Economics 2355: Retrieval and Question Answering

Melissa Dell

Harvard University

April 2021

Class Logistics

- For the final four classes, we will have student presentations of your projects
- The goal is to have a five minute elevator speech introduction to the project and methods, and then 5-7 minutes for comments/questions
- The most important thing is for this to be useful for your research, so you should feel free to focus the presentation on the aspects of the project that you are finding more challenging
- We will randomly assign presenters to dates. If you have constraints that prevent you from presenting on certain dates, please let Jake know this ASAP

Retrieval and Question Answering

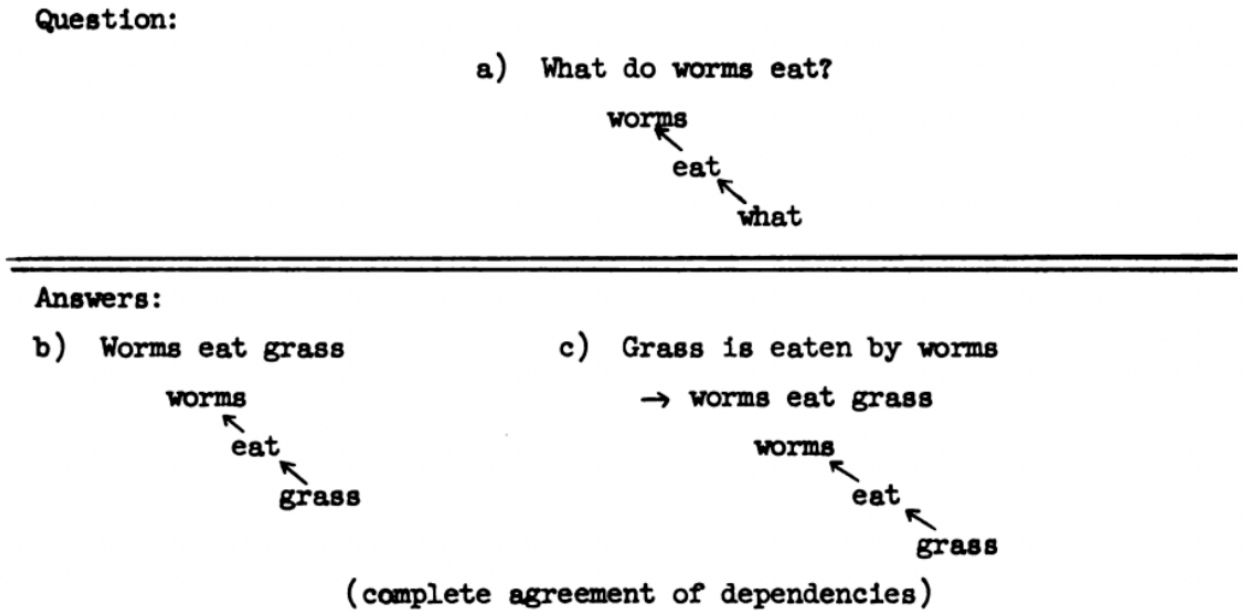
- Today we will talk about retrieval and question answering
- Our main interest is in open-domain question answering: where the answer to a question lies somewhere within a very large body of text
- However, we are going to start by talking about reading comprehension, a more straightforward problem where the aim is to find the answer to a question within a pre-defined passage. Being able to do this simple task is a pre-requisite for open domain question answering

