FALL 2020 CS 395T

TEXAS
The University of Texas at Austin

Open Domain QA

CS 395T: Topics in Natural Language Processing

November 5th, 2020

KAJ BOSTROM AND SHIVAM GARG
Department of Computer Science, The University of Texas at Austin

Open domain QA

In datasets like SQuAD, context passages containing the answer are provided. Models can expect to find the answer verbatim in this context.

In the open domain paradigm, however, tasks often don’t specify a particular context passage; if a model needs to refer to context it must be retrieved.

Language models for open domain QA

- The language modeling objective function doesn’t just encourage models to predict the structure of language well.
- If the model’s capacity is large enough, it will also learn to predict the actual content of the corpus, including factual knowledge.
- For example, BERT chooses “London” for “[MASK] is the capital of the UK”.

Language models for open domain QA

- This ‘implicit knowledge’ makes large pretrained language models an attractive starting point for open domain QA systems, since the language model may be able to answer many questions without context.
- However, there are questions that need to be answered:
  - How much knowledge is acquired this way?
  - Is there a better/more interpretable way to encode this knowledge?

Paper 1: REALM: Retrieval-Augmented Language Model Pre-Training

ICML 2020

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, Ming-Wei Chang
Google

Motivation

- Language models implicitly encode factual knowledge in their parameters
- If this knowledge was explicit, models would be more interpretable
- Can we learn to retrieve this kind of knowledge as text, and will that help model performance on open domain QA?

Implicit knowledge in language models

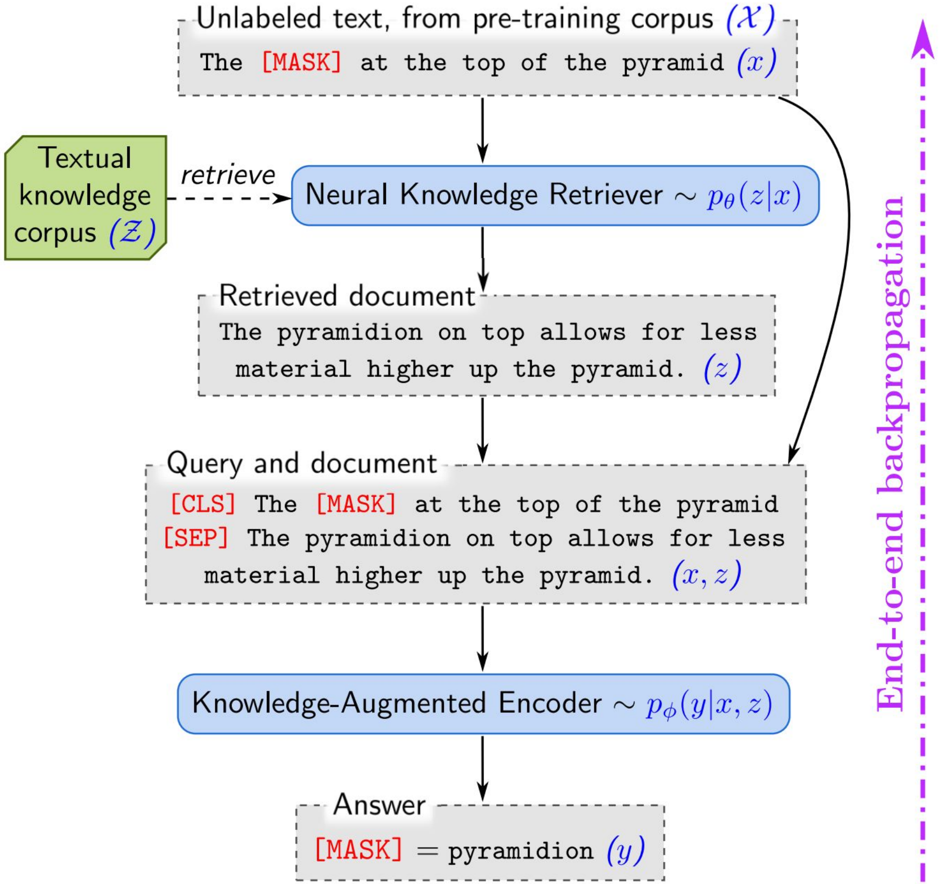
- Since they are optimized to predict language, LMs learn to predict facts that show up in their pretraining data
- However, they don't memorize *perfectly*, and there's no clear way to figure out how they're encoding or accessing this information
  - We can only "audit" this knowledge through probing tasks

REALM architecture

Neural Knowledge Retriever encodes knowledge corpus Z, retrieves extra context z based on input text x

Key issues:

- How to retrieve useful z?
- Z is large, how to update encodings as model params change?


Unlabeled text, from pre-training corpus $(\mathcal{X})$
The [MASK] at the top of the pyramid (x)
Textual knowledge corpus $(\mathcal{Z})$
retrieve
Neural Knowledge Retriever $\sim p_\theta(z|x)$
Retrieved document
The pyramidion on top allows for less material higher up the pyramid. (z)
Query and document
[CLS] The [MASK] at the top of the pyramid [SEP] The pyramidion on top allows for less material higher up the pyramid. (x, z)
Knowledge-Augmented Encoder $\sim p_\phi(y|x, z)$
Answer
[MASK] = pyramidion (y)
End-to-end backpropagation

Issue 1: How to retrieve useful knowledge

Prior work: retrieve based on standard heuristics (TF-IDF, fixed embedding similarity)

