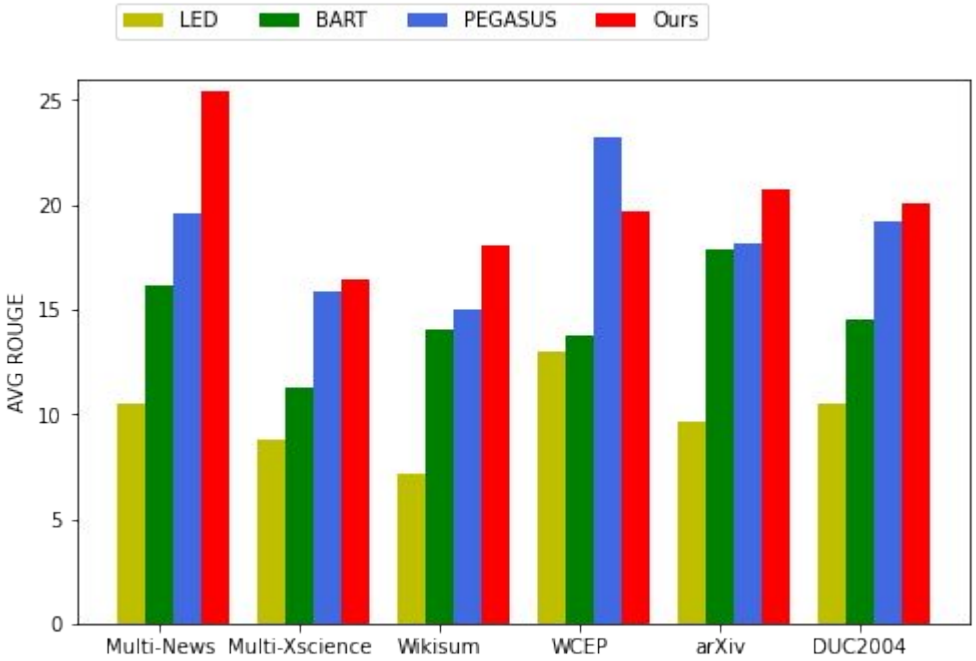


Results on Zero-shot



- PEGASUS is also pre-trained for summarization downstream task, thus it performs better than the other two models

Results on Zero-shot



- PEGASUS is also pre-trained for summarization downstream task, thus it performs better than the other two models
- Our model outperforms all the other pre-trained models on most of the datasets (up to 5 ROUGE points)

Experiments - Human Evaluation

- Datasets:
  - DUC 2007
  - TAC 2008
- Metrics:
  - Pyramid Evaluation
  - Fluency (following DUC guidelines*)

* https://www-nlpir.nist.gov/projects/duc/duc2007/quality-questions.txt

Human Evaluation - Pyramid & Fluency

	Model	Pyramid Evaluation				Fluency		
		S_r	R	P	F	Gram.	Ref.	Str.&Coh.
DUC 2007	PEGASUS	6.0	2.5	2.4	2.4	4.45	4.35	1.95
	LED	9.6	3.9	4.0	3.8	4.35	4.50	3.20
	PRIMERA	**12.5**	**5.1**	**5.0**	**5.0**	**4.70**	**4.65**	**3.70**
TAC 2008	PEGASUS	**8.7**	**9.1**	9.4	9.1	**4.40**	4.20	3.20
	LED	6.9	7.1	10.8	8.4	3.10	3.80	2.55
	PRIMERA	8.5	8.9	**10.0**	**9.3**	**4.40**	**4.45**	**4.10**

- PRIMERA also shows a better performance on human evaluation, regarding both pyramid evaluation and fluency evaluation.

Future Work

- Controllable generator to better control the length of generated summaries for zero-shot setting
- Evaluate PRIMERA and its Pyramid Entity strategy on other tasks with multiple documents as input, e.g. Multi-hop QA