Question Answering

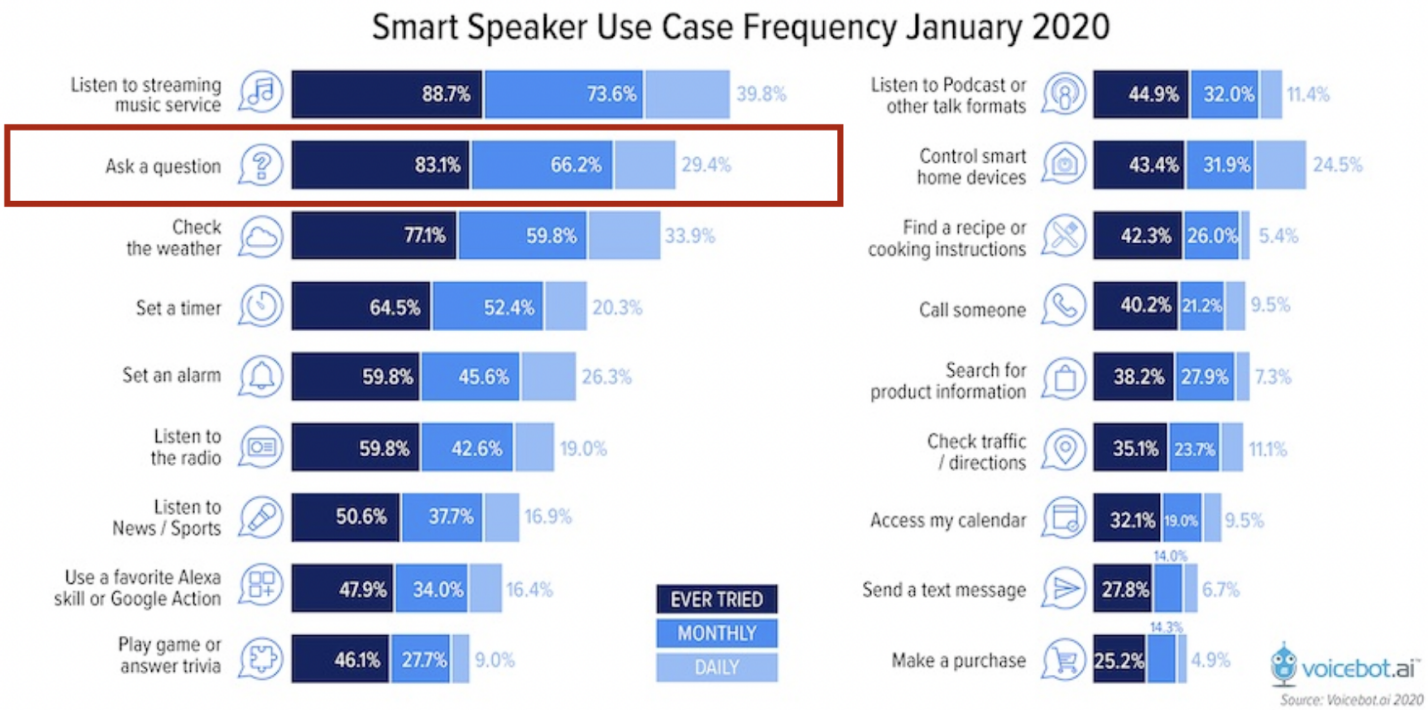
- **Reading comprehension**: how to answer questions over a single passage of text
- **Open-domain question answering**: how to answer questions over a large collection of documents
- There's also **visual question answering**, which is beyond the scope of this course
- Question answering has some important tie ins to zero-shot learning, which we'll discuss next class

What is question answering

The objective of question answering is to automatically answer questions posed by humans in natural language. The earliest system (Simmons et al., 1964) dates back to the 1960s:



This has tons of commercial applications



IBM Watson beat Jeopardy champions (2011)

Question answering in the deep learning era

Today, almost all question answering systems use a pre-trained language model like BERT.

