EE-608: Deep Learning for Natural Language Processing:

Question Answering and Model Analysis

James Henderson

Idiap Research Institute

DLNLP, Lecture 10

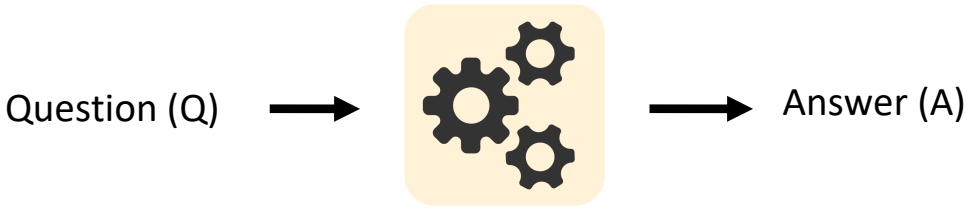
Outline

Question Answering

Model Analysis

Outline

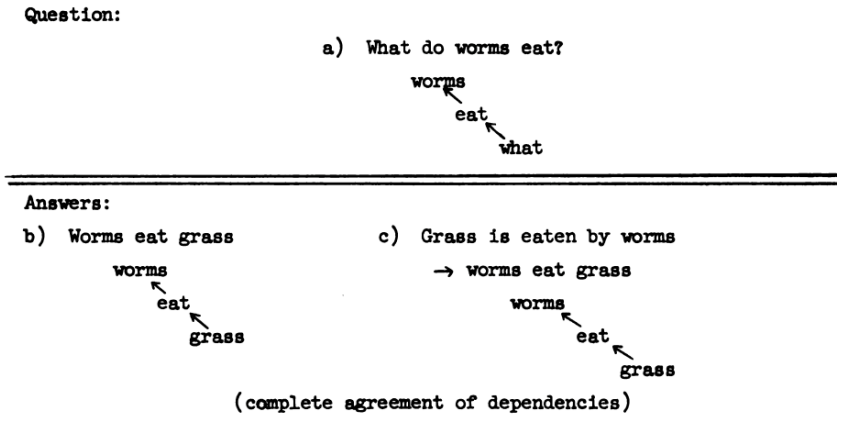
1. What is question answering?



The goal of question answering is to build systems that **automatically** answer questions posed by humans in a **natural language**

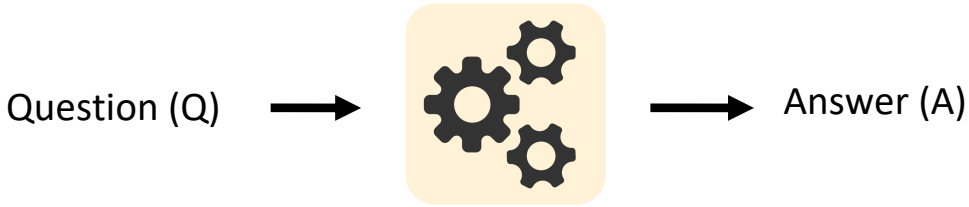
The earliest QA systems dated back to 1960s!

(Simmons et al., 1964)



Slide from John Hewitt (and Danqi Chen)

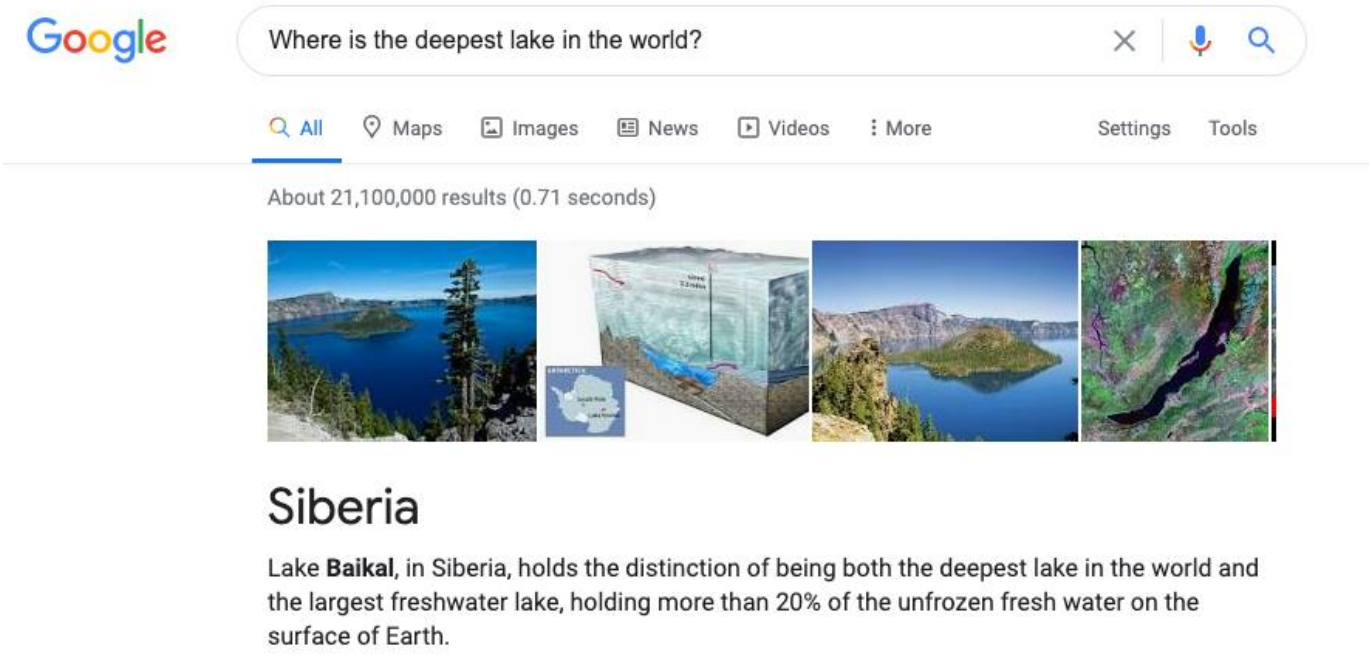
Question answering: a taxonomy



- What information source does a system build on?
  - A text passage, all Web documents, knowledge bases, tables, images..
- Question type
  - Factoid vs non-factoid, open-domain vs closed-domain, simple vs compositional, ..
- Answer type
  - A short segment of text, a paragraph, a list, yes/no, …

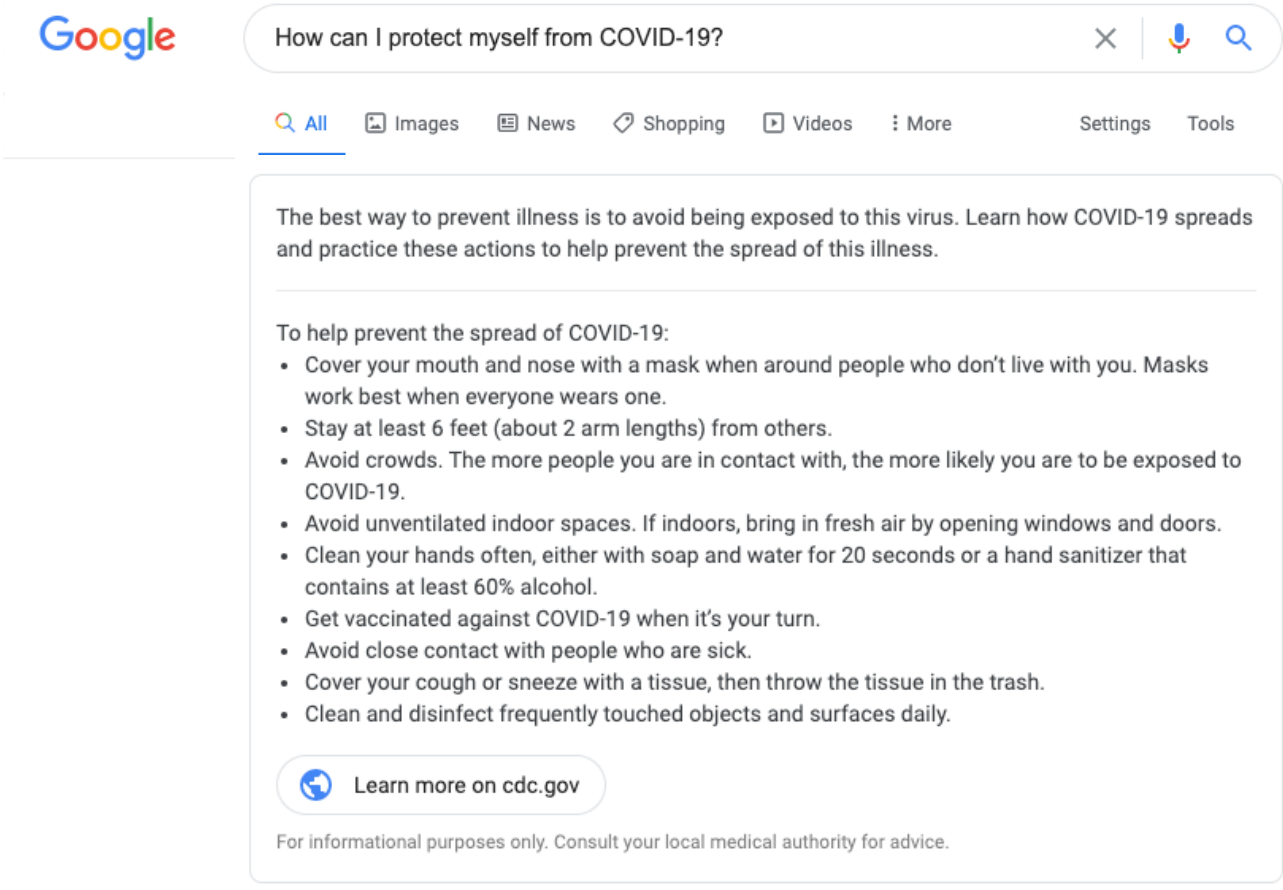
Slide from John Hewitt (and Danqi Chen)

Lots of practical applications



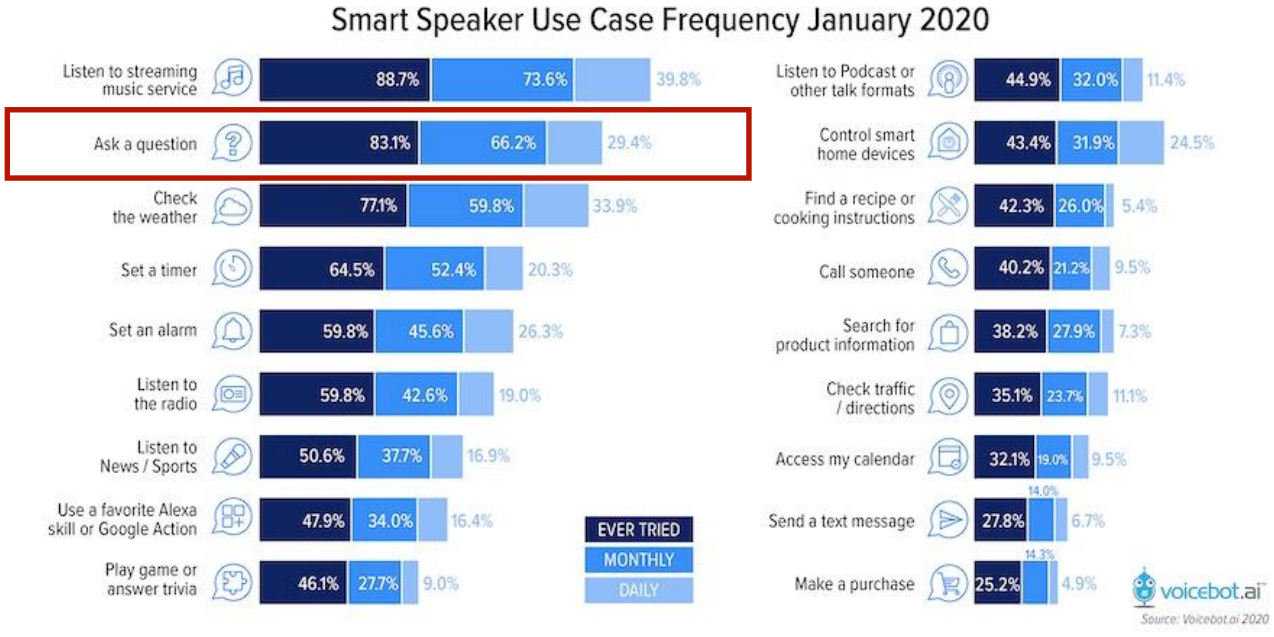
Slide from John Hewitt (and Danqi Chen)

Lots of practical applications



Slide from John Hewitt (and Danqi Chen)

Lots of practical applications



Slide from John Hewitt (and Danqi Chen)

2011: IBM Watson beat Jeopardy champions

IBM Watson defeated two of Jeopardy's greatest champions in 2011

Slide from John Hewitt (and Danqi Chen)

IBM Watson beat Jeopardy champions

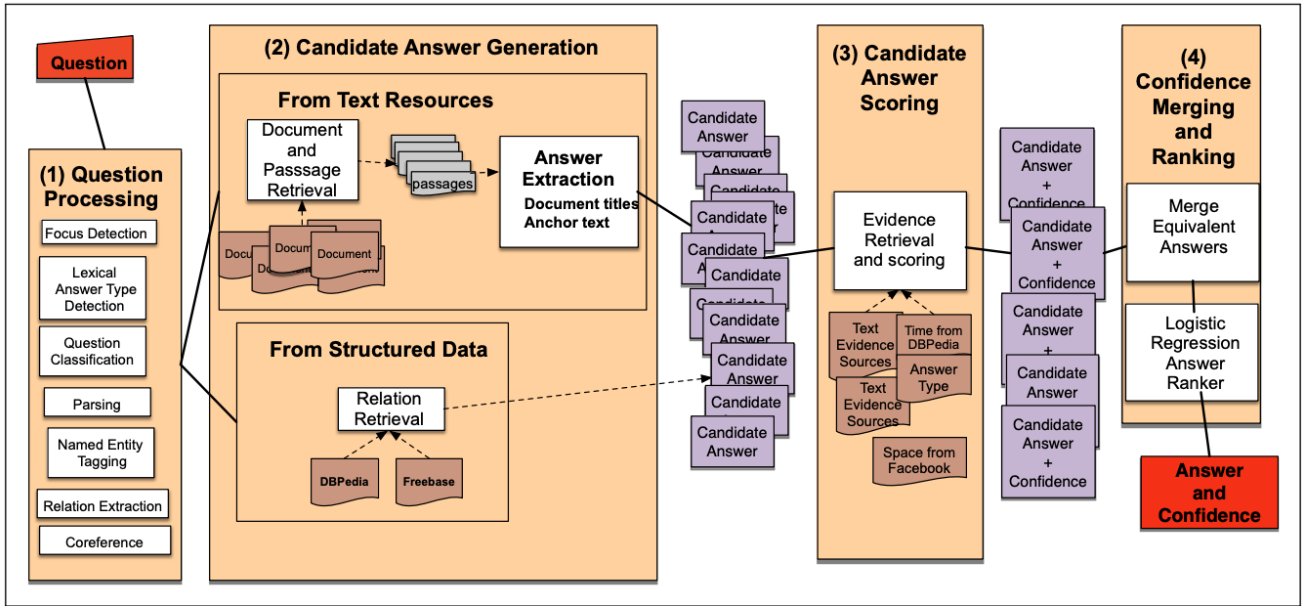


Image credit: J & M, edition 3

(1) Question processing, (2) Candidate answer generation, (3) Candidate answer scoring, and (4) Confidence merging and ranking.

Slide from John Hewitt (and Danqi Chen)

Question answering in deep learning era

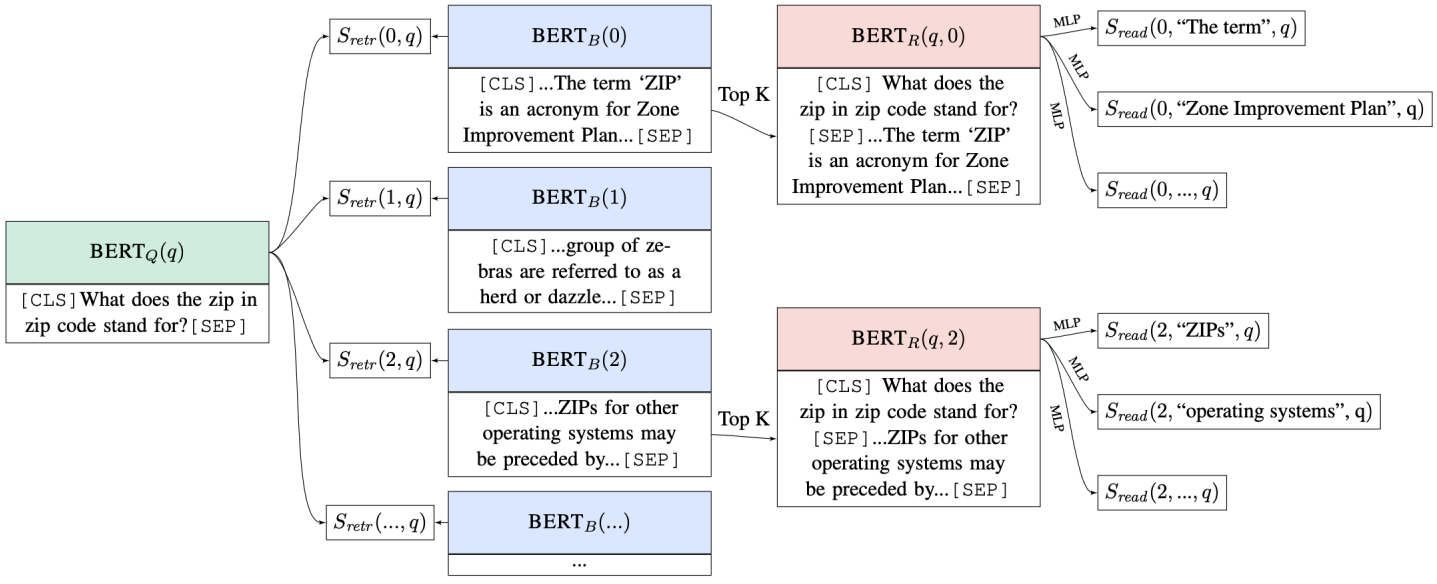


Image credit: (Lee et al., 2019)

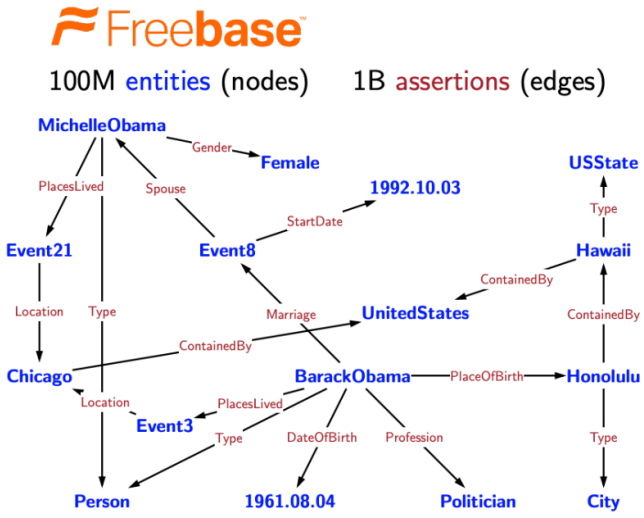
Almost all the state-of-the-art question answering systems are built on top of end-to-end training and pre-trained language models (e.g., BERT)!

Slide from John Hewitt (and Danqi Chen)

Beyond textual QA problems

Today, we will mostly focus on how to answer questions based on **unstructured text**.

Knowledge based QA



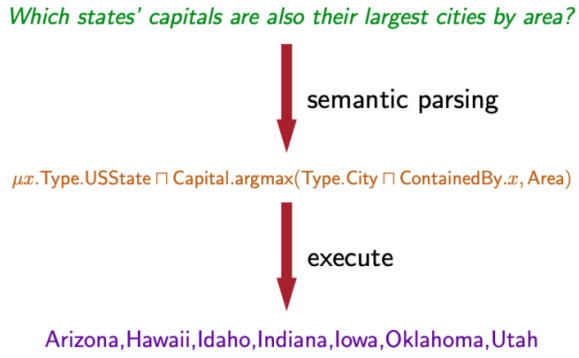


Image credit: Percy Liang

Slide from John Hewitt (and Danqi Chen)

Beyond textual QA problems

Today, we will mostly focus on how to answer questions based on **unstructured text**.

Visual QA

What color are her eyes?
What is the mustache made of?

How many slices of pizza are there?
Is this a vegetarian pizza?

(Antol et al., 2015): Visual Question Answering

Slide from John Hewitt (and Danqi Chen)

2. Reading comprehension

Reading comprehension = comprehend a passage of text and answer questions about its content (P, Q) ⟶ A

Tesla was the fourth of five children. He had an older brother named Dane and three sisters, Milka, Angelina and Marica. Dane was killed in a horse-riding accident when Nikola was five. In 1861, Tesla attended the "Lower" or "Primary" School in Smiljan where he studied German, arithmetic, and religion. In 1862, the Tesla family moved to Gospić, Austrian Empire, where Tesla's father worked as a pastor. Nikola completed "Lower" or "Primary" School, followed by the "Lower Real Gymnasium" or "Normal School."

Q: What language did Tesla study while in school?