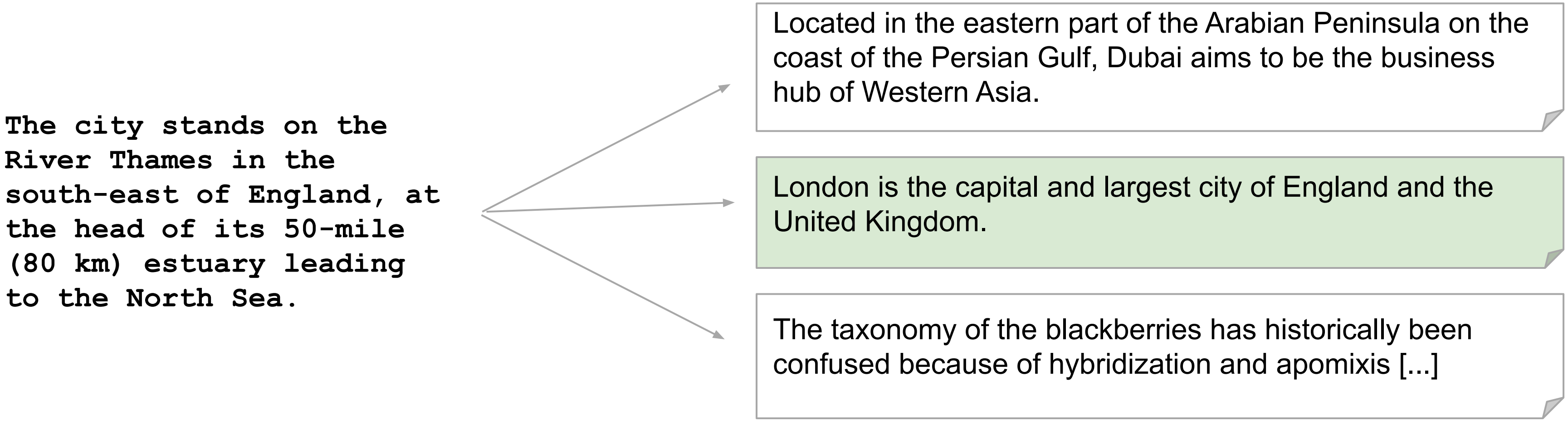
- Pros: Only need to precompute index once
- Cons: Behavior doesn’t adapt to model, may be too shallow

REALM: encode knowledge corpus using LM, retrieve with maximum inner-product search

- Pros: Retrieval is latent, optimized to maximize LM performance, takes advantage of rich LM state
- Cons: Difficult to keep corpus encodings up to date with model parameters

Training latent retrieval

Prior work by two of the main authors, **ORQA**, has the same latent retrieval setup but uses an ‘inverse cloze’ objective for training. The model is rewarded for retrieving a sentence’s original context.



The city stands on the River Thames in the south-east of England, at the head of its 50-mile (80 km) estuary leading to the North Sea.

Located in the eastern part of the Arabian Peninsula on the coast of the Persian Gulf, Dubai aims to be the business hub of Western Asia.

London is the capital and largest city of England and the United Kingdom.

The taxonomy of the blackberries has historically been confused because of hybridization and apomixis [...]

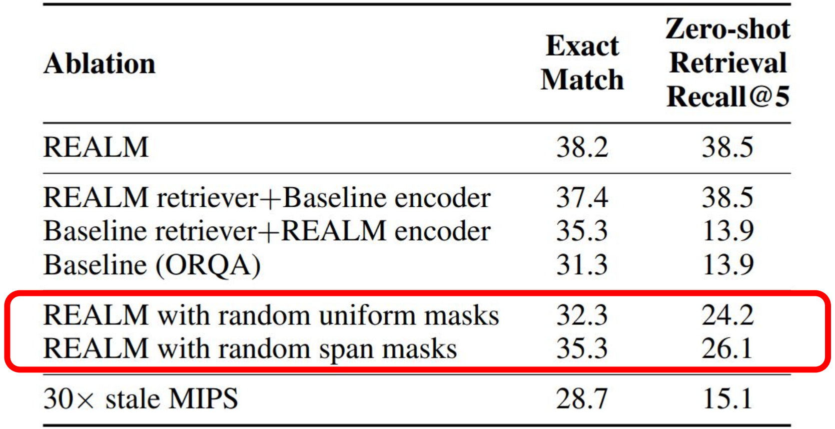
Training latent retrieval

In this work, retrieval parameters are trained via backpropagation from the LM loss.

The key issue is getting the LM to learn to use the retrieved text: **1**. normal masked LM is too easy, and **2**. the initial retrieval results aren't useful. Both of these issues cause the model to ignore retrieval.

- The authors use 'Salient Span Masking' to solve issue **1**; **named entities and dates** are masked instead of random tokens.
- Warm-starting retrieval using the ORQA inverse-cloze task solves **2**.

Table 2. Ablation experiments on NQ's development set.



Ablation	**Exact Match**	**Zero-shot Retrieval Recall@5**
REALM	38.2	38.5
REALM retriever+Baseline encoder	37.4	38.5
Baseline retriever+REALM encoder	35.3	13.9
Baseline (ORQA)	31.3	13.9
REALM with random uniform masks	32.3	24.2
REALM with random span masks	35.3	26.1
30× stale MIPS	28.7	15.1

Maximum inner-product search (MIPS)

1. Take the [CLS] embedding from a document *z* in the knowledge corpus and the [CLS] embedding from the input document
2. Linearly transform them into d dimensions and take the dot product
3. Choose the passages z with the highest scores $f(x, z)$

$$\begin{aligned}\mathtt{Embed}_{\mathtt{input}}(x) &= \mathbf{W}_{\mathtt{input}}\mathtt{BERT}_{\mathtt{CLS}}(\mathtt{join}_{\mathtt{BERT}}(x)) \\ \mathtt{Embed}_{\mathtt{doc}}(z) &= \mathbf{W}_{\mathtt{doc}}\mathtt{BERT}_{\mathtt{CLS}}(\mathtt{join}_{\mathtt{BERT}}(z_{\text{title}}, z_{\text{body}}))\end{aligned}$$

$$\begin{aligned}p(z \mid x) &= \frac{\exp f(x, z)}{\sum_{z'} \exp f(x, z')}, \\ f(x, z) &= \mathtt{Embed}_{\mathtt{input}}(x)^{\top}\mathtt{Embed}_{\mathtt{doc}}(z),\end{aligned}$$

Maximum inner-product search (MIPS)

- MIPS can be accelerated by computing an index over cached vectors Z for the knowledge corpus documents.
- This is critical to make the model feasible to train and run, however the cache raises another issue: as we train the model, the LM parameters change, and the cached encodings Z depend on those parameters.
- How do we keep Z up to date?

