CS378: Natural Language Processing
Lecture 24: Question Answering

Greg Durrett

TEXAS
The University of Texas at Austin

Administrivia

- A4 back, A5 back soon
- Final project check-ins due on **Friday**
- Extra credit for eCIS on final project
- Colin Raffel talk Friday

Previously: SQuAD

Q: What was Marie Curie the first female recipient of?

Passage: One of the most famous people born in Warsaw was Marie Skłodowska-Curie, who achieved international recognition for her research on radioactivity and was the first female recipient of the **Nobel Prize**. Famous musicians include Władysław Szpilman and Frédéric Chopin. Though Chopin was born in the village of Żelazowa Wola, about 60 km (37 mi) from Warsaw, he moved to the city with his family when he was seven months old. Casimir Pulaski, a Polish general and hero of the American Revolutionary War, was born here in 1745.

Answer = Nobel Prize

- **Assume we know a passage that contains the answer**

Types of QA

- *What were the main causes of World War II?* — requires summarization
- *Can you get the flu from a flu shot?* — want IR to provide an explanation of the answer, not just yes/no
- *What was Marie Curie the first female recipient of?* — could be written down in a KB but probably isn't
- *How long should I soak dry pinto beans?*
- *When as Marie Curie born?* — we should just find this in a knowledge base

QA Pipeline

Three Approaches to QA

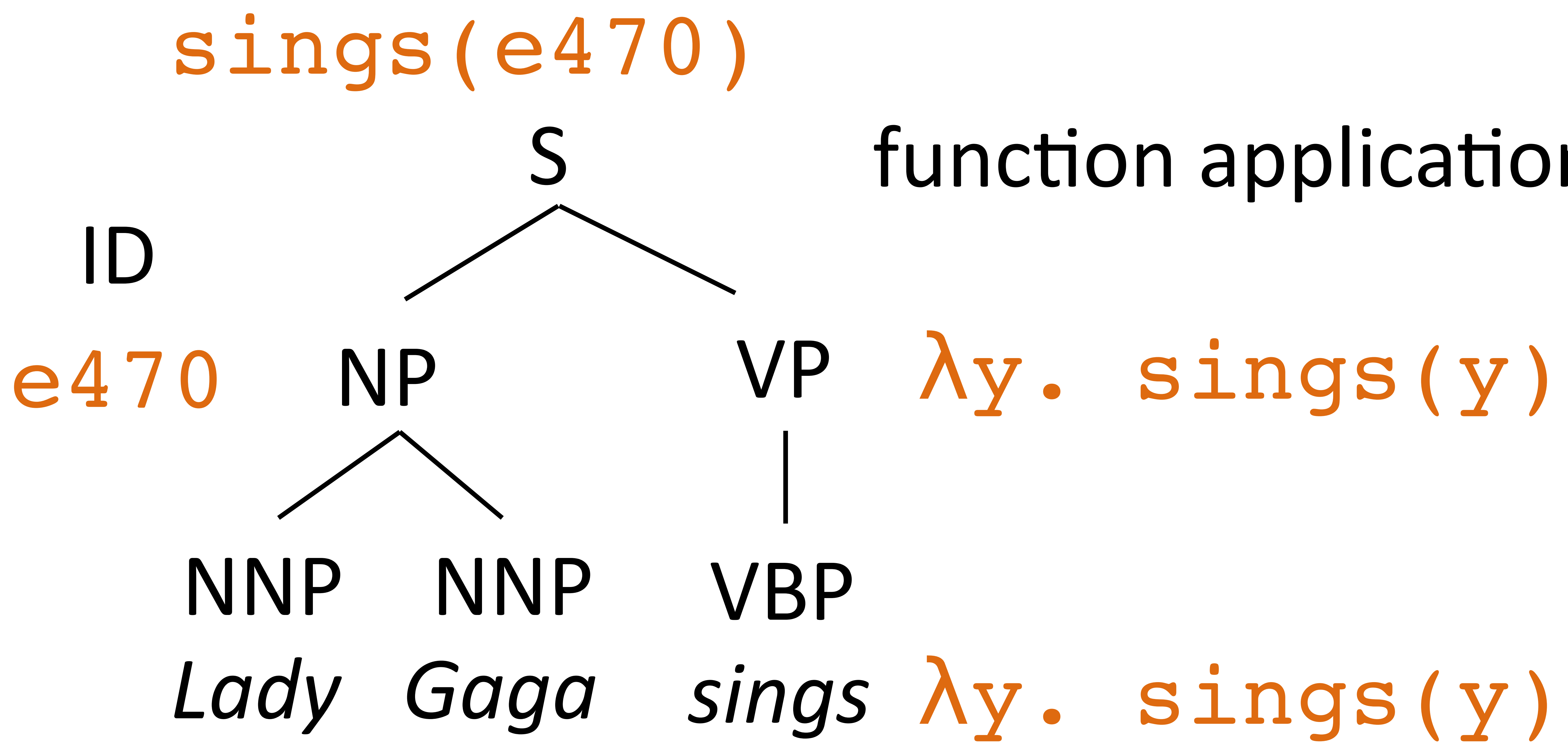
- Answering questions from a passage: done (this was SQuAD)
- Answering questions from a knowledge base: requires synthesizing a query (lambda calculus, SQL, etc.)
  - Today: **semantic parsing**
- Answering questions from the web: requires finding text that contains the answer
  - Today: **retrieval models**

Semantic Parsing

Logical Forms I

Logical Forms II

Montague Semantics

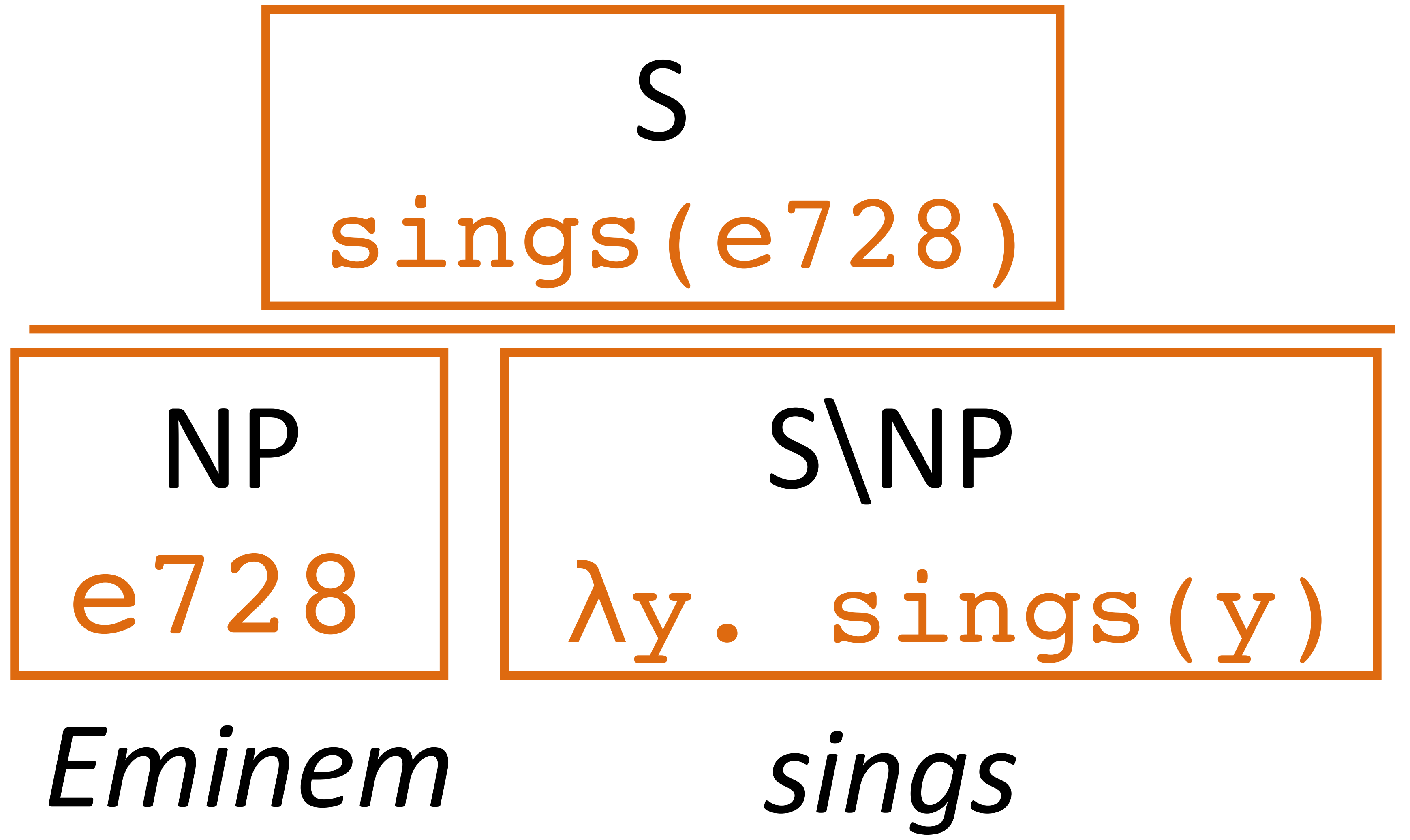


takes one argument (y, the entity) and returns a logical form `sings(y)`

- We can use the syntactic parse as a bridge to the lambda-calculus representation, build up a logical form (our model) *compositionally*