A: German

Slide from John Hewitt (and Danqi Chen)

2. Reading comprehension

Reading comprehension = comprehend a passage of text and answer questions about its content $(P, Q) \longrightarrow A$

Kannada language is the official language of Karnataka and spoken as a native language by about 66.54% of the people as of 2011. Other linguistic minorities in the state were Urdu (10.83%), Telugu language (5.84%), Tamil language (3.45%), Marathi language (3.38%), Hindi (3.3%), Tulu language (2.61%), Konkani language (1.29%), Malayalam (1.27%) and Kodava Takk (0.18%). In 2007 the state had a birth rate of 2.2%, a death rate of 0.7%, an infant mortality rate of 5.5% and a maternal mortality rate of 0.2%. The total fertility rate was 2.2.

Q: Which linguistic minority is larger, Hindi or Malayalam?

A: Hindi

Slide from John Hewitt (and Danqi Chen)

Why do we care about this problem?

- Useful for many practical applications
- Reading comprehension is an important testbed for evaluating how well computer systems 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**."
- Many other NLP tasks can be reduced to a reading comprehension problem:

Information extraction

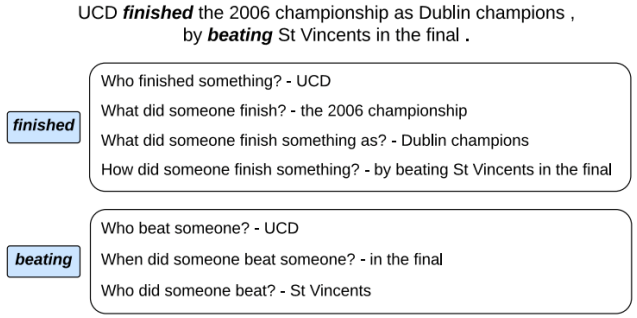
(Barack Obama, educated_at, ?)

Question: Where did Barack Obama graduate from?

Passage: Obama was born in Honolulu, Hawaii. After graduating from Columbia University in 1983, he worked as a community organizer in Chicago.

(Levy et al., 2017)

Semantic role labeling



(He et al., 2015)

Slide from John Hewitt (and Danqi Chen)

Stanford question answering dataset (SQuAD)

- 100k annotated (passage, question, answer) triples

  Large-scale supervised datasets are also a key ingredient for training effective neural models for reading comprehension!

- Passages are selected from English Wikipedia, usually 100~150 words.
- Questions are crowd-sourced.
- Each answer is a short segment of text (or span) in the passage.

  This is a limitation— not all the questions can be answered in this way!

- SQuAD was for years the most popular reading comprehension dataset; it is "almost solved" today (though the underlying task is not,) and the state-of-the-art exceeds the estimated human performance.

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under **gravity**. The main forms of precipitation include drizzle, rain, sleet, snow, **graupel** and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals **within a cloud**. Short, intense periods of rain in scattered locations are called "showers".

What causes precipitation to fall?
gravity

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
graupel

Where do water droplets collide with ice crystals to form precipitation?
within a cloud

(Rajpurkar et al., 2016): SQuAD: 100,000+ Questions for Machine Comprehension of Text

Slide from John Hewitt (and Danqi Chen)

Stanford question answering dataset (SQuAD)

- **Evaluation**: exact match (0 or 1) and F1 (partial credit).
- For development and testing sets, 3 gold answers are collected, because there could be multiple plausible answers.
- We compare the predicted answer to *each* gold answer (a, an, the, punctuations are removed) and take max scores. Finally, we take the average of all the examples for both exact match and F1.
- Estimated human performance: EM = 82.3, F1 = 91.2

Q: What did Tesla do in December 1878?

A: {left Graz, left Graz, left Graz and severed all relations with his family}

Prediction: {left Graz and served}

Exact match: max{0, 0, 0} = 0

F1: max{0.67, 0.67, 0.61} = 0.67

Slide from John Hewitt (and Danqi Chen)

Other question answering datasets

- TriviaQA: Questions and answers by trivia enthusiasts. Independently collected web paragraphs that contain the answer and seem to discuss question, but no human verification that paragraph supports answer to question

- Natural Questions: Question drawn from frequently asked Google search questions. Answers from Wikipedia paragraphs. Answer can be substring, yes, no, or NOT_PRESENT. Verified by human annotation.

- HotpotQA. Constructed questions to be answered from the whole of Wikipedia which involve getting information from two pages to answer a multistep query:
  Q: Which novel by the author of "Armada" will be adapted as a feature film by Steven Spielberg? A: *Ready Player One*

Slide from John Hewitt (and Danqi Chen)

Neural models for reading comprehension

How can we build a model to solve SQuAD?

(We are going to use **passage, paragraph and context,** as well as **question** and **query** interchangeably)

- Problem formulation
  - Input: $C = (c_1, c_2, \dots, c_N), Q = (q_1, q_2, \dots, q_M), c_i, q_i \in V$ N~100, M ~15
  - Output: $1 \leq \text{start} \leq \text{end} \leq N$ answer is a span in the passage

- A family of LSTM-based models with attention (2016–2018)

  Attentive Reader (Hermann et al., 2015), Stanford Attentive Reader (Chen et al., 2016), Match-LSTM (Wang et al., 2017), BiDAF (Seo et al., 2017), Dynamic coattention network (Xiong et al., 2017), DrQA (Chen et al., 2017), R-Net (Wang et al., 2017), ReasoNet (Shen et al., 2017)..

- Fine-tuning BERT-like models for reading comprehension (2019+)

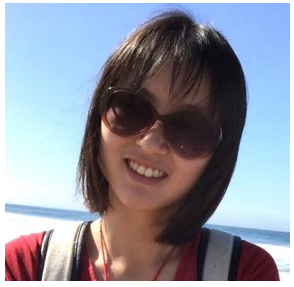
Slide from John Hewitt (and Danqi Chen)

2. Stanford Attentive Reader

[Chen, Bolton, & Manning 2016]

[Chen, Fisch, Weston & Bordes 2017] DrQA

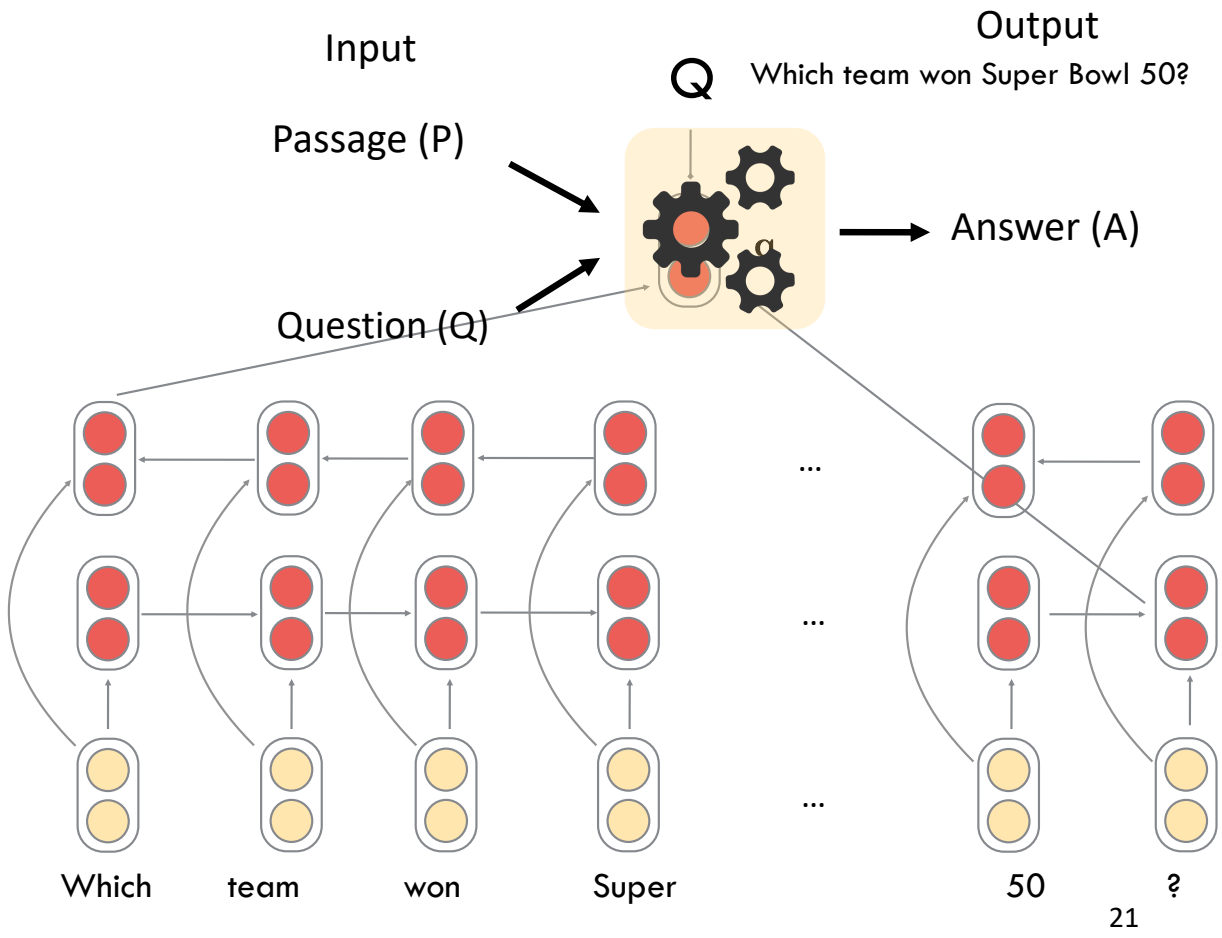
[Chen 2018]



- Demonstrated a minimal, highly successful architecture for reading comprehension and question answering
- Became known as the Stanford Attentive Reader

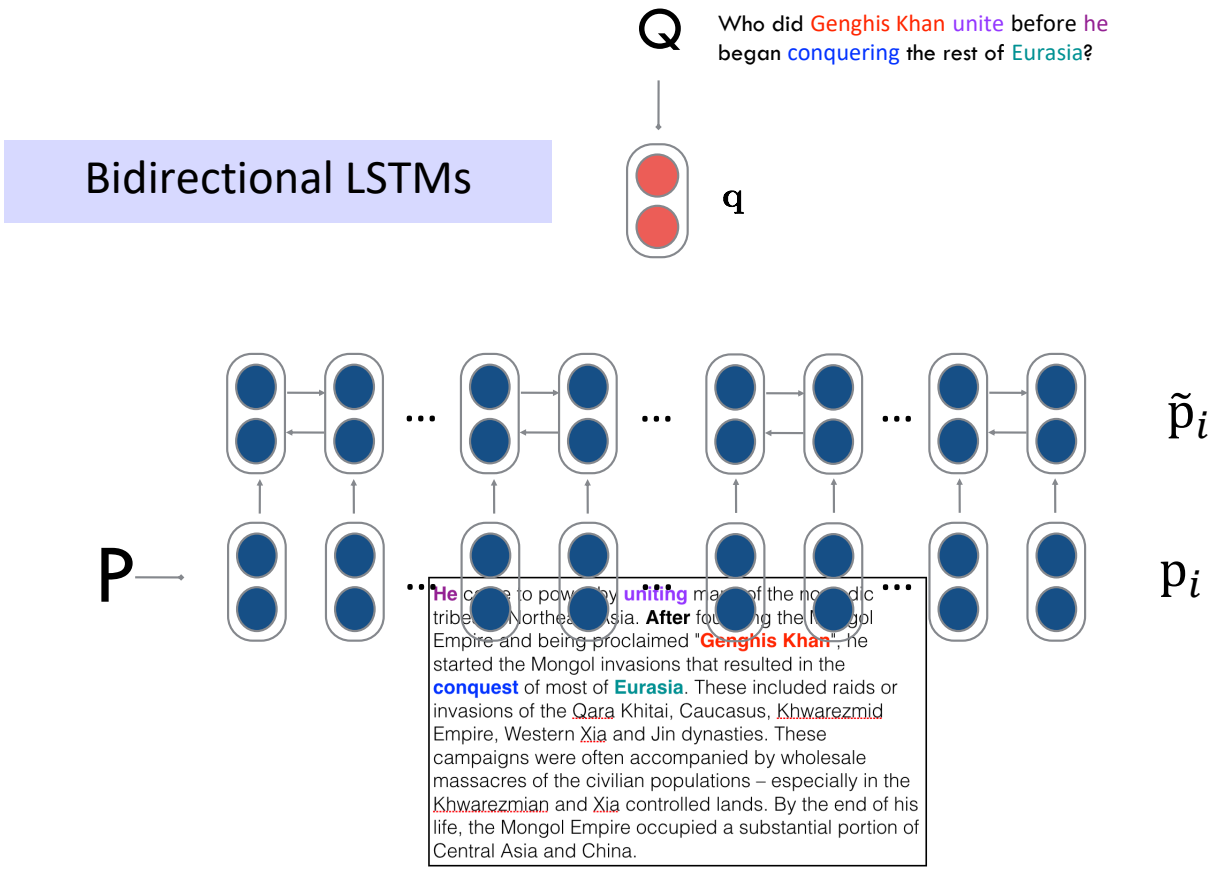
Slide from John Hewitt (and Danqi Chen)

The Stanford Attentive Reader



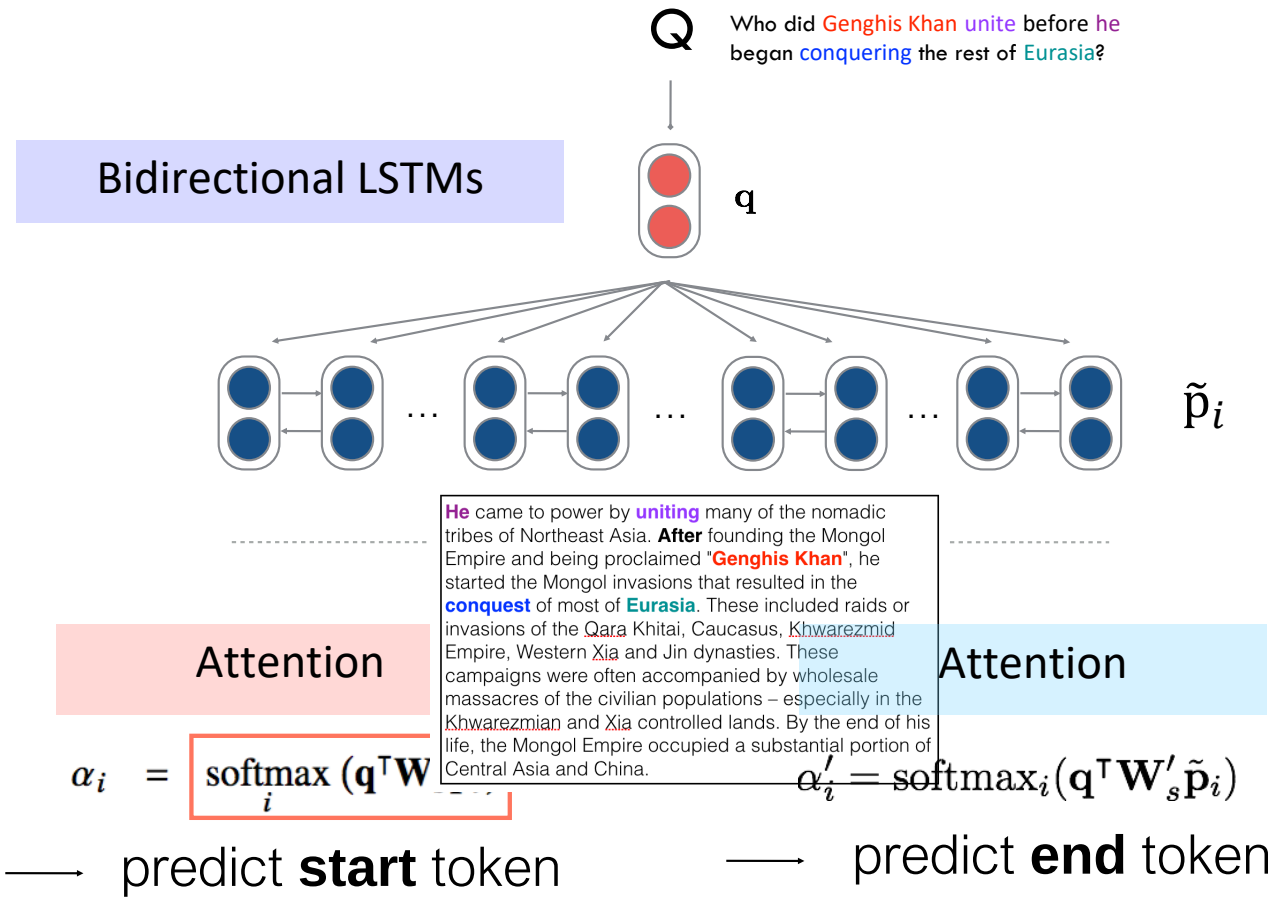
Slide from John Hewitt (and Danqi Chen)

Stanford Attentive Reader



Slide from John Hewitt (and Danqi Chen)

Stanford Attentive Reader



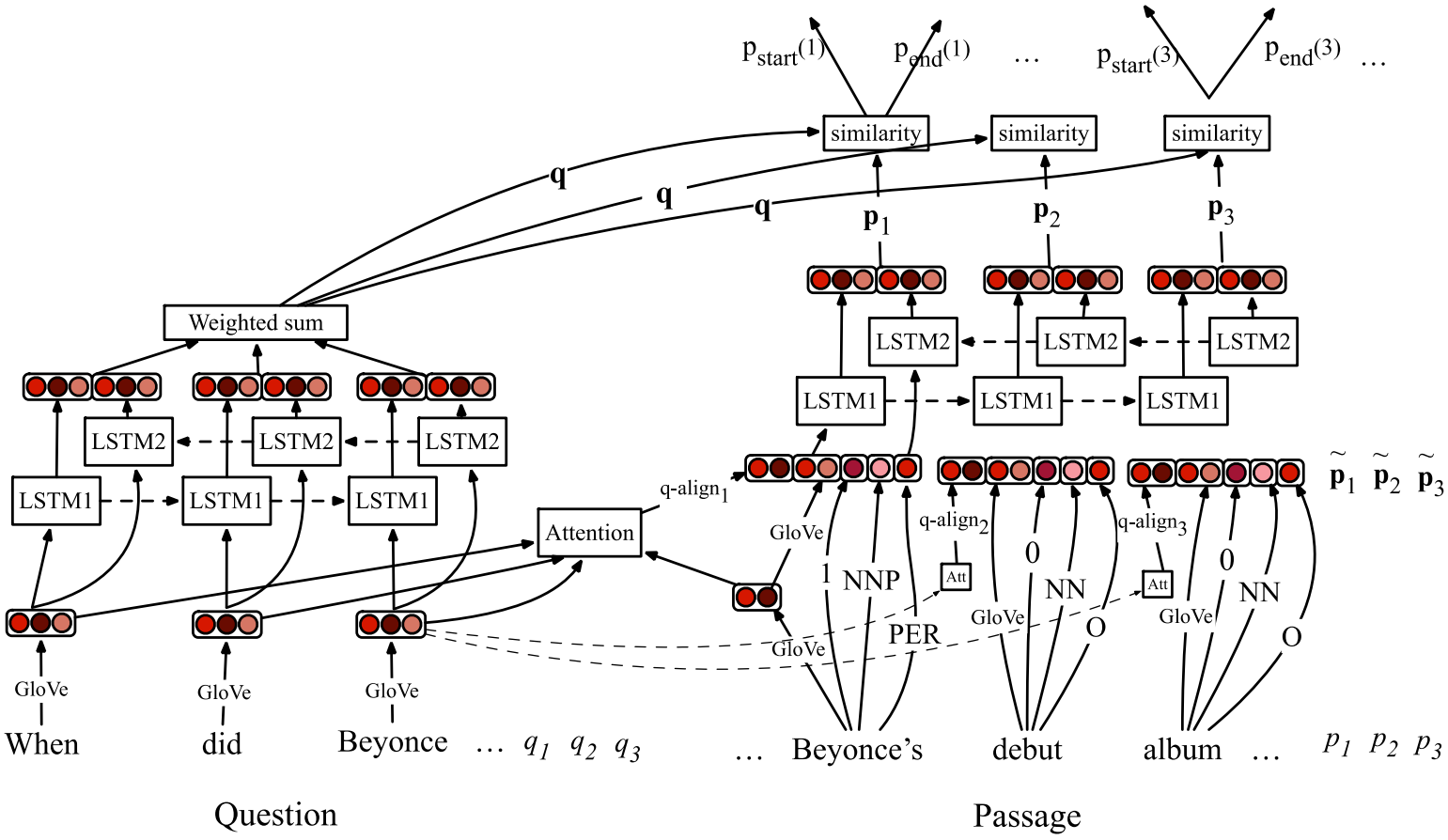
Slide from John Hewitt (and Danqi Chen)

SQuAD 1.1 Results (single model, c. Feb 2017)

	F1
Logistic regression	51.0
Fine-Grained Gating (Carnegie Mellon U)	73.3
Match-LSTM (Singapore Management U)	73.7
DCN (Salesforce)	75.9
BiDAF (UW & Allen Institute)	77.3
Multi-Perspective Matching (IBM)	78.7
ReasoNet (MSR Redmond)	79.4
DrQA (Chen et al. 2017)	79.4
r-net (MSR Asia) [Wang et al., ACL 2017]	79.7
Human performance	91.2

Pretrained + Finetuned Models circa 2021 >93.0

Slide from John Hewitt (and Danqi Chen)



Training objective: $\mathcal{L} = -\sum \log P^{(\text{start})}(a_{\text{start}}) - \sum \log P^{(\text{end})}(a_{\text{end}})$