Issue 2: Keeping Z updated

Too expensive to recalculate knowledge corpus embeddings every parameter update

Instead, update asynchronously (index fully refreshed every ~500 steps)

Ablations show that slower updates don’t work as well

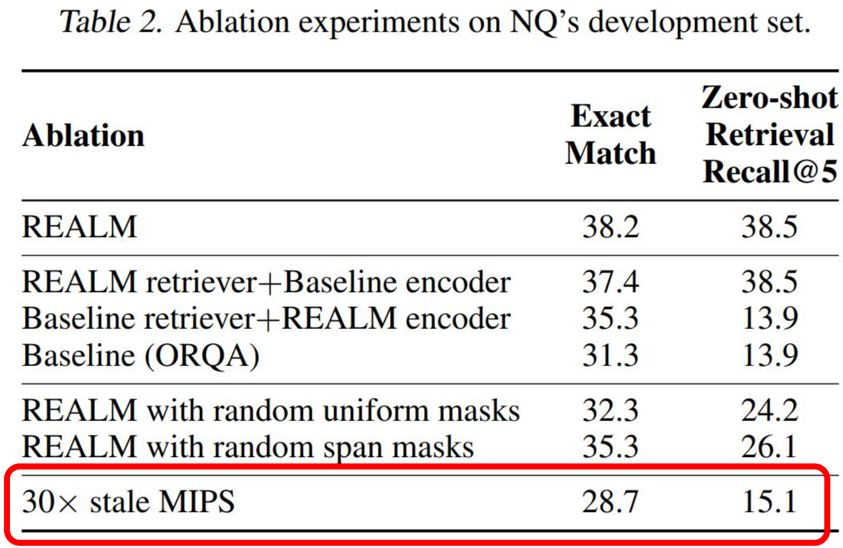
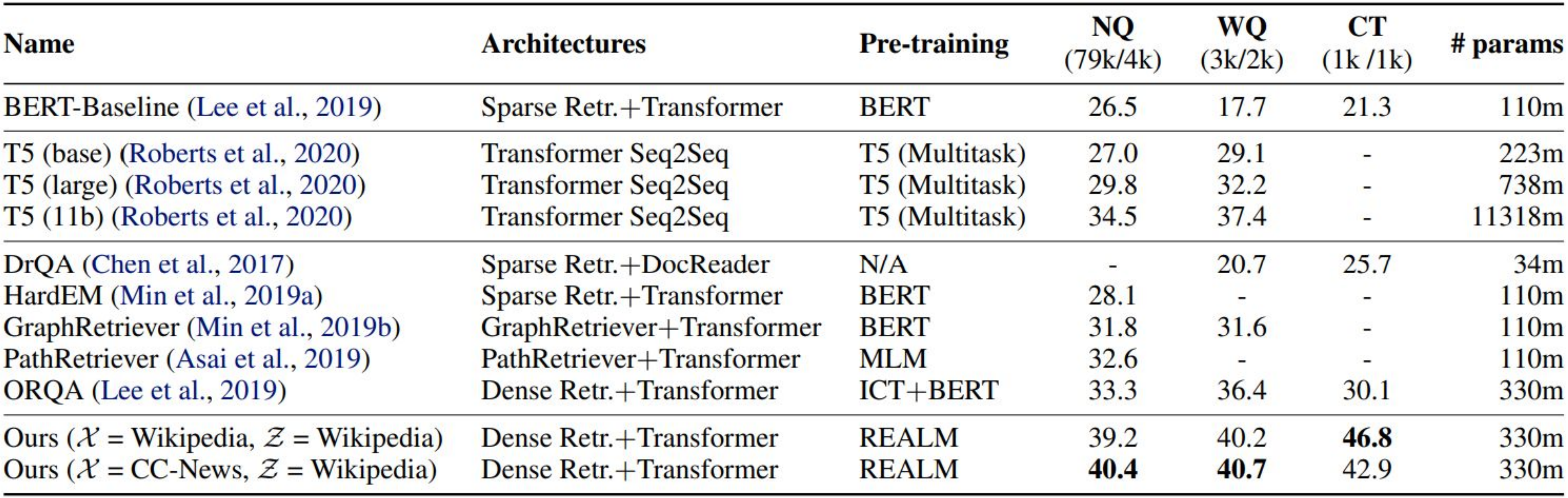


Table 2. Ablation experiments on NQ’s development set.

Ablation	**Exact Match**	**Zero-shot Retrieval Recall@5**
REALM	38.2	38.5
REALM retriever+Baseline encoder	37.4	38.5
Baseline retriever+REALM encoder	35.3	13.9
Baseline (ORQA)	31.3	13.9
REALM with random uniform masks	32.3	24.2
REALM with random span masks	35.3	26.1
30× stale MIPS	28.7	15.1

