Tencent AI Lab

Recent Advances in Retrieval-Augmented Text Generation

Deng Cai (蔡登)
The Chinese University of Hong Kong

Yan Wang (王琰)
Tencent AI Lab

Lemao Liu (刘乐茂)
Tencent AI Lab

Shuming Shi (史树明)
Tencent AI Lab

What is This Tutorial About?

- Integrating Information Retrieval (IR) Techniques in Text Generation

Information Retrieval

+

Text Generation

Close-book exam
(Hard mode)

=

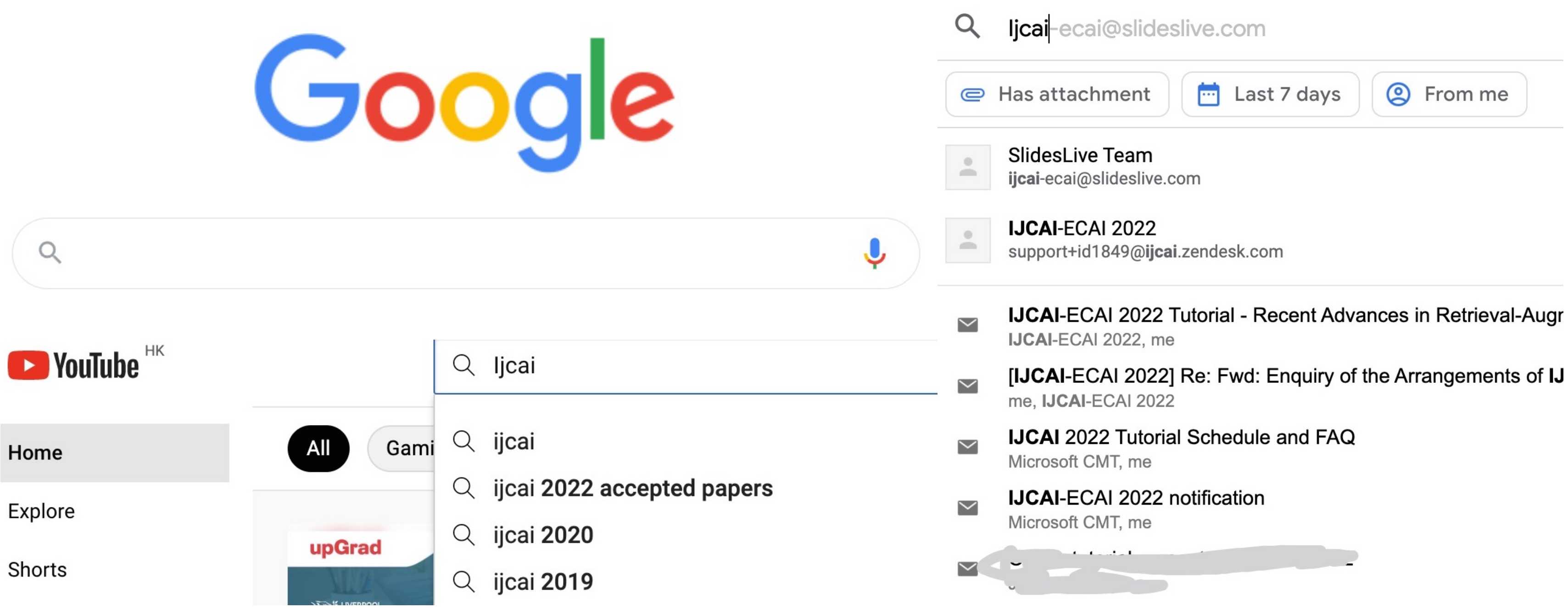
Retrieval-Augmented Text Generation

Open-book exam
(Easy mode)

Information Retrieval

- Information Retrieval (IR) is finding material of an unstructured nature (usually text) that satisfies an information need from large collections

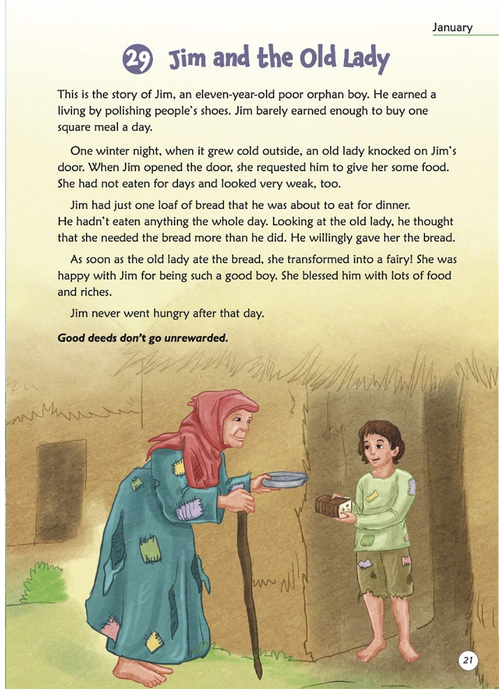
- Web Search
- Video Search
- E-mail Search



Text Generation

- Text generation, also known as natural language generation, is the task of generating text with the goal of appearing indistinguishable to human-written text

- Story Generation
- Dialogue Generation
- Machine Translation


January

29 Jim and the Old Lady

This is the story of Jim, an eleven-year-old poor orphan boy. He earned a living by polishing people's shoes. Jim barely earned enough to buy one square meal a day.

One winter night, when it grew cold outside, an old lady knocked on Jim's door. When Jim opened the door, she requested him to give her some food. She had not eaten for days and looked very weak, too.

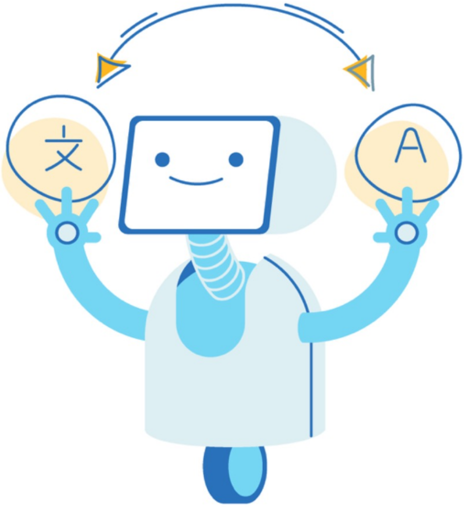
Jim had just one loaf of bread that he was about to eat for dinner. He hadn't eaten anything the whole day. Looking at the old lady, he thought that she needed the bread more than he did. He willingly gave her the bread.

As soon as the old lady ate the bread, she transformed into a fairy! She was happy with Jim for being such a good boy. She blessed him with lots of food and riches.

Jim never went hungry after that day.

Good deeds don't go unrewarded.

21



The Challenge

- Create is more difficult than judge!

Binary Classification

IJCAI-ECAI 2022 will be held on July?

- True
- False

Multi-Class Classification

When will IJCAI-ECAI 2022 be held?

- June
- July
- August
- September

Text Generation

Write about following topic

IJCAI-ECAI 2022 will be held at Vienna, Austria. What do you think about this conference? Will you attend this conference?

Write at least 250 words.

Require strong background information about IJCAI-ECAI 2022!

The information

Tencent AI Lab

- Where are these information?
  - In Training data
- How do we store these information
  - In Model parameters
  - This is why more data + bigger model always better in generation tasks
- Any alternative ways?
  - Endow model the capability to re-access its training data, or external resources

Close-book exam
(Hard mode)

Open-book exam
(Easy mode)

Successful Applications

- Language Modeling
- Open-Domain Dialogue Generation
- Machine Translation
- Question Answering
- Summarization
- Paraphrase Generation
- Text Style Transfer
- Data-to-Text Generation
- Image Caption
- Code Generation
- …

Outline

Language Modeling
(25 Min)

Dialogue Generation
(25 Min)

Machine Translation
(25 Min) +
Conclusion (5 Min)

Yan Wang (王琰)
Tencent AI Lab

Deng Cai (蔡登)
The Chinese University of Hong Kong

Lemao Liu (刘乐茂)
Tencent AI Lab

WARNING: this is a new research area, conclusions in this tutorial may be out-of-date soon!

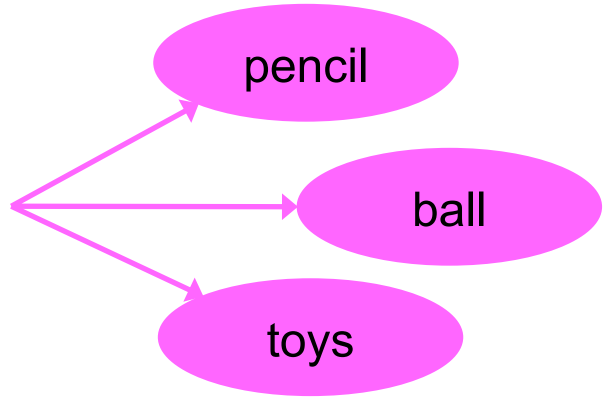
Outline

- Background and Introduction
- **Language Modeling**
- Open-Domain Dialogue Systems
- Neural Machine Translation
- Conclusion and Outlook

Language Modeling

- Language Modeling is a fundamental NLP task that predicting what word comes next

A boy is looking at his ______



- Formally: given a sequence of words $x^1, x^2, \dots, x^t$, compute the probability distribution of the next word x^{t+1}:

$$P(x^{t+1} | x^1, \dots, x^t)$$

Where x^{t+1} can be any word in the vocabulary$V = \{w_1, \dots, w_{|V|}\}$

- A system that does this is called a Language Model (LM)

Evaluation of Language Modeling

- Perplexity: an intrinsic evaluation method for LM
- Intuition: The probability of correct text (test set) should be high

Test Set

"Yesterday I went to the cinema"

"Hello, how are you?"