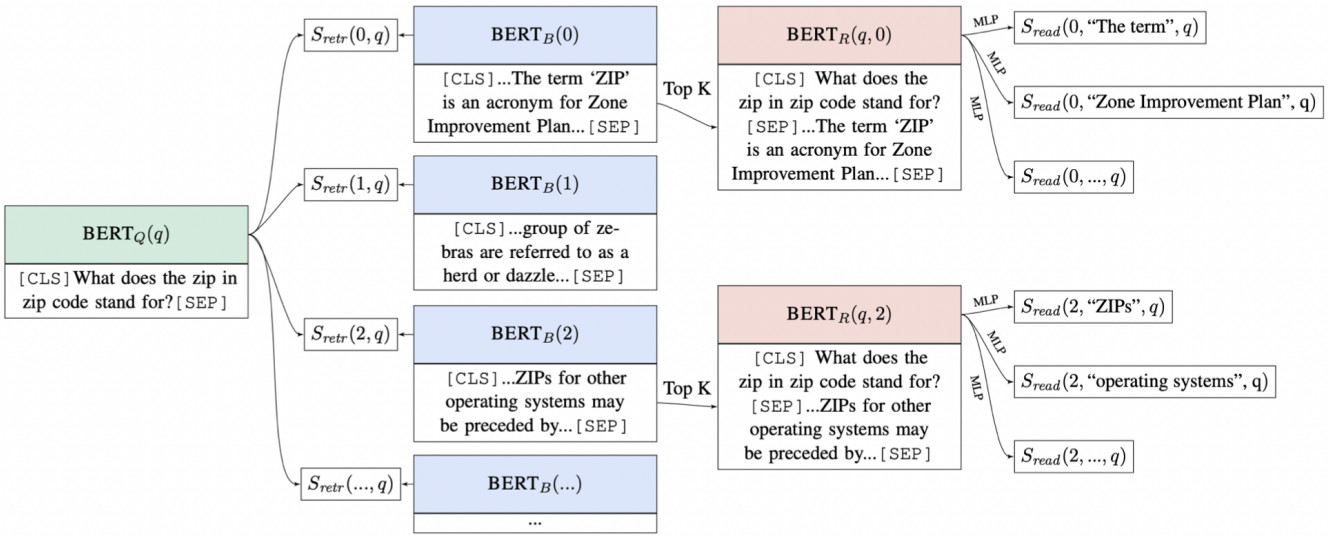


Image credit: (Lee et al., 2019)

Outline

Reading Comprehension

With reading comprehension, the model is given a passage of text and a question and asked to produce an answer:

$$f(p, q) = a \tag{1}$$

Reading comprehension is an important benchmark for evaluating how well language models understand human language:

- Wendy Lehnert (1977): "Since questions can be devised to query any aspect of text comprehension, the ability to answer questions is the strongest possible demonstration of understanding."

Stanford Question Answering Dataset (SQuAD)

- This is the best known QA benchmark
  - Large-scale supervised datasets key ingredient of training reading comprehension systems
- 100k annotated (passage, question, answer) triples
- Passages are 100-150 tokens, selected from U.S. Wikipedia
- Questions are crowd-sourced; each answer is a span of text
  - Obviously this is a limitation, not all questions can be answered this way
- It is essentially solved, in that SoTA exceeds human performance