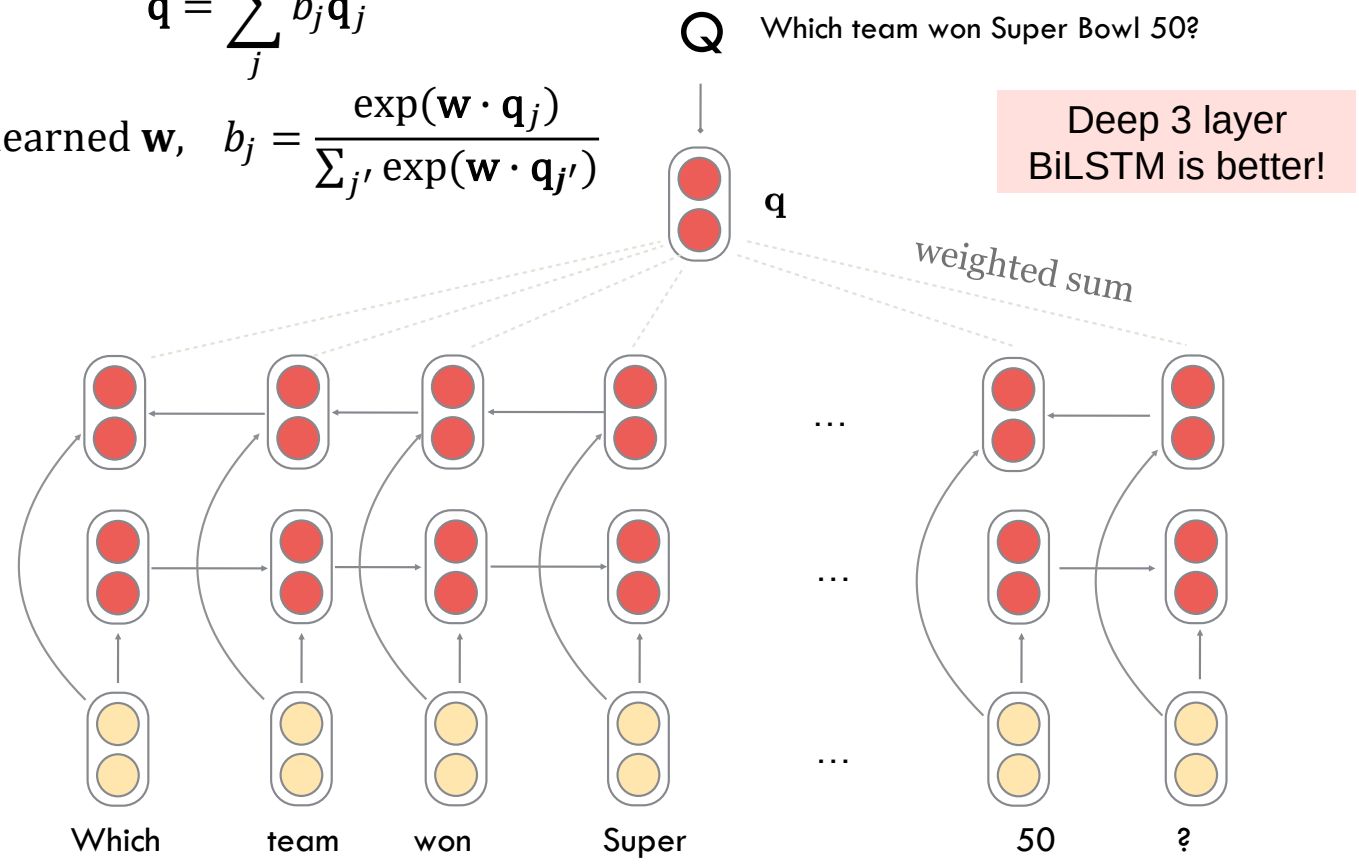
Slide from John Hewitt (and Danqi Chen)

(Chen et al., 2018)

Stanford Attentive Reader++

$$\mathbf{q} = \sum_j b_j \mathbf{q}_j$$

For learned $\mathbf{w}$, $b_j = \frac{\exp(\mathbf{w} \cdot \mathbf{q}_j)}{\sum_{j'} \exp(\mathbf{w} \cdot \mathbf{q}_{j'})}$



Slide from John Hewitt (and Danqi Chen)

Stanford Attentive Reader++

$\mathbf{p}\_i$: Vector representation of each token in passage

Made from concatenation of

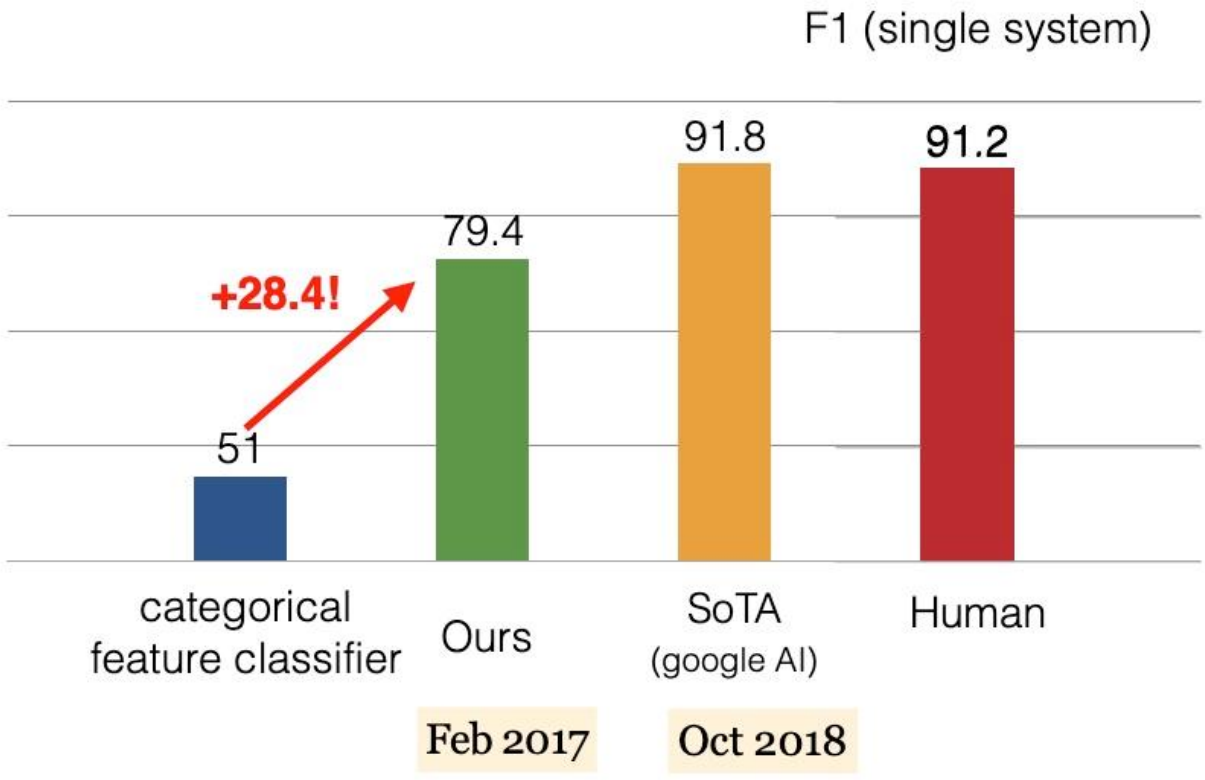
- Word embedding (GloVe 300d)
- Linguistic features: POS & NER tags, one-hot encoded
- Term frequency (unigram probability)
- Exact match: whether the word appears in the question
- 3 binary features: exact, uncased, lemma
- Aligned question embedding ("car" vs "vehicle")

$$f_{align}(p_i) = \sum_j a_{i,j}\mathbf{E}(q_j) \qquad a_{i,j} = \frac{\exp(\alpha(\mathbf{E}(p_i)) \cdot \alpha(\mathbf{E}(q_j)))}{\sum_{j'} \exp(\alpha(\mathbf{E}(p_i)) \cdot \alpha(\mathbf{E}(q'_j)))}$$

Where α is a simple one layer FFNN

Slide from John Hewitt (and Danqi Chen)

A big win for neural models

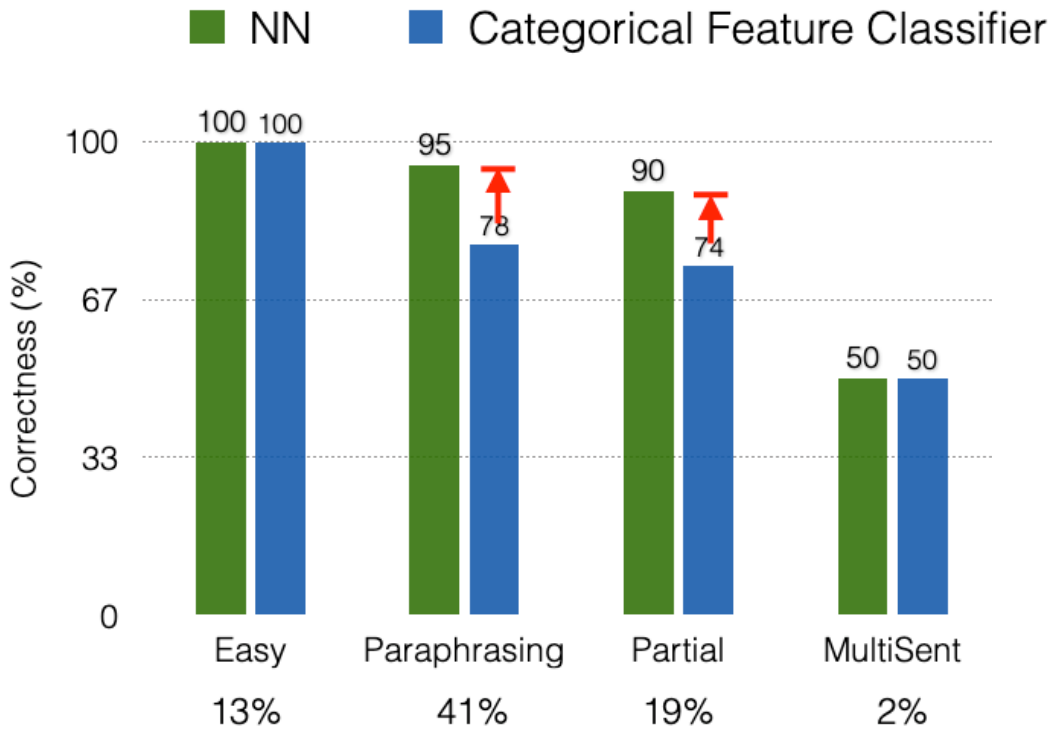


x

Slide from John Hewitt (and Danqi Chen)

(Chen, Bolton, Manning, 2016)

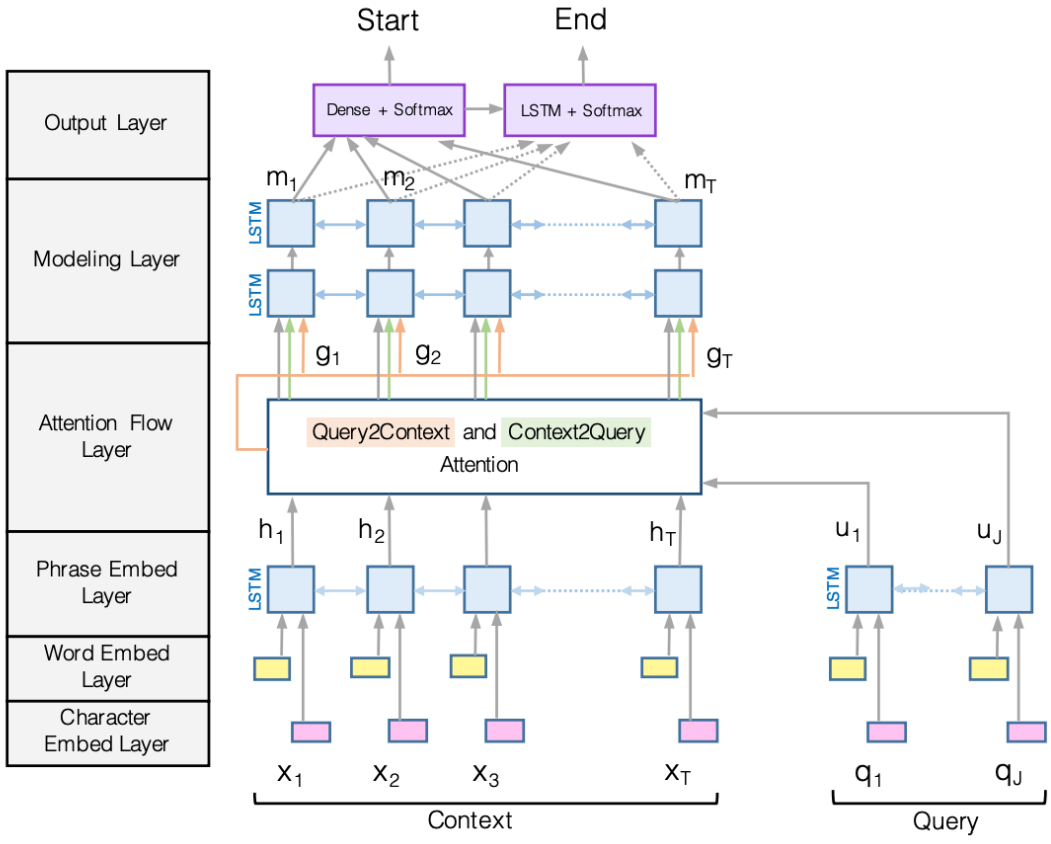
What do these neural models do?



X

Slide from John Hewitt (and Danqi Chen)

BiDAF: the Bidirectional Attention Flow model



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

Slide from John Hewitt (and Danqi Chen)

Attention visualization

Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season . The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super Bowl title . The game was played on February 7 , 2016 , at Levi 's Stadium in the San Francisco Bay Area at Santa Clara , California . As this was the 50th Super Bowl , the league emphasized the " golden anniversary " with various gold-themed initiatives , as well as temporarily suspending the tradition of naming each Super Bowl game with Roman numerals (under which the game would have been known as " Super Bowl L ") , so that the logo could prominently feature the Arabic numerals 50 .

Slide from John Hewitt (and Danqi Chen)

LSTM-based vs BERT models

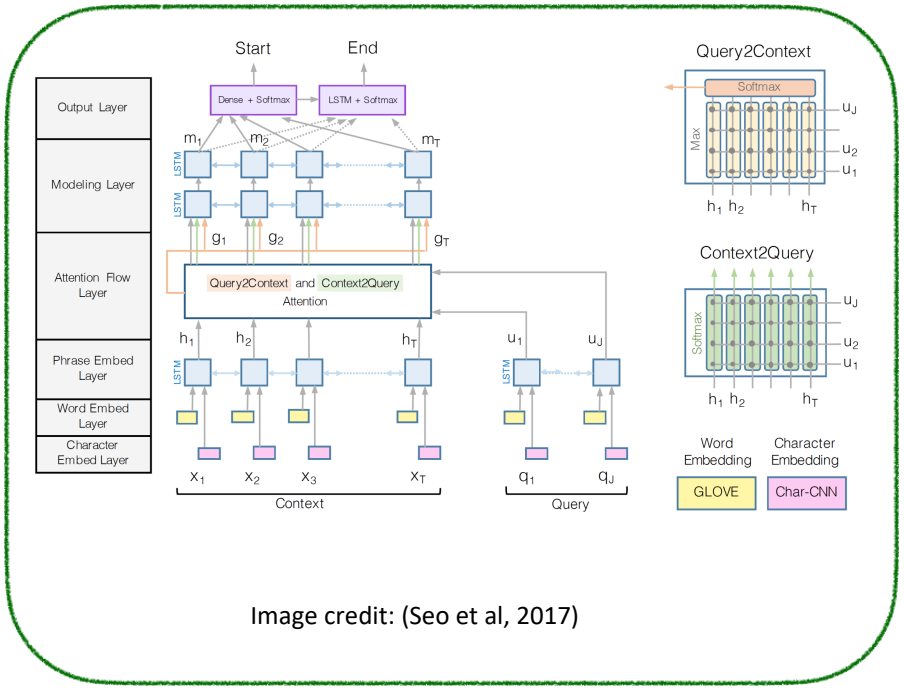


Image credit: (Seo et al, 2017)

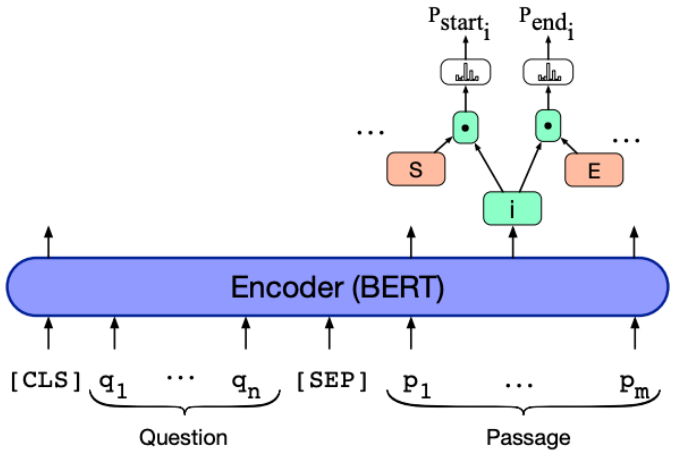
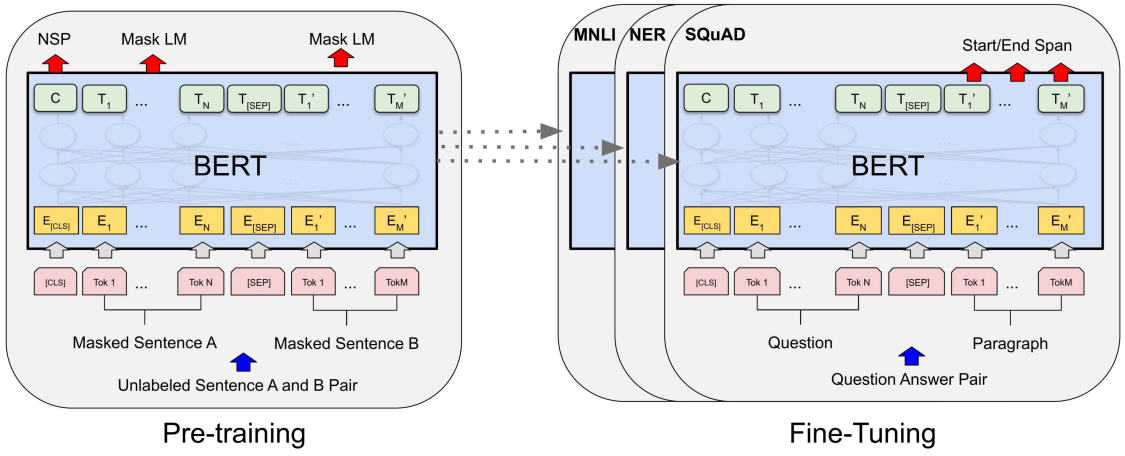


Image credit: J & M, edition 3

Slide from John Hewitt (and Danqi Chen)

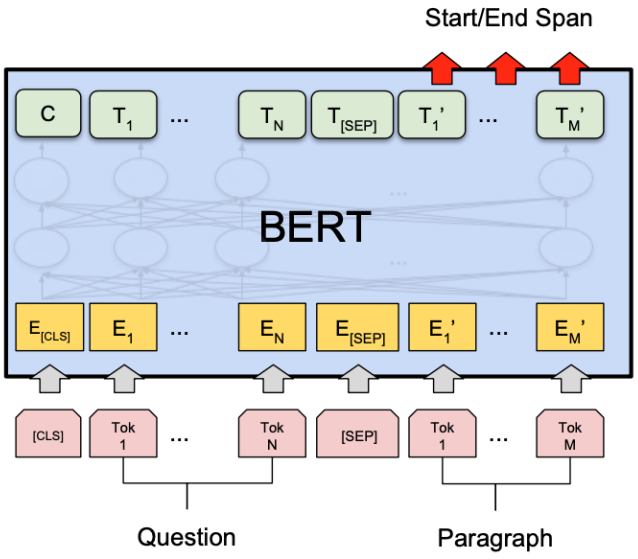
BERT for reading comprehension

- BERT is a deep bidirectional Transformer encoder pre-trained on large amounts of text (Wikipedia + BooksCorpus)
- BERT is pre-trained on two training objectives:
  - Masked language model (MLM)
  - Next sentence prediction (NSP)
- $\text{BERT}_{\text{base}}$ has 12 layers and 110M parameters, $\text{BERT}_{\text{large}}$ has 24 layers and 330M parameters



Slide from John Hewitt (and Danqi Chen)

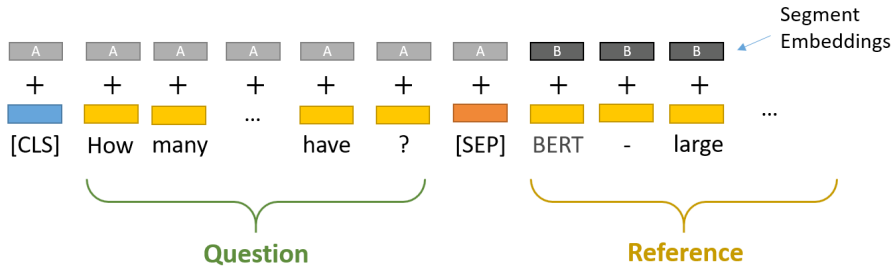
BERT for reading comprehension



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/

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

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

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

where $\mathbf{h}_i$ is the hidden vector of c_i, returned by BERT

Slide from John Hewitt (and Danqi Chen)

BERT for reading comprehension

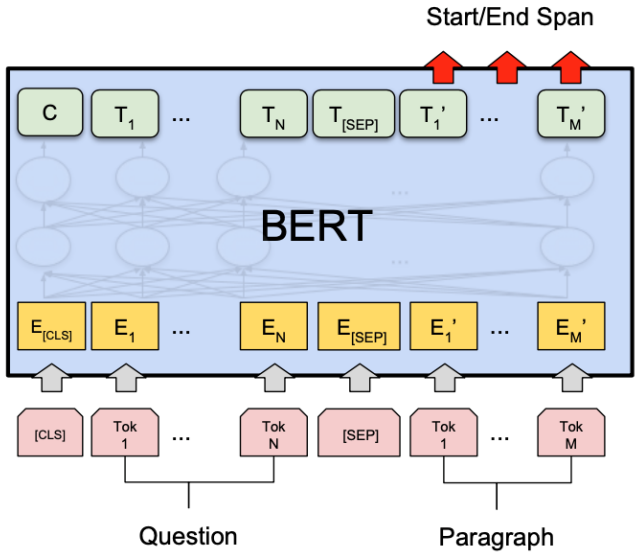
$$\mathcal{L} = -\log p_{\text{start}}(s^*) - \log p_{\text{end}}(e^*)$$

- All the BERT parameters (e.g., 110M) as well as the newly introduced parameters $\mathbf{h}_{\text{start}}, \mathbf{h}_{\text{end}}$ (e.g., 768 x 2 = 1536) are optimized together for $\mathcal{L}$.