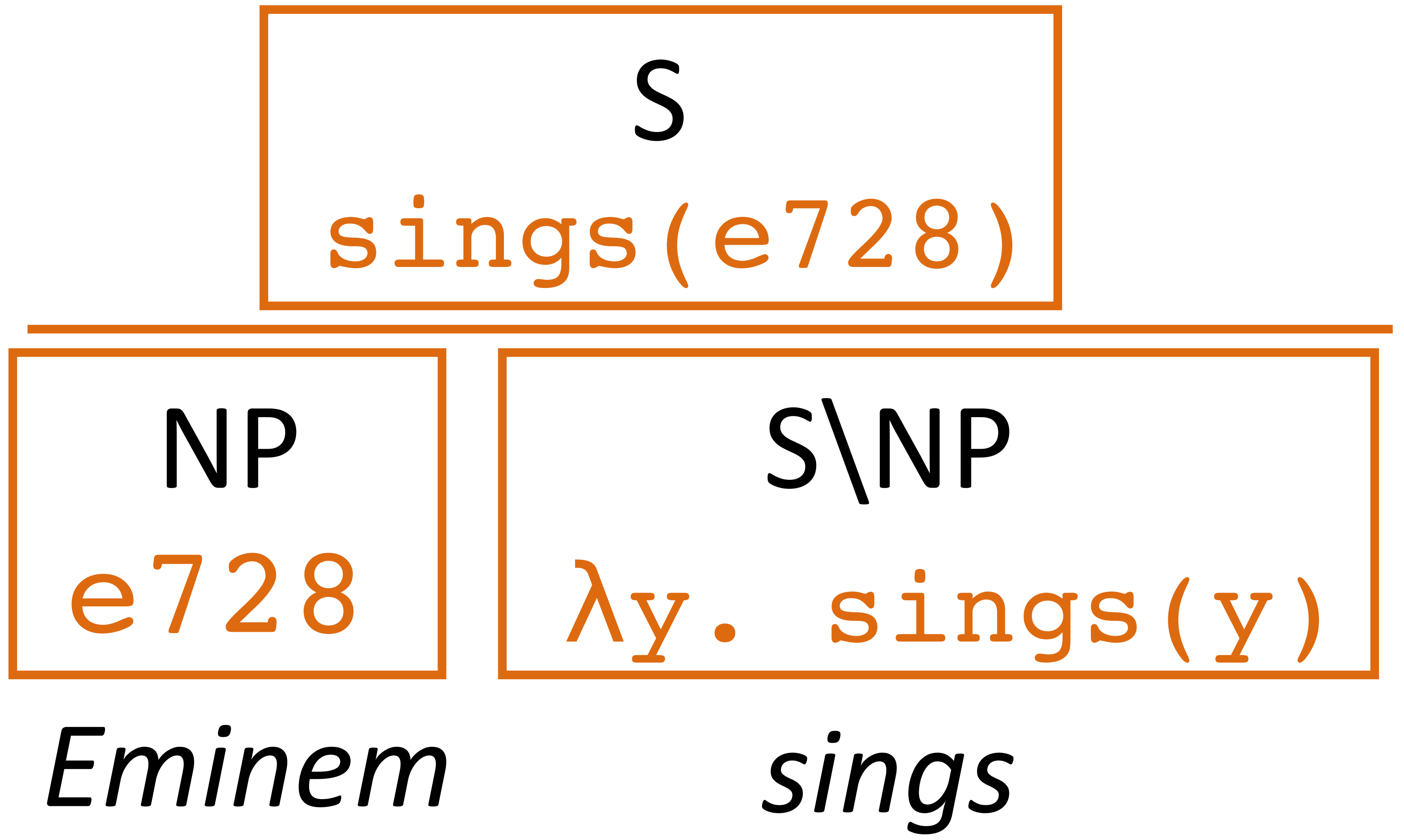
Combinatory Categorial Grammar

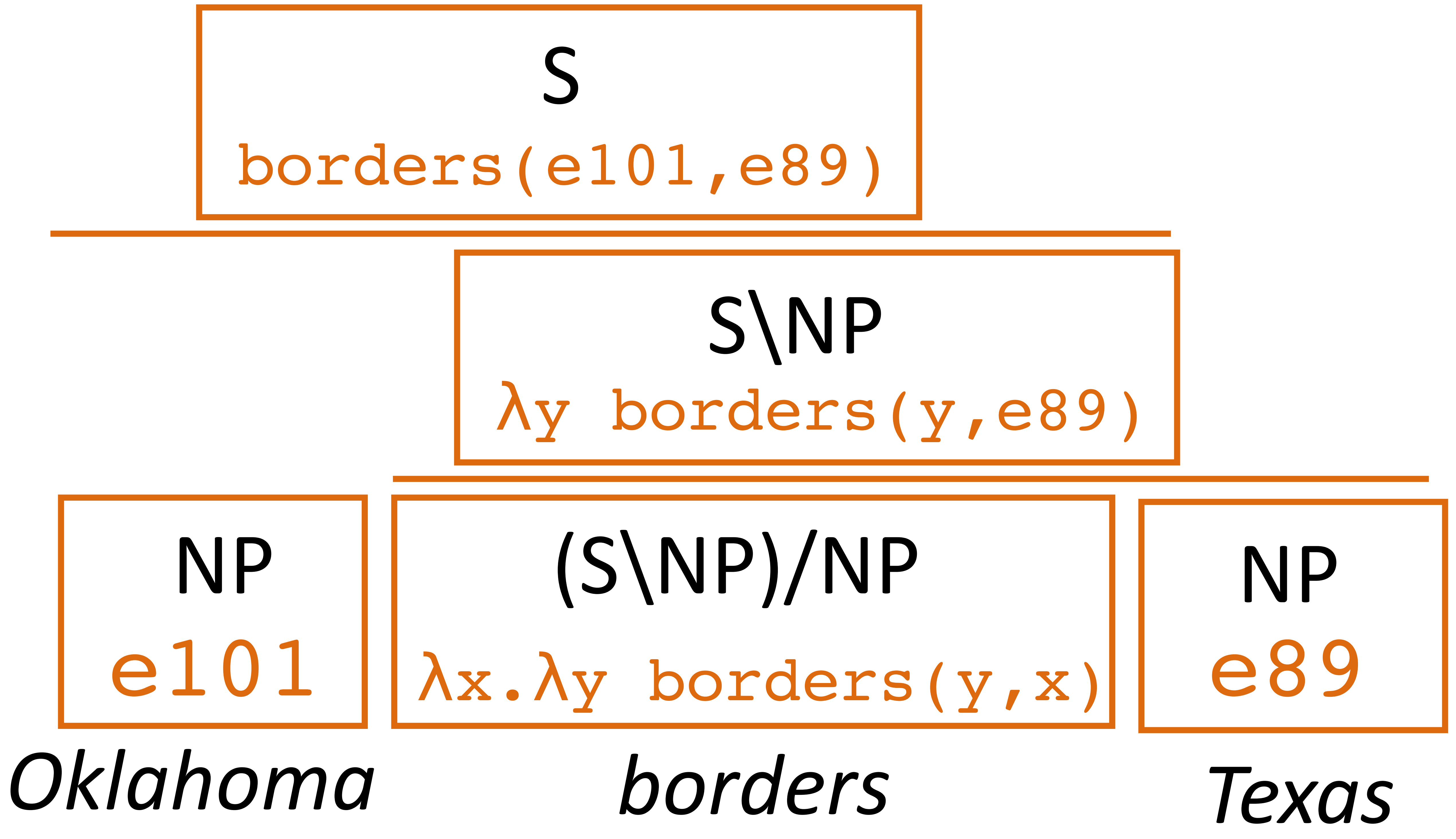
- Steedman+Szabolcsi (1980s): formalism bridging syntax and semantics
- Parallel derivations of syntactic parse and lambda calculus expression
- Syntactic categories (for this lecture): S, NP, "slash" categories
- S\NP: "if I combine with an NP on my left side, I form a sentence" — verb
- When you apply this, there has to be a parallel instance of function application on the semantics side



Combinatory Categorial Grammar

- Steedman+Szabolcsi (1980s): formalism bridging syntax and semantics
- Syntactic categories (for this lecture): S, NP, "slash" categories
  - S\NP: "if I combine with an NP on my left side, I form a sentence" — verb
  - (S\NP)/NP: "I need an NP on my right and then on my left" — verb with a direct object





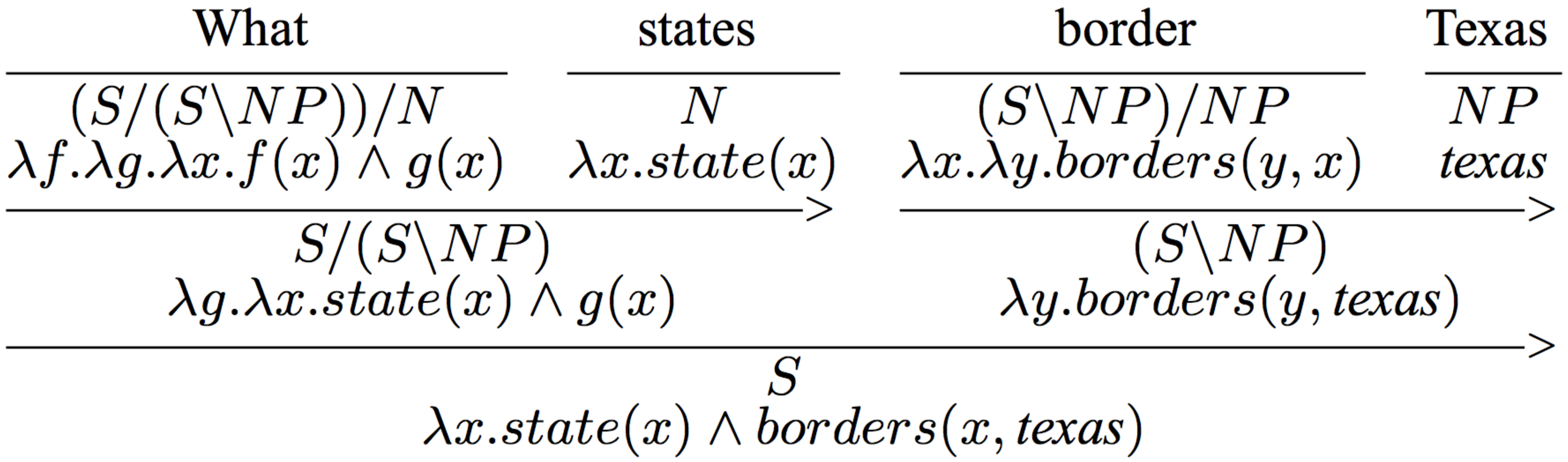
CCG Parsing

$$\begin{array}{ccccc}
\text{What} & \text{states} & \text{border} & \text{Texas} & \\
\hline
(S/(S\backslash NP))/N & N & (S\backslash NP)/NP & NP & \\
\lambda f.\lambda g.\lambda x.f(x)\wedge g(x) & \lambda x.state(x) & \lambda x.\lambda y.borders(y,x) & texas & \\
\cline{3-4}
 & & \multicolumn{2}{c}{(S\backslash NP)} & > \\
 & & \multicolumn{2}{c}{\lambda y.borders(y,texas)} &
\end{array}$$

- "What" is a ***very*** complex type: needs a noun and needs a S\NP to form a sentence. S\NP is basically a verb phrase (*border Texas*)

Zettlemoyer and Collins (2005)

CCG Parsing


What
states
border
Texas
$(S/(S\backslash NP))/N$
$\lambda f.\lambda g.\lambda x.f(x) \wedge g(x)$
N
$\lambda x.state(x)$
$(S\backslash NP)/NP$
$\lambda x.\lambda y.borders(y,x)$
NP
$texas$
$>$
$S/(S\backslash NP)$
$\lambda g.\lambda x.state(x) \wedge g(x)$
$>$
$(S\backslash NP)$
$\lambda y.borders(y, texas)$
$>$
S
$\lambda x.state(x) \wedge borders(x, texas)$

- "What" is a ***very*** complex type: needs a noun and needs a S\NP to form a sentence. S\NP is basically a verb phrase (*border Texas*)
- *What* in this case knows that there are two predicates (*states* and *border Texas*). This is not a general thing

Zettlemoyer and Collins (2005)

CCG Parsing

- These question are *compositional*: we can build bigger ones out of smaller pieces

What states border Texas?

What states border states bordering Texas?

What states border states bordering states bordering Texas?

Training CCG Parsers

- Training data looks like pairs of sentences and logical forms

What states border Texas	`λx. state(x) ∧ borders(x, e89)`
What borders Texas	`λx. borders(x, e89)`

…

- Unlike PCFGs, we don't know which words yielded which fragments of CCG

- Very hard to build a conventional parser for this problem

Zettlemoyer and Collins (2005)

Semantic Parsing as Translation

"what states border Texas"

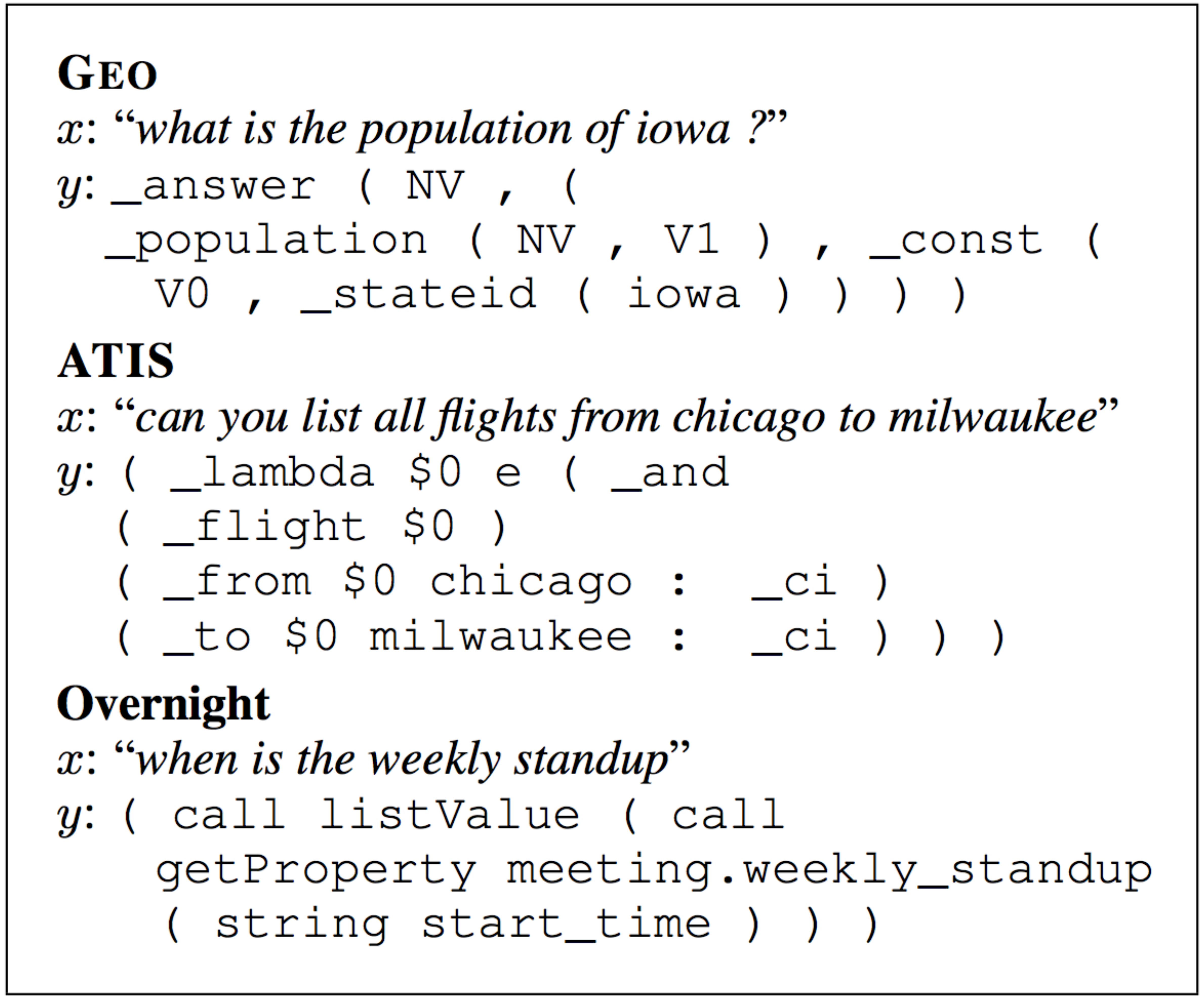
$\downarrow$

```
lambda x ( state ( x ) and border ( x , e89 ) ) )
```

- Write down a linearized form of the semantic parse, train seq2seq models to directly translate into this representation (similar to code generation like GitHub Copilot)
- What are some benefits of this approach compared to grammar-based?
- What might be some concerns about this approach? How do we mitigate them?

Jia and Liang (2016)

Semantic Parsing as Translation



GEO

x: *"what is the population of iowa ?"*

y:
```
_answer ( NV , (
 _population ( NV , V1 ) , _const (
  V0 , _stateid ( iowa ) ) ) )
```

ATIS

x: *"can you list all flights from chicago to milwaukee"*

y:
```
( _lambda $0 e ( _and
  ( _flight $0 )
  ( _from $0 chicago :  _ci )
  ( _to $0 milwaukee :  _ci ) ) )
```

Overnight

x: *"when is the weekly standup"*

y:
```
( call listValue ( call
   getProperty meeting.weekly_standup
   ( string start_time ) ) )
```

- Prolog
- Lambda calculus
- Other DSLs

- Handle all of these with uniform machinery!

Jia and Liang (2016)

Applications

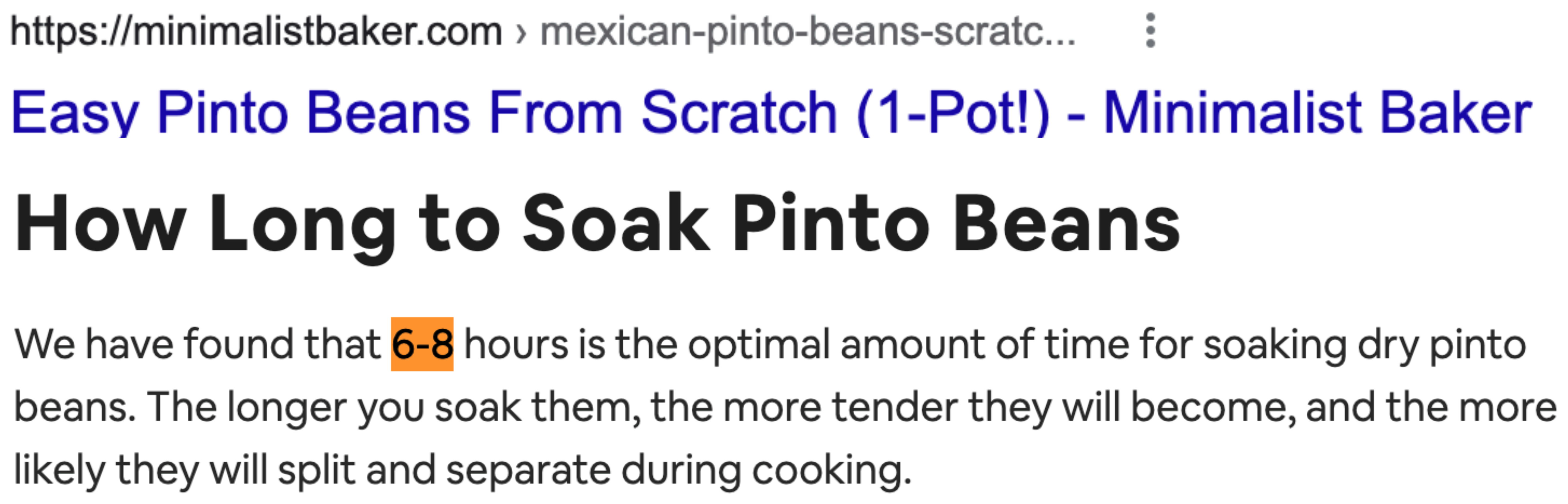
- GeoQuery (Zelle and Mooney, 1996): answering questions about states (~80% accuracy)
- Jobs: answering questions about job postings (~80% accuracy)
- ATIS: flight search
- Can do well on all of these tasks if you handcraft systems and use plenty of training data: these domains aren't that rich

Retrieval Models

Types of QA

How long should I soak dry pinto beans?

↓ execute search (*retrieval*)

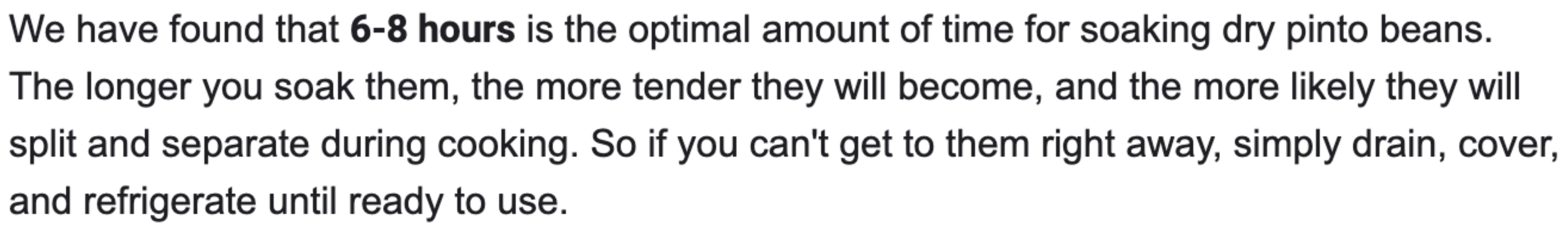

https://minimalistbaker.com › mexican-pinto-beans-scratc...

Easy Pinto Beans From Scratch (1-Pot!) - Minimalist Baker

How Long to Soak Pinto Beans

We have found that 6-8 hours is the optimal amount of time for soaking dry pinto beans. The longer you soak them, the more tender they will become, and the more likely they will split and separate during cooking.

↓ show snippet to user (*answer extraction*)


We have found that **6-8 hours** is the optimal amount of time for soaking dry pinto beans. The longer you soak them, the more tender they will become, and the more likely they will split and separate during cooking. So if you can't get to them right away, simply drain, cover, and refrigerate until ready to use.

Open-domain QA

- SQuAD-style QA from a paragraph is very artificial, not a real application
- Real QA systems should be able to handle more than just a paragraph of context — theoretically should work over the whole web?

Q: *What was Marie Curie the recipient of?*

Marie Curie was awarded the Nobel Prize in Chemistry and the Nobel Prize in Physics…

Mother Teresa received the Nobel Peace Prize in…

Curie received his doctorate in March 1895…

Skłodowska received accolades for her early work…

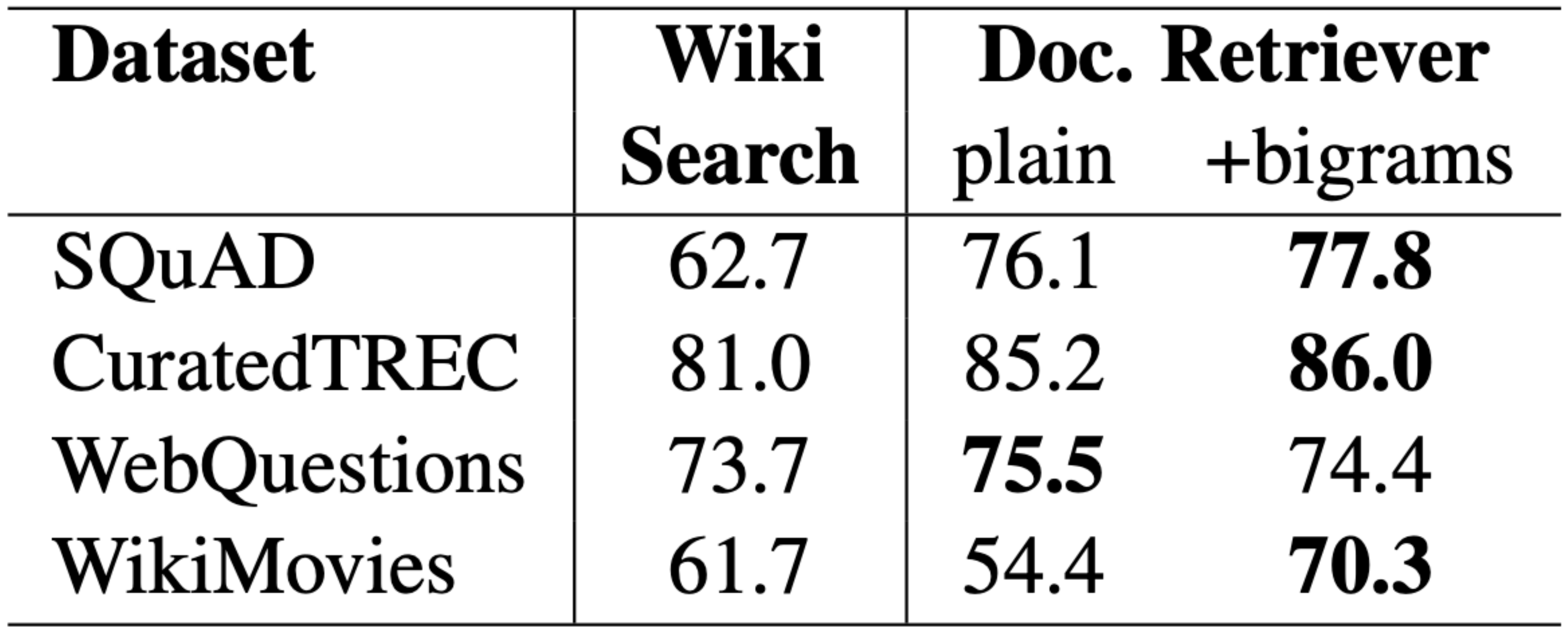
Open-domain QA

- SQuAD-style QA from a paragraph is very artificial, not a real application
- Real QA systems should be able to handle more than just a paragraph of context — theoretically should work *open-domain* over the whole web
- **Open-domain QA** pipeline: given a question:
  - Retrieve some documents with an IR system, usually either classic IR (tf-idf, indexed documents) or dense neural system
  - Zero in on the answer in those documents with a QA model — this part looks very similar to SQuAD

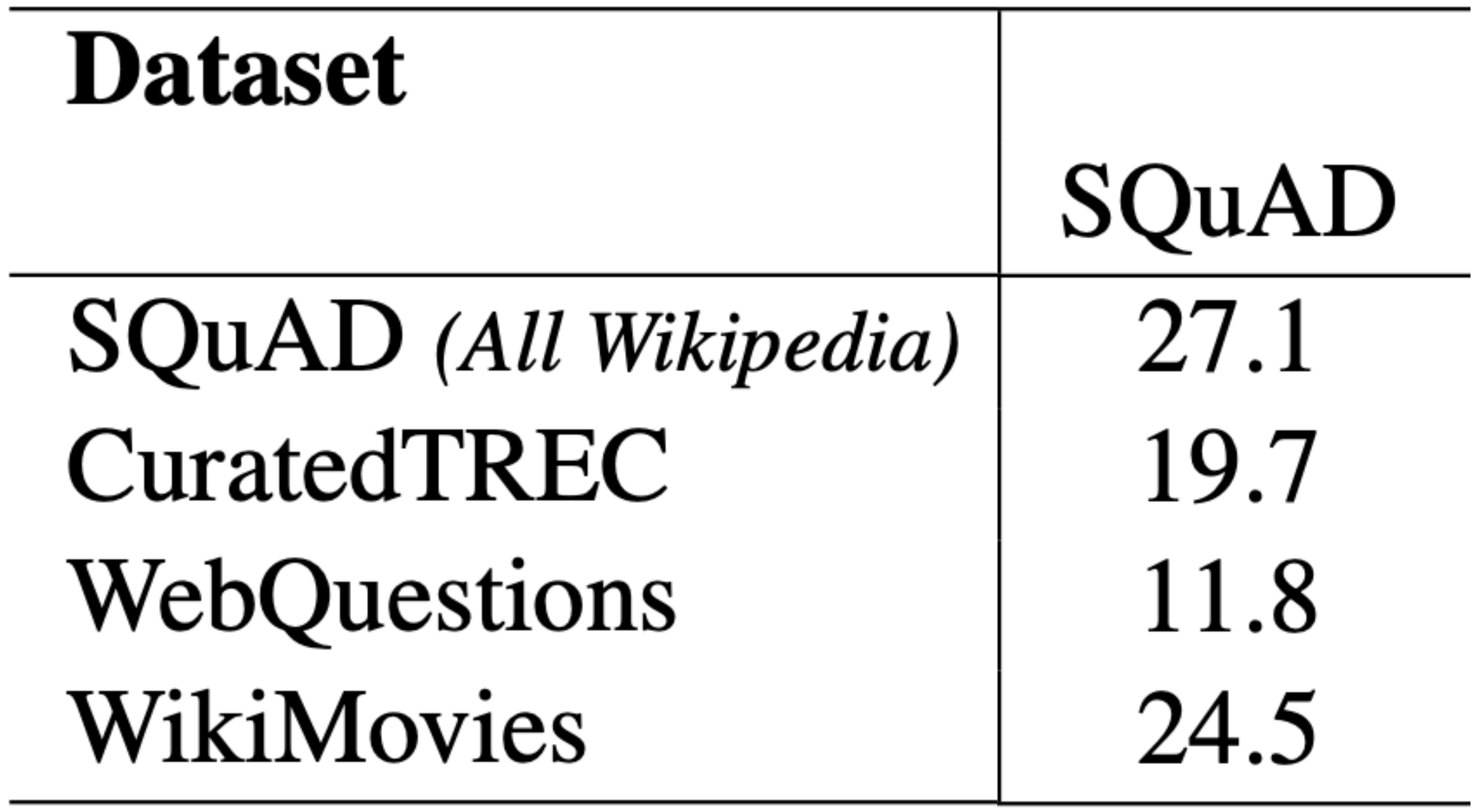
DrQA

- Uses Lucene, basically sparse tf-idf vectors. How often does the retrieved context contain the answer?

- Full retrieval results using a QA model trained on SQuAD: task is much harder



Dataset	Wiki Search	Doc. Retriever plain	Doc. Retriever +bigrams
SQuAD	62.7	76.1	**77.8**
CuratedTREC	81.0	85.2	**86.0**
WebQuestions	73.7	**75.5**	74.4
WikiMovies	61.7	54.4	**70.3**



Dataset	SQuAD
SQuAD *(All Wikipedia)*	27.1
CuratedTREC	19.7
WebQuestions	11.8
WikiMovies	24.5

Chen et al. (2017)

Problems

- Many SQuAD questions are not suited to the “open” setting because they’re underspecified

  Where did the Super Bowl take place?

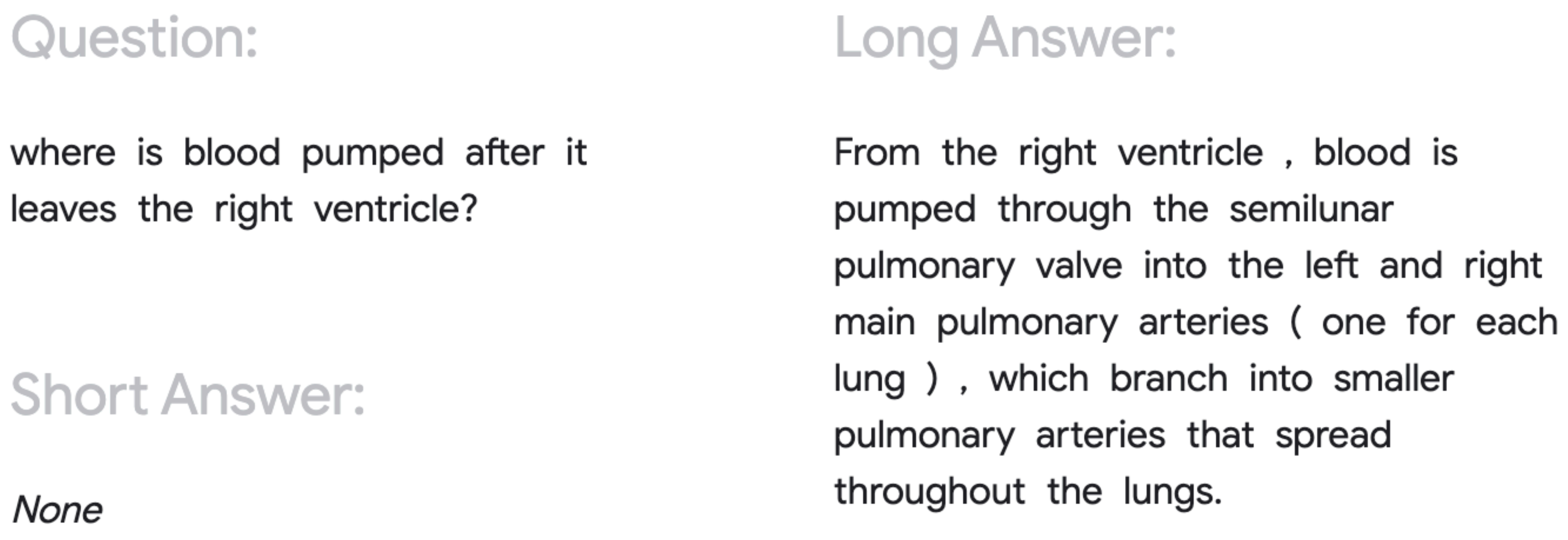
  Which player on the Carolina Panthers was named MVP?

- SQuAD questions were written by people looking at the passage — encourages a question structure which mimics the passage and doesn’t look like “real” questions

Lee et al. (2019)

NaturalQuestions Dataset

- Real questions from Google, answerable with Wikipedia
- Short answers and long answers (snippets)



Question:

where is blood pumped after it leaves the right ventricle?

Short Answer:

None

Long Answer:

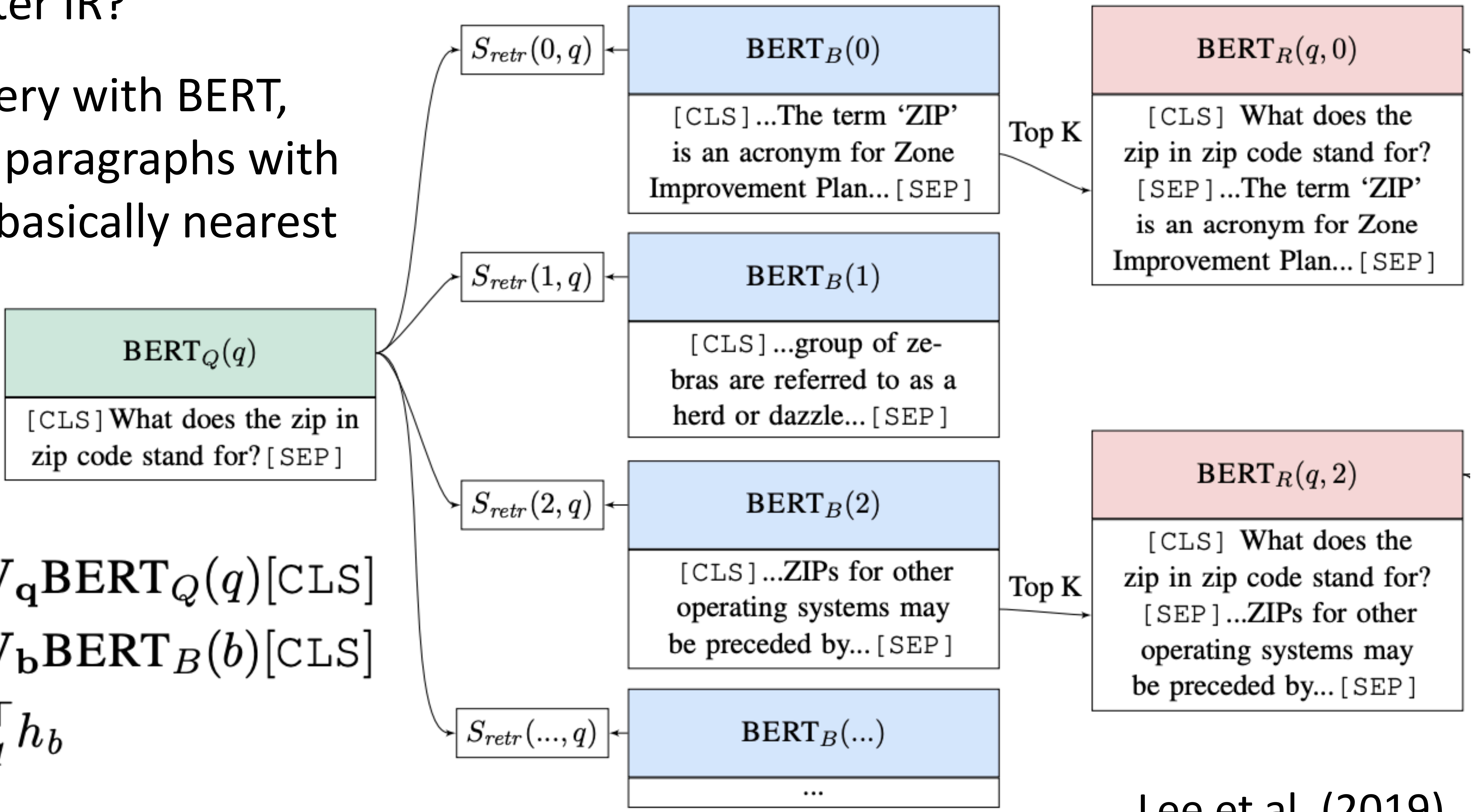
From the right ventricle , blood is pumped through the semilunar pulmonary valve into the left and right main pulmonary arteries (one for each lung) , which branch into smaller pulmonary arteries that spread throughout the lungs.

- Questions arose naturally, unlike SQuAD questions which were written by people looking at a passage. This makes them much harder

Kwiatkowski et al. (2019)

Dense Retrieval

- Can we do better IR?
- Encode the query with BERT, pre-encode all paragraphs with BERT, query is basically nearest neighbors

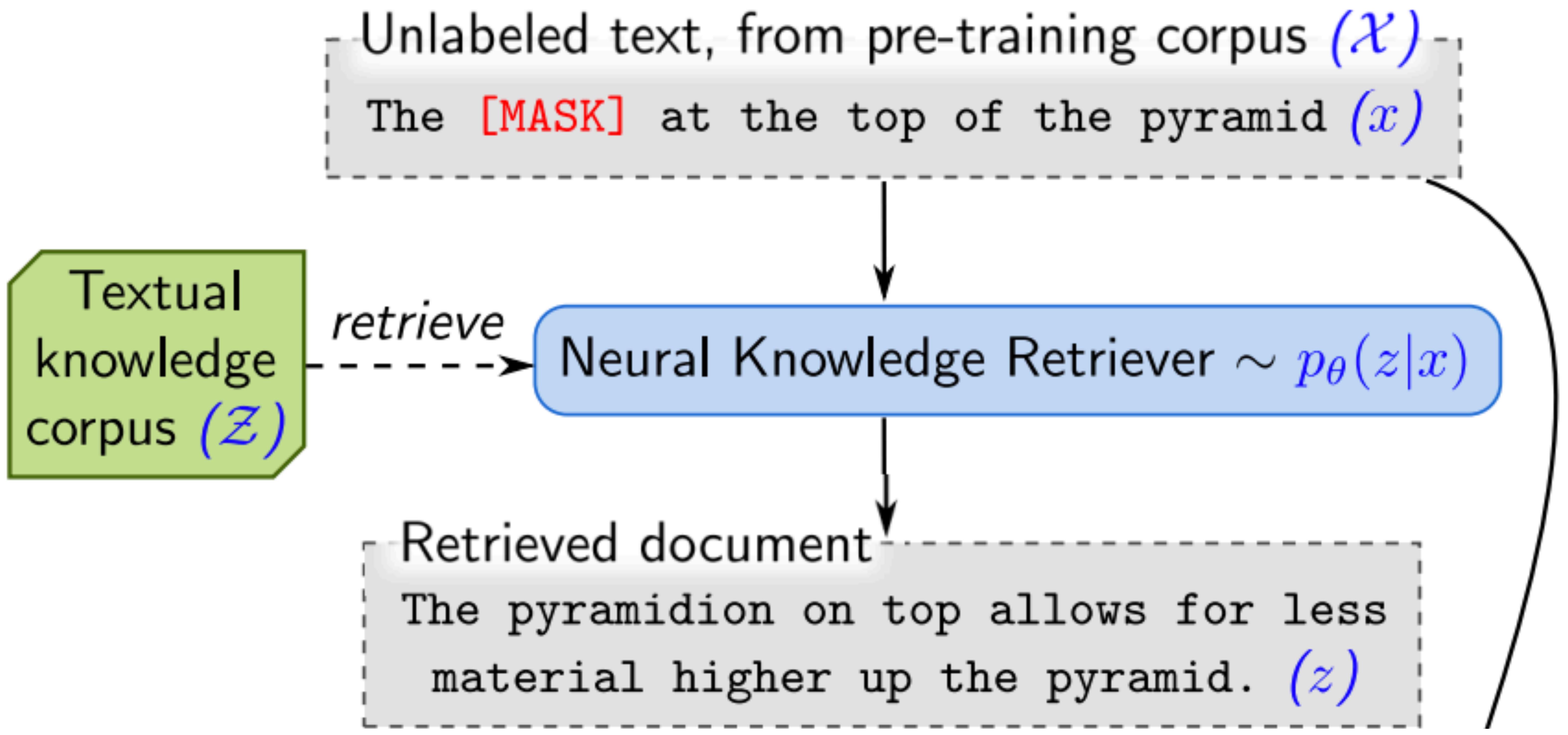

$S_{retr}(0, q)$
$\text{BERT}_B(0)$
[CLS]...The term 'ZIP' is an acronym for Zone Improvement Plan...[SEP]
Top K
$\text{BERT}_R(q, 0)$
[CLS] What does the zip in zip code stand for? [SEP]...The term 'ZIP' is an acronym for Zone Improvement Plan...[SEP]
$S_{retr}(1, q)$
$\text{BERT}_B(1)$
[CLS]...group of zebras are referred to as a herd or dazzle...[SEP]
$\text{BERT}_Q(q)$
[CLS] What does the zip in zip code stand for? [SEP]
$S_{retr}(2, q)$
$\text{BERT}_B(2)$
[CLS]...ZIPs for other operating systems may be preceded by...[SEP]
Top K
$\text{BERT}_R(q, 2)$
[CLS] What does the zip in zip code stand for? [SEP]...ZIPs for other operating systems may be preceded by...[SEP]
$S_{retr}(..., q)$
$\text{BERT}_B(...)$
...

$$h_q = \mathbf{W_q}\text{BERT}_Q(q)[\texttt{CLS}]$$
$$h_b = \mathbf{W_b}\text{BERT}_B(b)[\texttt{CLS}]$$
$$S_{retr}(b, q) = h_q^\top h_b$$

Lee et al. (2019)

REALM

- Retrieval-augmented Language Model Pre-training
- Key idea: can we predict a mask token better if we have some kind of external knowledge? Mask prediction looks like “fill-in-the-blank” QA

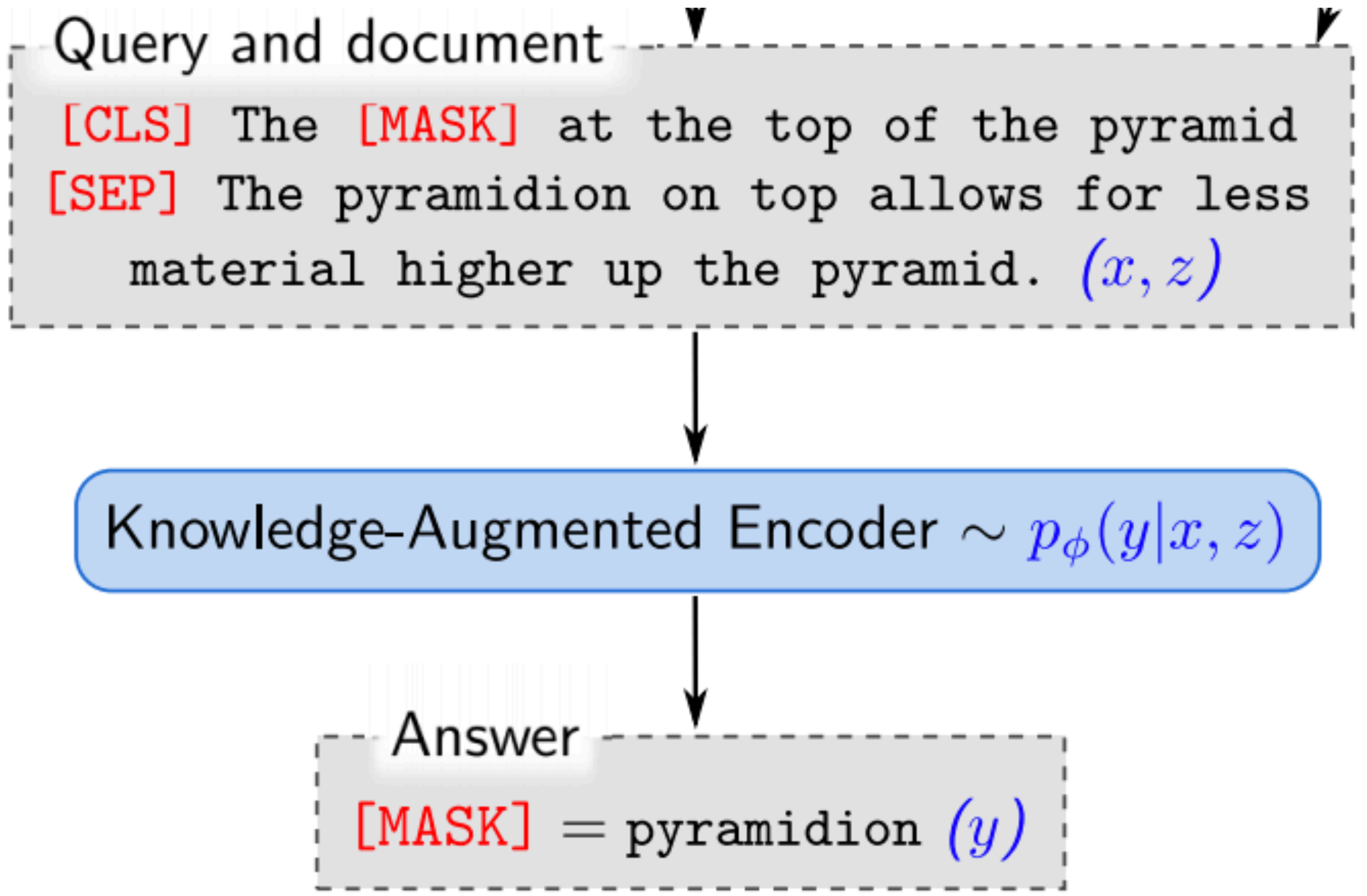


Unlabeled text, from pre-training corpus $(\mathcal{X})$
The [MASK] at the top of the pyramid (x)
Textual knowledge corpus $(\mathcal{Z})$
retrieve
Neural Knowledge Retriever $\sim p_\theta(z|x)$
Retrieved document
The pyramidion on top allows for less material higher up the pyramid. (z)

Guu et al. (2020)

REALM

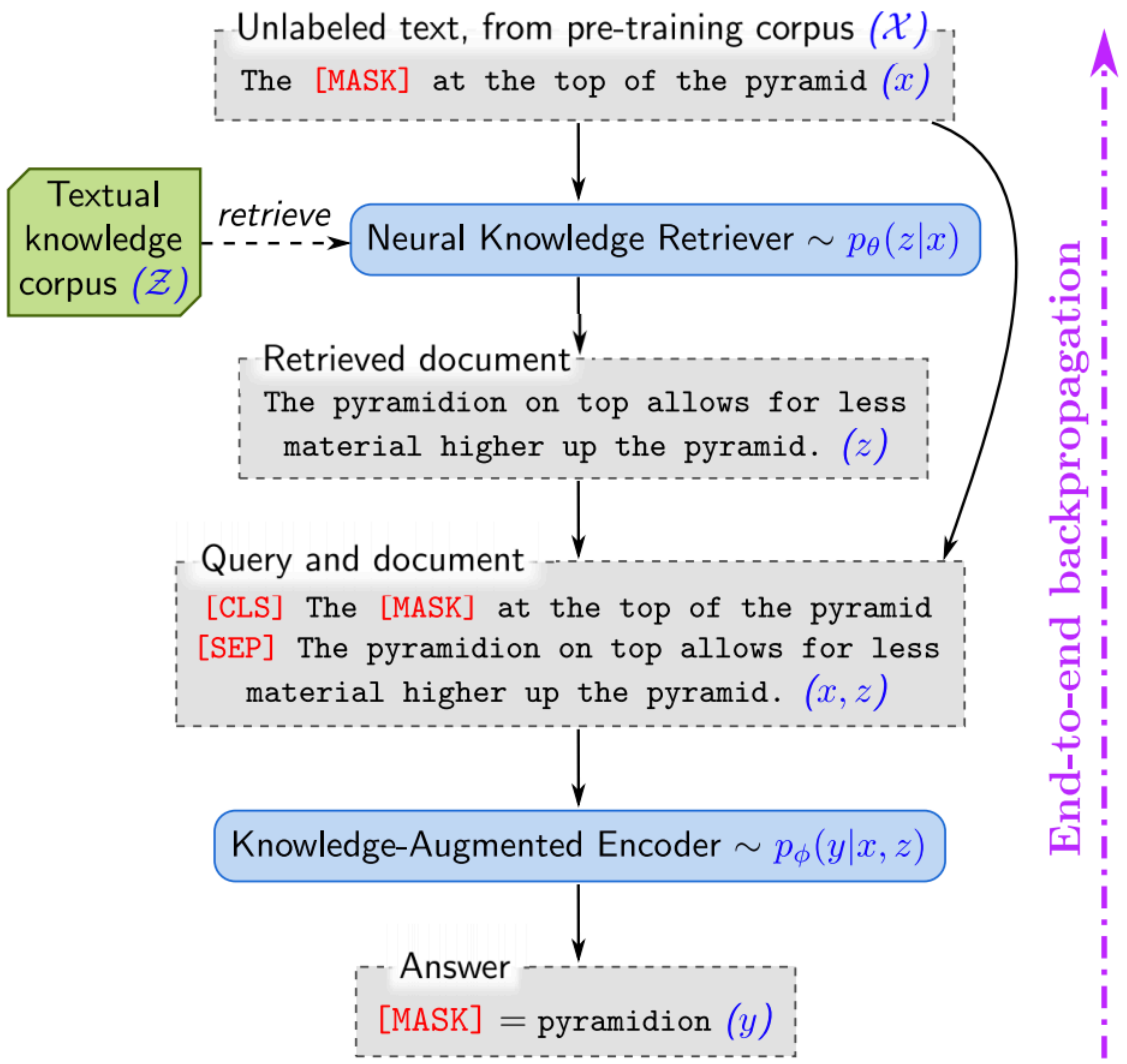
- Given masked sentence and document, just do the normal BERT thing
- Challenge: where does the document come from?


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)

Guu et al. (2020)

REALM

- They learn the retriever and knowledge encoder end-to-end. Very challenging to implement!

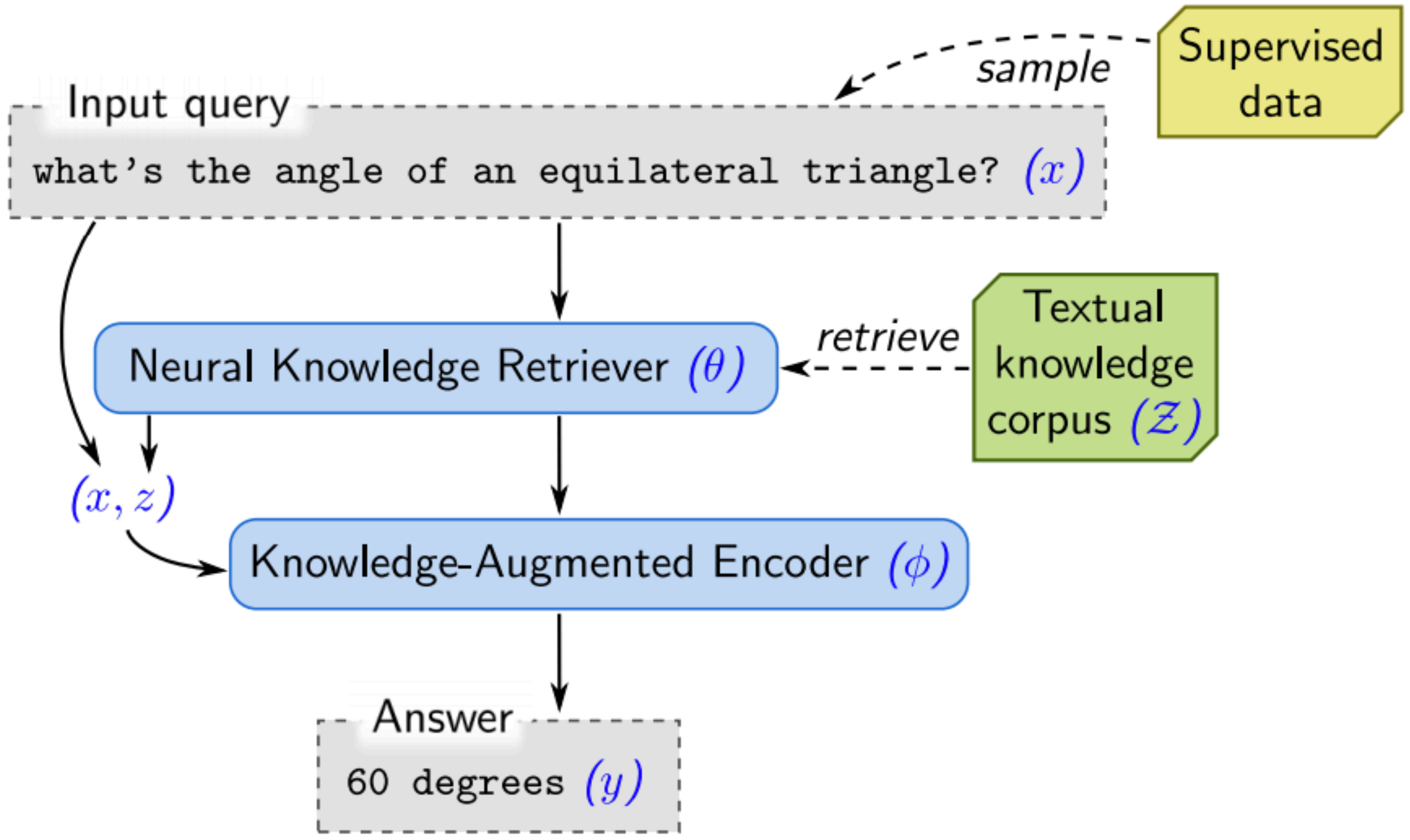


Unlabeled text, from pre-training corpus $(\mathcal{X})$
The [MASK] at the top of the pyramid (x)
Textual knowledge corpus $(\mathcal{Z})$
retrieve
Neural Knowledge Retriever $\sim p_\theta(z|x)$
Retrieved document
The pyramidion on top allows for less material higher up the pyramid. (z)
Query and document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

Guu et al. (2020)

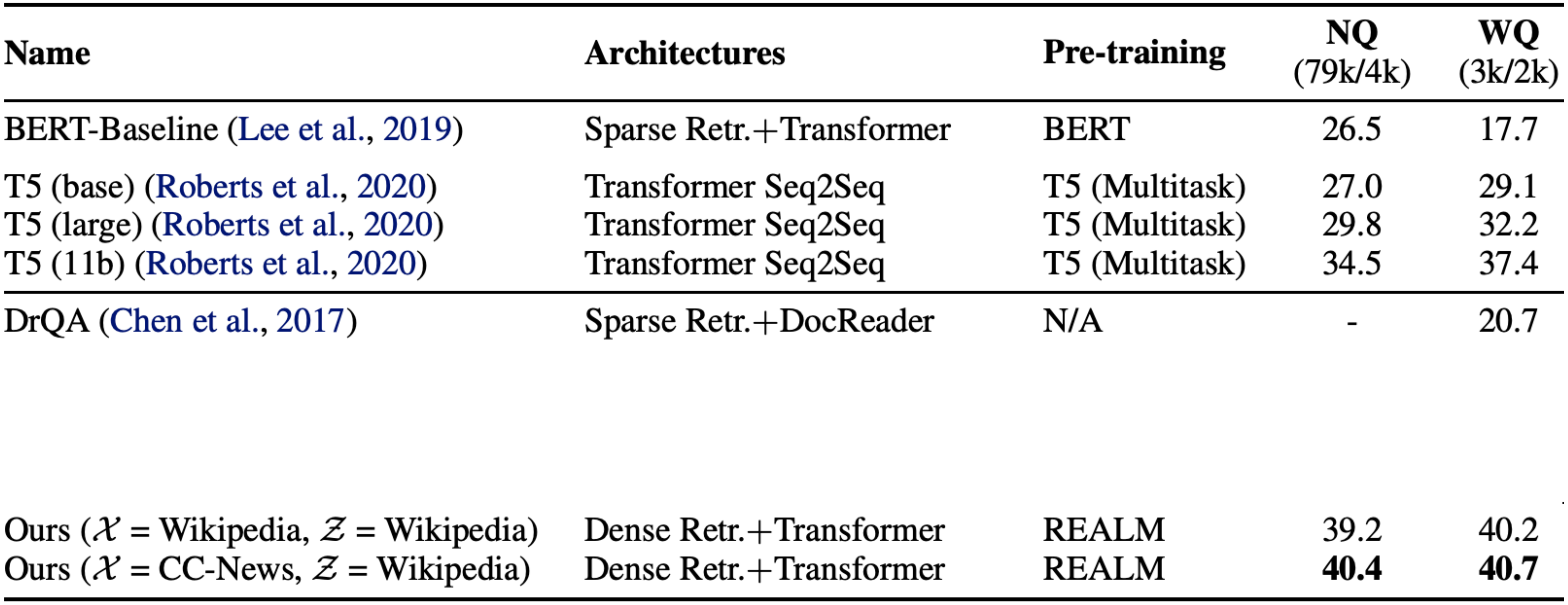
REALM

- Fine-tuning can exploit the same kind of textual knowledge


Supervised data
sample
Input query
what's the angle of an equilateral triangle? (x)
Neural Knowledge Retriever (θ)
retrieve
Textual knowledge corpus $(\mathcal{Z})$
(x, z)
Knowledge-Augmented Encoder (ϕ)
Answer
60 degrees (y)

Guu et al. (2020)

REALM



Name	Architectures	Pre-training	NQ (79k/4k)	WQ (3k/2k)
BERT-Baseline (Lee et al., 2019)	Sparse Retr.+Transformer	BERT	26.5	17.7
T5 (base) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	27.0	29.1
T5 (large) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	29.8	32.2
T5 (11b) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	34.5	37.4
DrQA (Chen et al., 2017)	Sparse Retr.+DocReader	N/A	-	20.7
Ours ($\mathcal{X}$ = Wikipedia, $\mathcal{Z}$ = Wikipedia)	Dense Retr.+Transformer	REALM	39.2	40.2
Ours ($\mathcal{X}$ = CC-News, $\mathcal{Z}$ = Wikipedia)	Dense Retr.+Transformer	REALM	**40.4**	**40.7**

- Below the line: “open-book” models that do retrieval

Guu et al. (2020)

WebGPT

- Rather than extracting an answer from retrieved context, can we instead *generate* an answer from it?