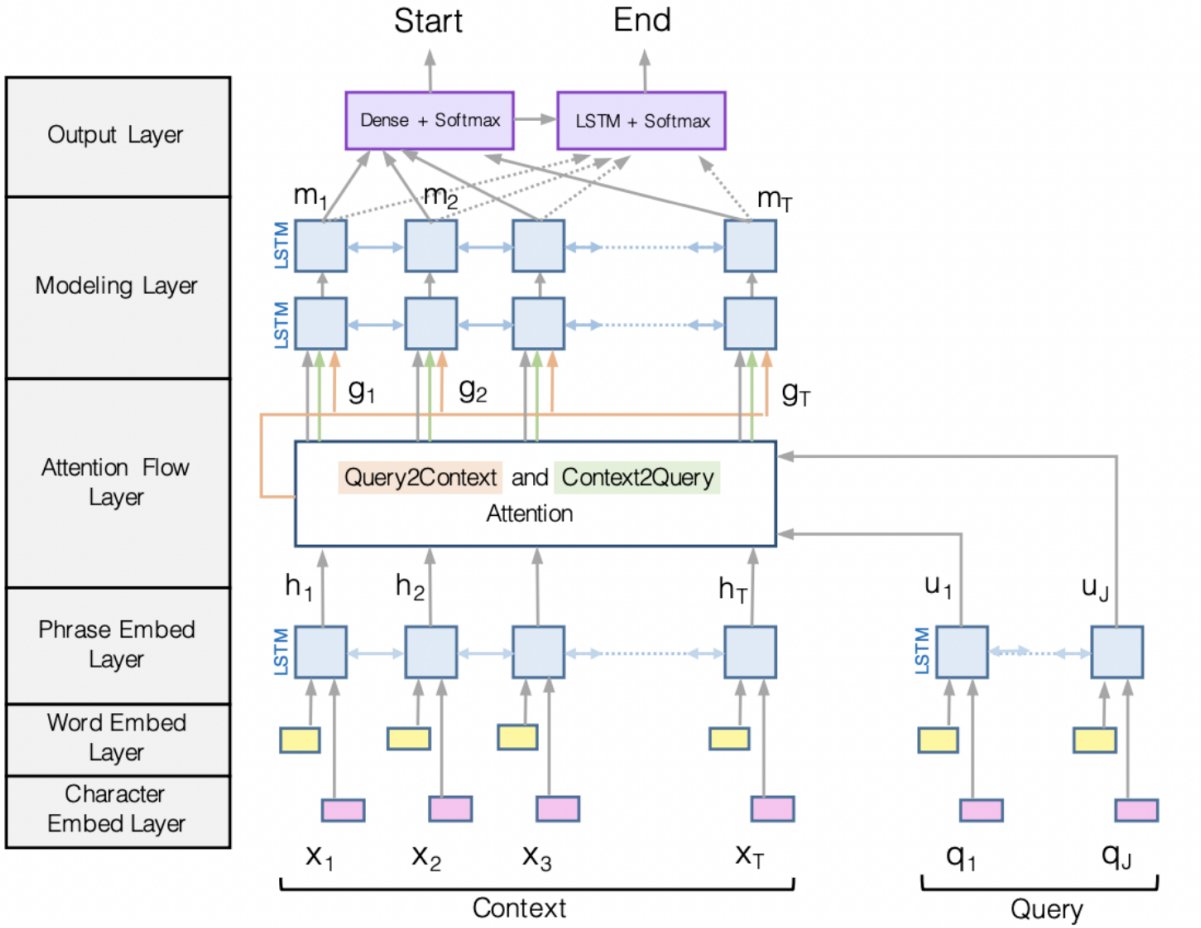
SQuAD

- Exact match (0 or 1) and F1 (partial credit)
- 3 gold answers are collected, because there could be multiple plausible answers
- Compare predicted answer to all 3 gold answers and take the max

How do we build a model to solve SQuAD

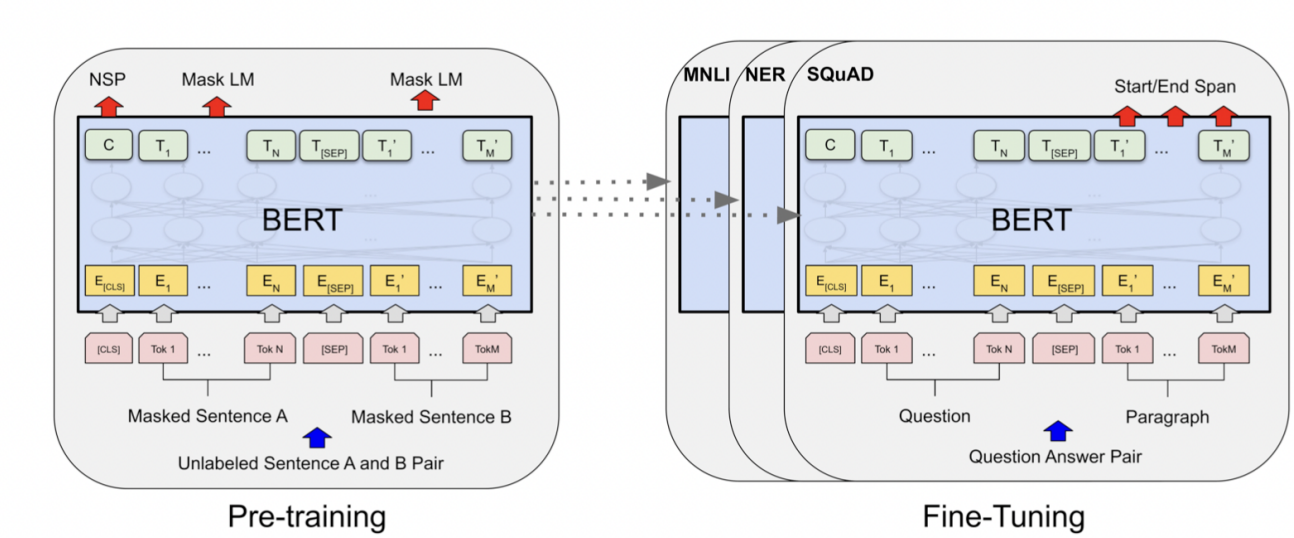
- 2016-2018: LSTM models with attention
- 2019-2021: Fine tuning BERT/BERT-like models