- It works amazing well. Stronger pre-trained language models can lead to even better performance and SQuAD becomes a standard dataset for testing pre-trained models.

	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

(dev set, except for human performance)



Slide from John Hewitt (and Danqi Chen)

Comparisons between BiDAF and BERT models

- BERT model has many many more parameters (110M or 330M)
  BiDAF has ~2.5M parameters.
- BiDAF is built on top of several bidirectional LSTMs while BERT is built on top of Transformers (no recurrence architecture and easier to parallelize).
- BERT is **pre-trained** while BiDAF is only built on top of GloVe (and all the remaining parameters need to be learned from the supervision datasets).

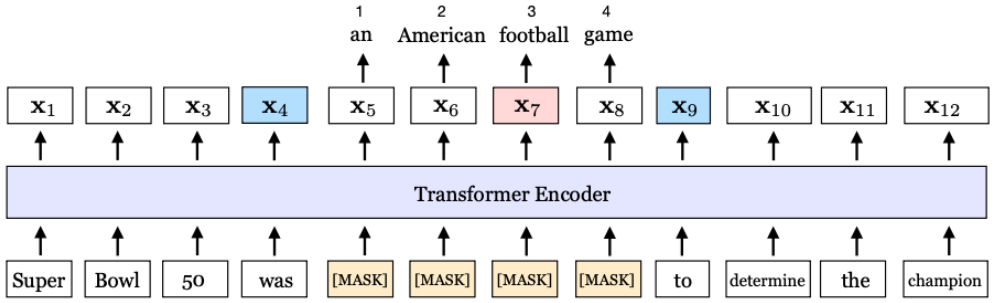
Pre-training is clearly a game changer but it is expensive..

Slide from John Hewitt (and Danqi Chen)

Can we design better pre-training objectives?

The answer is yes!

$$\mathcal{L}(\text{football}) = \mathcal{L}_{\text{MLM}}(\text{football}) + \mathcal{L}_{\text{SBO}}(\text{football})$$
$$= -\log P(\text{football} \mid \mathbf{x}_7) - \log P(\text{football} \mid \mathbf{x}_4, \mathbf{x}_9, \mathbf{p}_3)$$



Two ideas:

1) masking contiguous spans of words instead of 15% random words

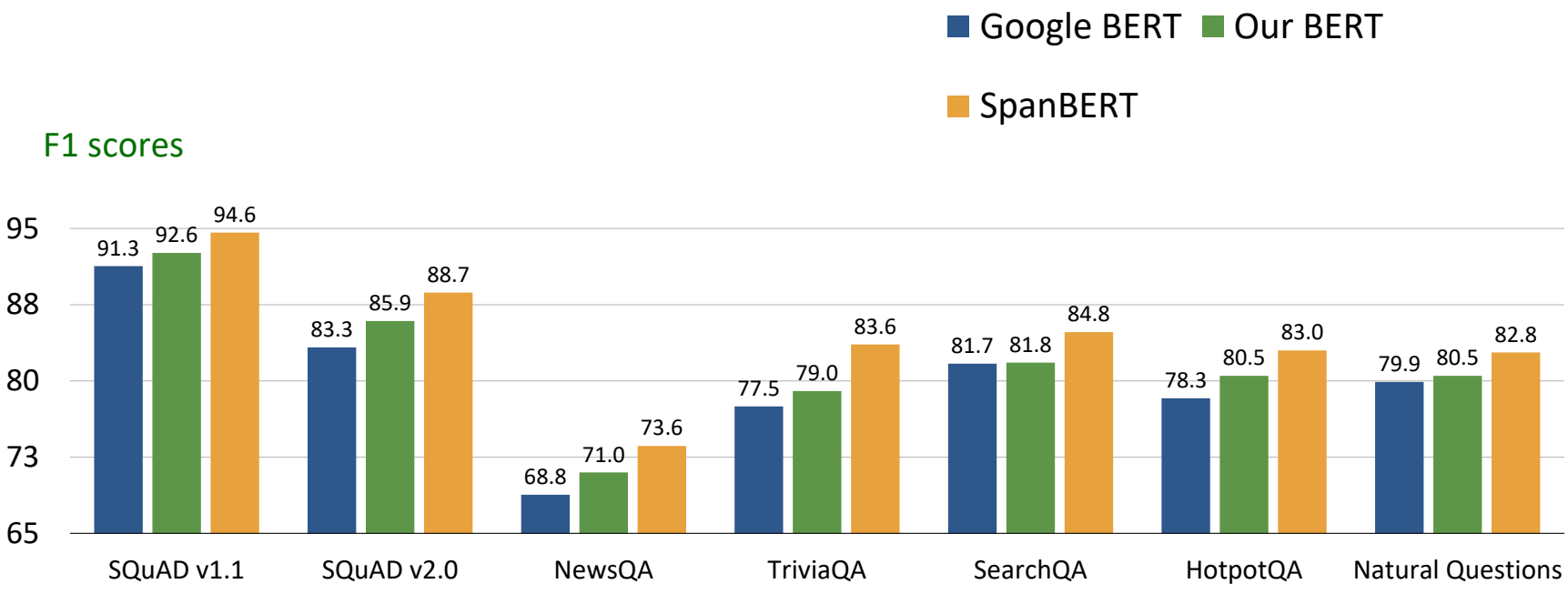
2) using the two end points of span to predict all the masked words in between = compressing the information of a span into its two endpoints

$$\mathbf{y}_i = f(\mathbf{x}_{s-1}, \mathbf{x}_{e+1}, \mathbf{p}_{i-s+1})$$

(Joshi & Chen et al., 2020): SpanBERT: Improving Pre-training by Representing and Predicting Spans

Slide from John Hewitt (and Danqi Chen)

SpanBERT performance



Slide from John Hewitt (and Danqi Chen)

Is reading comprehension solved?

- We have already surpassed human performance on SQuAD. Does it mean that reading comprehension is already solved? **Of course not!**
- The current systems still perform poorly on adversarial examples or examples from out-of-domain distributions

Article: Super Bowl 50
Paragraph: "*Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.*"
Question: "*What is the name of the quarterback who was 38 in Super Bowl XXXIII?*"
Original Prediction: John Elway
Prediction under adversary: Jeff Dean

	Match Single	Match Ens.	BiDAF Single	BiDAF Ens.
Original	71.4	75.4	75.5	80.0
ADDSENT	27.3	29.4	34.3	34.2
ADDONESENT	39.0	41.8	45.7	46.9
ADDANY	7.6	11.7	4.8	2.7
ADDCOMMON	38.9	51.0	41.7	52.6

(Jia and Liang, 2017): Adversarial Examples for Evaluating Reading Comprehension Systems

Slide from John Hewitt (and Danqi Chen)

Is reading comprehension solved?

Systems trained on one dataset can't generalize to other datasets:

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?

Slide from John Hewitt (and Danqi Chen)

Is reading comprehension solved?

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

Slide from John Hewitt (and Danqi Chen)

Is reading comprehension solved?

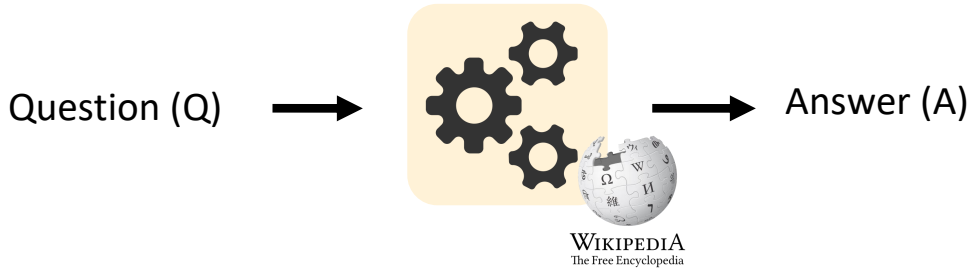
BERT-large model trained on SQuAD

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

Slide from John Hewitt (and Danqi Chen)

3. Open-domain question answering

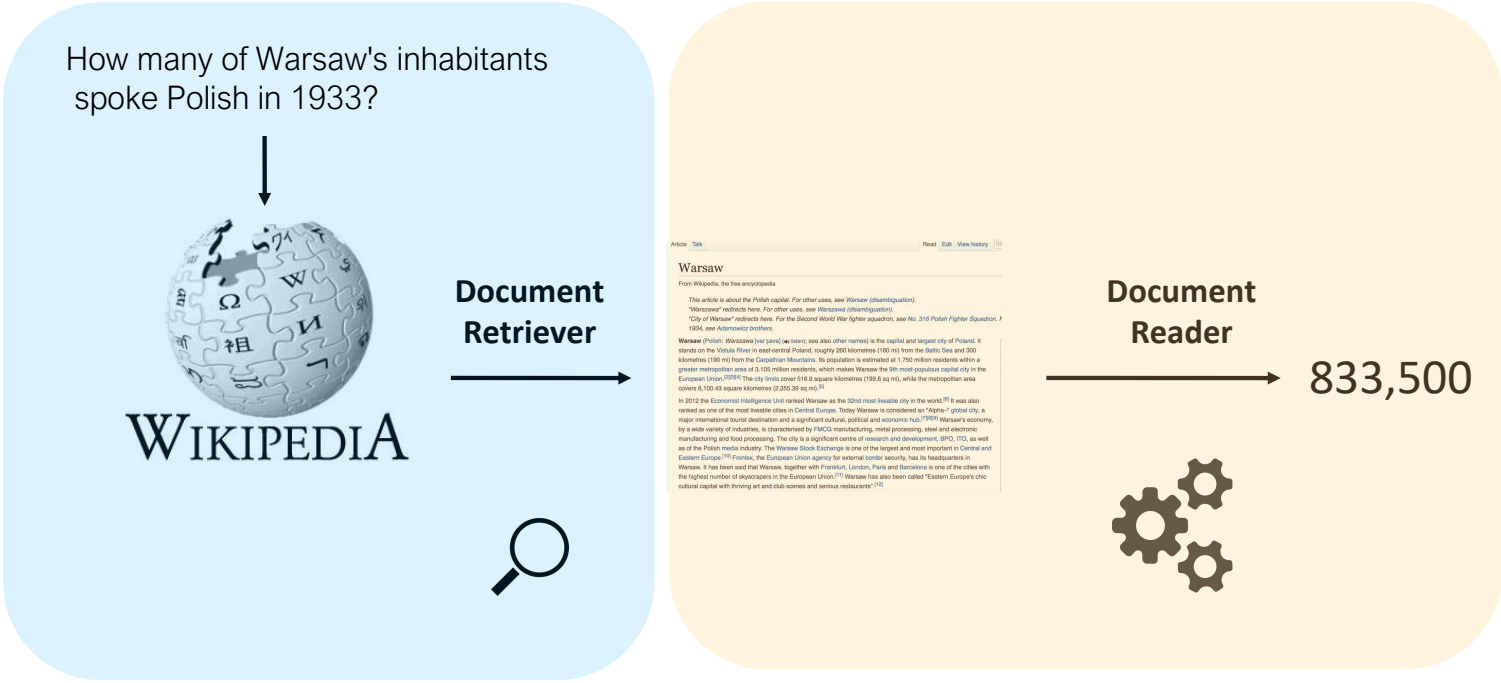


- Different from reading comprehension, we don't assume a given passage.
- Instead, we only have access to a large collection of documents (e.g., Wikipedia). We don't know where the answer is located, and the goal is to return the answer for any open-domain questions.
- Much more challenging and a more practical problem!

*In contrast to **closed-domain** systems that deal with questions under a specific domain (medicine, technical support).*

Slide from John Hewitt (and Danqi Chen)

Retriever-reader framework



https://github.com/facebookresearch/DrQA

Chen et al., 2017. Reading Wikipedia to Answer Open-domain Questions

Slide from John Hewitt (and Danqi Chen)

Retriever-reader framework

- Input: a large collection of documents $\mathcal{D} = D_1, D_2, \ldots, D_N$ and Q
- Output: an answer string A

- Retriever: $f(\mathcal{D}, Q) \longrightarrow P_1, \ldots, P_K$ K is pre-defined (e.g., 100)
- Reader: $g(Q, \{P_1, \ldots, P_K\}) \longrightarrow A$ A reading comprehension problem!

In DrQA,

- Retriever = A standard TF-IDF information-retrieval sparse model (a fixed module)
- Reader = a neural reading comprehension model that we just learned
  - Trained on SQuAD and other distantly-supervised QA datasets

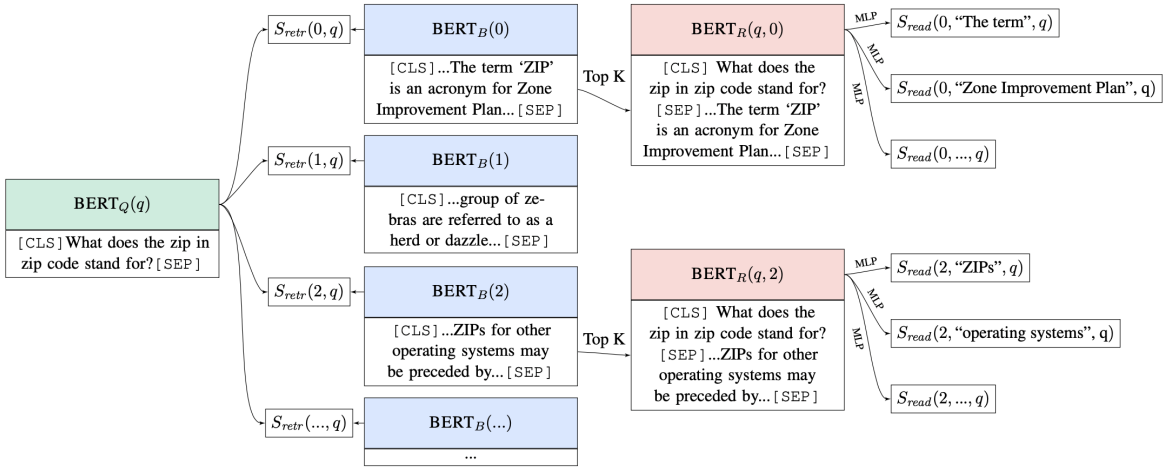
Distantly-supervised examples: (Q, A) ⟶ (P, Q, A)

Chen et al., 2017. Reading Wikipedia to Answer Open-domain Questions

Slide from John Hewitt (and Danqi Chen)

We can train the retriever too

- **Joint training** of retriever and reader



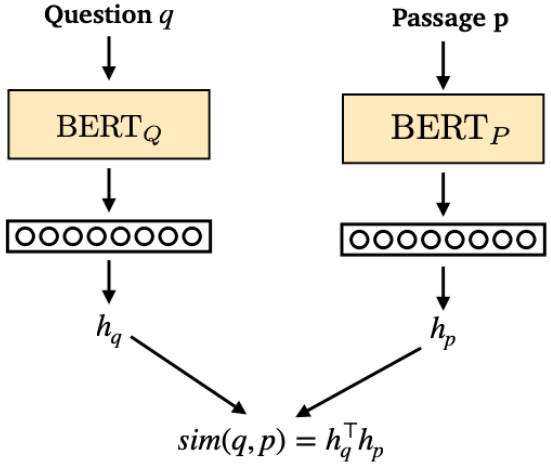
- Each text passage can be encoded as a vector using BERT and the retriever score can be measured as the dot product between the question representation and passage representation.
- However, it is not easy to model as there are a huge number of passages (e.g., 21M in English Wikipedia)

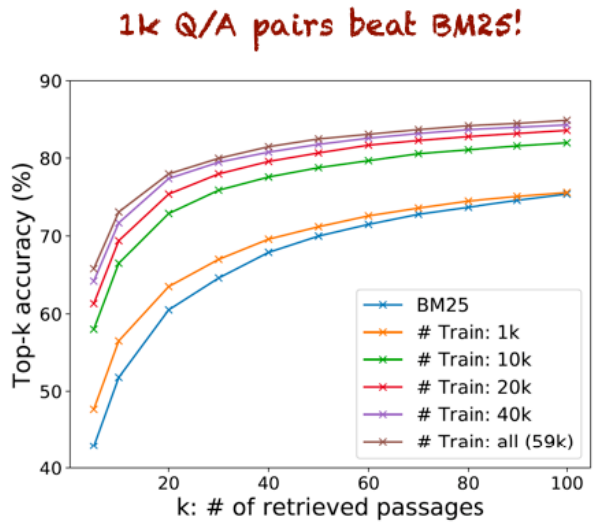
Lee et al., 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering

Slide from John Hewitt (and Danqi Chen)

We can train the retriever too

- Dense passage retrieval (DPR) - We can also just train the retriever using question-answer pairs!



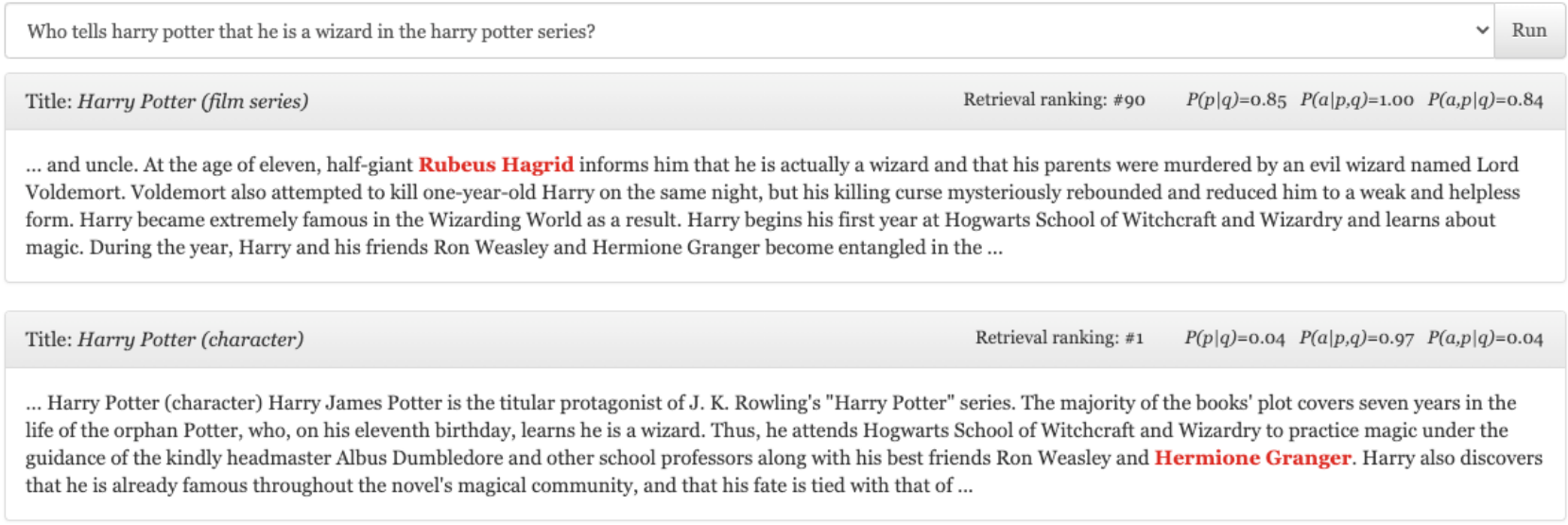


- Trainable retriever (using BERT) largely outperforms traditional IR retrieval models

Karpukhin et al., 2020. Dense Passage Retrieval for Open-Domain Question Answering

Slide from John Hewitt (and Danqi Chen)

We can train the retriever too



http://qa.cs.washington.edu:2020/

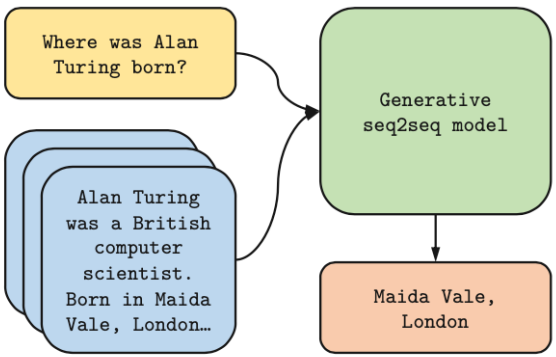
Karpukhin et al., 2020. Dense Passage Retrieval for Open-Domain Question Answering

Slide from John Hewitt (and Danqi Chen)

Dense retrieval + generative models

Recent work shows that it is beneficial to generate answers instead of to extract answers.