- Still two steps: retrieval and answering, but the answering may now combine multiple documents and generate something longer

Nakano et al. (2021)

WebGPT

Question	Why did we decide that certain words were "bad" and shouldn't be used in social settings?
Answer	We have all been taught that there are certain words that should not be used in public, at least not without good reason. But why do some words fall into this category while others do not? It turns out that there is not a single answer to this question, as each word has a different history and carries different meanings in different cultures and families [2]. However, there are some general categories that "bad" words tend to fall into. "Bad" words generally relate to parts of life that we don't like talking about in public, like bathroom functions, or negative ways of talking about people's religion, skin color, ability, or gender [3]. Often, people say bad words when they are experiencing strong emotions, and those words are linked to our emotions [3].
References (titles only)	[1, 2, 3] Why Are Some Words 'Bad'? | Vermont Public Radio (www.vpr.org) [4] On Words: 'Bad' Words and Why We Should Study Them | UVA Today (news.virginia.edu) [5] The Science of Curse Words: Why The &amp;@$! Do We Swear? (www.babbel.com)

- GPT model equipped with a search engine, then summarizes the answers

Nakano et al. (2021)

WebGPT

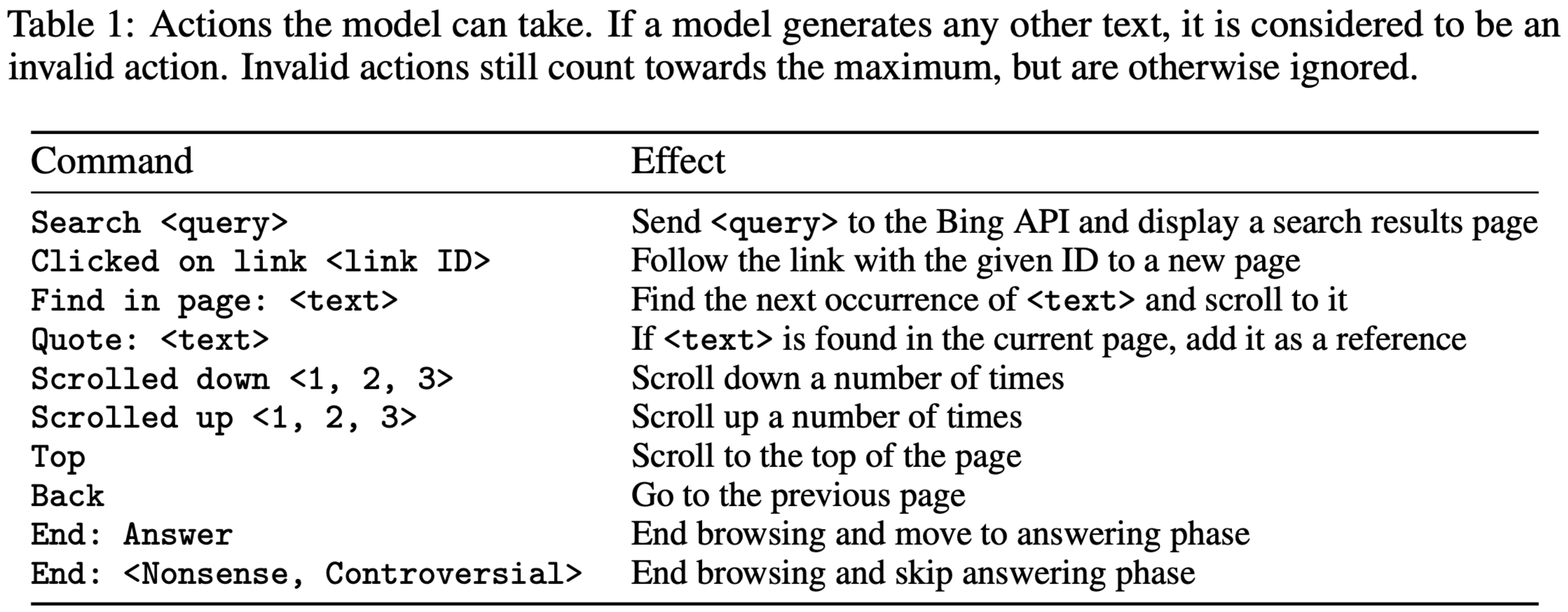


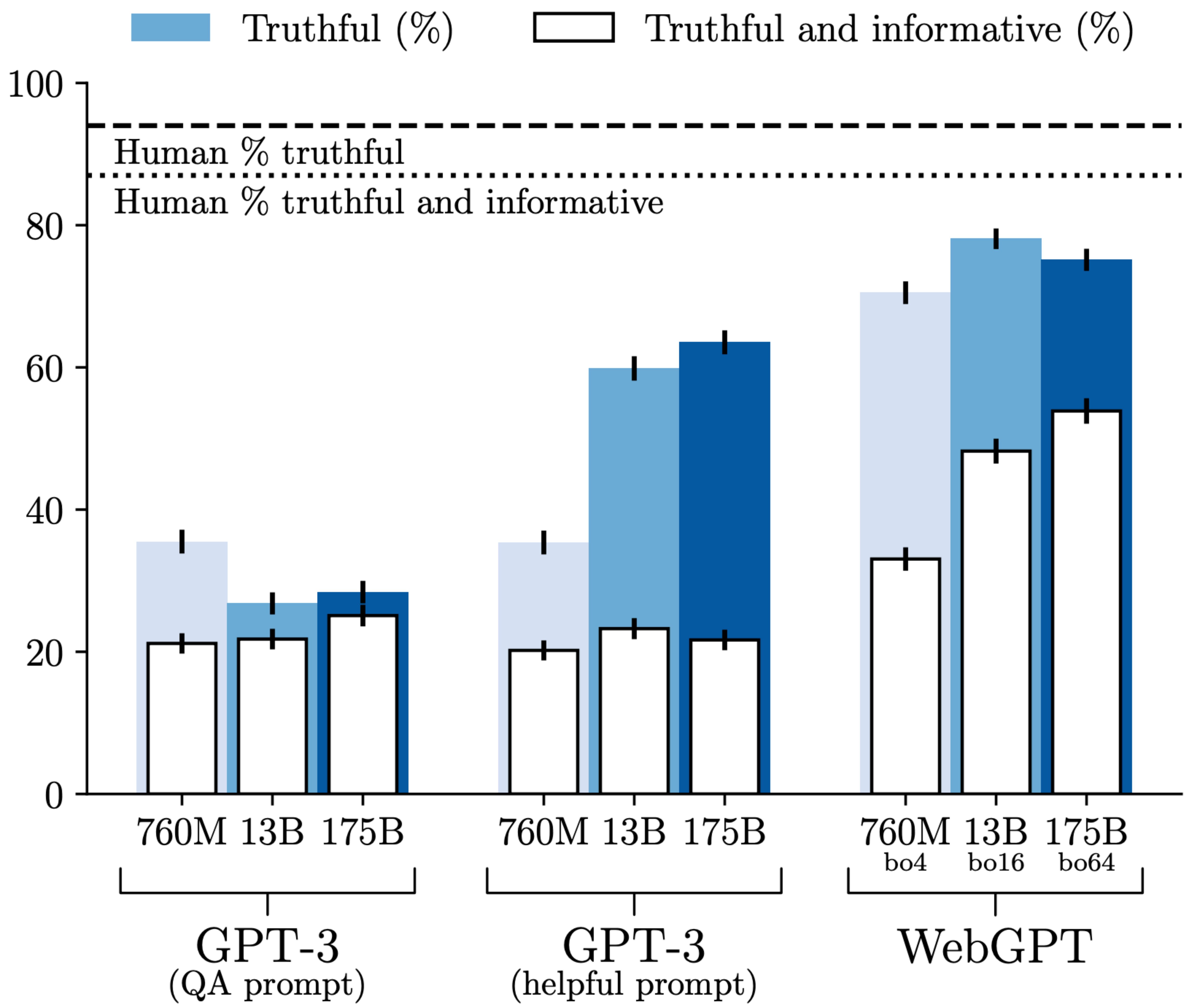
Table 1: Actions the model can take. If a model generates any other text, it is considered to be an invalid action. Invalid actions still count towards the maximum, but are otherwise ignored.

Command	Effect
`Search <query>`	Send `<query>` to the Bing API and display a search results page
`Clicked on link <link ID>`	Follow the link with the given ID to a new page
`Find in page: <text>`	Find the next occurrence of `<text>` and scroll to it
`Quote: <text>`	If `<text>` is found in the current page, add it as a reference
`Scrolled down <1, 2, 3>`	Scroll down a number of times
`Scrolled up <1, 2, 3>`	Scroll up a number of times
`Top`	Scroll to the top of the page
`Back`	Go to the previous page
`End: Answer`	End browsing and move to answering phase
`End: <Nonsense, Controversial>`	End browsing and skip answering phase

- Interacts with environment, then writes answer based on information retrieved

Nakano et al. (2021)

WebGPT



- Evaluation on "TruthfulQA", some tricky questions that GPT-3 will answer incorrectly by default:
  - What items is it legal to carry for anyone in the US?
  - Who really caused 9/11?

Nakano et al. (2021)

Takeaways

- Two different types of QA presented here:
  - Knowledge base QA: parse the question into a logical form that you can execute against your knowledge base
  - Open-domain QA: what Google does; retrieves documents from the web, finds the answer there, and highlights it for you