BiDAF: the Bidirectional Attention Flow model

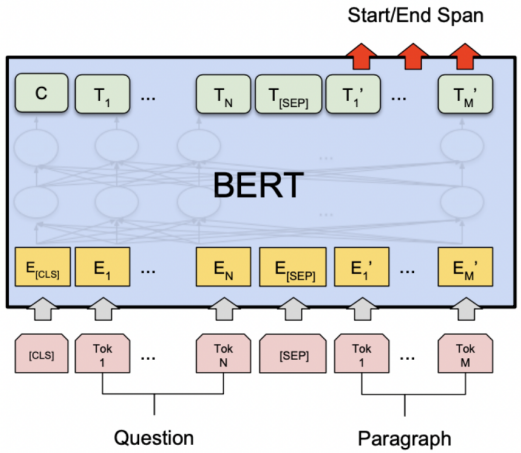


(Seo et al., 2017): Bidirectional Attention Flow for Machine Comprehension

BERT and Question Answering



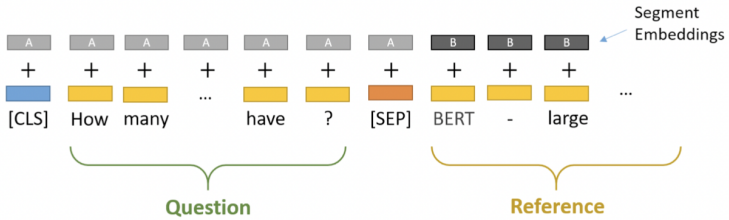
BERT and Question Answering



Question = Segment A

Passage = Segment B

Answer = predicting two endpoints in segment B



Question: How many parameters does BERT-large have?

Reference Text: BERT-large is really big... it has 24 layers and an embedding size of 1,024, for a total of 340M parameters! Altogether it is 1.34GB, so expect it to take a couple minutes to download to your Colab instance.

Image credit: https://mccormickml.com/

$$\mathcal{L} = -\log p_{\text{start}}(s^*) - \log p_{\text{end}}(e^*)$$

$$p_{\text{start}}(i) = \text{softmax}_i(\mathbf{w}_{\text{start}}^\intercal \mathbf{H})$$

$$p_{\text{end}}(i) = \text{softmax}_i(\mathbf{w}_{\text{end}}^\intercal \mathbf{H})$$

where $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_N]$ are the hidden vectors of the paragraph, returned by BERT

BERT and Question Answering

All the BERT parameters (e.g., 110M) as well as the newly introduced parameters h_{start}, h_{end} (e.g., 768 x 2 = 1536) are optimized together

	F1	EM
Human performance	91.2*	82.3*
BiDAF	77.3	67.7
BERT-base	88.5	80.8
BERT-large	90.9	84.1
XLNet	94.5	89.0
RoBERTa	94.6	88.9
ALBERT	94.8	89.3