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



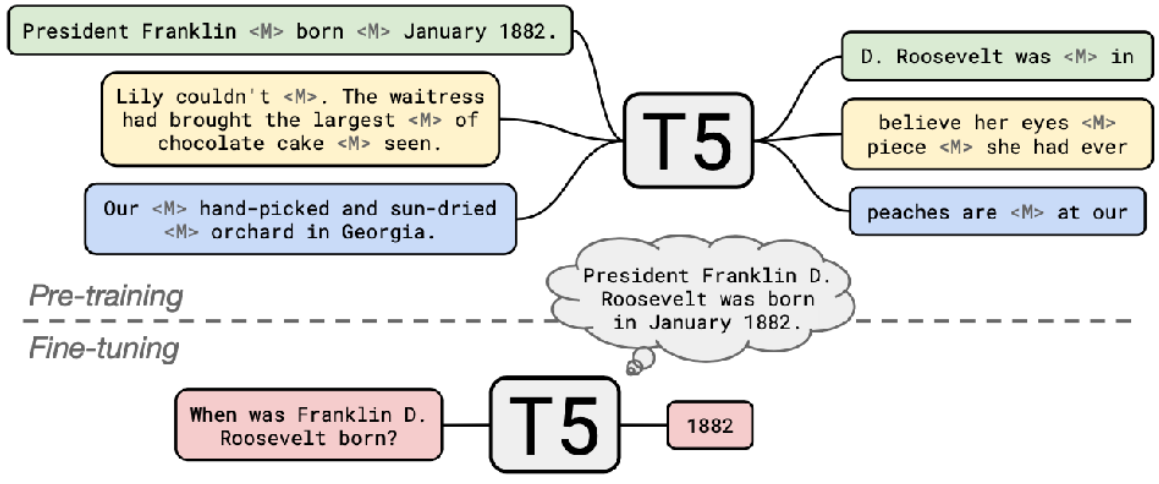
Model	NaturalQuestions	TriviaQA	
ORQA (Lee et al., 2019)	31.3	45.1	-
REALM (Guu et al., 2020)	38.2	-	-
DPR (Karpukhin et al., 2020)	41.5	57.9	-
SpanSeqGen (Min et al., 2020)	42.5	-	-
RAG (Lewis et al., 2020)	44.5	56.1	68.0
T5 (Roberts et al., 2020)	36.6	-	60.5
GPT-3 few shot (Brown et al., 2020)	29.9	-	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

Slide from John Hewitt (and Danqi Chen)

Large language models can do open-domain QA well

- … without an explicit retriever stage

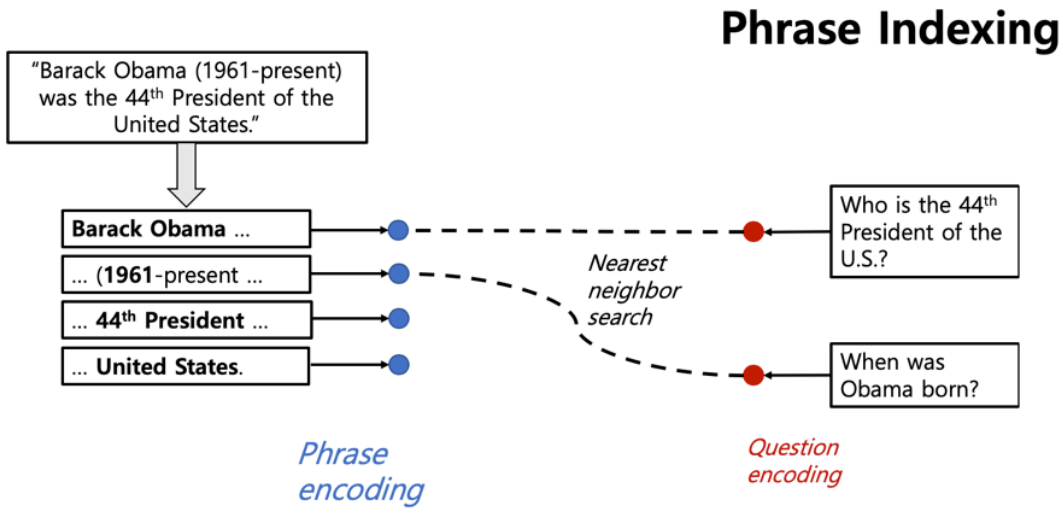


Roberts et al., 2020. How Much Knowledge Can You Pack Into the Parameters of a Language Model?

Slide from John Hewitt (and Danqi Chen)

Maybe the reader model is not necessary too!

It is possible to encode all the phrases (60 billion phrases in Wikipedia) using **dense** vectors and only do nearest neighbor search without a BERT model at inference time!

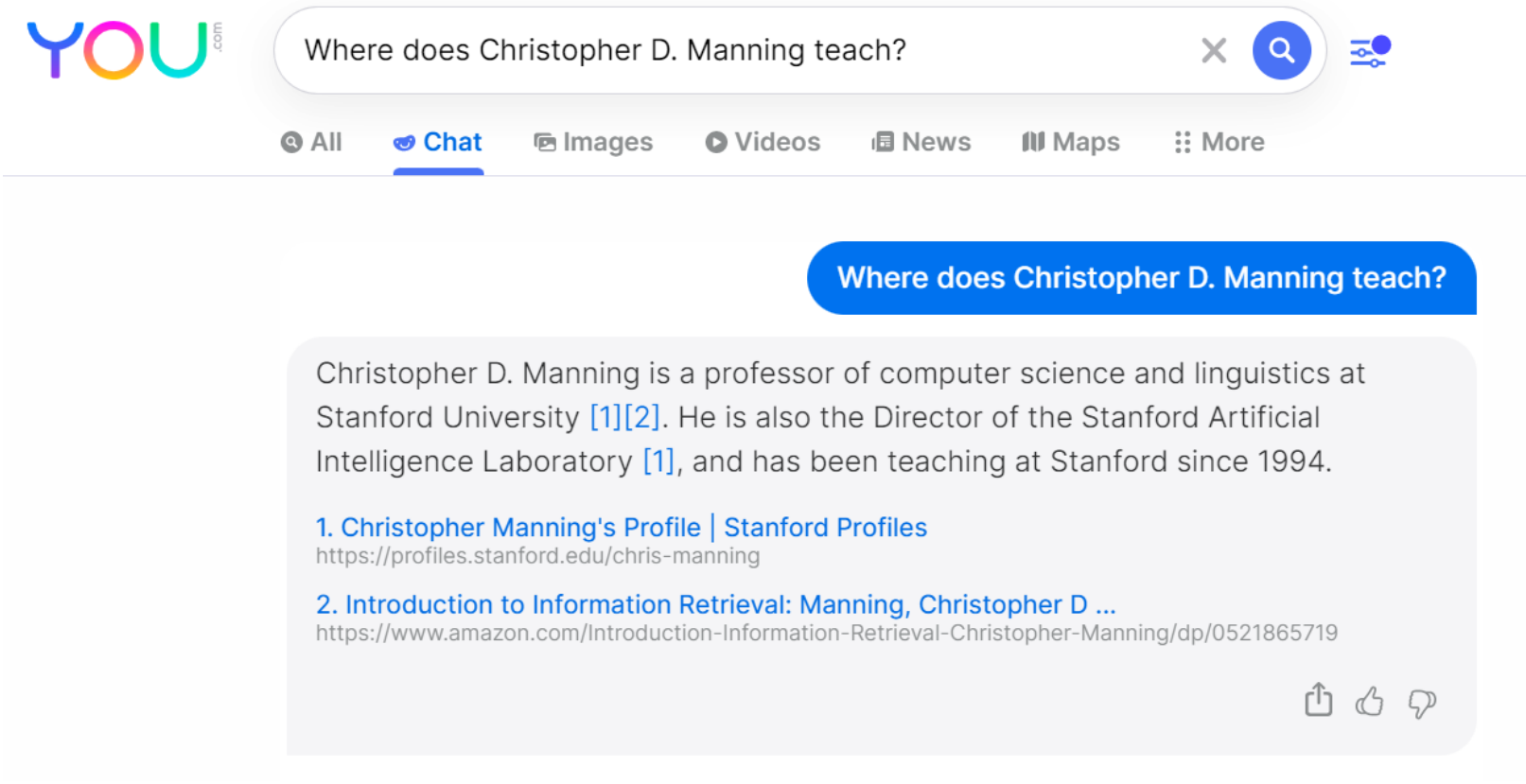


Seo et al., 2019. Real-Time Open-Domain Question Answering with Dense-Sparse Phrase Index

Lee et al., 2020. Learning Dense Representations of Phrases at Scale

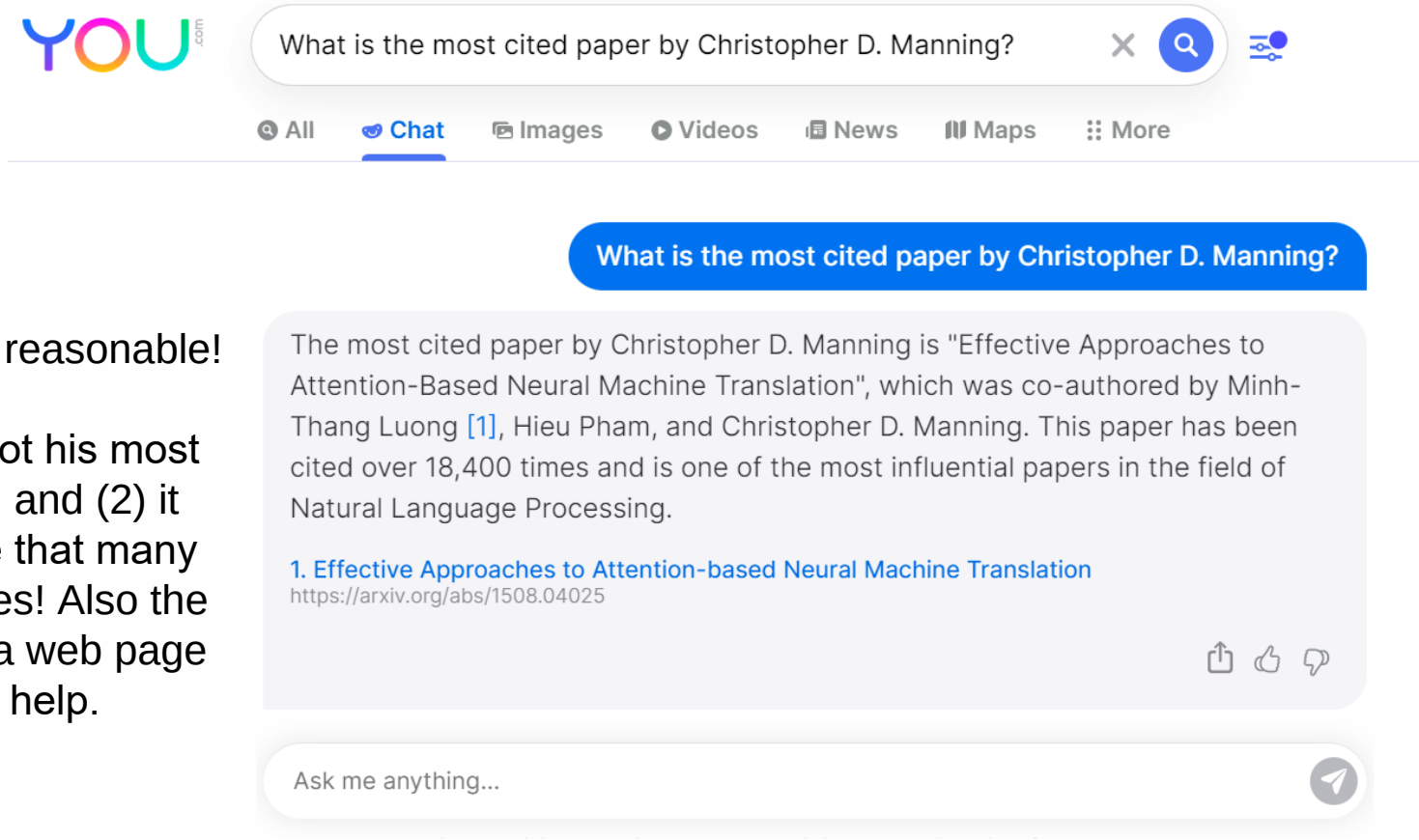
Slide from John Hewitt (and Danqi Chen)

Large language model-based QA (with web search!)



Slide from John Hewitt (and Danqi Chen)

Problems with large language model-based QA



Seems totally reasonable!

But (1) it's not his most cited paper, and (2) it doesn't have that many citations. Yikes! Also the reference to a web page doesn't help.

Slide from John Hewitt (and Danqi Chen)

Summary of Question Answering

- Question Answering combines Information Retrieval and Reading Comprehension
- QA is a very general task, with many different datasets
- Pretrained self-attention based DL models are SOTA
- Best models now generate the answer, rather than select a span from a text
- Very-large pretrained language models can generate an answer from their parameters

Outline

For links to papers cited in the slides, see
`http://web.stanford.edu/class/cs224n/slides/cs224n-2023-lecture18-analysis.pdf`

Lecture Plan

1. Motivating model analysis and explanation
2. One model at multiple levels of abstraction
3. Out-of-domain evaluation sets
    1. Testing for linguistic knowledge
    2. Testing for task heuristics
4. Influence studies and adversarial examples
    1. What part of my input led to this answer?
    2. How could I minimally modify this input to change the answer?
5. Analyzing representations
    1. Correlation in “interpretable” model components
    2. Probing studies: supervised analysis
6. Revisiting model ablations as analysis

Slide from John Hewitt

https://xkcd.com/1838/

Slide from John Hewitt

Motivation: what are our models doing?

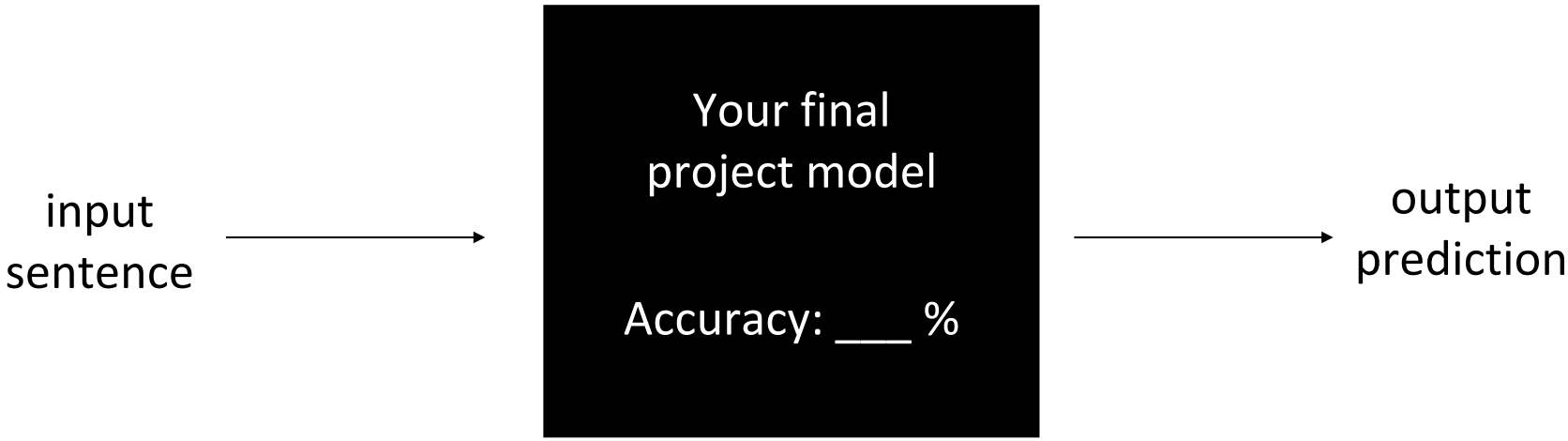


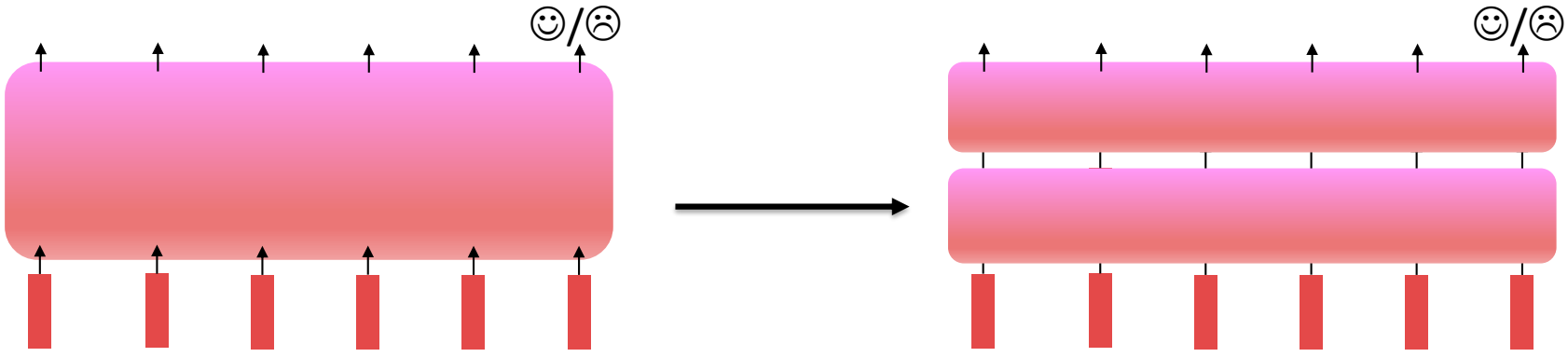
Fig 1. *A black box*

We summarize our models with one (or a handful) of accuracies metric numbers.

What do they learn? Why do they succeed and fail?

Slide from John Hewitt

Motivation: how do we make tomorrow's model?



Today's models: use recipes that work, but aren't perfect

Tomorrow's models: take what works and find what needs changing

Understanding **how far** we can get with incremental improvements on current methods is crucial to the eventual development of major improvements.

Slide from John Hewitt

Motivation: what biases are built into my model?

Man is to computer programmer as woman is to → Word2vec analogies (assignment 1) → homemaker

Fig 1. *A black box*

What did the model use in its decision?

What biases did it learn and possibly worsen?

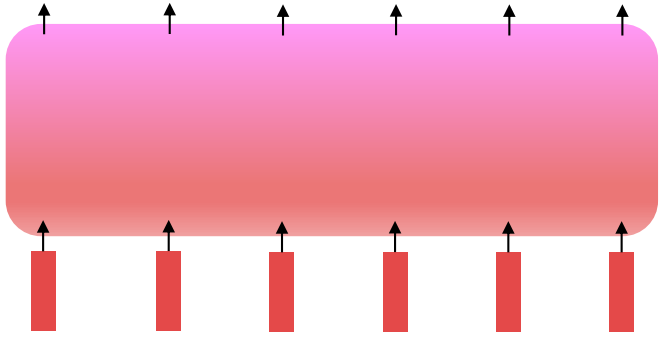
[Bolukbasi et al., 2016]

Slide from John Hewitt

Motivation: how do we make the next 25 years of models?

What can be learned via language model pretraining?

What will replace the Transformer?



What **can't** be learned via language model pretraining?

What does deep learning struggle to do?

How are our models affecting people, and transferring power?

What do neural models tell us about language?

Slide from John Hewitt

Model analysis at varying levels of abstraction

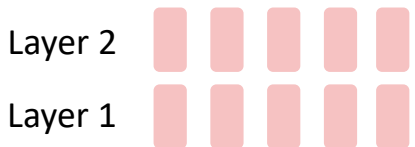
There is a **wide variety** of ways to analyze models; **none is perfect or provides total clarity.**

To start, at what level of **abstraction** do you want to reason about your model?

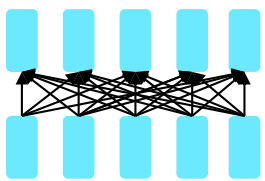
1. Your neural model as a probability distribution and decision function

$$p_{\text{model}}(y|x)$$

2. Your neural model as a sequence of vector representations in depth and time



3. Parameter weights, specific mechanisms like attention, dropout, +++



Slide from John Hewitt

Outline

1. Motivating model analysis and explanation
2. One model at multiple levels of abstraction
3. Out-of-domain evaluation sets
    1. Testing for linguistic knowledge
    2. Testing for task heuristics
4. Influence studies and adversarial examples
    1. What part of my input led to this answer?
    2. How could I minimally modify this input to change the answer?
5. Analyzing representations
    1. Correlation in “interpretable” model components
    2. Probing studies: supervised analysis
6. Revisiting model ablations as analysis

Slide from John Hewitt

Model evaluation as model analysis

When looking at the **behavior** of a model, we're not yet concerned with **mechanisms** the model is using. We want to ask *how does model behave in situations of interest?*

- You've trained your model on some samples $(x, y) \sim D$ from some distribution.
- How does the model behave on samples from the same distribution?
  - Aka *in-domain* or *i.i.d.* (independently and identically distributed)
  - **This is your test set accuracy / F1 / BLEU**

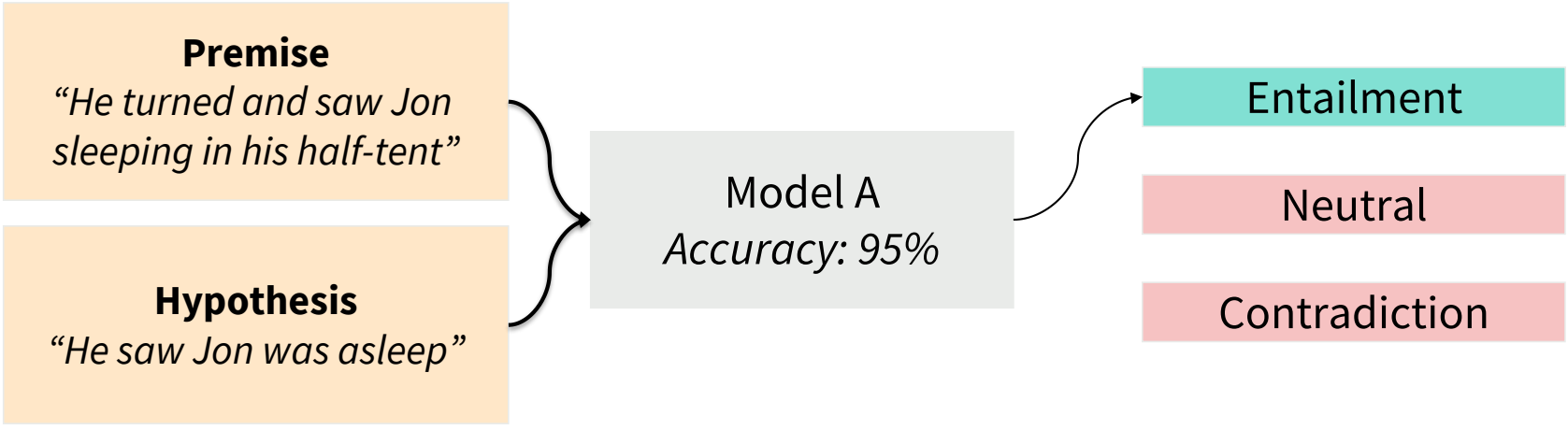
Model A		Model B
Accuracy: 95%	$\overset{?}{>}$	*Accuracy: 92%*

[Also, both models seem pretty good?]

Slide from John Hewitt

Model evaluation as model analysis in **natural language inference**

Recall the **natural language inference** task, as encoded in the Multi-NLI dataset.



[Likely to get the right answer, since the accuracy is 95%?]

[Williams et al., 2018]

Slide from John Hewitt

Model evaluation as model analysis in **natural language inference**

What if our model is using simple heuristics to get good accuracy?

A **diagnostic test set** is carefully constructed to test for a specific skill or capacity of your neural model.

For example, **HANS**: (Heuristic Analysis for NLI Systems) tests syntactic heuristics in NLI

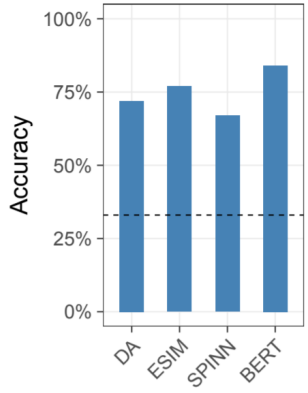
Heuristic	Definition	Example
Lexical overlap	Assume that a premise entails all hypotheses constructed from words in the premise	**The doctor** was **paid** by **the actor**. $\xrightarrow[\text{WRONG}]{}$ The doctor paid the actor.
Subsequence	Assume that a premise entails all of its contiguous subsequences.	The doctor near **the actor danced**. $\xrightarrow[\text{WRONG}]{}$ The actor danced.
Constituent	Assume that a premise entails all complete subtrees in its parse tree.	If **the artist slept**, the actor ran. $\xrightarrow[\text{WRONG}]{}$ The artist slept.

[McCoy et al., 2019]

Slide from John Hewitt

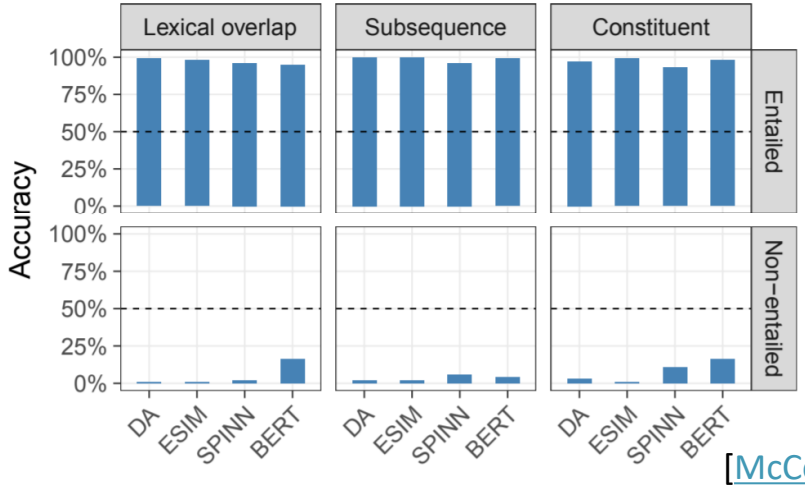
HANS model analysis in **natural language inference**

McCoy et al., 2019 took 4 strong MNLI models, with the following accuracies on the **original test set (in-domain)**



Evaluating on HANS, where syntactic heursitcs **work**, accuracy is high!

But where syntactic heuristics fail, accuracy is very very low…



[McCoy et al., 2019]

Slide from John Hewitt

Language models as linguistic test subjects

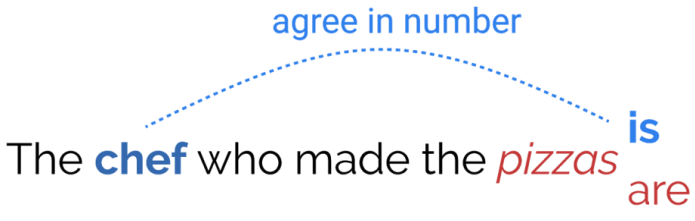
Humans

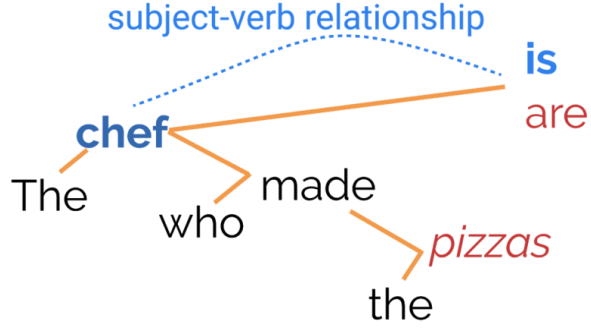
- How do we understand language behavior in humans?
- One method: *minimal pairs*. What sounds "okay" to a speaker, but doesn't with a small change?

The chef who made the pizzas is here. ← "Acceptable"

The chef who made the pizzas are here ← "Unacceptable"

Idea: English past-tense verbs *agree in number* with their subjects





[Linzen et al., 2016; Fig from Manning et al., 2020]

Slide from John Hewitt

Language models as linguistic test subjects

- What's the language model analogue of acceptability?

  The chef who made the pizzas is here. ← "Acceptable"

  The chef who made the pizzas are here ← "Unacceptable"

- Assign higher probability to the acceptable sentence in the minimal pair

  P(The chef who made the pizzas is here.) > P(The chef who made the pizzas are here)

- Just like in HANS, we can develop a **test set with carefully chosen properties**.
  - Specifically: can language models handle "attractors" in subject-verb agreement?
  - 0 Attractors: The chef is here.
  - 1 Attractor: The chef who made the pizzas is here.
  - 2 Attractors: The chef who made the pizzas and prepped the ingredients is here.
  - …

[Linzen et al., 2016]

Slide from John Hewitt

Language models as linguistic test subjects

- Kuncoro et al., 2018 train an LSTM language model on a small set of Wikipedia text.
- They evaluate it on and evaluate it *only* on sentences with specific numbers of agreement attractors.
- Numbers in this table: accuracy at predicting the correct number for the verb

Zero attractors: Easy

4 attractors: harder, but models still do pretty well!

	n=0	**n=1**	**n=2**	**n=3**	**n=4**
Random	50.0	50.0	50.0	50.0	50.0
Majority	32.0	32.0	32.0	32.0	32.0
Our LSTM, H=50	2.4	8.0	15.7	26.1	34.65
Our LSTM, H=150	1.5	4.5	9.0	14.3	17.6
Our LSTM, H=250	1.4	3.3	5.9	**9.7**	13.9
Our LSTM, H=350	**1.3**	**3.0**	**5.7**	**9.7**	**13.8**

The larger LSTMs learn subject-verb agreement better!

[Kuncoro et al., 2016]

Slide from John Hewitt

Language models as linguistic test subjects

Sample test examples for subject-verb agreement with attractors that a model got wrong

The **ship** that the player drives **has** a very high speed.

The **ship** that the player drives **have** a very high speed.

The **lead** is also rather long; 5 paragraphs **is** pretty lengthy …

The **lead** is also rather long; 5 paragraphs **are** pretty lengthy …

[Linzen et al., 2016]

Slide from John Hewitt

Careful test sets as unit test suites: CheckListing

- Small careful test sets sound like… unit test suites, but for neural networks!
- *Minimum functionality tests:* small test sets that target a specific behavior.

Test case	Expected	Predicted	Pass?
A Testing **Negation** with ***MFT***	Labels: negative, positive, neutral		
`Template: I {NEGATION} {POS_VERB} the {THING}.`			
I can't say I recommend the food.	neg	pos	X
I didn't love the flight.	neg	neutral	X
…			
Failure rate = 76.4%			

- Ribeiro et al., 2020 showed **ML engineers working on a sentiment analysis product** an interface with categories of linguistic capabilities and types of tests.
  - The engineers found a bunch of bugs (categories of high error) through this method!

[Ribeiro et al., 2020]

Slide from John Hewitt

Fitting the dataset vs learning the task

Across a wide range of tasks, high model accuracy on the in-domain test set does not imply the model will also do well on other, “reasonable” out-of-domain examples.

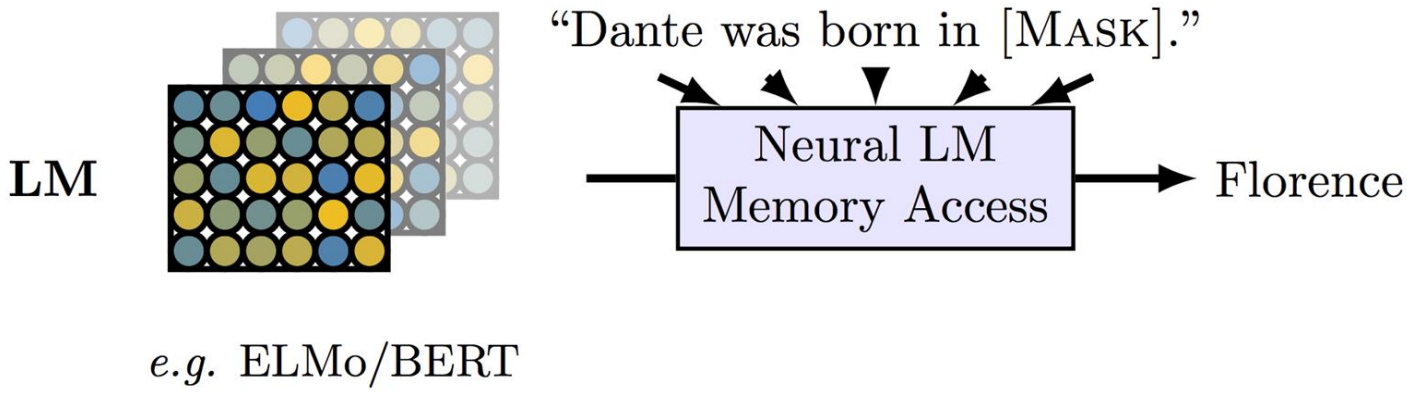
One way to think about this: models seem to be learning the *dataset* (like MNLI) not the *task* (like how humans can perform natural language inference).

[Ribeiro et al., 2020]

Slide from John Hewitt

Knowledge evaluation as model analysis

- What has a language model learned from pretraining?
- More on this later, but last lecture we saw one way of accessing some of the knowledge in the model by providing it with prompts.
- This fits into the set of *behavioral studies* we've seen so far!



[Petroni et al., 2020]

Slide from John Hewitt

Outline

1. Motivating model analysis and explanation
2. One model at multiple levels of abstraction
3. Out-of-domain evaluation sets (Your model as a probability distribution)
    1. Testing for linguistic knowledge
    2. Testing for task heuristics
4. Influence studies and adversarial examples
    1. What part of my input led to this answer?
    2. How could I minimally modify this input to change the answer?
5. Analyzing representations
    1. Correlations with simple model components
    2. Probing studies: supervised analysis
6. Revisiting model ablations as analysis

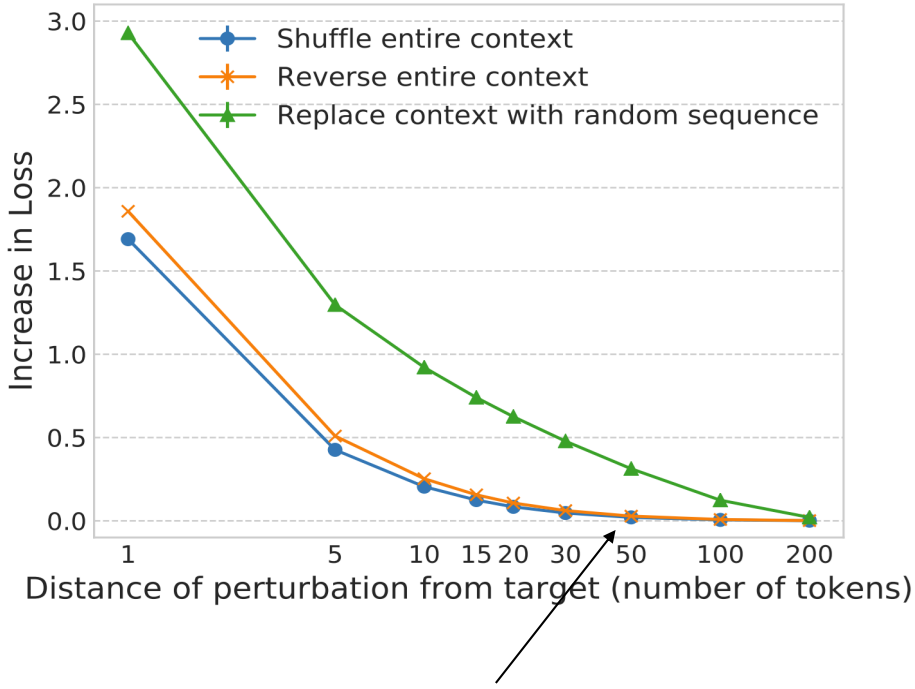
Slide from John Hewitt

Input influence: does my model *really* use long-distance context?

- We motivated LSTM language models through their theoretical ability to use long-distance context to make predictions. But how long really is the long short-term memory?

- Khandelwal et al., 2018's idea: shuffle or remove all contexts farther than k words away for multiple values of k and see at which k the model's predictions start to get worse!

- Loss is averaged across many examples.



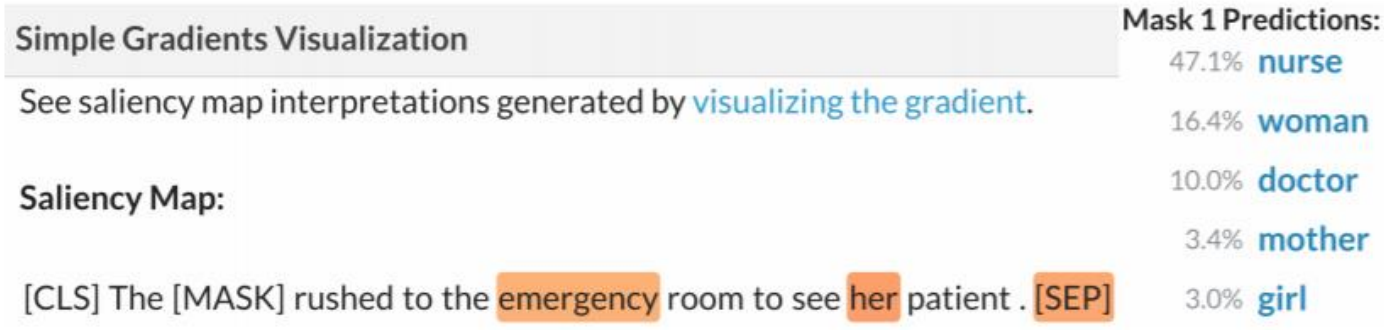
History farther than 50 words away treated as a bag of words.

[Khandelwal et al., 2018]

Slide from John Hewitt

Prediction explanations: what in the input led to this output?

- For a single example, what parts of the input led to the observed prediction?
- **Saliency maps**: a score for each input word indicating its importance to the model's prediction



- In the above example, BERT is analyzed, and interpretable words seem to contribute to the model's predictions (right).

[Simonyan et al., 2014, Wallace et al., 2019]

Slide from John Hewitt

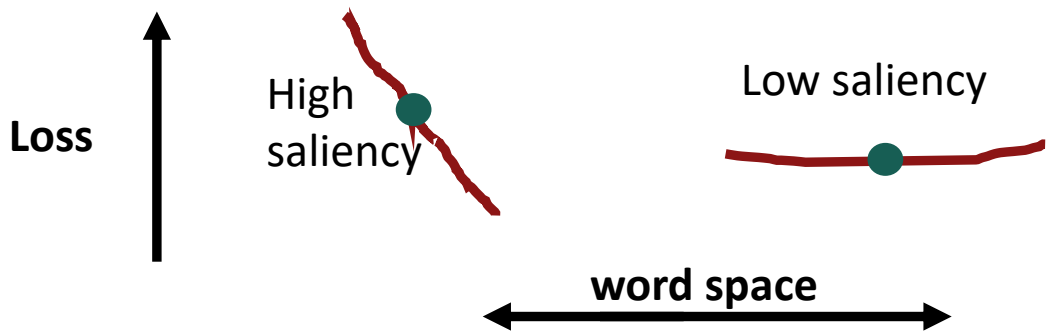
Prediction explanations: simple saliency maps

- How do we make a saliency map? Many ways to encode the intuition of "importance"
- **Simple gradient method:**

  For words $x_1, \ldots, x_n$ and the model's score for a given class (output label) $s_c(x_1, \ldots, x_n)$, take the norm of the gradient of the score w.r.t. each word:

$$\text{salience}(x_i) = ||\nabla_{x_i} s_c(x_1, \ldots, x_n)||$$

  Idea: **high gradient norm** means changing that word (locally) would affect the score a lot



[Li et al., 2016, Simonyan et al., 2014, Wallace et al., 2019]

Slide from John Hewitt

Prediction explanations: simple saliency maps

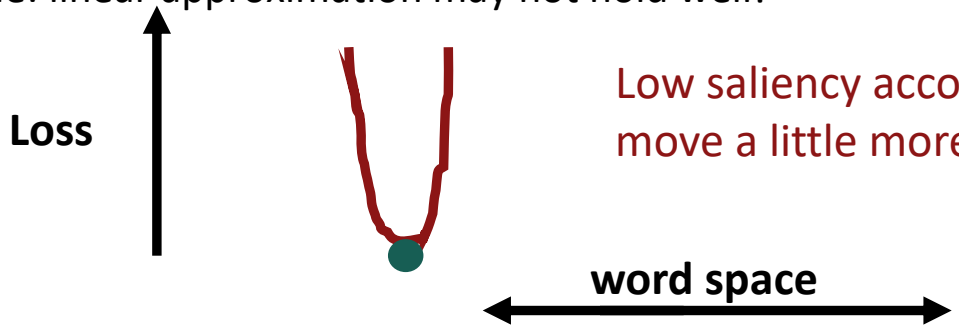
- How do we make a saliency map? Many ways to encode the intuition of "importance"
- **Simple gradient method:**

  For words $x_1, \ldots, x_n$ and the model's score for a given class (output label) $s_c(x_1, \ldots, x_n)$, take the norm of the gradient of the score w.r.t. each word:

$$\text{salience}(x_i) = |\nabla_{x_i} s_c(x_1, \ldots, x_n)|$$

Not a perfect method for saliency; many more methods have been proposed.

One issue: linear approximation may not hold well!



Low saliency according to the gradient… but move a little more and the loss skyrockets!

[Li et al., 2016, Simonyan et al., 2014, Wallace et al., 2019]

Slide from John Hewitt

Explanation by input reduction

What is the smallest part of the input I could keep and still get the same answer?

An example from SQuAD:

Passage: In 1899, John Jacob Astor IV invested $100,000 for Tesla to further develop and produce a new lighting system. Instead, Tesla used the money to fund his Colorado Springs experiments. [prediction]

Original Question: What did Tesla spend Astor's money on ?

Reduced Question did

In this example, the model had confidence 0.78 for the original question, and the same answer at confidence **0.91** for the reduced question!

[Feng et al., 2018]

Slide from John Hewitt

A method for explanation by input reduction

Idea: **run an input saliency method. Iteratively remove the most unimportant words.**

Passage: The Panthers used the San Jose State practice facility and stayed at the San Jose Marriott. The Broncos practiced at Stanford University and stayed at the Santa Clara Marriott. [prediction]

Original Question: Where did the Broncos practice for the Super Bowl ?

Steps of input reduction ↓

Where did the practice for the Super Bowl ?
Where did practice for the Super Bowl ?
Where did practice the Super Bowl ?
Where did practice the Super ?
Where did practice Super ?
did practice Super ?

[Note: beam search to find *k* least important words is an important addition]

Only here did the model stop being confident in the answer

[Feng et al., 2018]

Slide from John Hewitt

Analyzing models by breaking them

Idea: **Can we break models by making seemingly innocuous changes to the input?**

Passage: Peyton manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by **John Elway**, who led the Broncos to victory in Super Bowl XXXIII at age 38…

[prediction]

Question: What was the name of the quarterback who was 38 in Super Bowl XXXIII?

Looks good!

[Jia et al., 2017]

Slide from John Hewitt

Analyzing models by breaking them

Idea: **Can we break models by making seemingly innocuous changes to the input?**

Passage: Peyton manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38… Quarterback **Jeff Dean** had jersey number 37 in Champ Bowl XXXIV.

[prediction]

Question: What was the name of the quarterback who was 38 in Super Bowl XXXIII?

The sentence in orange hasn't changed the answer, but the model's prediction changed!

So, seems like the model wasn't performing question answering as we'd like?

[Jia et al., 2017]

Slide from John Hewitt

Analyzing models by breaking them

Idea: **Can we break models by making seemingly innocuous changes to the input?**

In the United States especially, several high-profile cases such as Debra LaFave, Pamela Rogers, and Mary Kay Letourneau have caused increased scrutiny on teacher misconduct.

(a) Input Paragraph

Q: What has been the result of this publicity?
A: increased scrutiny on teacher misconduct

(b) Original Question and Answer

Q: What haL been the result of this publicity?
A: teacher misconduct

(c) Adversarial Q & A (Ebrahimi et al., 2018)

Q: What's been the result of this publicity?
A: teacher misconduct

(d) **Semantically Equivalent Adversary**

This model's predictions look good!

This typo is annoying, but a reasonable human might ignore it.

Changing *what* to *what's* should never change the answer!

[Ribeiro et al., 2018]

Slide from John Hewitt

Are models robust to noise in their input?

Humans

"Aoccdrnig to a rscheearch at Cmabrigde Uinervtisy, it deosn't mttaer in waht oredr the ltteers in a wrod are, the olny iprmoetnt tihng is taht the frist and lsat ltteer be at the rghit pclae."

Seemingly so!

Slide from John Hewitt

Are models robust to noise in their input?

Noise of various kinds is an inevitable part of the inputs to NLP systems. How do models trained on (relatively) clean text perform when typo-like noise is added?

Belinkov and Bisk, 2018 performed a study on popular machine translation models.

BLEU scores are high on in-domain clean text

Character-swaps like we just saw break the model!

(More) natural typo noise also breaks the models.

			Synthetic				
		Vanilla	Swap	Mid	Rand	Key	Nat
French	charCNN	42.54	10.52	9.71	1.71	8.26	17.42
German	charCNN	34.79	9.25	8.37	1.02	6.40	14.02
	char2char	29.97	5.68	5.46	0.28	2.96	12.68
	Nematus	34.22	3.39	5.16	0.29	0.61	10.68
Czech	charCNN	25.99	6.56	6.67	1.50	7.13	10.20
	char2char	25.71	3.90	4.24	0.25	2.88	11.42
	Nematus	29.65	2.94	4.09	0.66	1.41	11.88

[Belinkov and Bisk, 2018]

Slide from John Hewitt

Outline

1. Motivating model analysis and explanation
2. One model at multiple levels of abstraction
3. Out-of-domain evaluation sets
    1. Testing for linguistic knowledge
    2. Testing for task heuristics
4. Influence studies and adversarial examples
    1. What part of my input led to this answer?
    2. How could I minimally modify this input to change the answer?
5. Analyzing representations
    1. Correlation in “interpretable” model components
    2. Probing studies: supervised analysis
6. Revisiting model ablations as analysis

Slide from John Hewitt

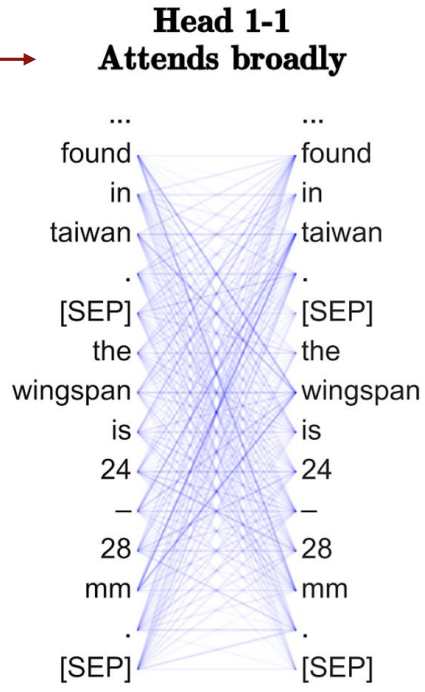
Analysis of “interpretable” architecture components

Idea: **Some modeling components lend themselves to inspection.**

For example, can we try to characterize each attention head of BERT?

Attention head 1 of layer 1.

This head performs this kind of behavior on most sentences.



[Why is “interpretable” in quotes? It’s hard to tell exactly how/whether the model is performing an interpretable function, especially deep in the network.]

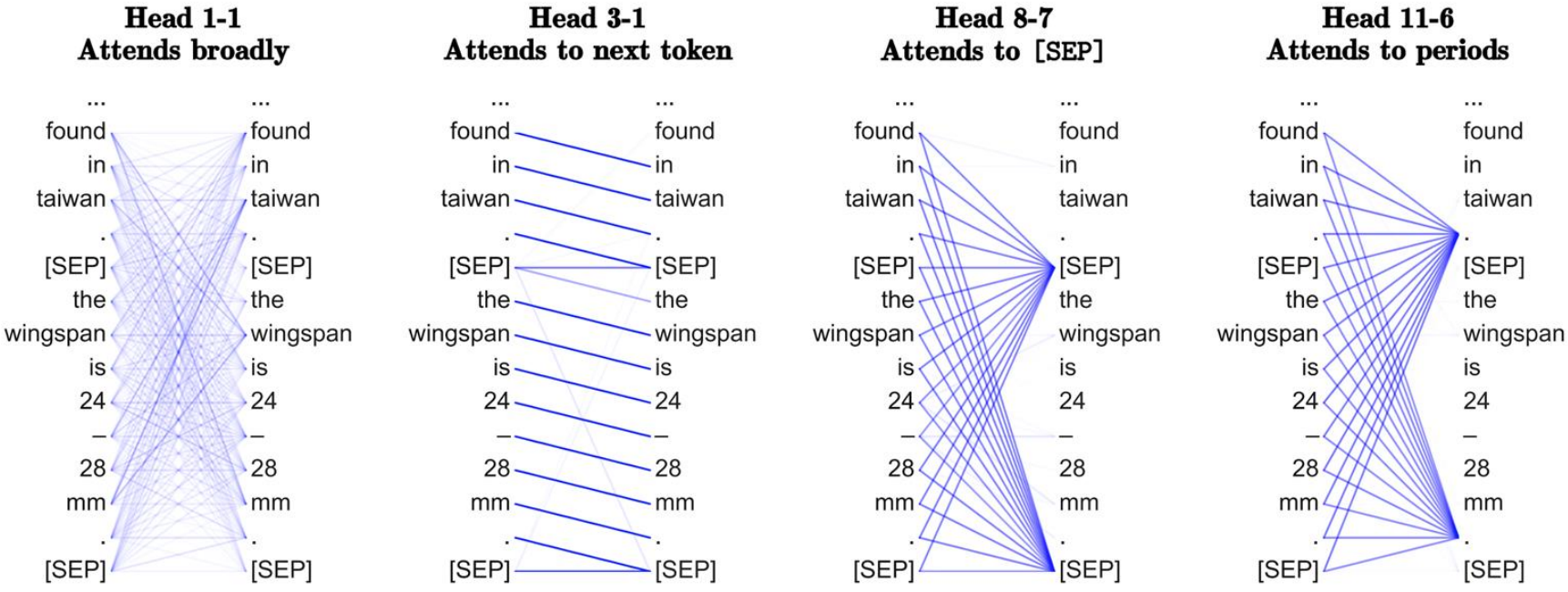
[Clark et al., 2018]

Slide from John Hewitt

Analysis of “interpretable” architecture components

Idea: **Some modeling components lend themselves to inspection.**

Some attention heads seem to perform simple operations.



[Clark et al., 2018]

Slide from John Hewitt

Analysis of “interpretable” architecture components

Idea: **Some modeling components lend themselves to inspection.**

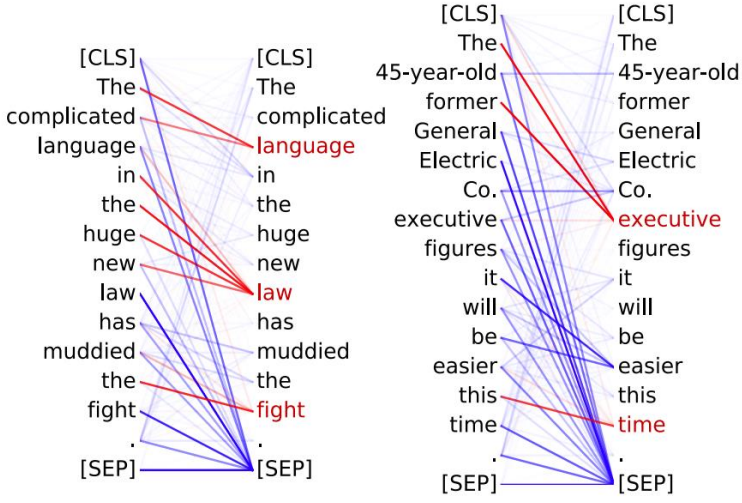
Some heads are correlated with linguistic properties!

Approximate interpretation + quantitative analysis

Model behavior

Head 8-11

- **Noun modifiers** (e.g., determiners) attend to their noun
- 94.3% accuracy at the `det` relation



[Clark et al., 2018]

Slide from John Hewitt

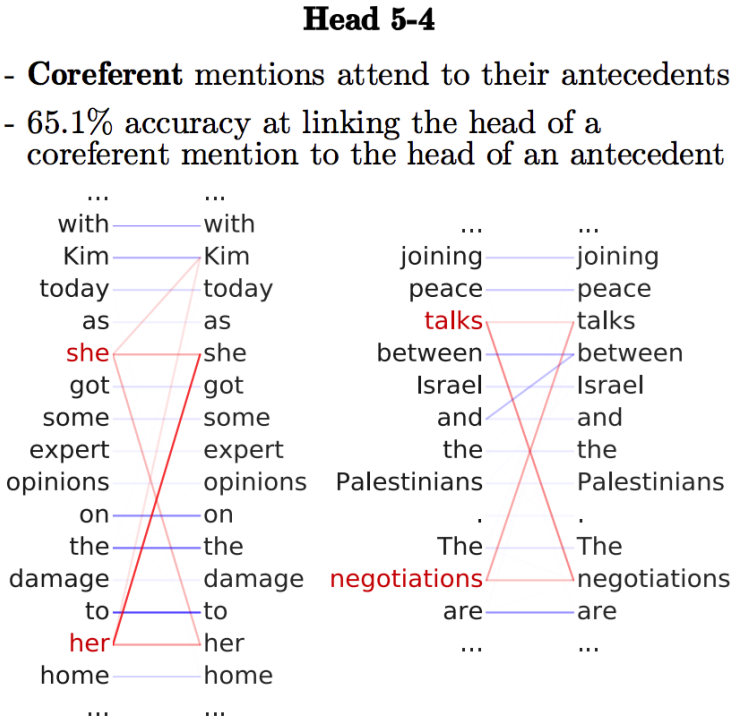
Analysis of “interpretable” architecture components

Idea: **Some modeling components lend themselves to inspection.**

We saw coreference before; one head often matches coreferent mentions!

Approximate interpretation + quantitative analysis

Model behavior



[Clark et al., 2018]

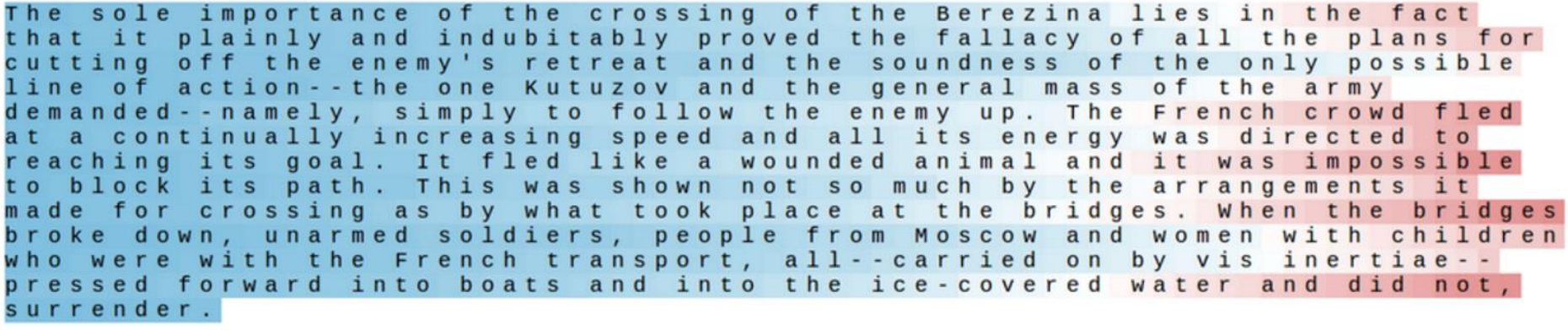
Slide from John Hewitt

Analysis of “interpretable” architecture components

Idea: **Individual hidden units can lend themselves to an interpretable meaning.**

This model: a character-level LSTM language model.

Cell sensitive to position in line:



Here, “cell” refers to a single dimension of the cell state of the LSTM.

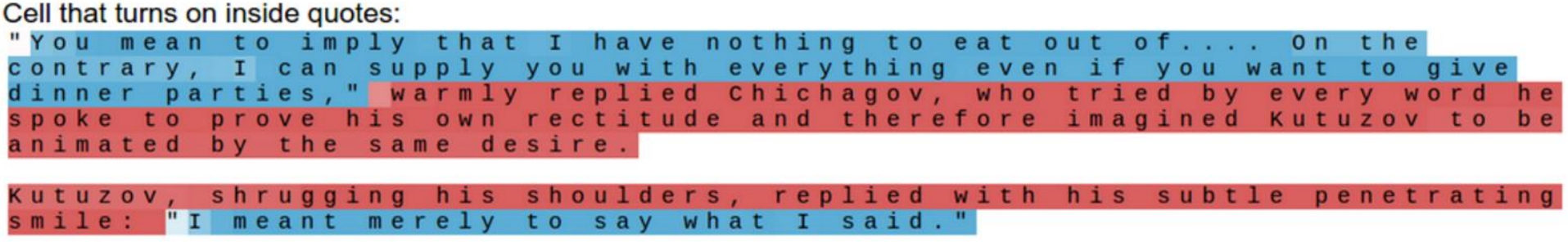
[Karpathy et al., 2016]

Slide from John Hewitt

Analysis of “interpretable” architecture components

Idea: **Individual hidden units can lend themselves to an interpretable meaning.**

This model: a character-level LSTM language model.



Here, “cell” refers to a single dimension of the cell state of the LSTM.

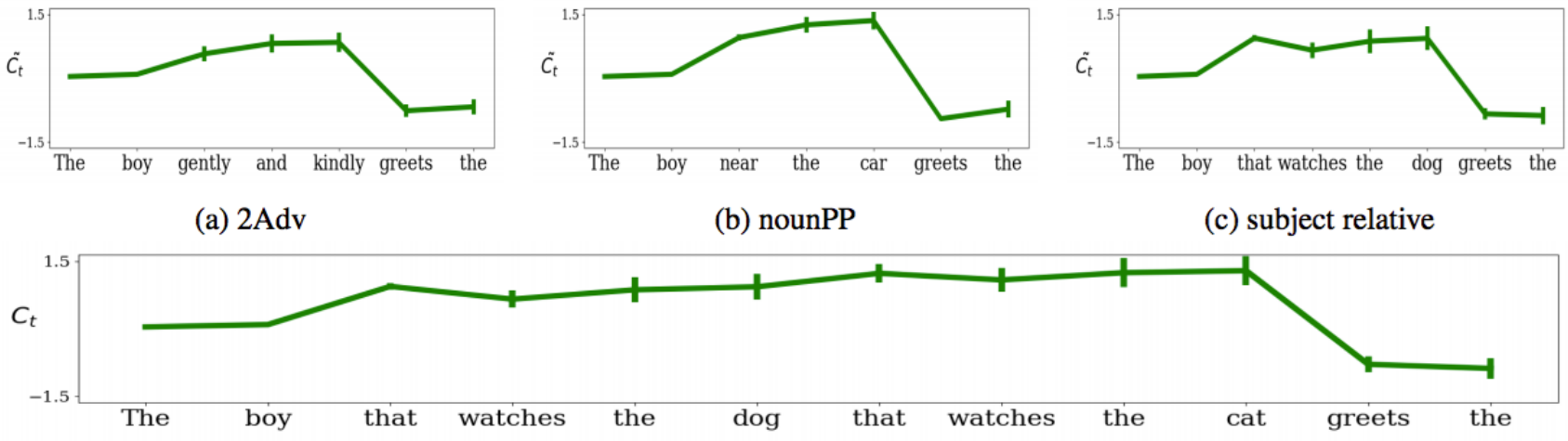
[Karpathy et al., 2016]

Slide from John Hewitt

Analysis of “interpretable” architecture components

Idea: Let’s go back to **subject-verb agreement**. What’s the mechanism by which LSTMs solve the task?

This model: a word-level LSTM language model.

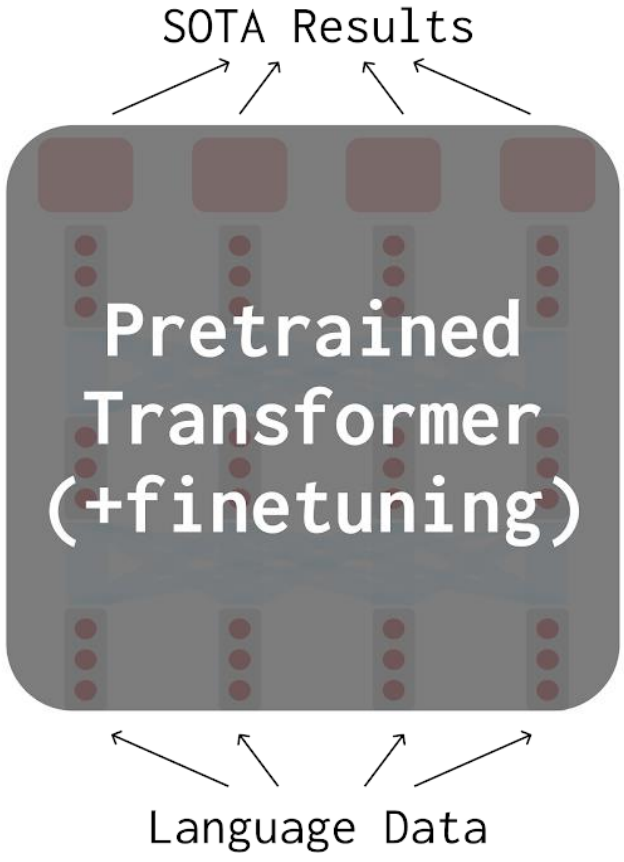


This is neuron 1150 in the LSTM, which seems to track the scope of the grammatical number of the subject! Removing this unit harms subject-verb agreement much more than removing a random unit.

[Lakretz et al., 2019]

Slide from John Hewitt

Probing: supervised analysis of neural networks



Premise:

Pretrained Transformers provide wildly **general-purpose language representations**

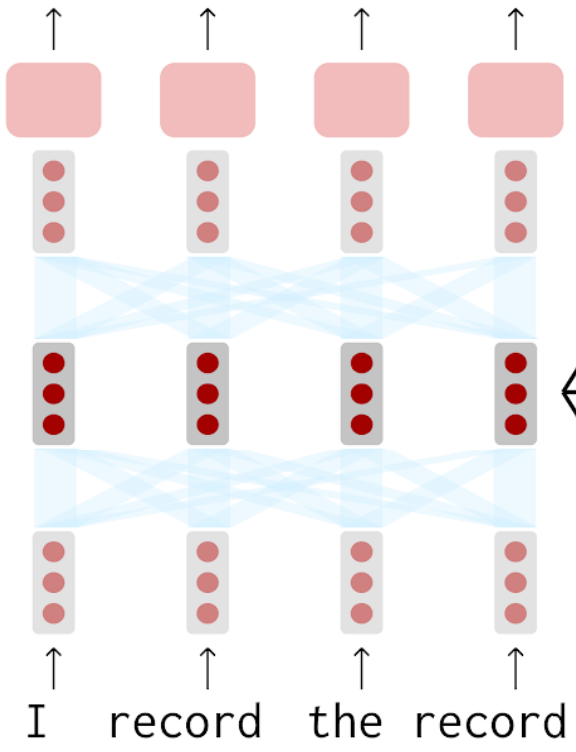
Question:

What do their representations **encode about language**?

[SOTA means "state-of-the-art," the best method for a given problem.]