"The dog was wagging its tail"

High probability
Low perplexity

Fake/incorrect sentences

"Can you does it?"

"For wall a driving"

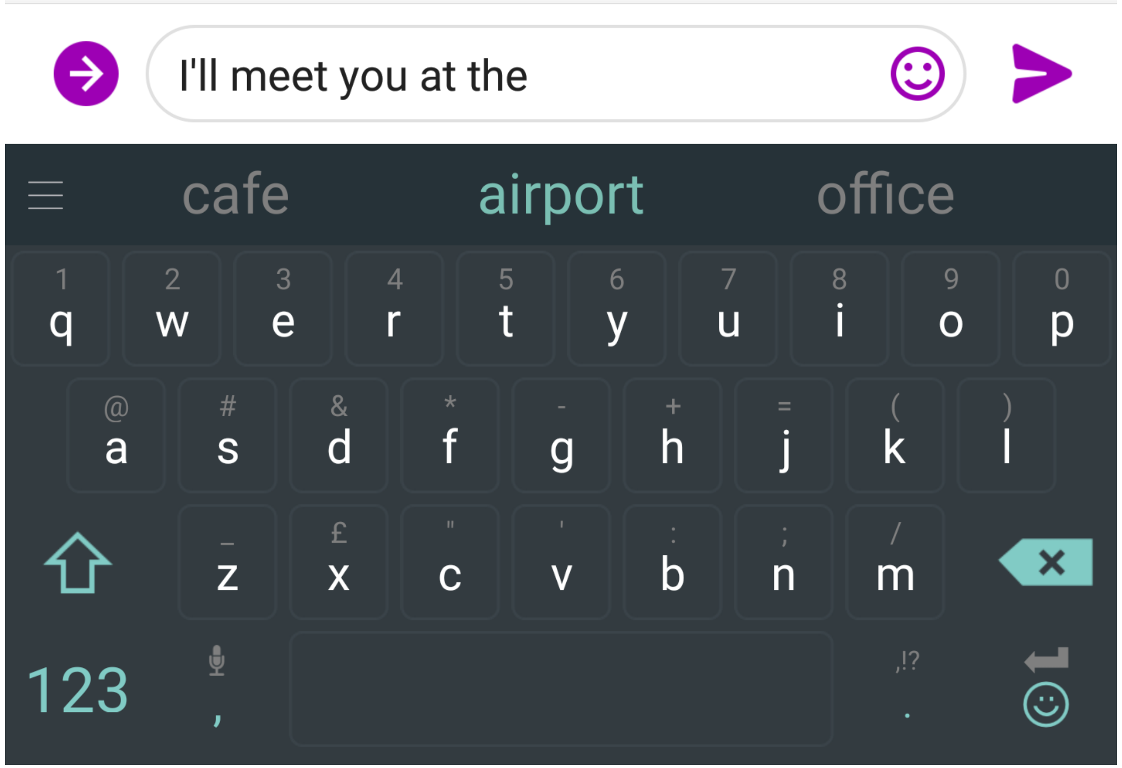
"She said me this"

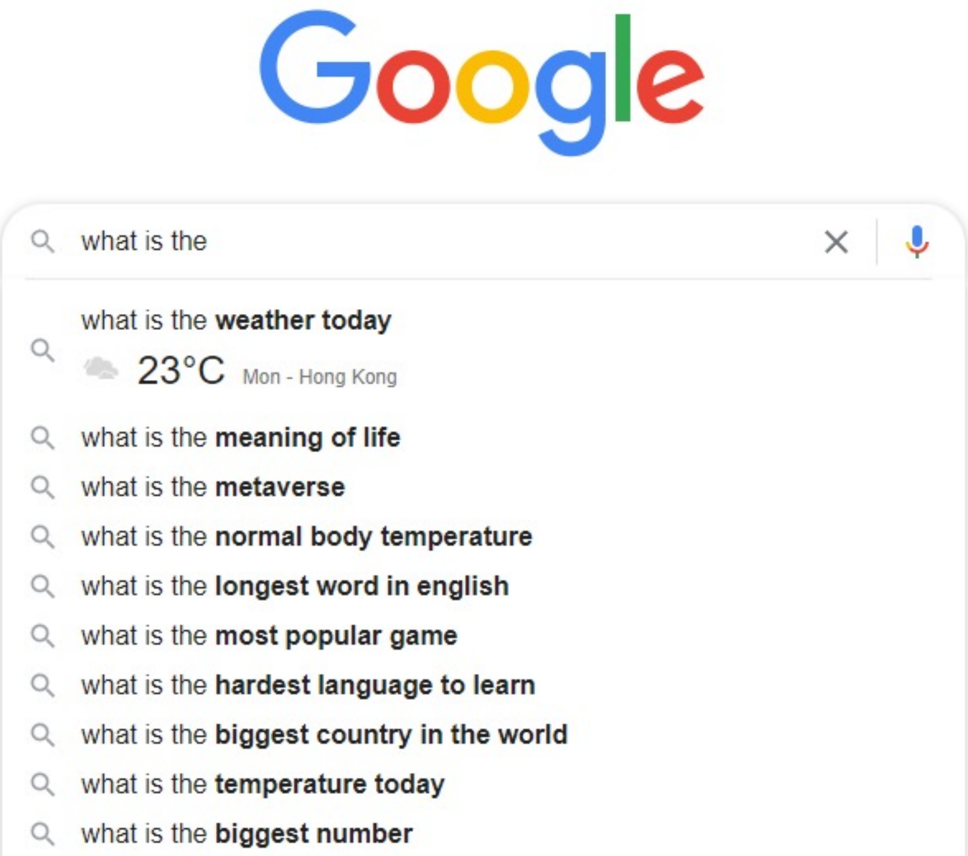
Low probability
High perplexity

- Formal definition:

$$PP(W) = \sqrt[N]{\frac{1}{P(w_1, w_2, \ldots, w_N)}}$$

We use LM every day!





Traditional (Pre-Deep Learning) way: n-gram LM

A boy is looking at his ______

- N-gram Language Model
- Definition: A *n-gram* is a chunk of n consecutive words.
  - 1-gram: “a”, “boy”, “is”, ”looking”, “at”, “his”
  - 2-grams: “a boy”, “boy is”, “is looking”, ”looking at”, “at his”
  - 3-grams:“a boy is”, “boy is looking”, “is looking at”, ”looking at his”
  - …
  - 6-grams: “a boy is looking at his ”
- N-gram LM: Collect statistics about how frequent different n-grams are

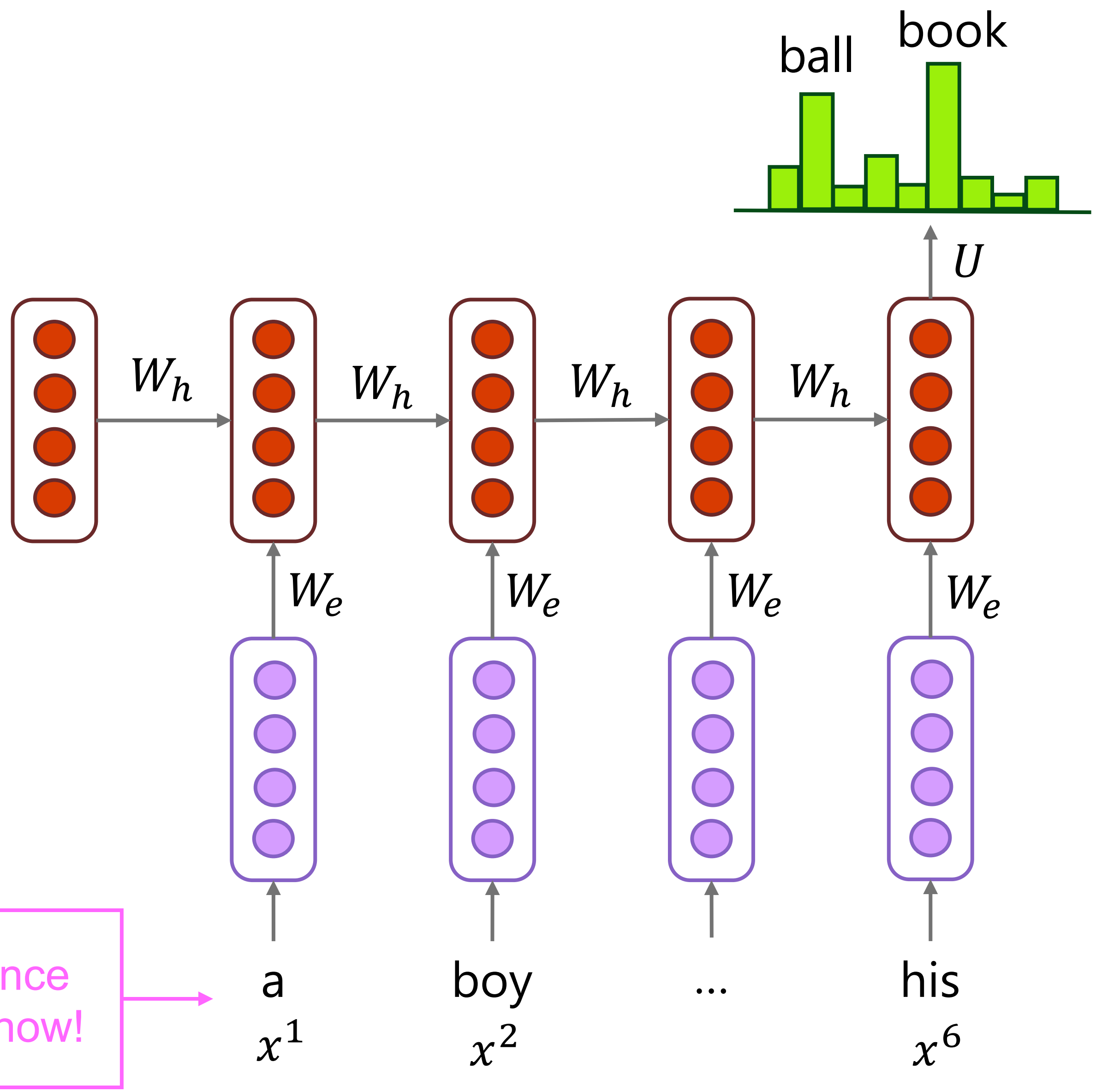
$$P(x^{t+1}|x^t,\dots,x^1) = P(x^{t+1}|x^t,\dots,x^{t-n+2}) \approx \frac{count(x^{t+1},x^t,\dots,x^{t-n+2})}{count(x^t,\dots,x^{t-n+2})}$$

Problems of n-gram LM

- Sparsity
  - Hard to compute the probability of unseen text
- Storage
  - Need to store count for all n-grams. Increasing n or corpus increases model size!