Experiments

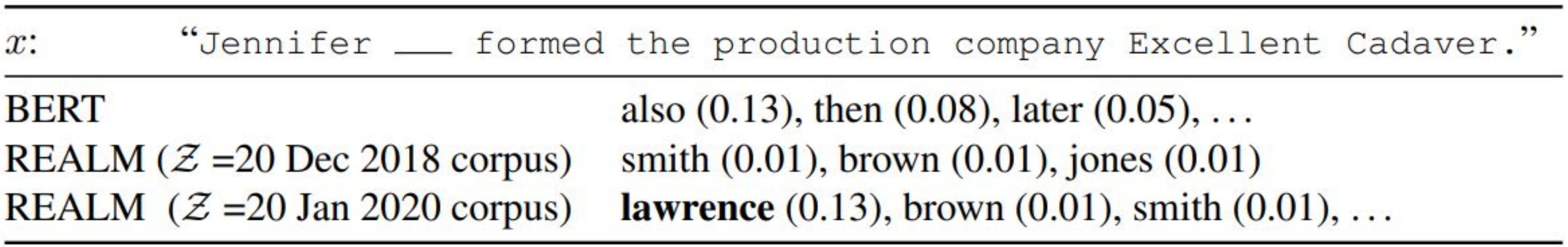


Name	**Architectures**	**Pre-training**	**NQ** (79k/4k)	**WQ** (3k/2k)	**CT** (1k /1k)	**# params**
BERT-Baseline (Lee et al., 2019)	Sparse Retr.+Transformer	BERT	26.5	17.7	21.3	110m
T5 (base) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	27.0	29.1	-	223m
T5 (large) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	29.8	32.2	-	738m
T5 (11b) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	34.5	37.4	-	11318m
DrQA (Chen et al., 2017)	Sparse Retr.+DocReader	N/A	-	20.7	25.7	34m
HardEM (Min et al., 2019a)	Sparse Retr.+Transformer	BERT	28.1	-	-	110m
GraphRetriever (Min et al., 2019b)	GraphRetriever+Transformer	BERT	31.8	31.6	-	110m
PathRetriever (Asai et al., 2019)	PathRetriever+Transformer	MLM	32.6	-	-	110m
ORQA (Lee et al., 2019)	Dense Retr.+Transformer	ICT+BERT	33.3	36.4	30.1	330m
Ours ($\mathcal{X}$ = Wikipedia, $\mathcal{Z}$ = Wikipedia)	Dense Retr.+Transformer	REALM	39.2	40.2	**46.8**	330m
Ours ($\mathcal{X}$ = CC-News, $\mathcal{Z}$ = Wikipedia)	Dense Retr.+Transformer	REALM	**40.4**	**40.7**	42.9	330m

NQ = NaturalQuestions, WQ = WebQuestions, CT = CuratedTREC

Analysis: Modular knowledge

In some cases, the model is able to use updated information in its knowledge corpus to adapt to new information without retraining (the article in this example was not present in the 2018 corpus):

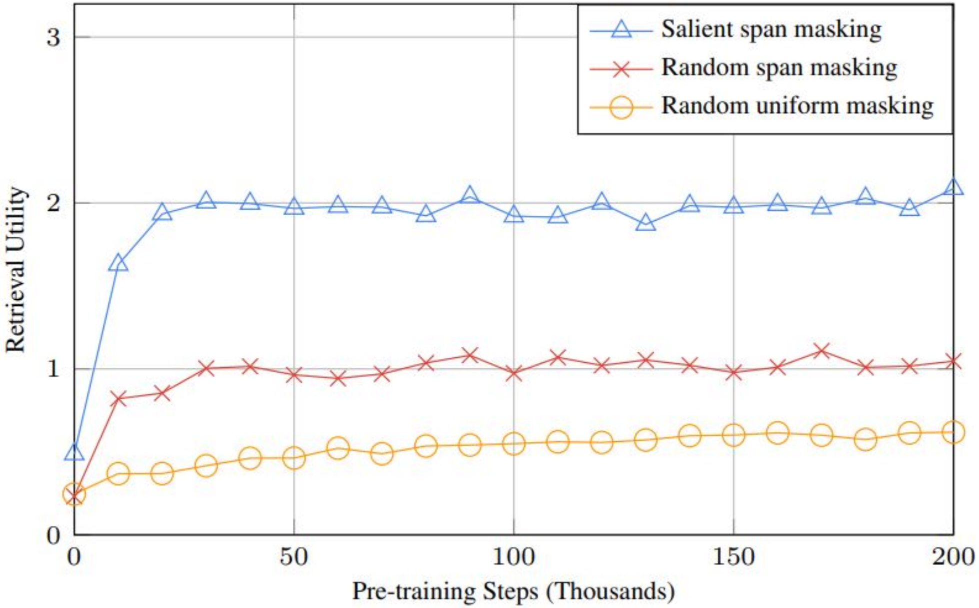


x:	“Jennifer ___ formed the production company Excellent Cadaver.”
BERT	also (0.13), then (0.08), later (0.05), …
REALM ($\mathcal{Z}$ =20 Dec 2018 corpus)	smith (0.01), brown (0.01), jones (0.01)
REALM ($\mathcal{Z}$ =20 Jan 2020 corpus)	**lawrence** (0.13), brown (0.01), smith (0.01), …

However, for information with strong distributional support (such as Margaret Thatcher being the prime minister of the UK), the model still predicts based on its implicit knowledge, regardless of the retrieved knowledge.

It’s unclear how to fix this in practice without sabotaging the language modeling objective.

Analysis: Usefulness of retrieval



Retrieval Utility measures the improvement in masked prediction log-likelihood resulting from incorporating retrieval results.

Using salient span masking instead of random span masking results in a much higher average retrieval utility, indicating that it is driving the model to make less trivial predictions.

The authors note that salient span masking has not been helpful in previous non-retrieval language model pretraining setups, but it is “crucial” for REALM, likely since it is important for providing informative gradients to the retrieval parameters.

Computational cost

- Pretraining: 200k steps on 64(!) TPUs, 16 of which are used for MIPS indexing

- Despite the high cost of training, during inference the MIPS cache doesn’t need to be updated and the whole model fits in 12gb of GPU memory

Discussion

- Do you think this method is interpretable, given that the model doesn’t necessarily rely on retrieval results?
- What are some other domains you think could benefit from using latent retrieval instead of traditional IR?
- Starting from inverse cloze + masking only named entities and dates may steer the retrieval into a ‘local maximum’ behavior. Can you think of other ways to encourage the model to use retrieval results early in training?

YOU
BELONG
HERE

CNS Diversity and Inclusion • cns.utexas.edu/diversity

Questions?

Paper 2: How Much Knowledge Can You Pack Into the Parameters of a Language Model?

Adam Roberts, Collin Raffel, Noam Shazeer
Google

Focus of the paper

- Measure the implicit knowledge encoded in a pretrained language model.
- Studies the feasibility of using language models as knowledge bases for open domain QA.
- Studies whether large models have more knowledge.

Motivation

- The knowledge is encoded by pre-training on unstructured and unlabeled text data.

- Informal natural language queries can be used to extract information from language models.

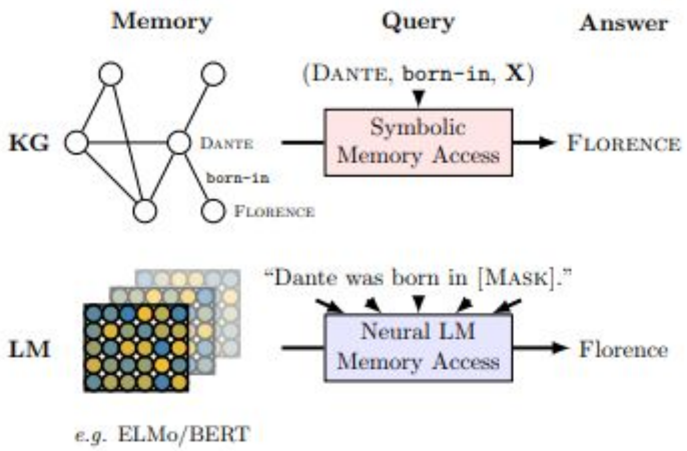


Figure 1: Querying knowledge bases (KB) and language models (LM) for factual knowledge.

Image Source: Petroni et. al.

Related Work

Language Models as Knowledge Bases? Petroni et. al.

- Language Model Analysis(LAMA) probe.
- A set of facts : sub-rel-obj triples or QA-pairs.
- Facts converted to cloze statement.
- MLE of target word used to quantify the information in LM.

Related Work

How Can We Know What Language Models Know? Jiang et. al.

- Uses LAMA probe, prompts are cloze statements.
- Automatic prompt creation:
  - Mining templates from dependency parses
  - Paraphrasing the prompts for diversity.

Related Work

oLMpics - On what Language Model Pre-training Captures. Talmor et. al.

- Three aspects studied:
  - Zero-shot knowledge
  - Finetune knowledge
  - Controlled zero-shot, vary certain aspects of input .
- Two probing setups:
  - Masked slot filling
  - QA- Multiple Choice

Related Work

Language Models are Unsupervised Multitask Learners. Radford et. al.

- GPT-2 models the problem as a zero-shot text generation task.

Learning crosscontext entity representations from text. Ling et. al.

Entities as Experts: Sparse Memory Access with Entity Supervision. Fevry et. al.

- Learn representations of an explicitly defined set of entities.

This paper

- Uses open-domain QA to evaluate the language models extent of information storage.
  - Context of a question not used to evaluate the LM.
- Uses T5 encoder-decoder transformer for closed-book open-domain QA.
- Measures real-world usable information in the model.
  - Not manually designed probes

Modelling

- T5 (Text-toText Transfer Transformer) Raffel et. al. used to model the task and to study the implicit information encoded by it.

T5

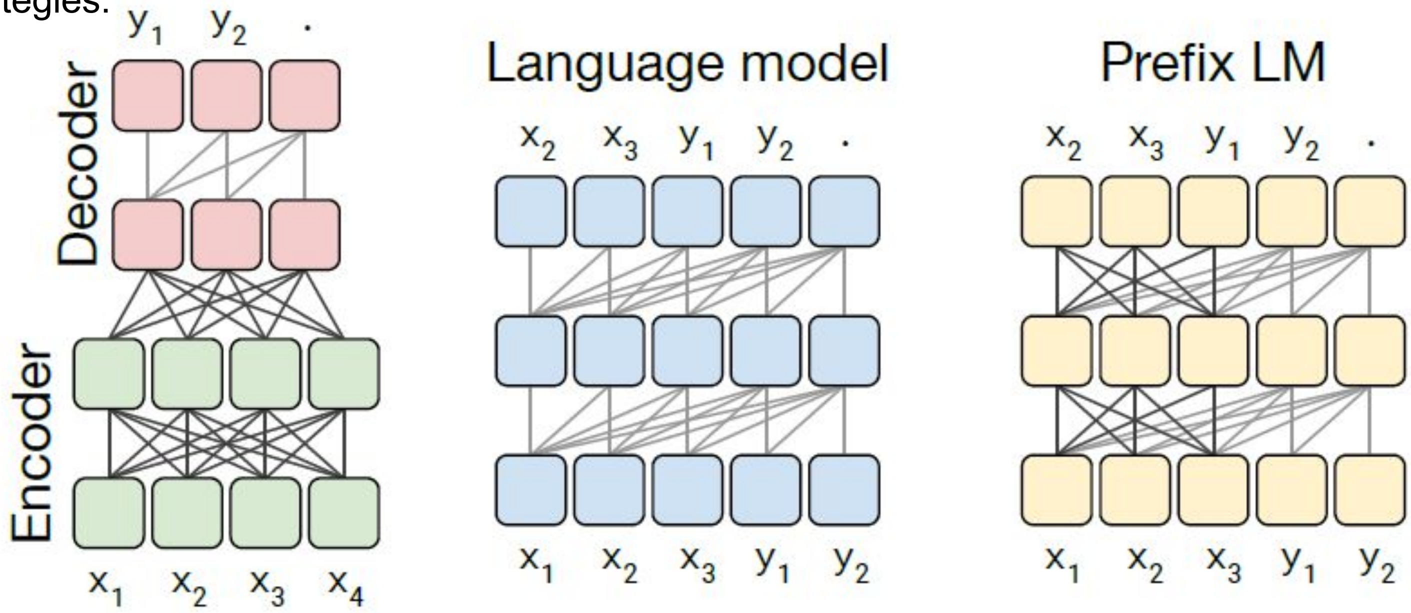
- A unified framework that combines all language problems in a text-to-text format.
- A single model is trained for all downstream tasks.
- The model training is agnostic for the task being trained for.