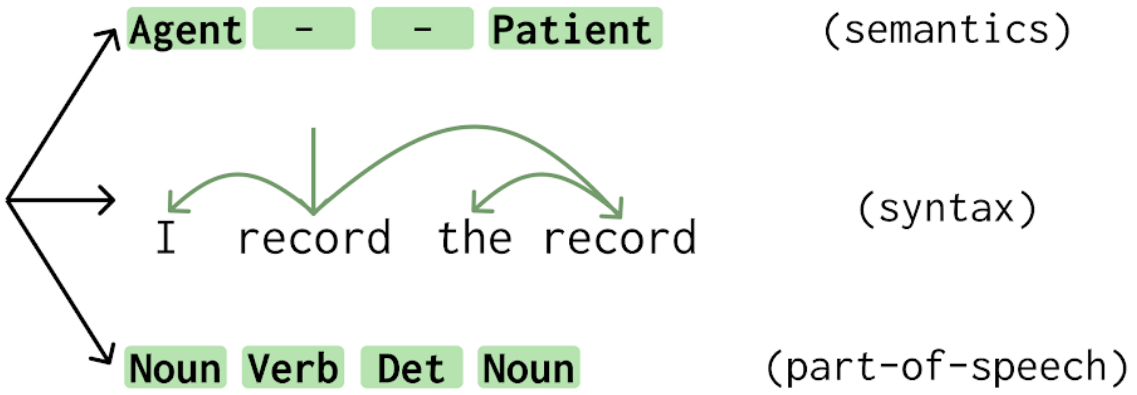
Slide from John Hewitt

Probing: supervised analysis of neural networks



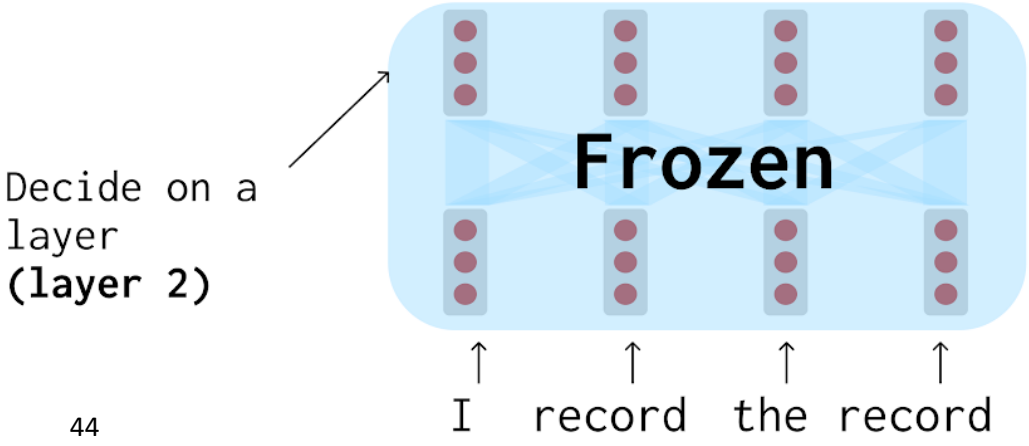
Idea:

What do pretrained representations encode about **linguistic properties** that we have annotated data for?



Slide from John Hewitt

Probing: supervised analysis of neural networks



Slide from John Hewitt

Probing: supervised analysis of neural networks

Let's take a second to think more about probing.

- We have some property *y* (like part-of-speech)
- We have the model's word representations at a fixed layer: $h_1, \ldots, h_T$, where $h_i \in \mathbb{R}^d$, where the words are at indices $1, \ldots, T$.
- We have a function family F like the set of *linear models* or *1-layer feed-forward networks (with fixed hyperparmaters.)*
- We **freeze** the parameters of the model, so it's not finetuned. Then, we train our probe, a function

$$\hat{y} \sim f(h_i) \qquad f \in F$$

The extent to which we can predict y from h_i is a measure of the accessibility of that feature in the representation.

- This helps gain a rough understanding into how the model processes its inputs.
- Also may help in the search for causal mechanisms.

Slide from John Hewitt

Probing: supervised analysis of neural networks

BERT (and other pretrained LMs) make some linguistic properties predictable to very high accuracy with a simple linear probe.

Syntactic roles

Part-of-speech

Named entity recognition

Pretrained Representation			POS			
	Avg.	CCG	PTB	EWT	Chunk	NER
BERT (base, cased) best layer	84.09	93.67	96.95	95.21	92.64	82.71
BERT (large, cased) best layer	**85.07**	**94.28**	96.73	**95.80**	**93.64**	**84.44**
GloVe (840B.300d)	59.94	71.58	90.49	83.93	62.28	53.22
Previous state of the art (without pretraining)	83.44	94.7	97.96	95.82	95.77	91.38

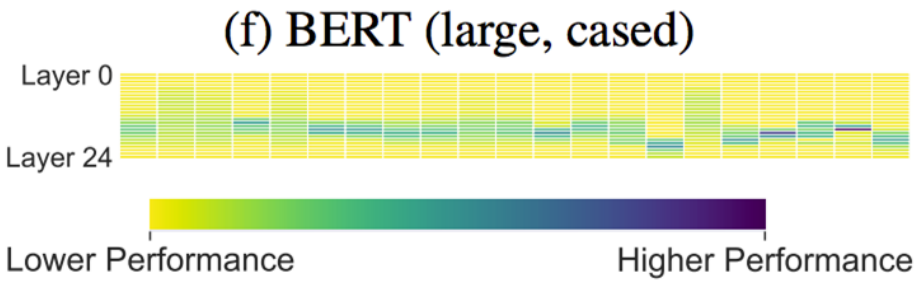
Slide from John Hewitt

Layerwise trends of probing accuracy

- Across a wide range of linguistic properties, the middle layers of BERT yield the best probing accuracies.

Input words here

MLM objective here



Consistently best a bit past the mid point

Figure 3: A visualization of layerwise patterns in task performance. Each column represents a probing task, and each row represents a contextualizer layer.

[Liu et al., 2019]

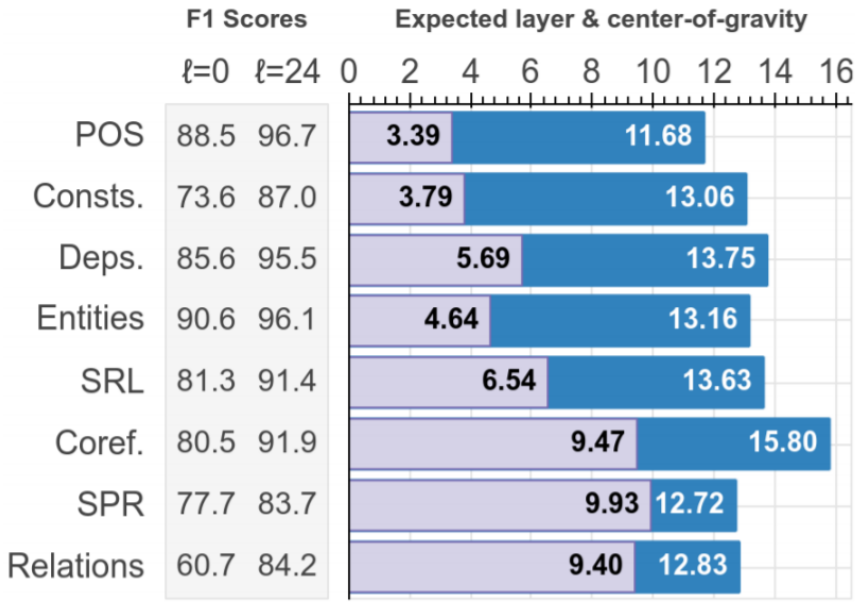
Slide from John Hewitt

Layerwise trends of probing accuracy

- Increasingly abstract linguistic properties are more accessible later in the network.

Increasing depth in the network

Increasing abstractness of linguistic properties

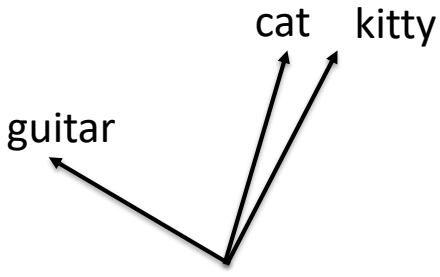


[Tenney et al., 2019]

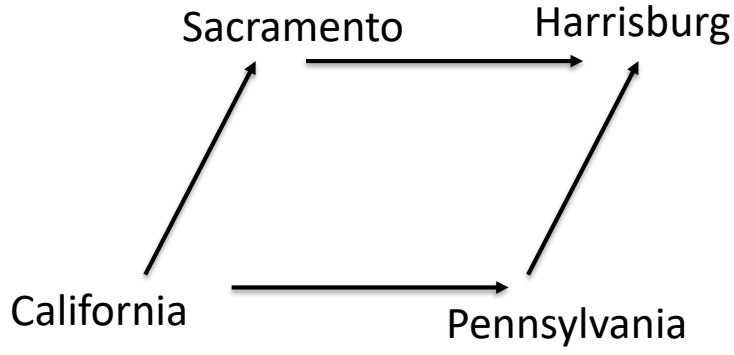
Slide from John Hewitt

Emergent simple structure in neural networks

- Recall word2vec, and the intuitions we built around its vectors



We interpret **cosine similarity** as *semantic similarity.*



Some *relationships* are encoded as **linear offsets**

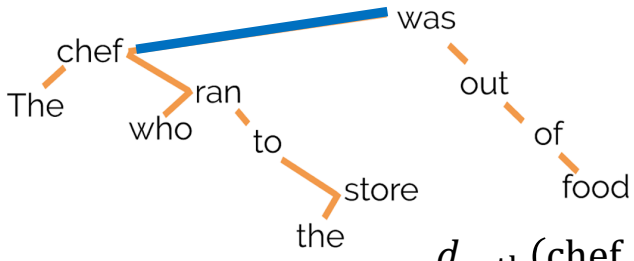
- It's hard to the dimensions of word2vec vectors, but it's fascinating that interpretable concepts approximately map onto simple functions of the vectors

[Mikolov et al., 2013]

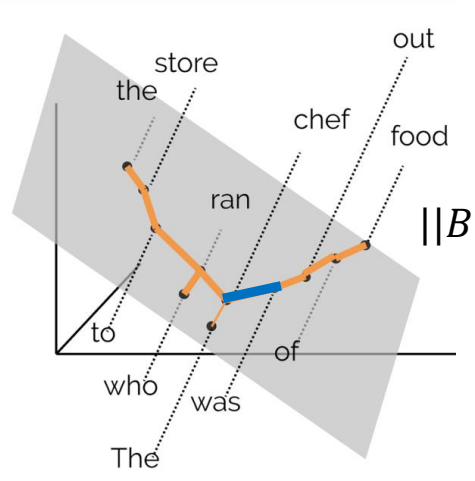
Slide from John Hewitt

Probing: trees simply recoverable from BERT representations

- Recall dependency parse trees. They describe underlying syntactic structure in sentences.
- Hewitt and Manning 2019 show that BERT models make dependency parse **tree structure** easily accessible.



$d_{\text{path}}(\text{chef}, \text{was}) = 1$



$||B(h_{\text{chef}} - h_{\text{was}})||_2^2 \approx 1$

$d_{\text{path}}(w_1, w_2)$

Tree path distance: the number of edges in the path between the words

$||B(h_{w_1} - h_{w_2})||_2^2$

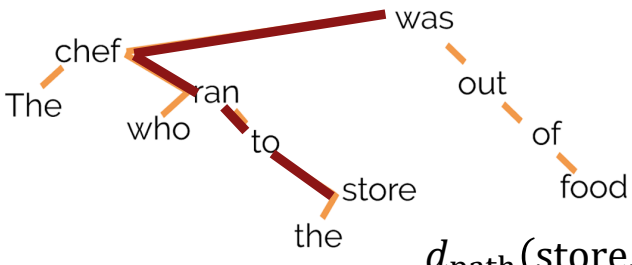
Squared Euclidean distance of BERT vectors after transformation by the (probe) matrix B.

[Hewitt and Manning, 2019]

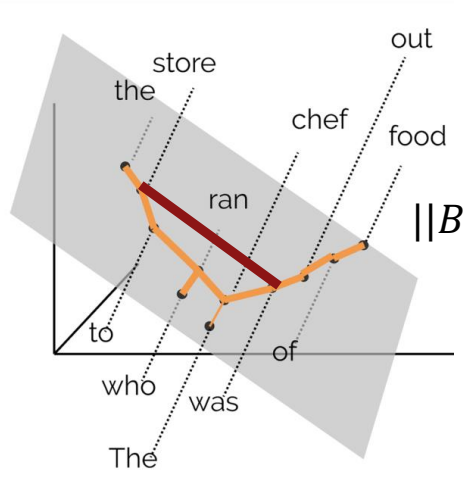
Slide from John Hewitt

Probing: trees simply recoverable from BERT representations

- Recall dependency parse trees. They describe underlying syntactic structure in sentences.
- Hewitt and Manning 2019 show that BERT models make dependency parse **tree structure** easily accessible.



$d_{\text{path}}(\text{store}, \text{was}) = 4$



$||B(h_{\text{store}} - h_{\text{was}})||_2^2 \approx 4$

$d_{\text{path}}(w_1, w_2)$

Tree path distance: the number of edges in the path between the words

$||B(h_{w_1} - h_{w_2})||_2^2$

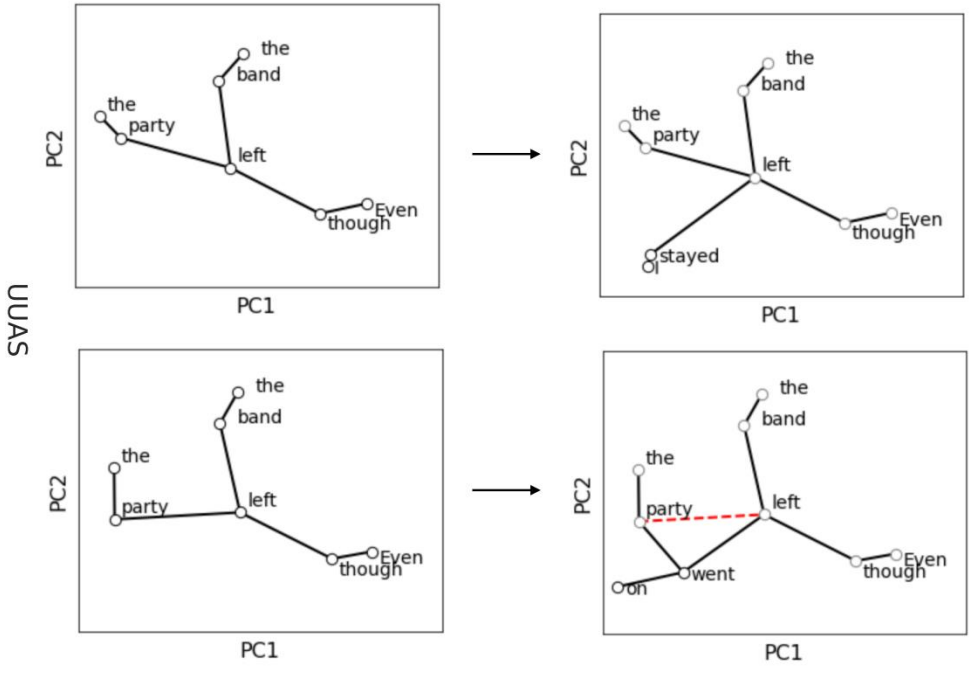
Squared Euclidean distance of BERT vectors after transformation by the (probe) matrix B.

[Hewitt and Manning, 2019]

Slide from John Hewitt

Probing: helping design hypotheses for causal analysis

- The structural probe was used to design and test hypotheses as to how LMs *incrementally* parse sentences (and causally intervene on network behavior)!



Even though the band left the party I stayed

Even though the band left the party went on

[Eisape et al., 2022]

Slide from John Hewitt

Final thoughts on probing and correlation studies

- Probing shows that properties are accessible to your probe family, not that they're used by the neural model you're studying.
- Correlation studies (like attention maps) likewise.
- For example:
  - Hewitt and Liang, 2019 show that under certain conditions, probes can achieve high accuracy on random labels.
  - Ravichander et al., 2021 show that probes can achieve high accuracy on a property even when the model is trained to know the property isn't useful.
- Some efforts (Vig et al., 2020, Eisape et al., 2022) have gone towards causal studies. Interesting and harder!

[Hewitt and Liang, 2019, Ravichander et al., 2021, Vig et al., 2020]

Slide from John Hewitt

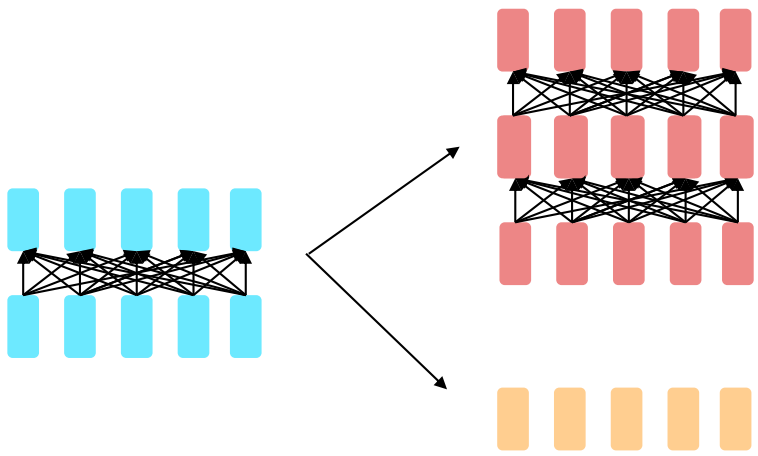
Outline

1. Motivating model analysis and explanation
2. One model at multiple levels of abstraction
3. Out-of-domain evaluation sets
    1. Testing for linguistic knowledge
    2. Testing for task heuristics
4. Influence studies and adversarial examples
    1. What part of my input led to this answer?
    2. How could I minimally modify this input to change the answer?
5. Analyzing representations
    1. Correlation in “interpretable” model components
    2. Probing studies: supervised analysis
6. Revisiting model ablations as analysis

Slide from John Hewitt

Recasting model tweaks and ablations as analysis

- Consider the usual neural network improvement process:
  - You have a network, which works okay.
  - You see whether you can tweak it in simple ways to improve it.
  - You see whether you can remove any complex things and have it still work as well.
- This can be thought of as a kind of model analysis!

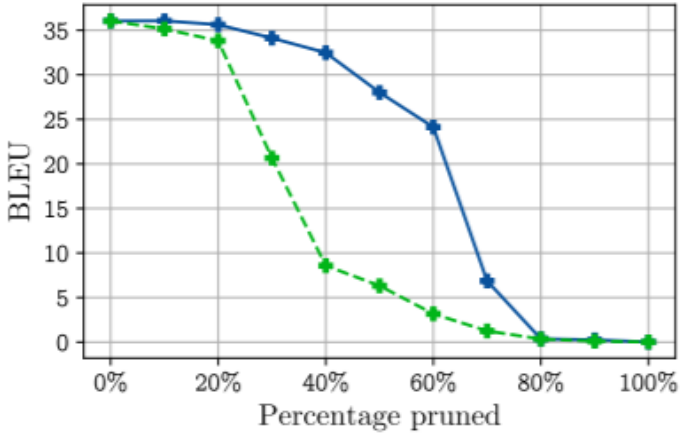


[Would it be better for this part of my model to be deeper? Or can I get away with making it shallower?]

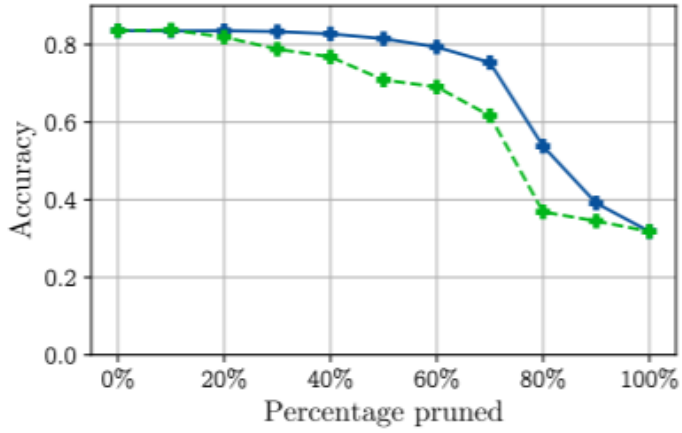
Slide from John Hewitt

Ablation analysis: do we need all these attention heads?

- Michel et al., 2019 train transformers with multi-headed attention on machine translation and natural language inference.
- After training, they find many attention heads can be **removed** with no drop in accuracy!



(a) Evolution of BLEU score on `newstest2013` when heads are pruned from WMT.



(b) Evolution of accuracy on the MultiNLI-matched validation set when heads are pruned from BERT.

[Green and blue lines indicate two different ways to choose the order to prune attention heads.]

[Michel et al., 2019]

Slide from John Hewitt

What's the right layer order for a Transformer?

- We saw that Transformer models are sequences of layers
  - Self-attention → Feed-forward → Self-attention → Feed-forward → ….
  - (Layer norm and residual connections omitted)
- Press et al., 2019 asked, why? Is there a better ordering of self-attention and feed-forward layers?
- Here's that sequence of layers again:

Achieves 18.40 perplexity on a language modeling benchmark

Achieves 17.96 perplexity on a language modeling benchmark

Many self-attention layers first

Many feed-forward layers last

[Press et al., 2019]

Slide from John Hewitt

Parting thoughts

- Neural models are complex, and difficult to characterize. A single accuracy metric doesn't cut it.
- We struggle to find intuitive descriptions of model behaviors, but we have a many tools at many levels of abstraction to give insight.
- Engage critically when someone claims a (neural) NLP model is interpretable – in what ways is it interpretable? In what ways is it still opaque?
- Bring this analysis and explanation way of thinking with you to your model building efforts even if analysis isn't your main goal.

Good luck on finishing your final projects! We're really appreciative of your efforts.

Slide from John Hewitt