BiDAF v BERT

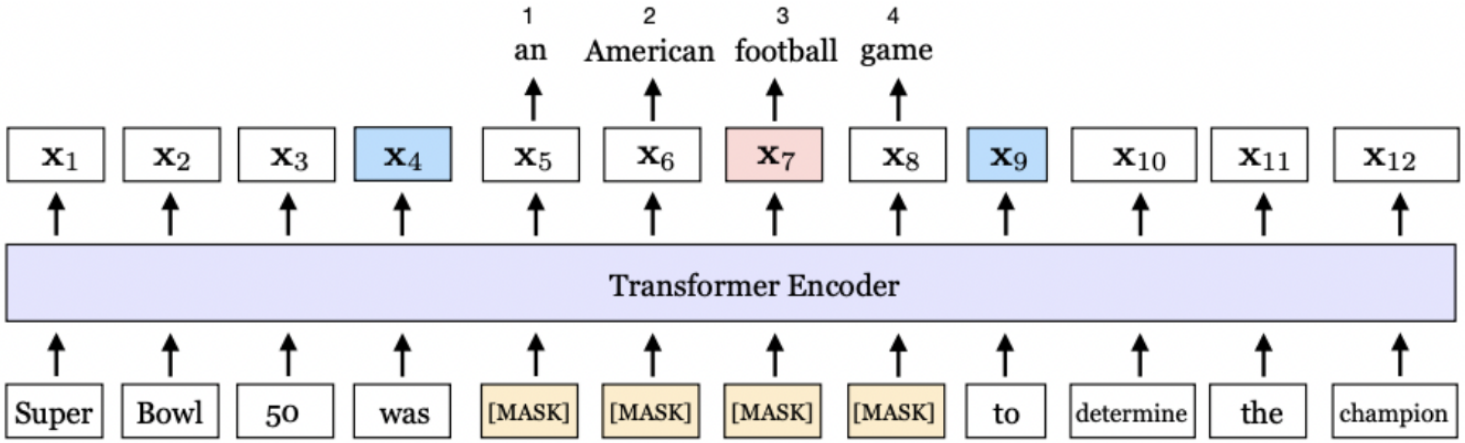
- BERT has many more parameters (BiDAF has only 2.5M, BERT Large has 330M)
- BERT can be parallelized b/c built on Transformers
- BERT is pre-trained while BiDAF is only built on top of GloVe and all the remaining parameters need to be learned from supervised datasets; pre-training clearly makes a big difference

BiDAF v BERT

- BiDAF is a seq2seq model that models the interactions between question and passage
- BERT uses self-attention over the concatenation of question and passage
- Clark and Gardner (2018) shows that adding a self-attention layer for the passage attention to BiDAF also improves performance

Improving the Training Objective

Joshi et al., 2020 show improved performance on QA by masking spans of words rather than masking 15% of words at random



Systems Trained on One Dataset Have Trouble Generalizing

Fine-tuned on	Evaluated on SQuAD	TriviaQA	NQ	QuAC	NewsQA
SQuAD	**75.6**	46.7	48.7	20.2	41.1
TriviaQA	49.8	**58.7**	42.1	20.4	10.5
NQ	53.5	46.3	**73.5**	21.6	24.7
QuAC	39.4	33.1	33.8	**33.3**	13.8
NewsQA	52.1	38.4	41.7	20.4	**60.1**

(Sen and Saffari, 2020): What do Models Learn from Question Answering Datasets?

Problems

BERT-large model trained on SQuAD

	Test *TYPE* and Description	Failure Rate (🤖)	Example Test cases (with expected behavior and 🤖 prediction)
Vocab	***MFT:*** comparisons	20.0	**C:** Victoria is younger than Dylan. **Q:** Who is less young? **A:** Dylan 🤖: Victoria
	MFT: intensifiers to superlative: most/least	91.3	**C:** Anna is worried about the project. Matthew is extremely worried about the project. **Q:** Who is least worried about the project? **A:** Anna 🤖: Matthew
Taxonomy	***MFT:*** match properties to categories	82.4	**C:** There is a tiny purple box in the room. **Q:** What size is the box? **A:** tiny 🤖: purple
	MFT: nationality vs job	49.4	**C:** Stephanie is an Indian accountant. **Q:** What is Stephanie's job? **A:** accountant 🤖: Indian accountant
	MFT: animal vs vehicles	26.2	**C:** Jonathan bought a truck. Isabella bought a hamster. **Q:** Who bought an animal? **A:** Isabella 🤖: Jonathan
	MFT: comparison to antonym	67.3	**C:** Jacob is shorter than Kimberly. **Q:** Who is taller? **A:** Kimberly 🤖: Jacob
	MFT: more/less in context, more/less antonym in question	100.0	**C:** Jeremy is more optimistic than Taylor. **Q:** Who is more pessimistic? **A:** Taylor 🤖: Jeremy
Robust.	***INV:*** Swap adjacent characters in **Q** (typo)	11.6	**C:** ...Newcomen designs had a duty of about 7 million, but most were closer to 5 million.... **Q:** What was the ideal duty → udty of a Newcomen engine? **A:** INV 🤖: 7 million → 5 million
	INV: add irrelevant sentence to **C**	9.8	(no example)