T5

Source: Google AI Blog

T5

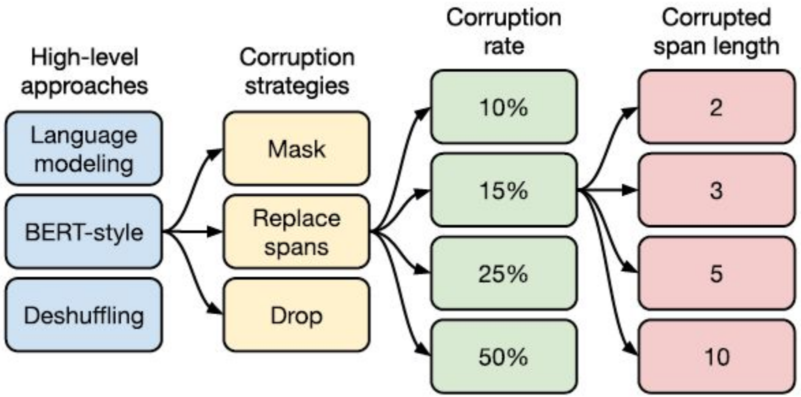
Training strategies:



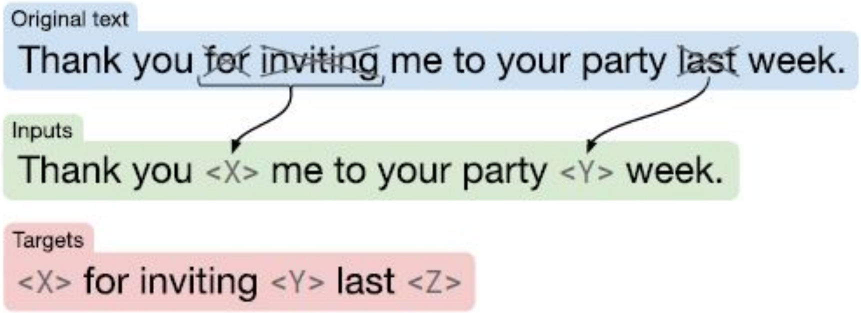
Source: T5 paper

T5

Training objectives:



Span Corruption:


Original text
Thank you ~~for inviting~~ me to your party ~~last~~ week.
Inputs
Thank you <X> me to your party <Y> week.
Targets
<X> for inviting <Y> last <Z>

Source: T5 paper

T5

Datasets

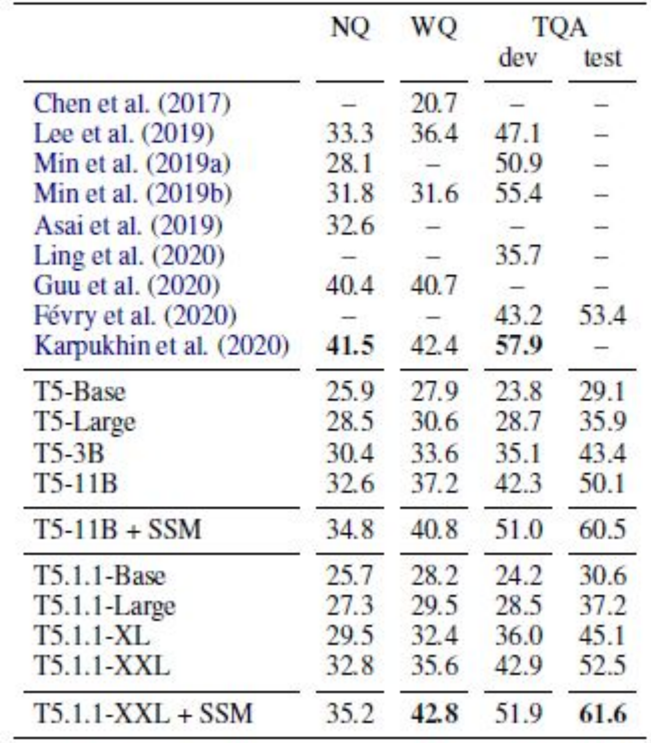
- Colossal Clean Crawled Corpus(C4) used for pretraining.
  - Derived from Common Crawl
  - 750GB of cleaned text
- Multi-task training for downstream tasks.

Implementation Details

- Only question prompt used in both training and testing.
- T5 finetuned on each dataset separately. Two different pretrained T5 used:
  - Pretrained on C4 + downstream tasks (T5)
  - Pretrained only C4 (T5 1.1)
- Greedy sampling used to sample the output.
- Salient Span Masking pretraining done before finetuning on QA datasets.

Results

- Larger models have better performance.
- SSM pretraining gives a huge boost in performance.
- T5 approach competitive on all datasets.
- T51.1XXL+SSM beats all existing works in WQ and TQA (including the ones using external knowledge sources).
- The approach is computationally equivalent to existing approaches, since they involve an expensive lookup in the external KB.

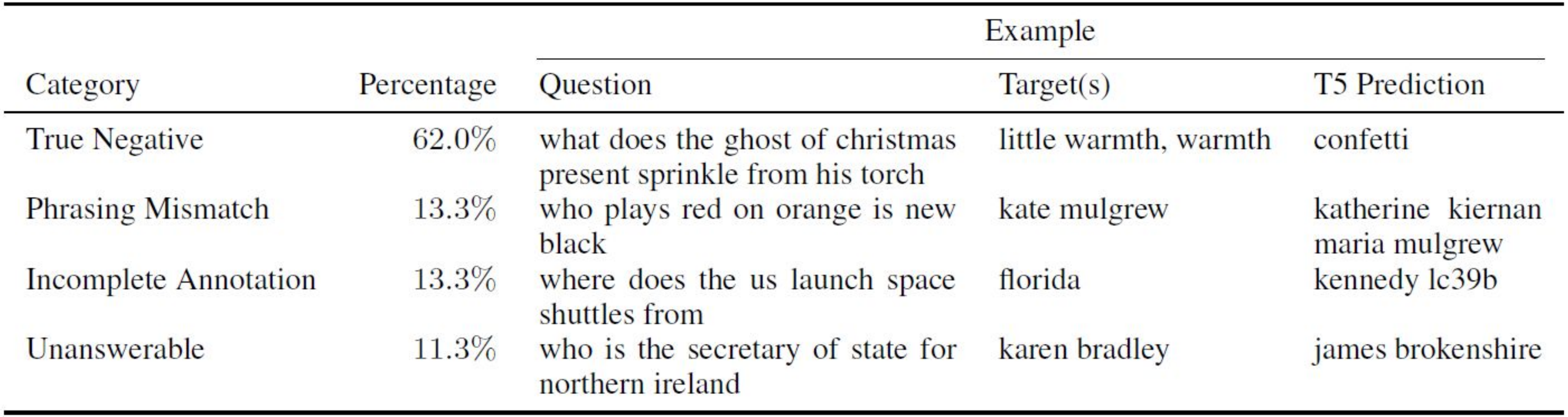


	NQ	WQ	TQA	
			dev	test
Chen et al. (2017)	–	20.7	–	–
Lee et al. (2019)	33.3	36.4	47.1	–
Min et al. (2019a)	28.1	–	50.9	–
Min et al. (2019b)	31.8	31.6	55.4	–
Asai et al. (2019)	32.6	–	–	–
Ling et al. (2020)	–	–	35.7	–
Guu et al. (2020)	40.4	40.7	–	–
Févry et al. (2020)	–	–	43.2	53.4
Karpukhin et al. (2020)	**41.5**	42.4	**57.9**	–
T5-Base	25.9	27.9	23.8	29.1
T5-Large	28.5	30.6	28.7	35.9
T5-3B	30.4	33.6	35.1	43.4
T5-11B	32.6	37.2	42.3	50.1
T5-11B + SSM	34.8	40.8	51.0	60.5
T5.1.1-Base	25.7	28.2	24.2	30.6
T5.1.1-Large	27.3	29.5	28.5	37.2
T5.1.1-XL	29.5	32.4	36.0	45.1
T5.1.1-XXL	32.8	35.6	42.9	52.5
T5.1.1-XXL + SSM	35.2	**42.8**	51.9	**61.6**

Image Source: Roberts et. al.

Results

- Eval metrics of all benchmarks uses “exact match” to compare predictions with ground truth.
- Human eval for 150 examples to study the errors.



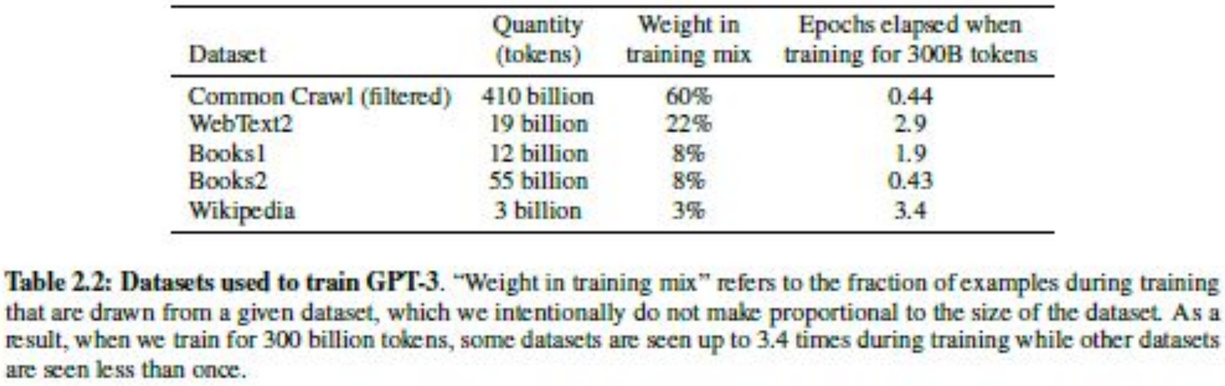
		Example		
Category	Percentage	Question	Target(s)	T5 Prediction
True Negative	62.0%	what does the ghost of christmas present sprinkle from his torch	little warmth, warmth	confetti
Phrasing Mismatch	13.3%	who plays red on orange is new black	kate mulgrew	katherine kiernan maria mulgrew
Incomplete Annotation	13.3%	where does the us launch space shuttles from	florida	kennedy lc39b
Unanswerable	11.3%	who is the secretary of state for northern ireland	karen bradley	james brokenshire

- Removing “Unanswerable” questions from val set boosts the accuracy to 57.8% .

Image Source: Roberts et. al.

Recent Work – GPT3

- Same model and architecture as GPT-2.



Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

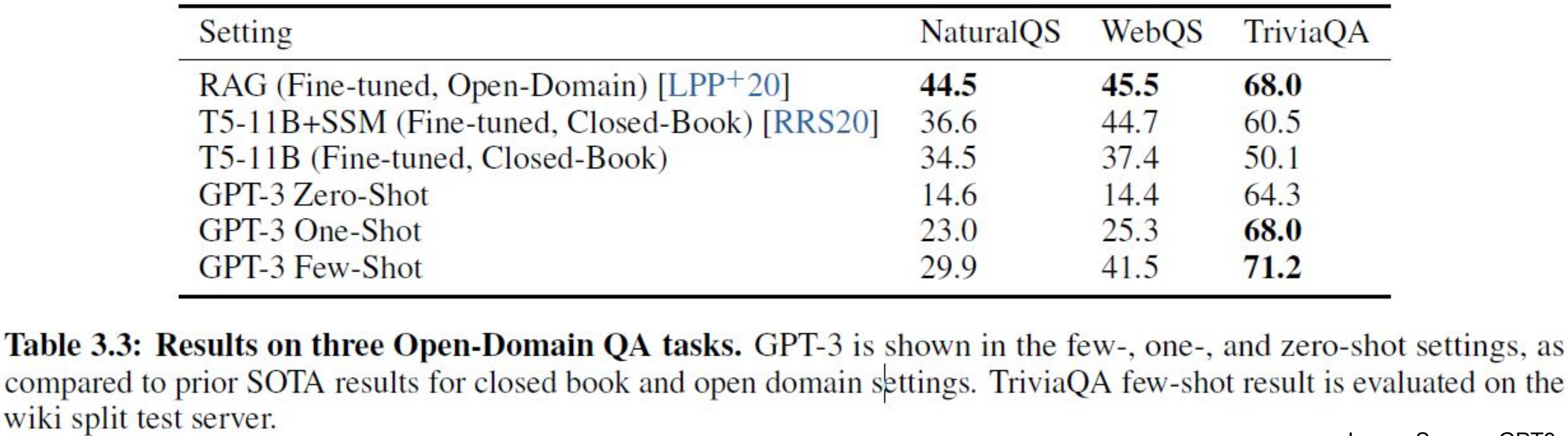
Table 2.2: Datasets used to train GPT-3. "Weight in training mix" refers to the fraction of examples during training that are drawn from a given dataset, which we intentionally do not make proportional to the size of the dataset. As a result, when we train for 300 billion tokens, some datasets are seen up to 3.4 times during training while other datasets are seen less than once.

Image Source: GPT3 paper

Recent Work – GPT3

GPT3 few-shot beats T5 on TriviaQA.

No finetuning involved, indicating higher knowledge encoding in GPT3.



Setting	NaturalQS	WebQS	TriviaQA
RAG (Fine-tuned, Open-Domain) [LPP+20]	**44.5**	**45.5**	**68.0**
T5-11B+SSM (Fine-tuned, Closed-Book) [RRS20]	36.6	44.7	60.5
T5-11B (Fine-tuned, Closed-Book)	34.5	37.4	50.1
GPT-3 Zero-Shot	14.6	14.4	64.3
GPT-3 One-Shot	23.0	25.3	**68.0**
GPT-3 Few-Shot	29.9	41.5	**71.2**

Table 3.3: Results on three Open-Domain QA tasks. GPT-3 is shown in the few-, one-, and zero-shot settings, as compared to prior SOTA results for closed book and open domain settings. TriviaQA few-shot result is evaluated on the wiki split test server.

Image Source: GPT3 paper

Recent Work – GPT3

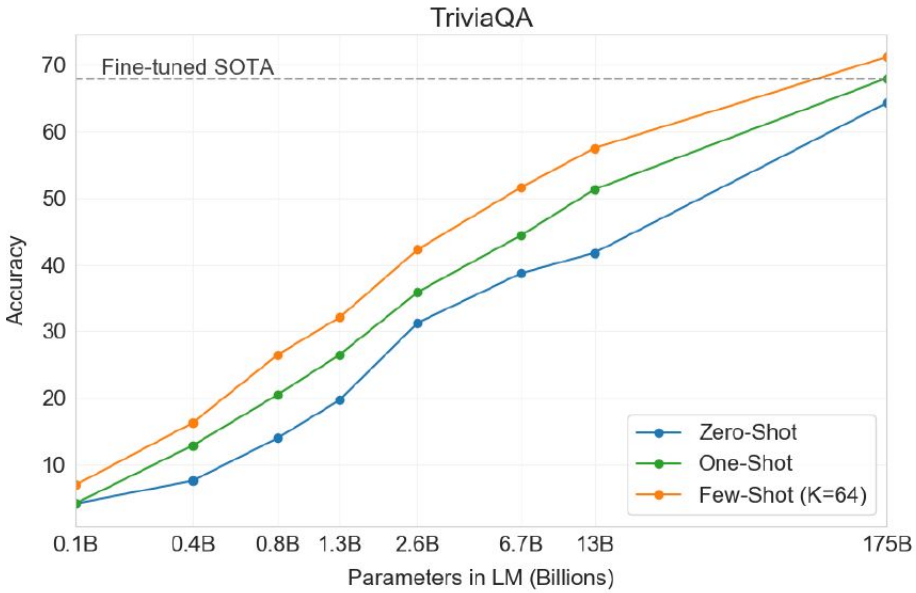


Figure 3.3: On TriviaQA GPT3's performance grows smoothly with model size, suggesting that language models continue to absorb knowledge as their capacity increases. One-shot and few-shot performance make significant gains over zero-shot behavior, matching and exceeding the performance of the SOTA fine-tuned open-domain model, RAG [LPP+20]

Image Source: GPT3 paper

Negative Results

- Continued Pretraining T5 on Wikipedia
  - No effect on performance.
  - C4 contains many Wikipedia articles.
- Pre-training from scratch on Wikipedia
  - Huge impact on performance as compared to C4 pertaining.
  - Wikipedia too small, LM overfitting

Negative Results

- Span Corruption Pre-Training on Wikipedia Sentences with Salient Spans
  - No effect on performance.
  - SSM is needed together with sentences with salient entities.
- Fine Tuning on all QA tasks together
  - Performance drops on WebQ and TriviaQA, increases slightly on NQ

Negative Results

- Randomly Sampling Answers for Natural Questions.
  - NQ has questions with multiple correct answers.
  - Experimented two answer choosing strategies:
    - Random
    - First
  - Both performed equally well.

Discussion

- How efficient are large transformer models when compared to SOTA approaches on open-domain QA using external KBs?
- How to encode interpretability in the language models?
- How to encode new information/add new sources in the model?
- How to extend the system for multi-hop answering problems like DROP?

YOU
BELONG
HERE
CNS Diversity and Inclusion • cns.utexas.edu/diversity

Questions?