RNN Language Model

- Advantages:
  - Can process any length input
  - Theoretically, can consider very long context
  - Model size doesn't increase for longer input context
- Disadvantage:
  - Recurrent computation is slow
  - Difficult to access very long context in practice

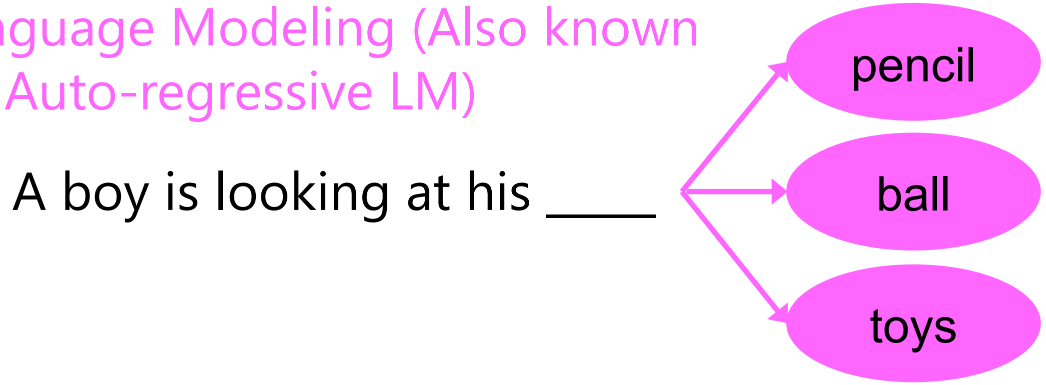


Note: this input sequence could be much longer now!

Pre-trained Language Model (PLM)

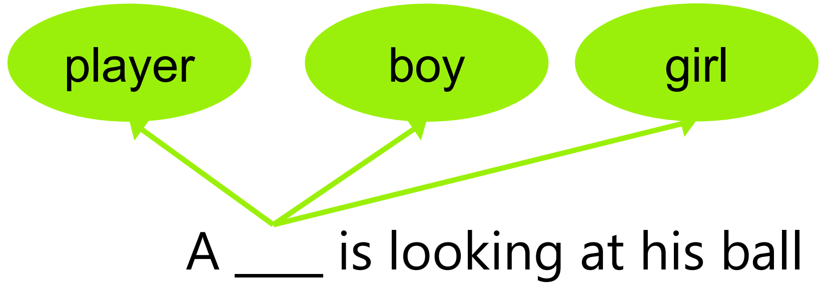
- Two pretraining objectives:

Language Modeling (Also known as Auto-regressive LM)



- Condition on the past only
- Representatives: GPT, GPT2, Retro
- It's helpful when the output is a sequence:
  - Dialogue (Condition on dialogue history)
  - Story Generation (Condition on story title)

Masked Language Modeling



- Condition on both the past and the future
- Representatives: BERT, and its variants
- It's helpful on Natural Language Understanding tasks
  - Sequence Labeling & Semantic Matching

PLM for Text Generation

Tencent AI Lab

- Open-Ended Text Generation: Fluent, informative, and coherent

Context (human-written): In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

GPT-2: The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

[Radford + 19]

Why So Good?

- Why so good?
  - Big: big model, big corpus
  - A way that teach the model remember knowledge in corpus

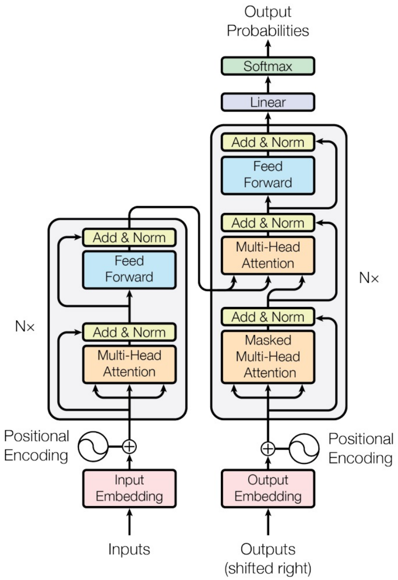
- What's bad?
  - Big->High cost on both time and space

Motivation of Retrieval-Augmented LM

Remember? This is the Expertise of IR

- Store knowledge in LM

Knowledge



- Store knowledge in non-parametric index

Knowledge

Full List of Retrieval-Augmented LM

- Interpolation-based LM
  - Improving neural language models with a continuous cache. ICLR 2017
  - Generalization through memorization: Nearest neighbor language models. ICLR 2020
  - Adaptive semiparametric language models. TACL 2021
- Masked LM and QA*
  - Dense passage retrieval for open-domain question answering. EMNLP 2020
  - Latent Retrieval for Weakly Supervised Open Domain Question Answering. ACL 2019
  - Retrieval augmented language model pre-training. ICML 2020
  - Retrieval-augmented generation for knowledge-intensive NLP tasks. NeuiPS 2020
  - Leveraging passage retrieval with generative models for open domain question answering. EACL 2021
- Huge-Index but Small-Size LM
  - Improving language models by retrieving from trillions of tokens. DeepMind 2022

*Retrieval-Augmented QA is not the core of this tutorial, one may refer to ACL tutorial "Knowledge-Augmented Methods for Natural Language Processing" for more details about this area

Full List of Retrieval-Augmented LM

- Interpolation-based LM
  - Improving neural language models with a continuous cache. ICLR 2017
  - Generalization through memorization: Nearest neighbor language models. ICLR 2020 ★
  - Adaptive semiparametric language models. TACL 2021
- Masked LM and QA*
  - Dense passage retrieval for open-domain question answering. EMNLP 2020
  - Latent Retrieval for Weakly Supervised Open Domain Question Answering. ACL 2019
  - Retrieval augmented language model pre-training. ICML 2020 ★
  - Retrieval-augmented generation for knowledge-intensive NLP tasks. NeuiPS 2020
  - Leveraging passage retrieval with generative models for open domain question answering. EACL 2021
- Huge-Index but Small-Size LM
  - Improving language models by retrieving from trillions of tokens. DeepMind 2022 ★

*Retrieval-Augmented QA is not the core of this tutorial, one may refer to ACL tutorial "Knowledge-Augmented Methods for Natural Language Processing" for more details about this area

Interpolation-based Method: KNN-LM

Generalization through Memorization: Nearest Neighbor Language Models

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, Mike Lewis
Stanford University, Facebook AI Research

Stanford | NLP

facebook Artificial Intelligence

KNN-LM: Intuition

x = Obama's birthplace is ____

Language Model (GPT2)

$q = f(x) =$

Nearest Neighbors

Keys	Values
f(Obama was senator for)	Illinois
f(Obama was born in)	Hawaii
…	…

P_{LM} on vocabulary	
Hawaii	0.2
Illinois	0.2
…	…

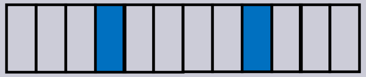
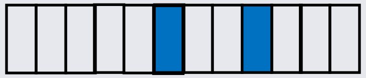
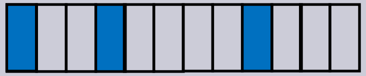
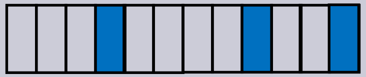
$$(1-\lambda)P_{LM}+\lambda P_{KNN}$$

P_{KNN} on vocabulary	
Hawaii	0.6
Illinois	0.2
…	…

Constructing the Index

Training Contexts c_i	Targets v_i
Obama was senator for	Illinois
Barack is married to	Michelle
Obama was born in	Hawaii
…	…
Obama is a native of	Hawaii