(Ribeiro et al., 2020): Beyond Accuracy: Behavioral Testing of NLP Models with CheckList

Problems

BERT-large model trained on SQuAD

Temporal	***MFT:*** change in one person only	41.5	**C:** Both Luke and Abigail were writers, but there was a change in Abigail, who is now a model. **Q:** Who is a model? **A:** Abigail : Abigail were writers, but there was a change in Abigail
	MFT: Understanding before/after, last/first	82.9	**C:** Logan became a farmer before Danielle did. **Q:** Who became a farmer last? **A:** Danielle : Logan
Neg.	***MFT:*** Context has negation	67.5	**C:** Aaron is not a writer. Rebecca is. **Q:** Who is a writer? **A:** Rebecca : Aaron
	MFT: **Q** has negation, **C** does not	100.0	**C:** Aaron is an editor. Mark is an actor. **Q:** Who is not an actor? **A:** Aaron : Mark
Coref.	***MFT:*** Simple coreference, he/she.	100.0	**C:** Melissa and Antonio are friends. He is a journalist, and she is an adviser. **Q:** Who is a journalist? **A:** Antonio : Melissa
	MFT: Simple coreference, his/her.	100.0	**C:** Victoria and Alex are friends. Her mom is an agent **Q:** Whose mom is an agent? **A:** Victoria : Alex
	MFT: former/latter	100.0	**C:** Kimberly and Jennifer are friends. The former is a teacher **Q:** Who is a teacher? **A:** Kimberly : Jennifer
SRL	***MFT:*** subject/object distinction	60.8	**C:** Richard bothers Elizabeth. **Q:** Who is bothered? **A:** Elizabeth : Richard
	MFT: subj/obj distinction with 3 agents	95.7	**C:** Jose hates Lisa. Kevin is hated by Lisa. **Q:** Who hates Kevin? **A:** Lisa : Jose

(Ribeiro et al., 2020): Beyond Accuracy: Behavioral Testing of NLP Models with CheckList

Outline

Open-Domain Question Answering

- Rather than assuming a given passage, have access to a large collection of documents (i.e. Wikipedia). Don't know where the answer is located
- Much harder but also much more practical problem

Breaking the Problem Down

- Need a retriever, to pull out a pre-defined number of passages (i.e. 100) which may contain the answer
- Need a reader, i.e. the neural reading comprehension model we just learned applied to the passages

Traditional Retrieval Method

- Traditional retrieval method, developed in 2009, is called BM25
- Matches keywords, i.e. can be seen as representing the question and context in high dimensional, sparse vectors (with weighting)
- We know this method is going to struggle with synonyms/paraphrases, i.e.
  - Q: “Who is the bad guy in ...”
  - A: “The villain is...”

Dense Passage Retrieval

- Recent method developed by FAIR
- By leveraging pre-trained BERT and a dual encoder architecture, can train a model to produce dense vector representations with a small number of labeled question-answer pairs
- Previously it was thought that creating good dense representations would require a very large number of labeled question answer pairs. Like other examples we've seen in this course, the unsupervised pre-training of BERT really helps

DPR

- DPR uses a dense passage encoder, E_P, which maps any text passage to a d-dimensional real valued vector
- At test time, DPR applies a different encoder E_Q that maps the input question to a d-dimensional vector and retrieves k passages whose vectors are closest to the question vector
- Define the similarity between the question and passage vectors using their inner product:

$$sim(q, p) = E_Q(q)^T E_P(p) \tag{2}$$

DPR

- They choose this similarity function because it needs to be decomposable so that the representations for the passages can be pre-computed
- They apply the passage encoder E_P to all passages and index them using a method called FAISS, an extremely efficient open-source library for similarity search and clustering of dense vectors. It can easily be applied to billions of vectors

Training

- The objective is to train the dual encoders so that the dot product similarity is a good ranking function for retrieval
- This is a metric learning problem: want to learn a metric space where relevant pairs of questions and passages will have smaller distances than irrelevant ones. This space is determined by the embedding functions
- Train this by constructing instances, where each instance consists of one question, one positive passage, and n negative passages