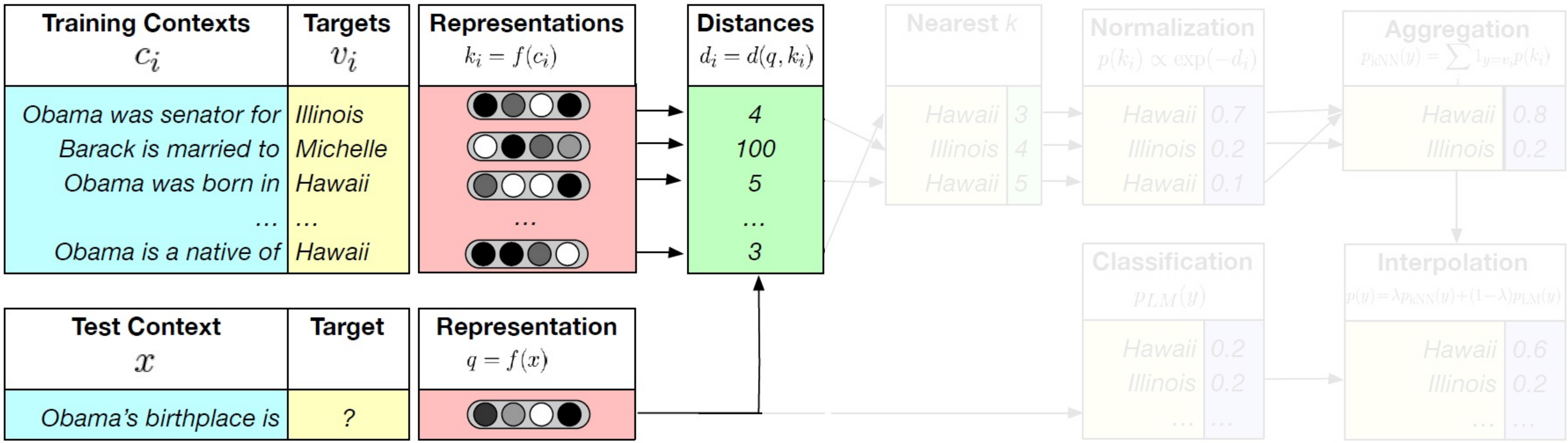
Constructing the Index

Training Contexts c_i	Representations $c_i = f(c_i)$	Targets v_i
Obama was senator for		Illinois
Barack is married to		Michelle
Obama was born in		Hawaii
…	…	…
Obama is a native of		Hawaii

The size of the datastore = The number of tokens in training corpus

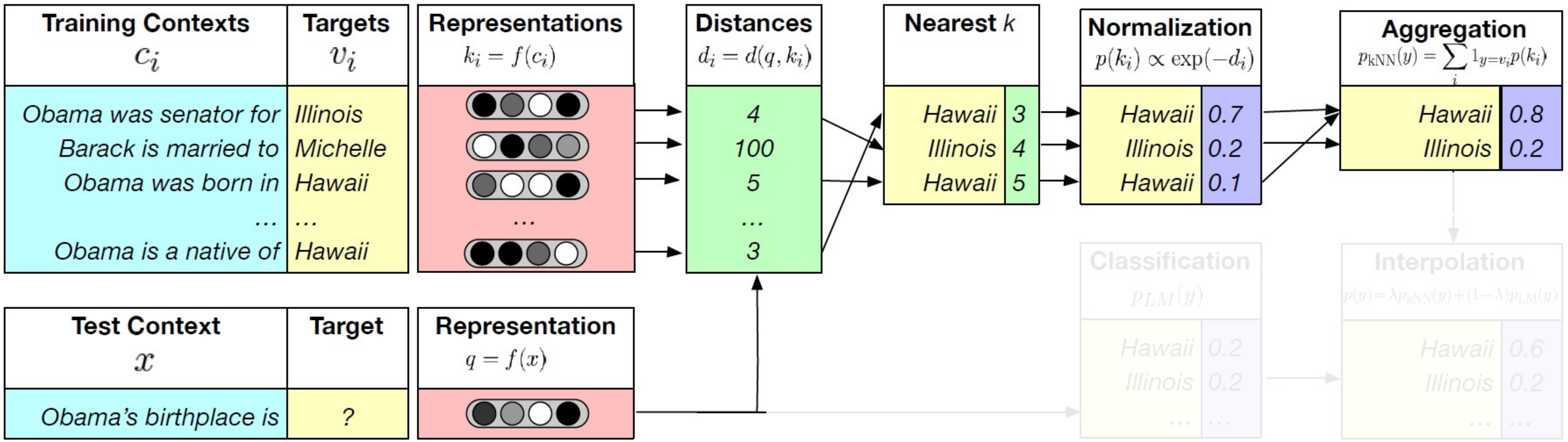
Retrieval nearest contexts according to current context $\boldsymbol{c_i}$

Back to Inference



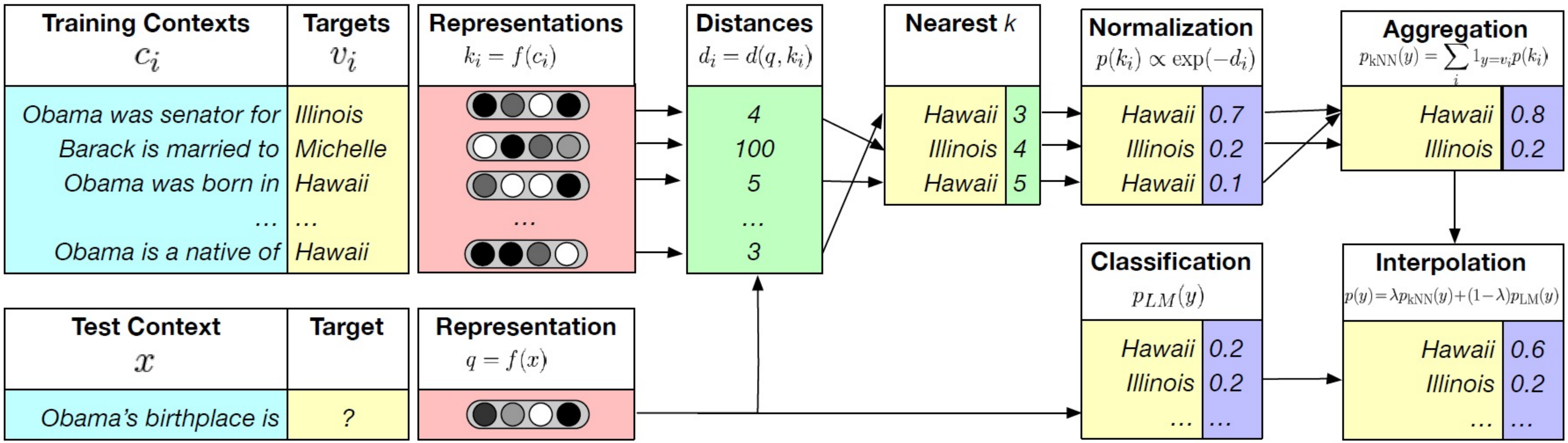
[Khandelwal+ 19]

Back to Inference



[Khandelwal+ 19]

Back to Inference



[Khandelwal+ 19]

Key Results

Explicitly memorizing the training data helps generation

LMs can scale to larger text collections without the added cost of training, by simply adding the data to the index

A single LM can adapt to multiple domains without the in-domain training, by adding domain-specific data to the index

Key Results

Tencent AI Lab

Memorizing with Wikitext-103: 103M tokens, $\lambda = 0.25$

Model	Perplexity↓
Previous Best (Luo et al., 2019)	17.40
Base LM	18.65
KNN-LM	16.12 ★
KNN-LM + Cont. Cache*	15.79 ★

*Edouard Grave, Armand Joulin, and Nicolas Usunier. Improving neural language models with a continuous cache. In ICLR, 2017

Key Results

Tencent AI Lab

Explicitly memorizing the training data helps generation

LMs can scale to larger text collections without the added cost of training, by simply adding the data to the index

A single LM can adapt to multiple domains without the in-domain training, by adding domain-specific data to the index

Key Results

From Wiketext-103 (100M tokens) to En-Wiki (3B tokens)

LM Training Data	Index	Perplexity↓
En-Wiki-3B	-	15.17
Wiki-100M	-	19.59
Wiki-100M	En-Wiki	13.73

Retrieving from corpus VS training on corpus

Key Results

Explicitly memorizing the training data helps generation

LMs can scale to larger text collections without the added cost of training, by simply adding the data to the index

A single LM can adapt to multiple domains without the in-domain training, by adding domain-specific data to the index

Key Results

Domain Adaptation from Wiki to Books

LM Training Data	Index	Perplexity↓
Books	-	11.89
Wiki-3B	-	34.84
Wiki-3B	Books	20.47

Domain adaptation in a plug-and-play manner!

Summary

Explicitly memorizing the training data helps generation

LMs can scale to larger text collections without the added cost of training, by simply adding the data to the index

A single LM can adapt to multiple domains without the in-domain training, by adding domain-specific data to the index

Limitations of KNN-LM

High index cost: Index size = Token number!

High inference cost: times of retrieval = generation length

Gap between training and inference: No retrieval in training

Retrieval-Augmented MLM Pretraining

REALM: Retrieval-Augmented Language Model Pre-training

Kelvin Guu*, **Kenton Lee***, Zora Tung, Ice Pasupat, Ming-Wei Chang

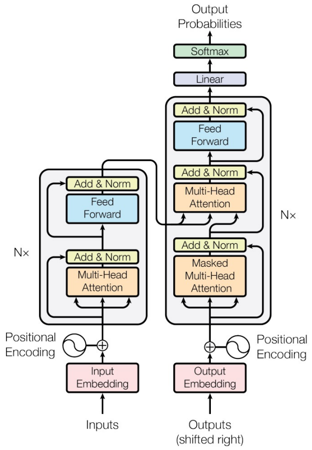
Google Research

* equal contribution

Introducing Explicit World Knowledge

Typical encoder: $p(y|x)$

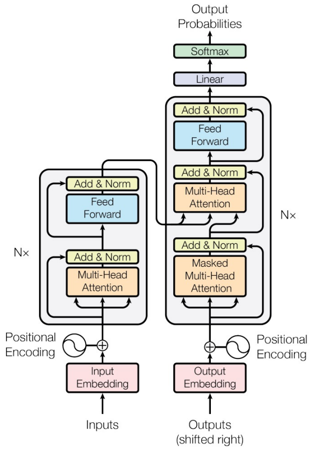
y= pounds



x: we paid 20 __ at the Buckingham Palace gift shop

Knowledge-augmented encoder:$p(y|x,z)$

y= pounds



x: we paid 20 __ at the Buckingham Palace gift shop

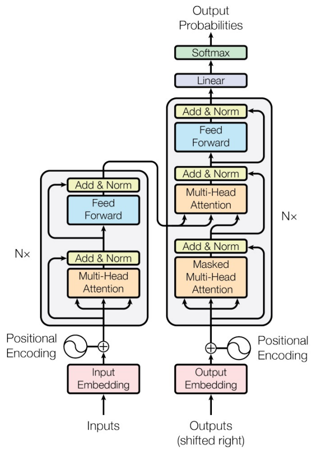
z: Buckingham Palace is home to the British monarchy

explicit knowledge

Problem: How to Select Right Knowledge

Knowledge-augmented encoder: $p(y|x, z)$

y= pounds



x: we paid 20 __ at the Buckingham Palace gift shop

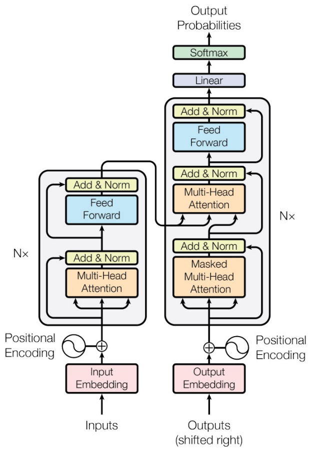
z: ???

No golden labels

Solution: try different documents

High

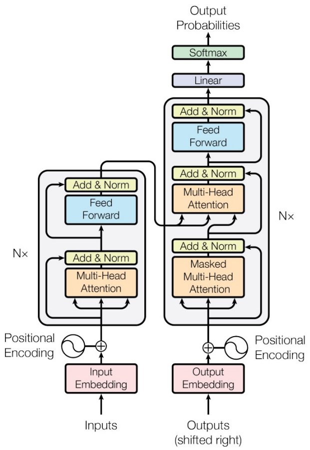
$$p(y = \text{'}pounds\text{'} | x, z_1)$$



z_1: Buckingham Palace is home to…

Low

$$p(y = \text{'}pounds\text{'} | x, z_2)$$



z_2: The Wall Street …

Neural Retriever: $p(z|x)$

x: we paid 20 __ at the Buckingham…

The Model

$$p(y|x) = \sum_z p(y|x,z)p(z|x)$$

Knowledge-Augmented Encoder

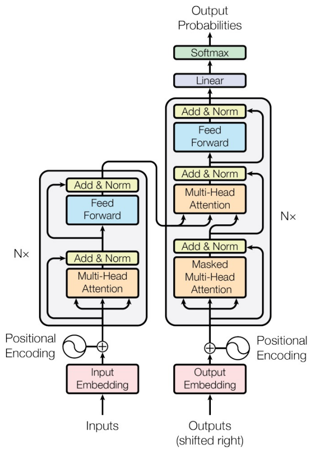
Neural Retriever

Challenge: Summation over millions of documents!
(for every sample, ever gradient step)

Approximation: Dual-Encoder + MIPS

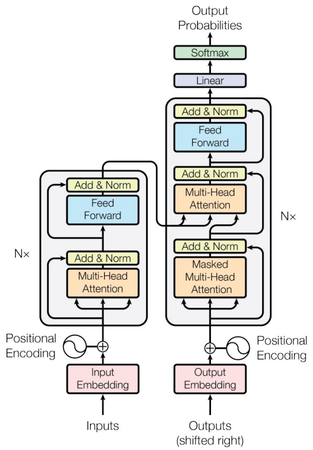
Retriever: $p(z|x) \propto h(x)^T h(z)$

$h(x)$



x: we paid 20 __ at the Buckingham…

$h(z)$



z: Buckingham Palace is home to…

- Search top-k candidates via MIPS tool:

$$p(y|x) = \sum_{z} p(y|x,z)p(z|x)$$

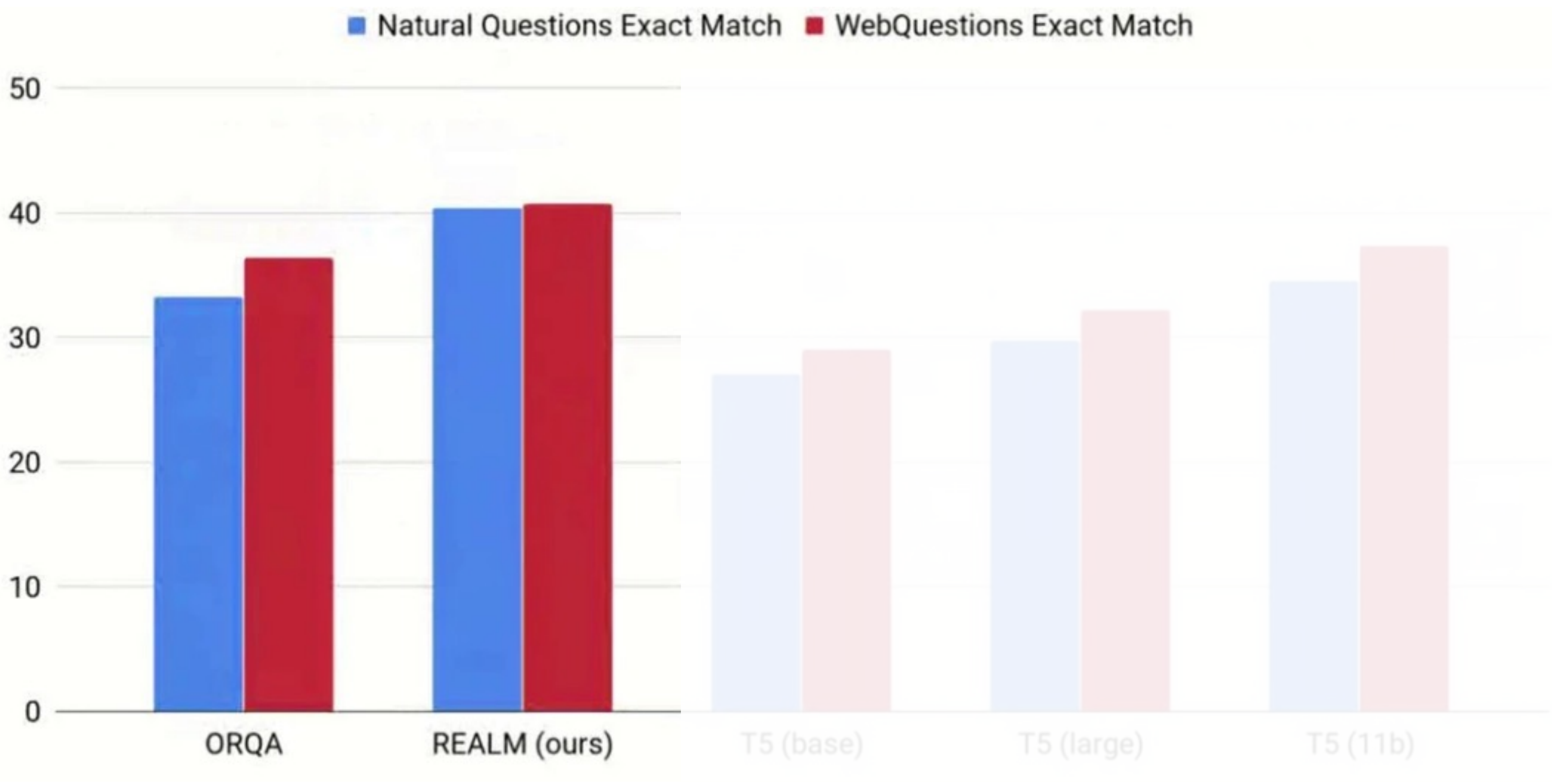
$$= \sum_{z \in MIPS(x)} p(y|x,z)p(z|x)$$

Key Results

- 3 open-domain QA datasets:
  - Natural Questions, WebQuestions, CuratedTrec

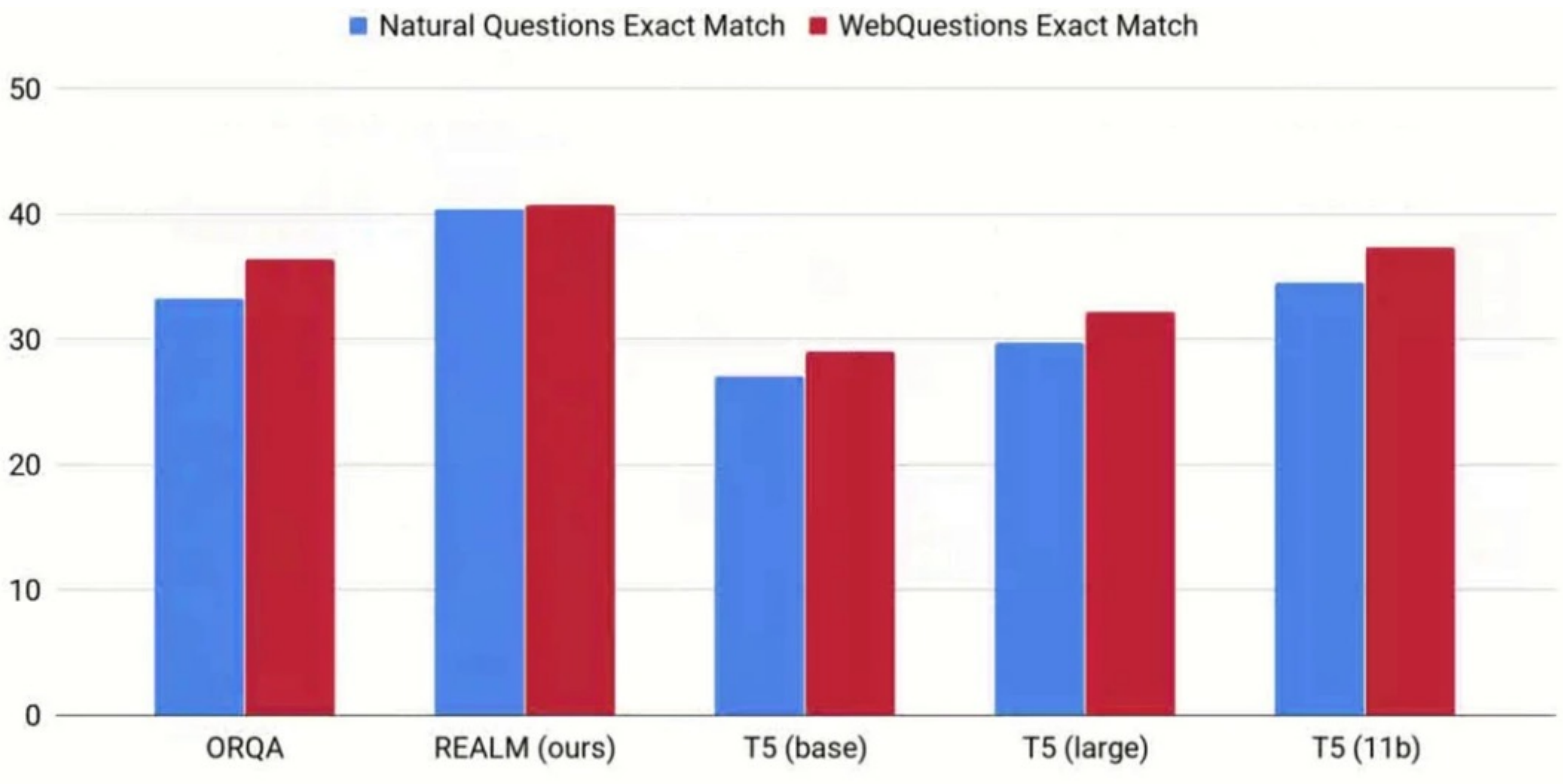
- Baselines
  - QRQA (Lee et al. 2019) – 330M paras
    - Equivalent to REALM without joint training
  - T5-base (220M), L (770M), XL (11B) (Raffel et al. 2019)

Key Results



[Guu+ 20]

Key Results



[Guu+ 20]

Comparison with KNN-LM

- Learnable Retriever and Joint Training Matters!

- Limitation:
  - Masked Language Model is unfriendly to Sequence Generation Tasks
  - Retrieval in very coarse-grained (document) level

DeepMind

Improving language models by retrieving from trillions of tokens

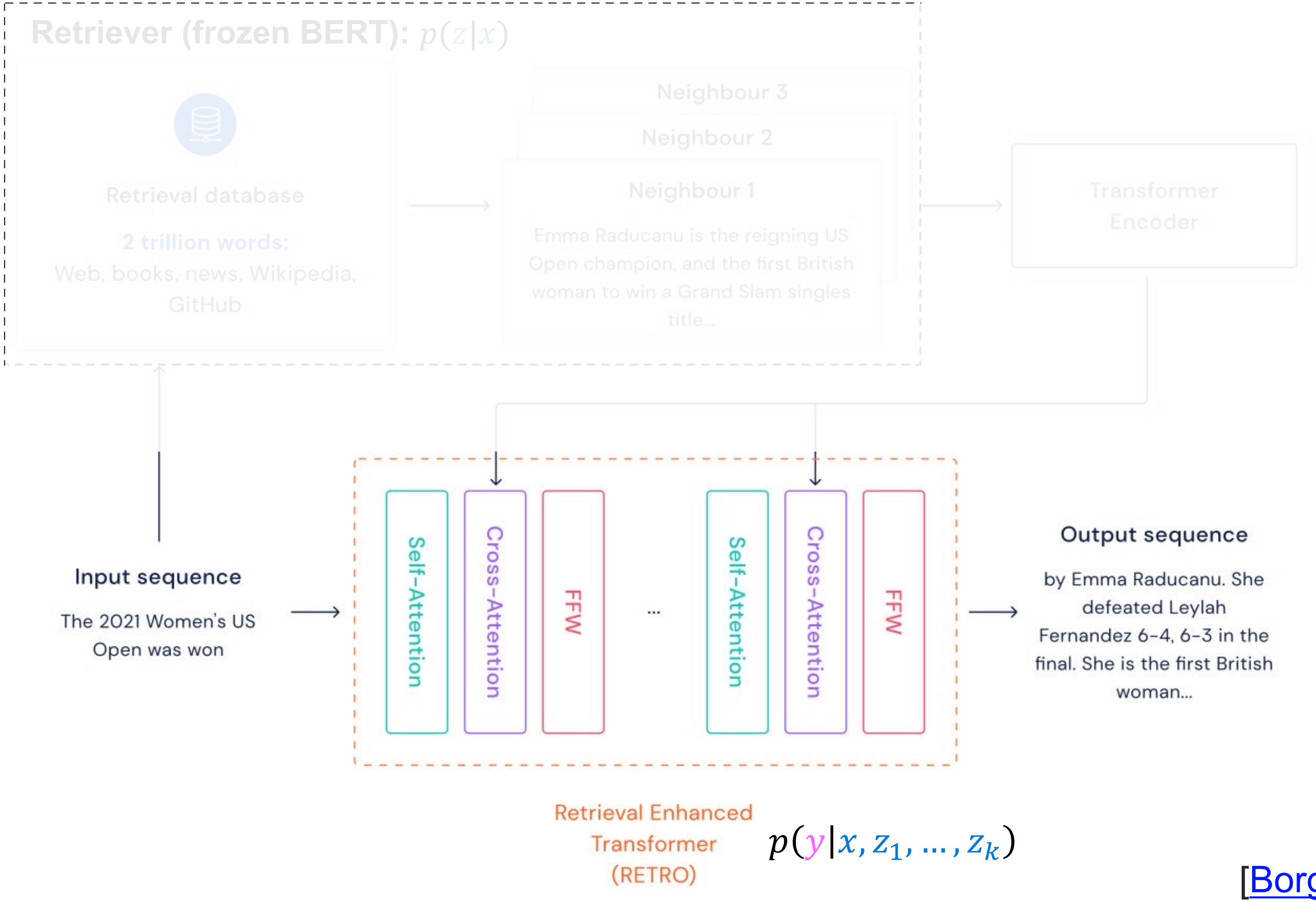
Sebastian Borgeaud[†], Arthur Mensch[†], Jordan Hoffmann[†], Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae[‡], Erich Elsen[‡] and Laurent Sifre[†,‡]

All authors from DeepMind, [†]Equal contributions, [‡]Equal senior authorship

Big Index + Small model

- RETRO: Retrieval-Enhanced transformer
  - Bigger and Bigger index:
    - from 200M~2B tokens (KNN-LM, REALM) to 2T tokens (RETRO)
  - Smaller and Smaller Model:
    - From 175B parameters (GPT3) to 172M ~ 7.5B parameters (RETRO)
  - Efficient training:
    - Works well without joint training

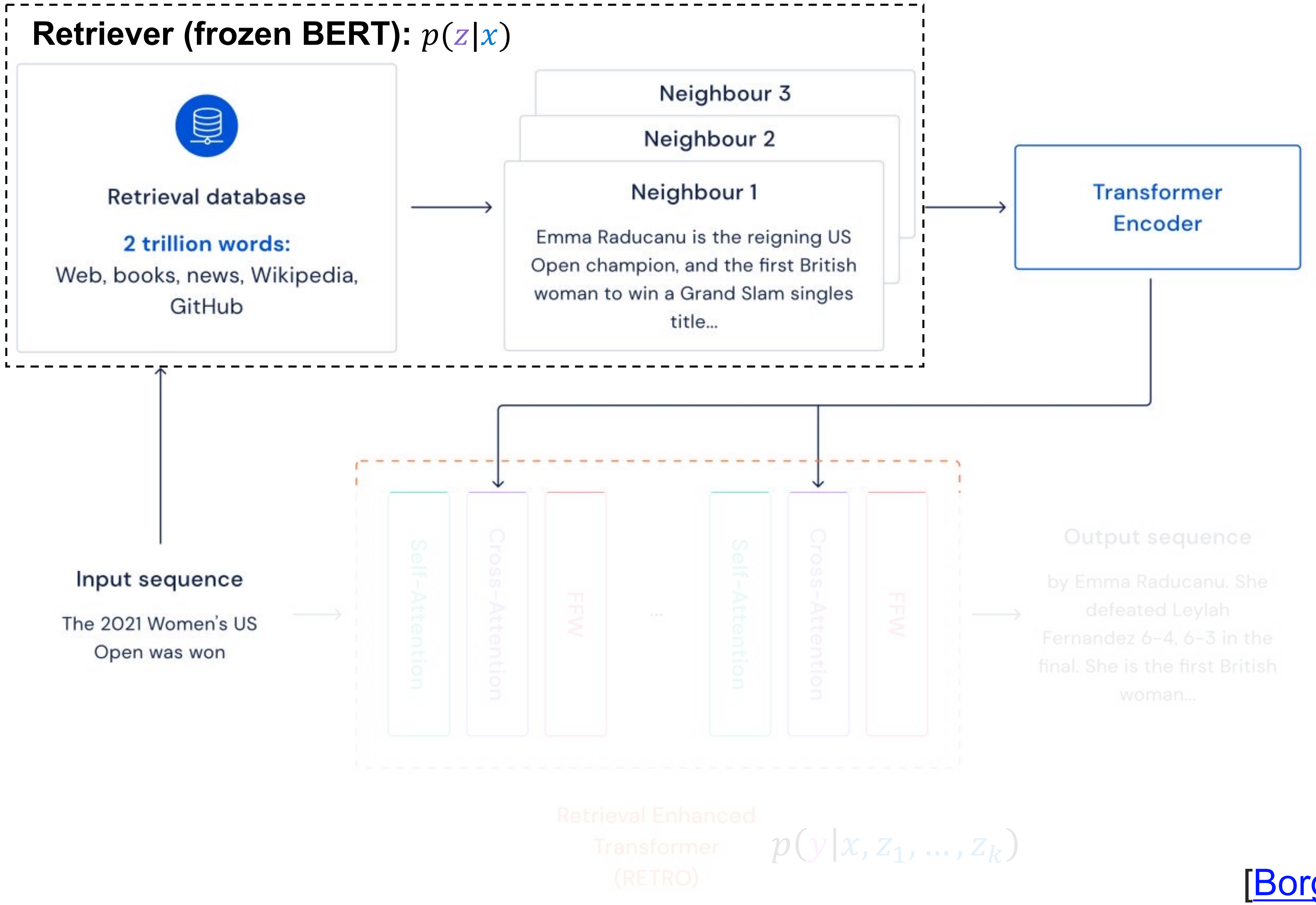
Main Framework: Decoder



$p(y|x, z_1, \dots, z_k)$

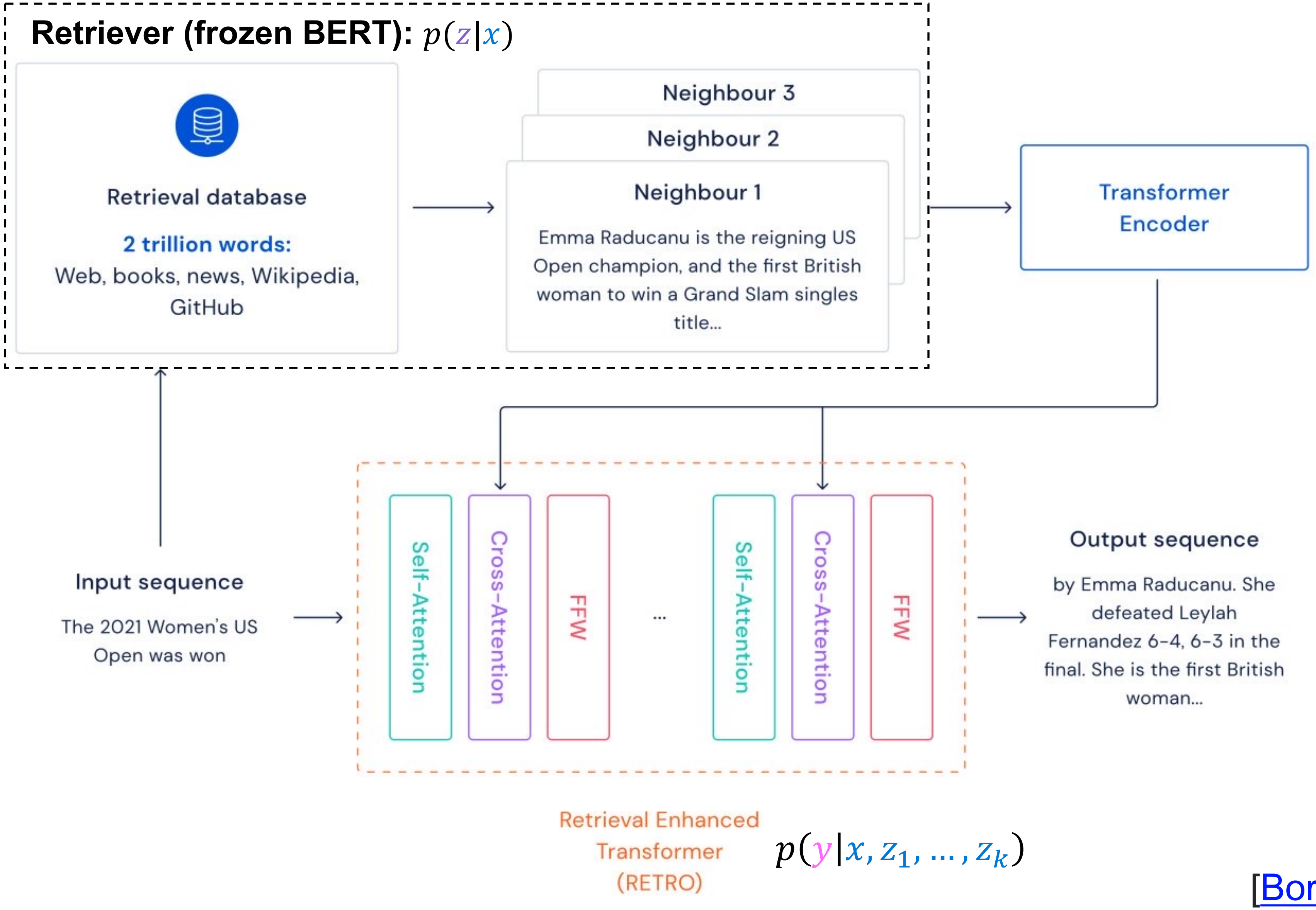
[Borgeaud+ 22]

Main Framework: Memory-Encoder



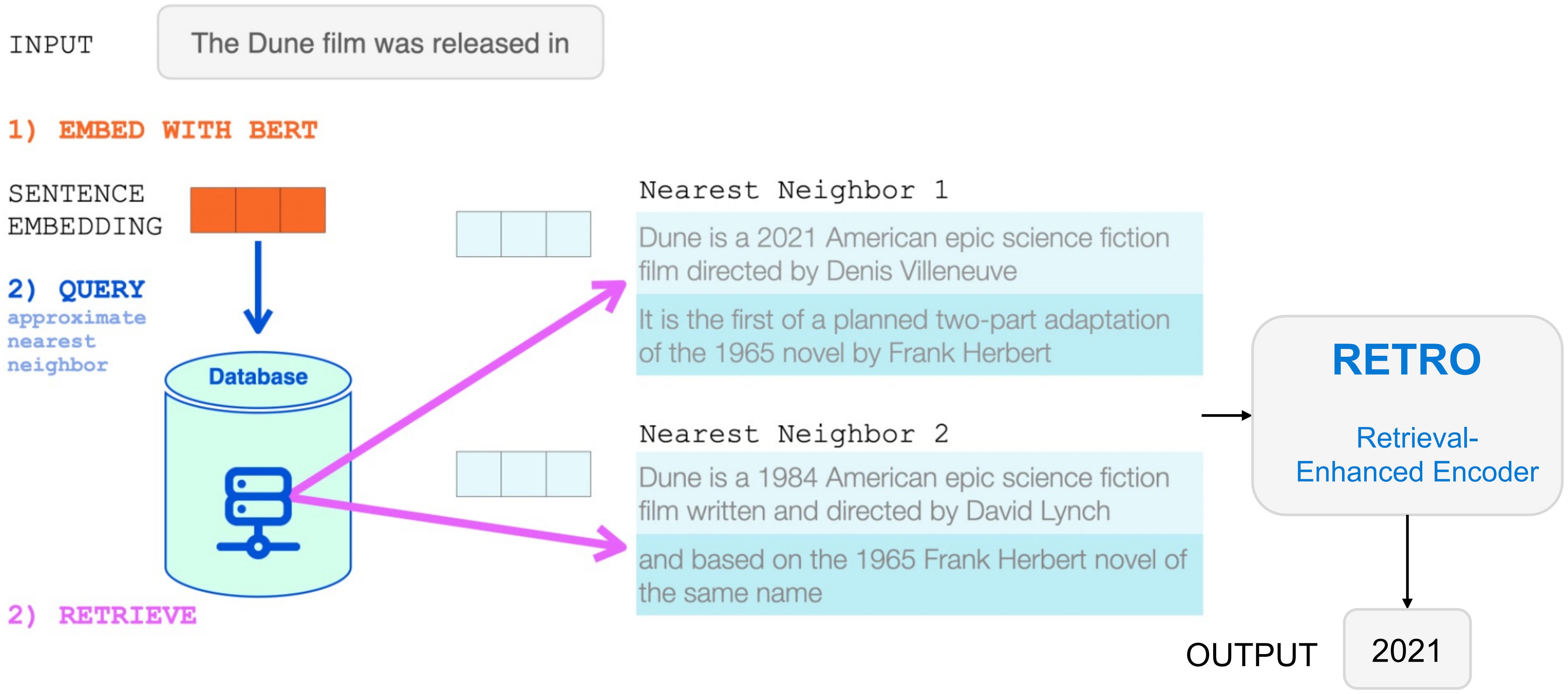
[Borgeaud+ 22]

Main Framework: Encoder-Decoder



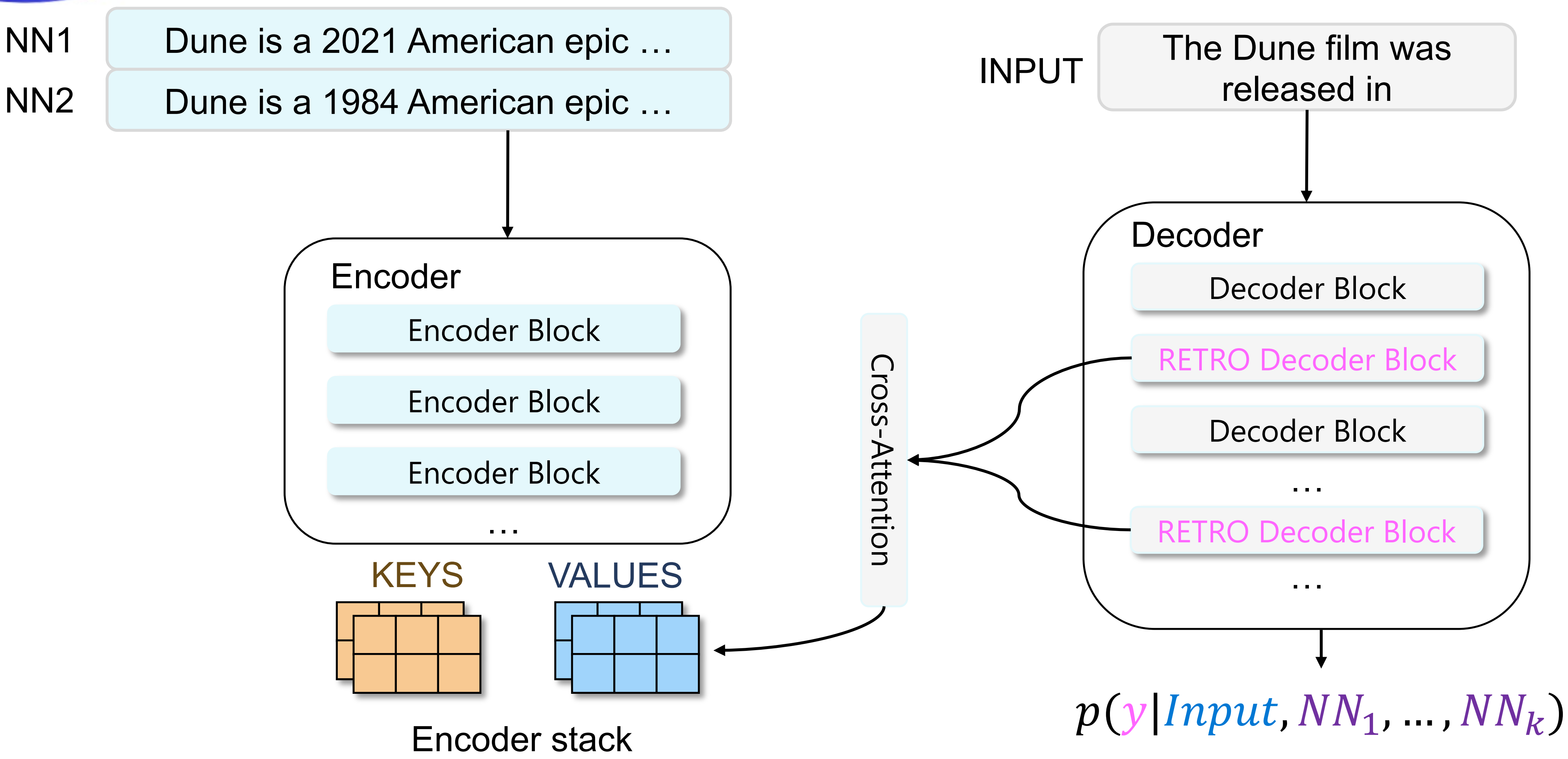
[Borgeaud+ 22]

Nearest Neighbor Search



https://jalammar.github.io/illustrated-retrieval-transformer/

Retrieval-Augmented Generation



Experimental Baselines

Tencent AI Lab

- Baselines:
  - Small models:

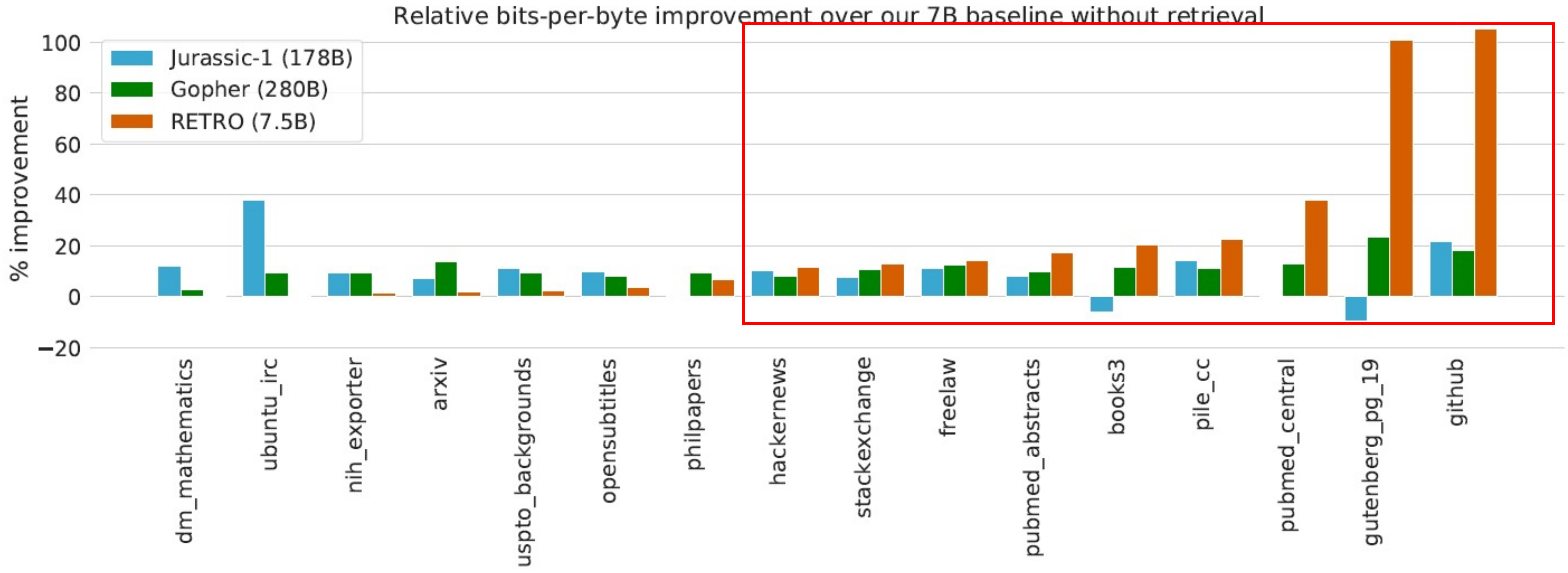
Baseline parameters	RETRO	d	d_{ffw}	# heads	Head size	# layers
132M	172M (+30%)	896	3,584	16	64	12
368M	425M (+15%)	1,536	6,144	12	128	12
1,309M	1,451M (+11%)	2,048	8,192	16	128	24
6,982M	7,532M (+8%)	4,096	16,384	32	128	32

  - Jurasic-1 (Lieber et al., 2021): 178B parameters
  - Gopher (Rae et al., 2021): 280B parameters

Gopher and Jurrasic-1 outperforms GPT-3 in most tasks!

Comparison with Large Models

- Outperforms 7B baselines on all test sets
- Comparable with two very large models (178B and 280B) in 16 domains



Comparison with Other Retrieval-Augmented Models

- Performance gain from big Database

Model	Retrieval Set	#Database tokens	#Database keys	Valid	Test
Adaptive Inputs (Baevski and Auli, 2019)	-	-	-	17.96	18.65
SPALM (Yogatama et al., 2021)	Wikipedia	3B	3B	17.20	17.60
*k*NN-LM (Khandelwal et al., 2020)	Wikipedia	3B	3B	16.06	16.12
Megatron (Shoeybi et al., 2019)	-	-	-	-	10.81
Baseline transformer (ours)	-	-	-	21.53	22.96
*k*NN-LM (ours)	Wikipedia	4B	4B	18.52	19.54
RETRO	Wikipedia	4B	0.06B	18.46	18.97
RETRO	C4	174B	2.9B	12.87	10.23
RETRO	MassiveText (1%)	18B	0.8B	18.92	20.33
RETRO	MassiveText (10%)	179B	4B	13.54	14.95
RETRO	MassiveText (100%)	1792B	28B	**3.21**	**3.92**

An Interesting Sample

The RETRO model stays more on-topic than the baseline sample

Input prompt

Beavers are interesting animals that live near rivers. They build

Baseline 7.1B sample

dams to create ponds. Frogs are amphibians so they can live in both land and water. They have great camouflage to hide from predators. The Golden Retriever, or Golden, is a very big dog...

RETRO 7.5B sample

their houses called beaver dams in the riverbeds. They also live on land. Beavers use their strong teeth and strong jaws to cut down trees and branches to build their homes. They also use their teeth and jaws to chew up food.

The Evolution of Retrieval-Augmented LM

- Three types:
  - Token-level and Interpolation-based model
  - Document-level and Joint-Training model
  - Chunk-level, Frozen-Retriever, huge index model

	# Retrieval tokens	Granularity	Retriever training	Retrieval integration
Continuous Cache	$\mathcal{O}(10^3)$	Token	Frozen (LSTM)	Add to probs
kNN-LM	$\mathcal{O}(10^9)$	Token	Frozen (Transformer)	Add to probs
SPALM	$\mathcal{O}(10^9)$	Token	Frozen (Transformer)	Gated logits
DPR	$\mathcal{O}(10^9)$	Prompt	Contrastive proxy	Extractive QA
REALM	$\mathcal{O}(10^9)$	Prompt	End-to-End	Prepend to prompt
RAG	$\mathcal{O}(10^9)$	Prompt	Fine-tuned DPR	Cross-attention
FID	$\mathcal{O}(10^9)$	Prompt	Frozen DPR	Cross-attention
EMDR^2	$\mathcal{O}(10^9)$	Prompt	End-to-End (EM)	Cross-attention
RETRO (ours)	$\mathcal{O}(10^{12})$	**Chunk**	**Frozen (BERT)**	**Chunked cross-attention**

Thanks!