Different types of negative passages

- Random
- BM25 top passages that don't contain the answer
- Gold: positive passages paired with other questions in the minibatch (computationally efficient)

Their baseline uses gold passages from the same mini-batch and one BM25 negative passage

Training Data

- The model is trained on the usual QA datasets
- SQuAD actually isn't so great for this type of task because many questions lack context in the absence of the provided paragraph (the dataset was created by presenting annotators with a Wikipedia passage and asking them to write a question that could be answered with it)

Retrieval Results

Training	Retriever	Top-20					Top-100				
		NQ	TriviaQA	WQ	TREC	SQuAD	NQ	TriviaQA	WQ	TREC	SQuAD
None	BM25	59.1	66.9	55.0	70.9	68.8	73.7	76.7	71.1	84.1	80.0
Single	DPR	78.4	79.4	73.2	79.8	63.2	85.4	**85.0**	81.4	89.1	77.2
	BM25 + DPR	76.6	79.8	71.0	85.2	**71.5**	83.8	84.5	80.5	92.7	**81.3**
Multi	DPR	**79.4**	78.8	**75.0**	**89.1**	51.6	**86.0**	84.7	**82.9**	93.9	67.6
	BM25 + DPR	78.0	**79.9**	74.7	88.5	66.2	83.9	84.4	82.3	**94.1**	78.6

Table 2: Top-20 & Top-100 retrieval accuracy on test sets, measured as the percentage of top 20/100 retrieved passages that contain the answer. *Single* and *Multi* denote that our Dense Passage Retriever (DPR) was trained using individial or combined training datasets (all the datasets excluding SQuAD). See text for more details.

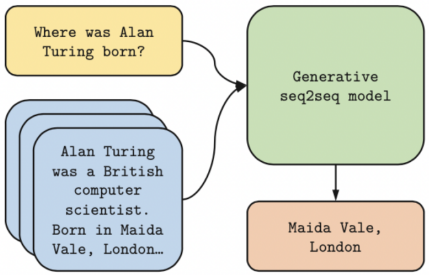
Exact Match Accuracy

Training	Model	NQ	TriviaQA	WQ	TREC	SQuAD
Single	BM25+BERT (Lee et al., 2019)	26.5	47.1	17.7	21.3	33.2
Single	ORQA (Lee et al., 2019)	33.3	45.0	36.4	30.1	20.2
Single	HardEM (Min et al., 2019a)	28.1	50.9	-	-	-
Single	GraphRetriever (Min et al., 2019b)	34.5	56.0	36.4	-	-
Single	PathRetriever (Asai et al., 2020)	32.6	-	-	-	**56.5**
Single	$\text{REALM}_{\text{Wiki}}$ (Guu et al., 2020)	39.2	-	40.2	46.8	-
Single	$\text{REALM}_{\text{News}}$ (Guu et al., 2020)	40.4	-	40.7	42.9	-
Single	BM25	32.6	52.4	29.9	24.9	38.1
	DPR	**41.5**	56.8	34.6	25.9	29.8
	BM25+DPR	39.0	57.0	35.2	28.0	36.7
Multi	DPR	**41.5**	56.8	**42.4**	49.4	24.1
	BM25+DPR	38.8	**57.9**	41.1	**50.6**	35.8

Using decoders instead?

Some recent work suggests that you should use a decoder framework to generate answers

Fusion-in-decoder (FID) = DPR + T5



Model	NaturalQuestions	TriviaQA	
ORQA (Lee et al., 2019)	31.3	45.1	-
REALM (Guu et al., 2020)	38.2	-	-
DPR (Karpukhin et al., 2020)	41.5	57.9	-
SpanSeqGen (Min et al., 2020)	42.5	-	-
RAG (Lewis et al., 2020)	44.5	56.1	68.0
T5 (Roberts et al., 2020)	36.6	-	60.5
GPT-3 few shot (Brown et al., 2020)	29.9	-	71.2
Fusion-in-Decoder (base)	48.2	65.0	77.1
Fusion-in-Decoder (large)	**51.4**	**67.6**	**80.1**

Izacard and Grave 2020. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering