

Data: Collection, Annotation, and Biases

Antoine Bosselut



Announcements

- **Course Feedback:** Indicative Feedback this week! Please fill it out!
- **Assignment 1:** Due this past Sunday, 17/03/2024 at 11:59 PM
- **Assignment 2: Released Today.** Due Friday, 07/04/2024 at 11:59 PM
 - Assignment Q&A Sessions: **27.03.2024**, 1 PM & **28.03.2024**, 2 PM
- **Project Group Registration:** Please fill out the form ASAP.

Assignment 1 Feedback

- Difficulty — content, tools, environment.
 - Background. Timing. Benefit.
- Assignments aren't designed for you to immediately translate instruction into code
 - Not closed form — might be multiple ways to solve the problem
 - If there's ambiguity, we're open to different strategies (as long as it's not just "ChatGPT solved it for me")
- **"Ed-iquette"**: Let's remember to be kind to each other on the discussion board!
 - Please read Ed before asking questions — lots of repeat inquiries; not efficient use of TA time!
 - **FYI:** "Anonymous" posting is anonymous to other students, not to the course staff
 - Also, if you post long code snippets, please consider making your post private

Assignment 1 Next Steps

- Assignment 1 Grading will begin at the end of the week
 - typically, we let people exhaust late day opportunities before beginning the grading
- Course registration deadline was the final time to drop course
 - Life happens; priorities change
- NA Option
 - If we don't grade any of your course deliverables (assignments, project), you will receive **NA**
 - If we do grade any course deliverables, you will get at least a **1**
 - If you turned in Assignment 1, but would now prefer to get **NA** for the course, message us by end of the week.

Changes to Assignment Sessions

- Both TAs that organise an assignment will go to both sessions going forward
- To streamline, please ask questions on Ed ahead of time
 - We'll create a special tag for Assignment session questions
 - Please vote on questions you'd like to see answered
 - TAs will focus on most upvoted questions (likely to be conceptual questions)
- For very specific questions, keep using Ed.

Assignment 2

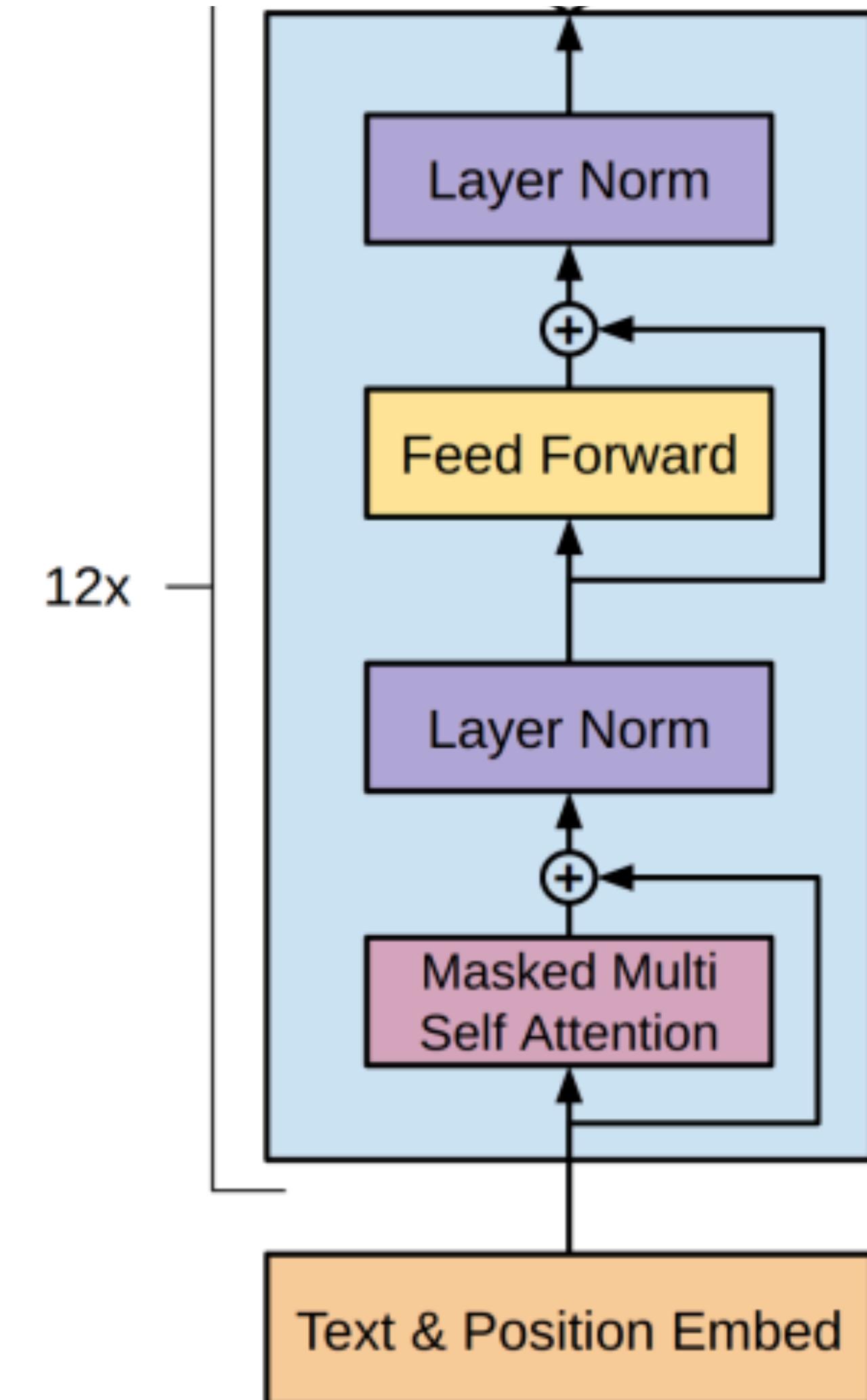
- After today, you will have seen all of the content you need to do A2
- Due: **07.04.2024**
 - Take a look early!
 - Due date is at end of break to give you maximum flexibility, rather than making the due date over Easter Weekend. **You are welcome to submit sooner!**
 - **There will be no extensions to this due date.** You are welcome to use your late days!
- There's a collaborative element to it where you have to coordinate with someone else from the class.

Today's Outline

- **Lecture**
 - **Quick Recap:** Finetuning
 - **Data Annotation:** Process, Biases, and effect on Fine-tuning
- **Tomorrow:** Exercise Session
 - **Review of Week 4 Exercise Session:** Finetuning pretrained models
 - **Week 5 Exercise Session:** Robustness & Prompting

GPT: Generative Pretrained Transformer

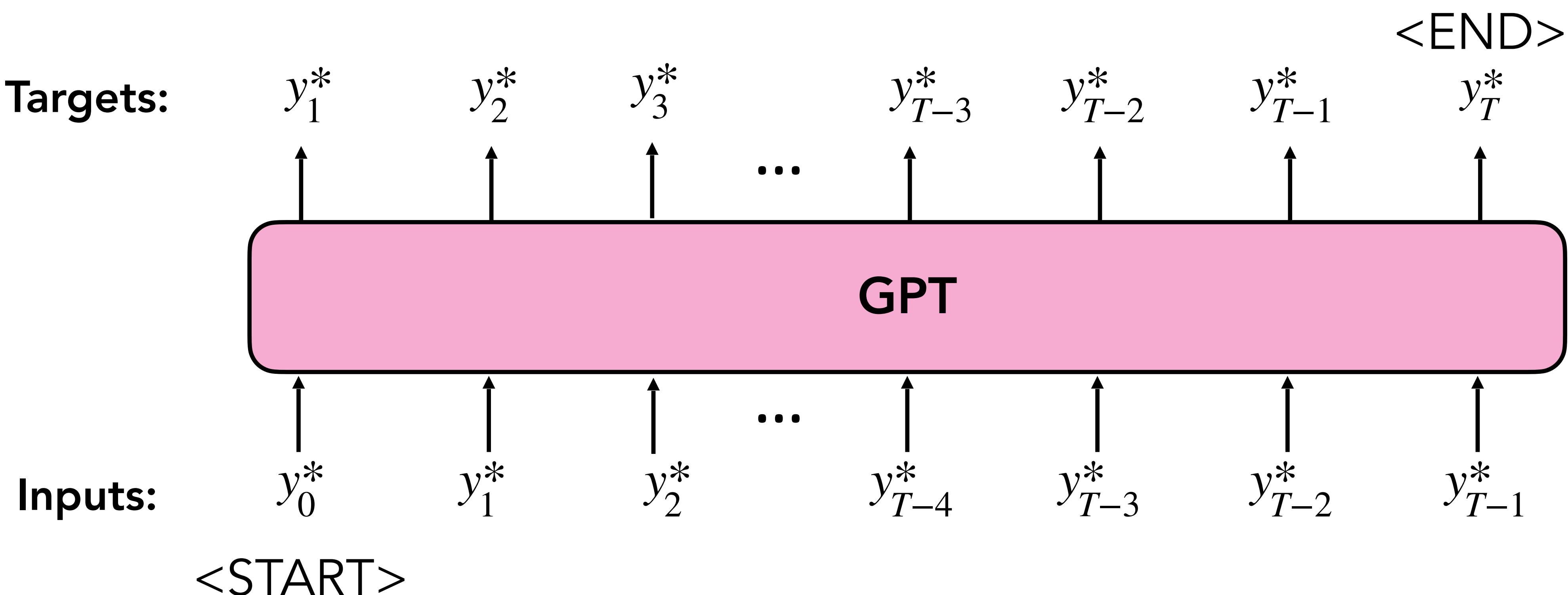
- Called a *decoder* transformer
- GPT block mixes design of encoder and decoder from original transformer
- Uses masked multi-headed self-attention (decoder)
 - Can't see future
- No cross-attention; only computes a self-attention over its history in each block (encoder)



Pretraining

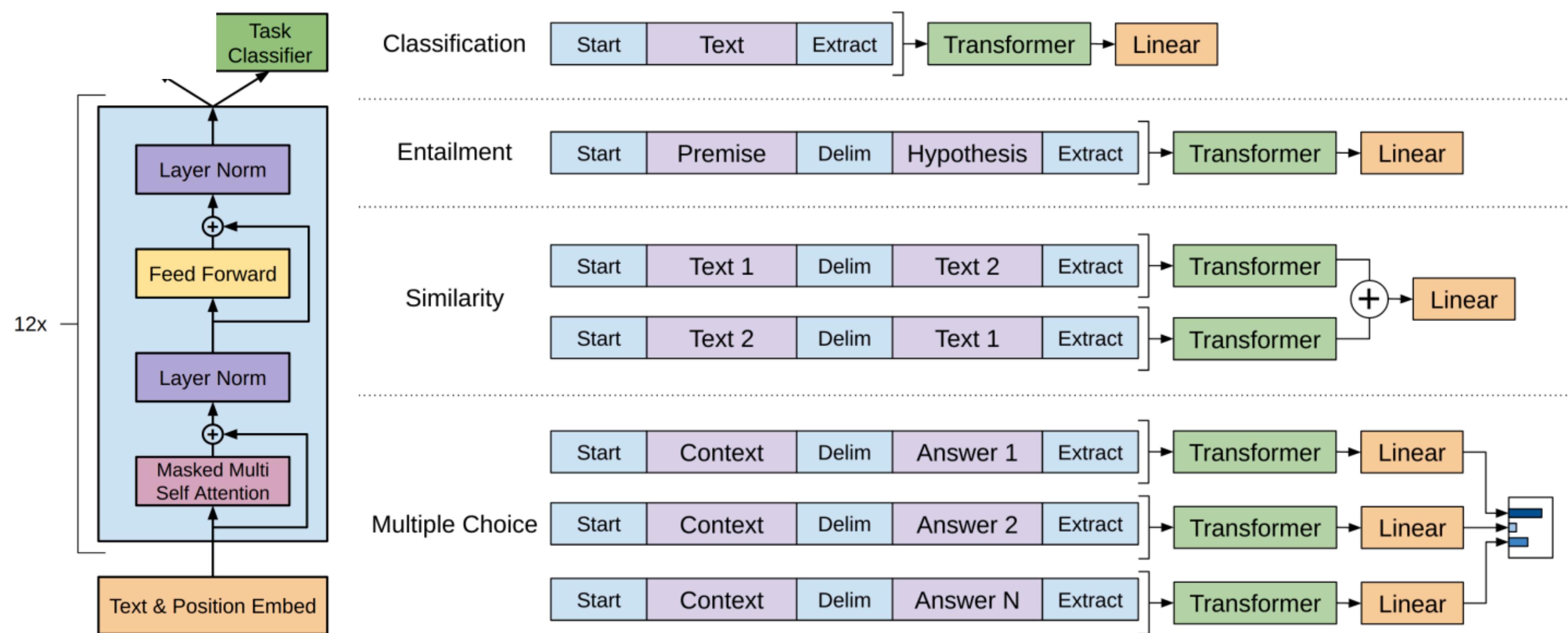
- Minimize the negative log probability of the gold* sequences in your dataset

$$\mathcal{L} = - \sum_{t=1}^T \log P(y_t^* | \{y_s^*\}_{s < t})$$



Fine-tuning

- After pre-training, model can be fine-tuned by training on individual datasets
- Pretrained model used as initialisation for training on individual tasks



Question

What's an advantage of a full pretrained language model compared to pretrained word embeddings ?

Question

Words have different meanings in different contexts!

- 1) Chico Ruiz made a spectacular **play** on Alusik's grounder {. . . }
- 2) Olivia De Havilland signed to do a Broadway **play** for Garson {. . . }
- 3) Kieffer was commended for his ability to hit in the clutch , as well as his all-round excellent **play** {. . . }
- 4) {. . . } they were actors who had been handed fat roles in a successful **play** {. . . }
- 5) Concepts **play** an important role in all aspects of cognition {. . . }

Question

What's an advantage of the word embeddings algorithms we learned compared to GPT ?

They learn representations using future tokens too !

Where might this be useful ?

Can we get the best of both worlds ?

Pretraining: Two Approaches

(Causal, Left-to-right)
Language Modeling

*I really enjoyed the movie we
watched on _____*



OpenAI

(Radford et al., 2018, 2019, many others)

Masked
Language Modeling

*I really enjoyed the _____ we
watched on Saturday!*



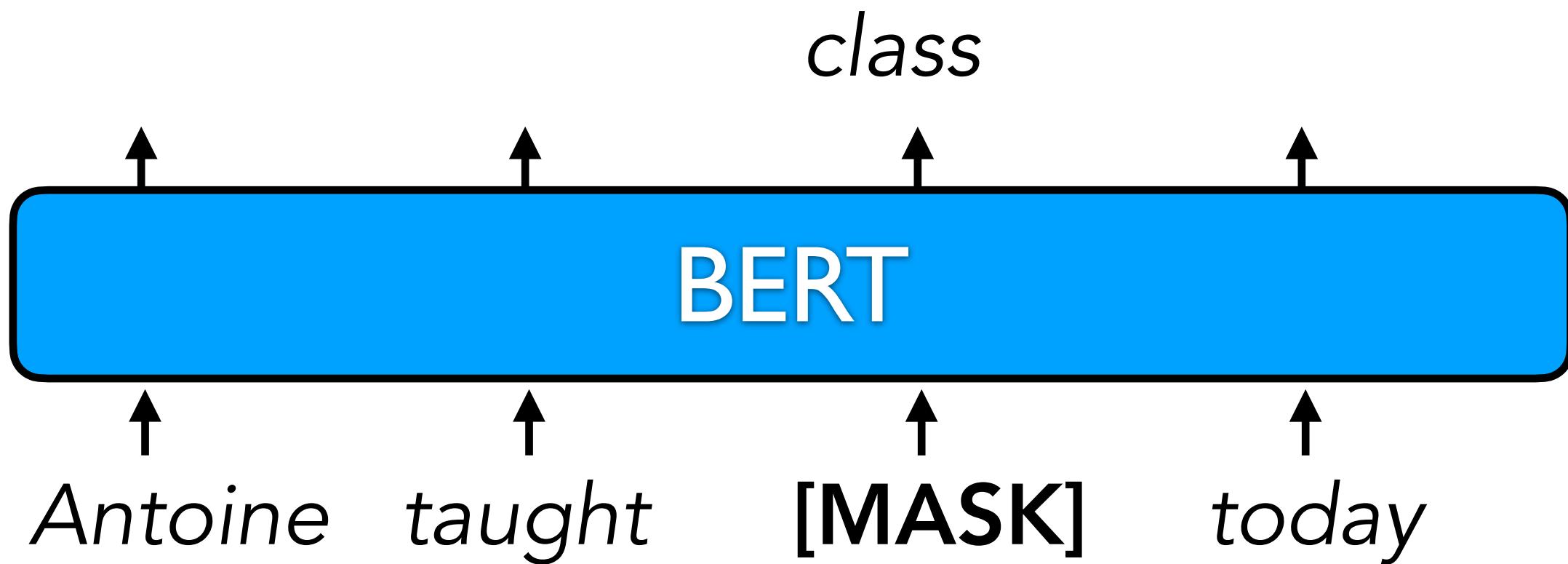
(Devlin et al., 2018; Liu et al., 2020)

Masked Language Modeling (BERT)

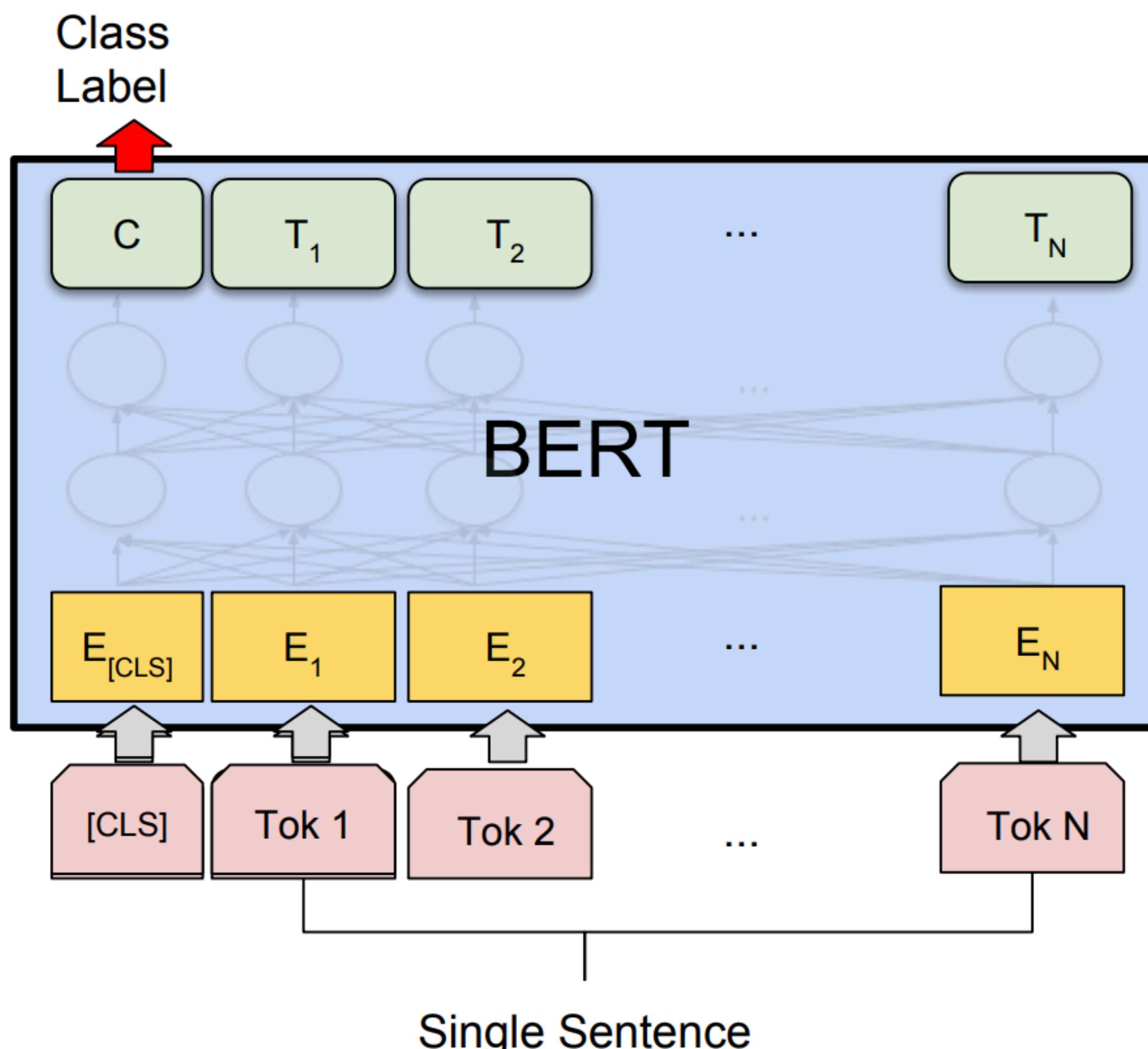
- **Training:** take a sequence of text, and predict 15% of the tokens

- **When predicting:**

- Replace input token with [MASK]
- Predict the masked token at the output
- Similar to CBOW

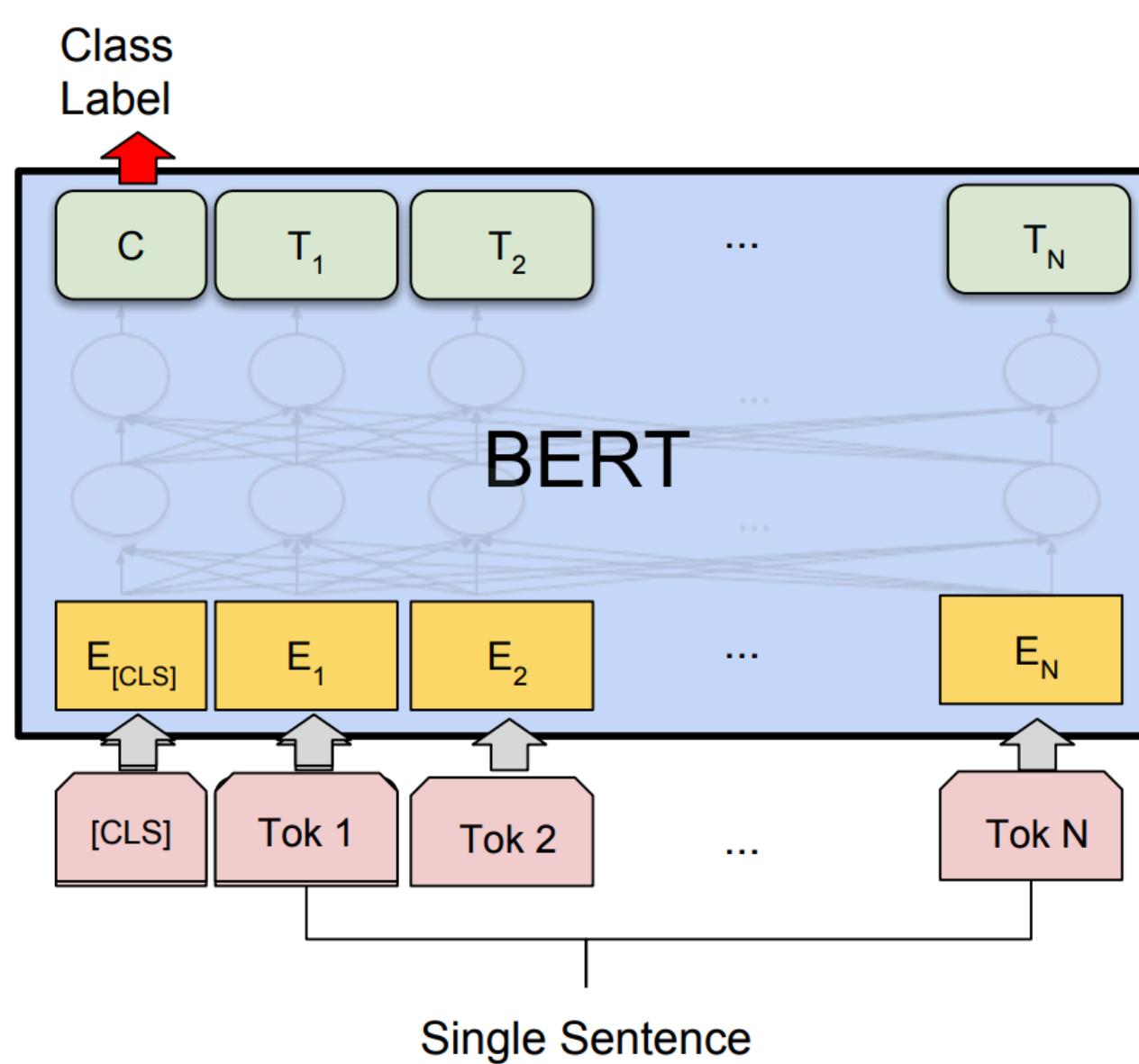


Fine-tuning BERT

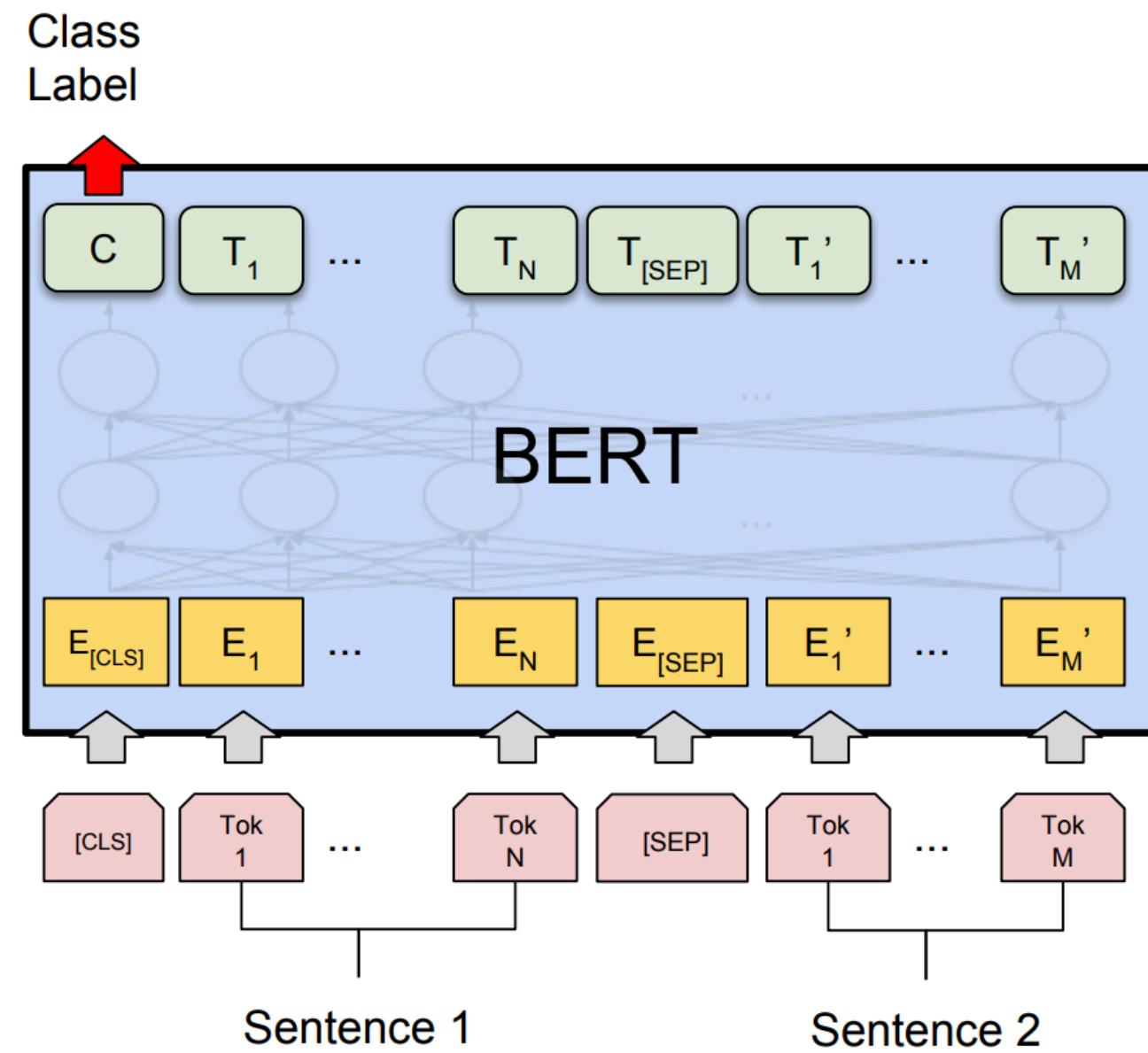


- Done after BERT has been pretrained (no more pretraining objectives)
- Select a task with supervised data (i.e., classification for sentiment analysis)
- Prepend a special **token** **[CLS]** to the front of the sequence to classify
- **Learn** to classify the **output embedding** for this **token**
- **During fine-tuning**, we update the parameters of the BERT model to learn the task

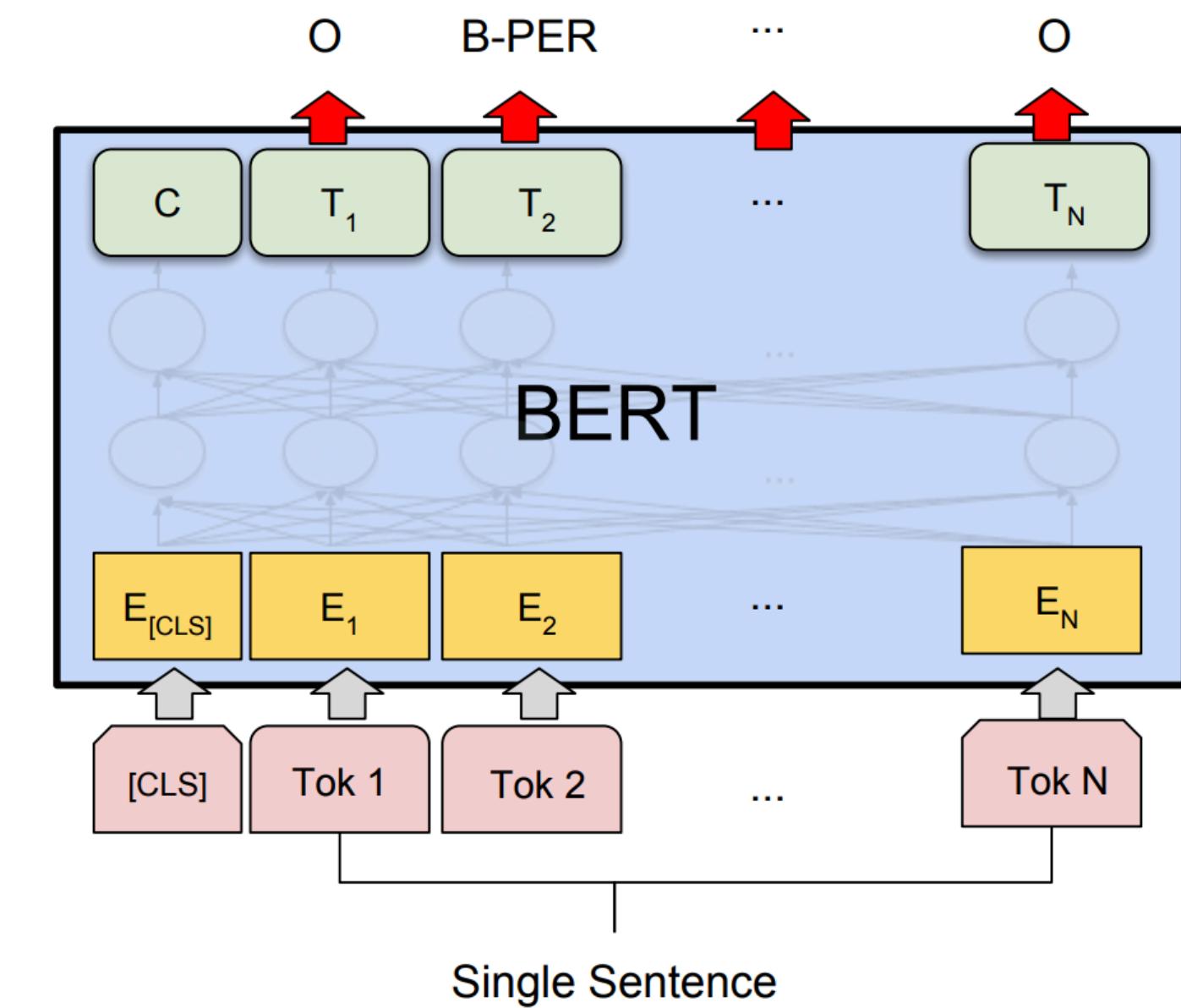
Single model starting point for many tasks



(b) Single Sentence Classification Tasks:
SST-2, CoLA



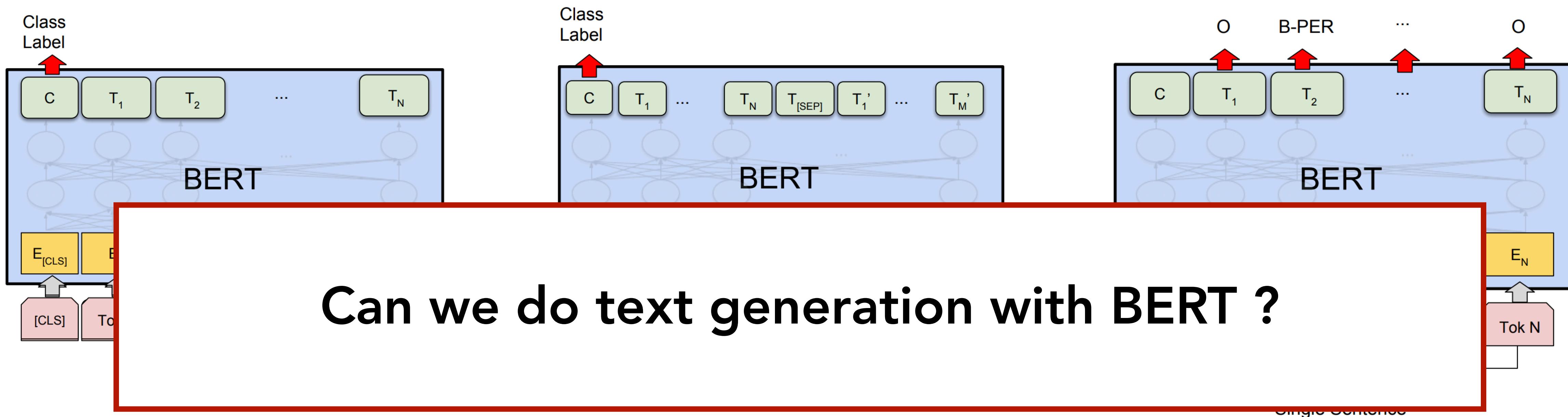
(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG



(d) Single Sentence Tagging Tasks:
CoNLL-2003 NER

- Re-using the same pretrained BERT model for fine-tuning on many tasks:
 - **Classification:** Take [CLS] output embedding as input features to classification model
 - **Sequence labeling:** Take output embedding for each token and classify individually

Single model starting point for many tasks



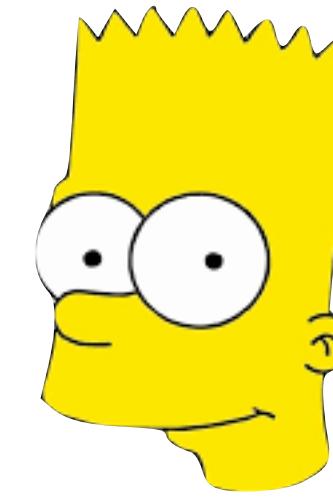
(b) Single Sentence Classification Tasks:
SST-2, CoLA

(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG

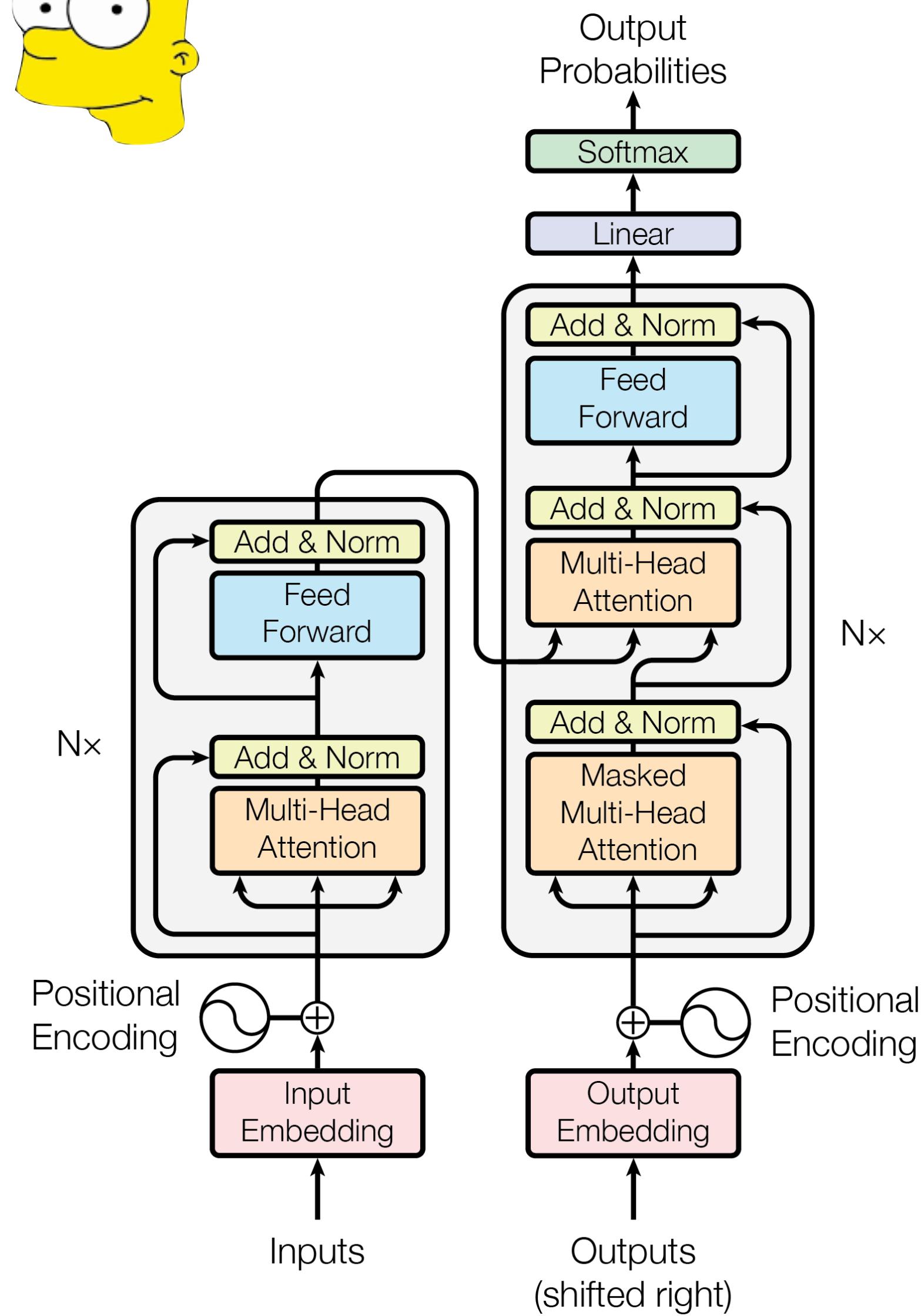
(d) Single Sentence Tagging Tasks:
CoNLL-2003 NER

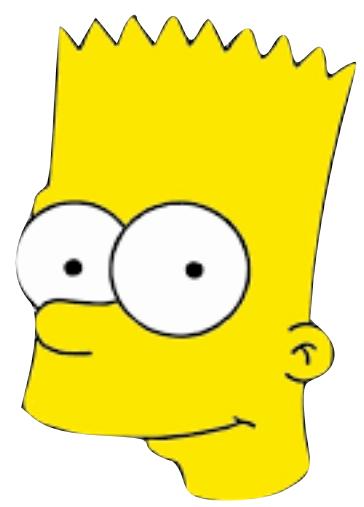
- Re-using the same pretrained BERT model for fine-tuning on many tasks:
 - **Classification:** Take [CLS] output embedding as input features to classification model
 - **Sequence labeling:** Take output embedding for each token and classify individually

BART



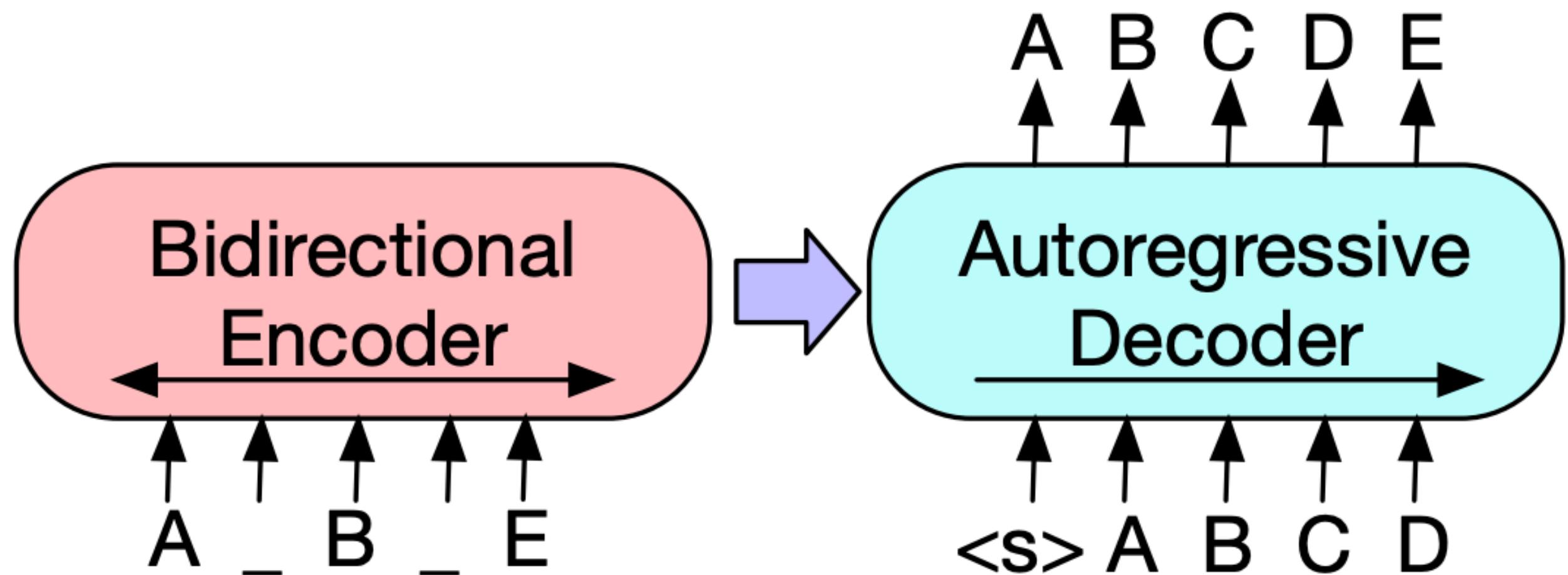
- Classic transformer architecture
- Bidirectional encoder feeds into autoregressive decoder
- Cross-attention layers in decoder are back!
- **BART-base**: 6-layers each in encoder and decoder; 140M parameters
- **BART-large**: 12 layers each in encoder and decoder; 400M parameters





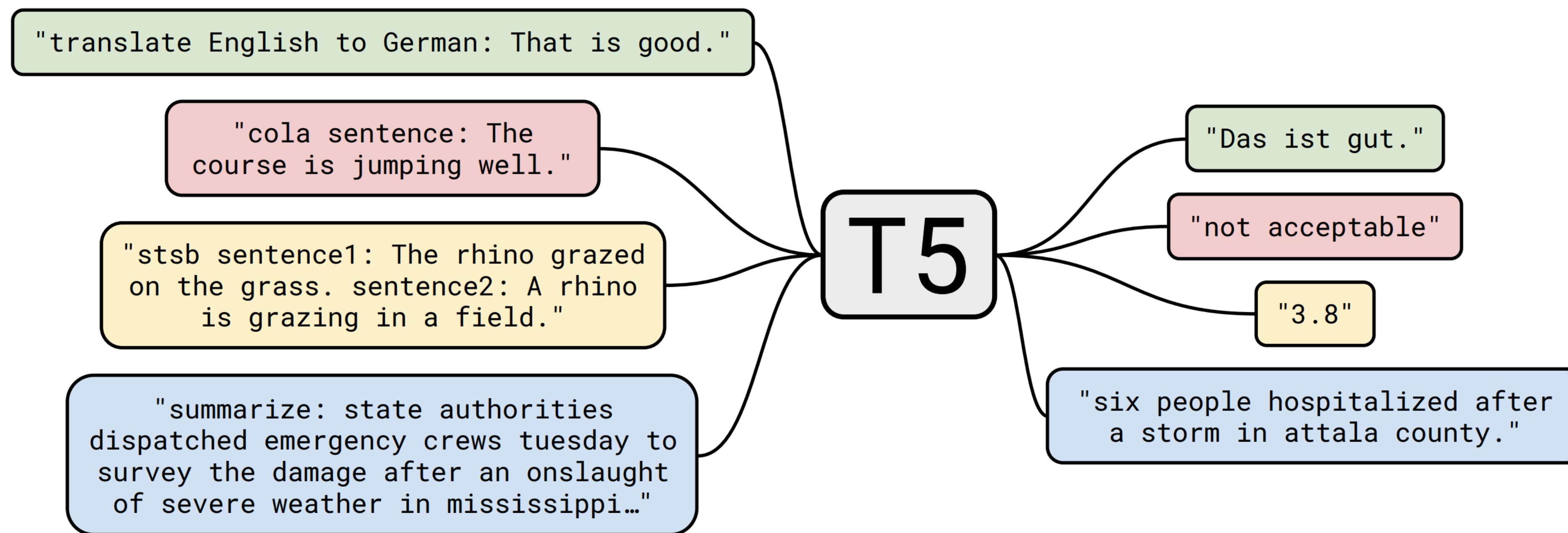
BART Pretraining

- Pretraining BART combines elements of BERT and GPT!
- **BERT-style:** input texts corrupted before they are passed to bidirectional encoder
- **GPT-style:** model is trained with a language modelling objective in the decoder: predict the next word!



T5

- Similar idea as BART: Any problem can be cast as sequence-to-sequence



Recap

- **Contextual representations:** Let us model words and sequences conditioned on the context around them
- **ELMo:** Based on bidirectional LSTMs. **Good for pretrained embeddings.**
- **GPT:** Uses a transformer decoder. **Good for generating text as a language model.**
- **BERT:** Uses a transformer encoder. **Good for classification and sequence labelling.**
- **BART + T5:** Pretraining sequence-to-sequence transformer models. **Extendable to all task types!**

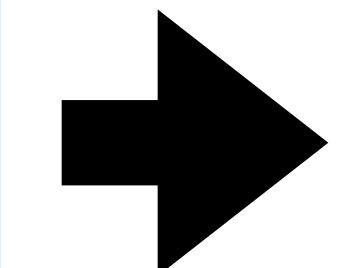
Transfer Learning

Pretraining

Learn embeddings that can be used to seed a downstream model (ELMo)

-or-

Learn a model that can be fine-tuned for many downstream tasks (GPT, BERT)



Fine-tuning

Design a new model architecture whose embeddings are initialised with pretrained embeddings. Train this model on a task of interest

- or -

Take a pretrained model and train it further on data from a task of interest

Transfer Learning

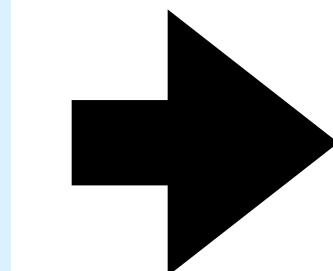
Pretraining

Uses simple training objectives

Requires tons of data

Resultant model often not useful yet

Slow & expensive; can often only do once



Fine-tuning

Done on smaller datasets

Trained on data with a more complex structure

Resultant model applied to task of interest

Typically cheaper; can afford multiple runs, hyper parameter tuning, etc.

Benchmark Performance: SuperGLUE

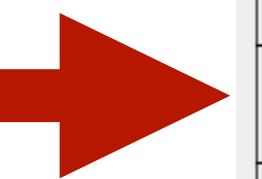
Rank	Name	Model	URL	Score	BoolQ	CB	COPA	MultiRC	ReCoRD	RTE	WiC	WSC	AX-b	AX-g	
1	JDExplore d-team	Vega v2		91.3	90.5	98.6/99.2	99.4	88.2/62.4	94.4/93.9	96.0	77.4	98.6	-0.4	100.0/50.0	
+	2	Liam Fedus	ST-MoE-32B		91.2	92.4	96.9/98.0	99.2	89.6/65.8	95.1/94.4	93.5	77.7	96.6	72.3	96.1/94.1
+	3	Microsoft Alexander v-team	Turing NLR v5		90.9	92.0	95.9/97.6	98.2	88.4/63.0	96.4/95.9	94.1	77.1	97.3	67.8	93.3/95.5
+	4	ERNIE Team - Baidu	ERNIE 3.0		90.6	91.0	98.6/99.2	97.4	88.6/63.2	94.7/94.2	92.6	77.4	97.3	68.6	92.7/94.7
+	5	Yi Tay	PaLM 540B		90.4	91.9	94.4/96.0	99.0	88.7/63.6	94.2/93.3	94.1	77.4	95.9	72.9	95.5/90.4
+	6	Zirui Wang	T5 + UDG, Single Model (Google Brain)		90.4	91.4	95.8/97.6	98.0	88.3/63.0	94.2/93.5	93.0	77.9	96.6	69.1	92.7/91.9
+	7	DeBERTa Team - Microsoft	DeBERTa / TuringNLVR4		90.3	90.4	95.7/97.6	98.4	88.2/63.7	94.5/94.1	93.2	77.5	95.9	66.7	93.3/93.8
	8	SuperGLUE Human Baselines	SuperGLUE Human Baselines		89.8	89.0	95.8/98.9	100.0	81.8/51.9	91.7/91.3	93.6	80.0	100.0	76.6	99.3/99.7
+	9	T5 Team - Google	T5		89.3	91.2	93.9/96.8	94.8	88.1/63.3	94.1/93.4	92.5	76.9	93.8	65.6	92.7/91.9

T5 model at #9

Humans at #8

7 models better
than humans!!!

On GLUE,
22 models
better than
humans!!!



Rank	Name	Model	URL	Score	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m	MNLI-mm	QNLI	RTE	WNLI	AX
1	Microsoft Alexander v-team	Turing ULR v6	🔗	91.3	73.3	97.5	94.2/92.3	93.5/93.1	76.4/90.9	92.5	92.1	96.7	93.6	97.9	55.4
2	JDExplore d-team	Vega v1		91.3	73.8	97.9	94.5/92.6	93.5/93.1	76.7/91.1	92.1	91.9	96.7	92.4	97.9	51.4
3	Microsoft Alexander v-team	Turing NLR v5	🔗	91.2	72.6	97.6	93.8/91.7	93.7/93.3	76.4/91.1	92.6	92.4	97.9	94.1	95.9	57.0
4	DIRL Team	DeBERTa + CLEVER		91.1	74.7	97.6	93.3/91.1	93.4/93.1	76.5/91.0	92.1	91.8	96.7	93.2	96.6	53.3
5	ERNIE Team - Baidu	ERNIE	🔗	91.1	75.5	97.8	93.9/91.8	93.0/92.6	75.2/90.9	92.3	91.7	97.3	92.6	95.9	51.7
6	AliceMind & DIRL	StructBERT + CLEVER	🔗	91.0	75.3	97.7	93.9/91.9	93.5/93.1	75.6/90.8	91.7	91.5	97.4	92.5	95.2	49.1
7	DeBERTa Team - Microsoft	DeBERTa / TuringNLRv4	🔗	90.8	71.5	97.5	94.0/92.0	92.9/92.6	76.2/90.8	91.9	91.6	99.2	93.2	94.5	53.2
8	HFL iFLYTEK	MacALBERT + DKM		90.7	74.8	97.0	94.5/92.6	92.8/92.6	74.7/90.6	91.3	91.1	97.8	92.0	94.5	52.6
9	PING-AN Omni-Sinitic	ALBERT + DAAF + NAS		90.6	73.5	97.2	94.0/92.0	93.0/92.4	76.1/91.0	91.6	91.3	97.5	91.7	94.5	51.2
10	T5 Team - Google	T5	🔗	90.3	71.6	97.5	92.8/90.4	93.1/92.8	75.1/90.6	92.2	91.9	96.9	92.8	94.5	53.1
11	Microsoft D365 AI & MSR AI & GATECH	MT-DNN-SMART	🔗	89.9	69.5	97.5	93.7/91.6	92.9/92.5	73.9/90.2	91.0	90.8	99.2	89.7	94.5	50.2
12	Huawei Noah's Ark Lab	NEZHA-Large		89.8	71.7	97.3	93.3/91.0	92.4/91.9	75.2/90.7	91.5	91.3	96.2	90.3	94.5	47.9
13	LG AI Research	ANNA	🔗	89.8	68.7	97.0	92.7/90.1	93.0/92.8	75.3/90.5	91.8	91.6	96.0	91.8	95.9	51.8
14	Zihang Dai	Funnel-Transformer (Ensemble B10-10-10H1024)	🔗	89.7	70.5	97.5	93.4/91.2	92.6/92.3	75.4/90.7	91.4	91.1	95.8	90.0	94.5	51.6
15	ELECTRA Team	ELECTRA-Large + Standard Tricks	🔗	89.4	71.7	97.1	93.1/90.7	92.9/92.5	75.6/90.8	91.3	90.8	95.8	89.8	91.8	50.7
16	David Kim	2digit LANet		89.3	71.8	97.3	92.4/89.6	93.0/92.7	75.5/90.5	91.8	91.6	96.4	91.1	88.4	54.6
17	倪仕文	DropAttack-RoBERTa-large		88.8	70.3	96.7	92.6/90.1	92.1/91.8	75.1/90.5	91.1	90.9	95.3	89.9	89.7	48.2
18	Microsoft D365 AI & UMD	FreeLB-RoBERTa (ensemble)	🔗	88.4	68.0	96.8	93.1/90.8	92.3/92.1	74.8/90.3	91.1	90.7	95.6	88.7	89.0	50.1
19	Junjie Yang	HIRE-RoBERTa	🔗	88.3	68.6	97.1	93.0/90.7	92.4/92.0	74.3/90.2	90.7	90.4	95.5	87.9	89.0	49.3
20	Shiwen Ni	ELECTRA-large-M (bert4keras)		88.3	69.3	95.8	92.2/89.6	91.2/91.1	75.1/90.5	91.1	90.9	93.8	87.9	91.8	48.2
21	Facebook AI	RoBERTa	🔗	88.1	67.8	96.7	92.3/89.8	92.2/91.9	74.3/90.2	90.8	90.2	95.4	88.2	89.0	48.7
22	Microsoft D365 AI & MSR AI	MT-DNN-ensemble	🔗	87.6	68.4	96.5	92.7/90.3	91.1/90.7	73.7/89.9	87.9	87.4	96.0	86.3	89.0	42.8
23	GLUE Human Baselines	GLUE Human Baselines	🔗	87.1	66.4	97.8	86.3/80.8	92.7/92.6	59.5/80.4	92.0	92.8	91.2	93.6	95.9	-

Is natural language understanding solved?

No! So why is this happening?

Where does our data come from?

What is a benchmark?

- What is a benchmark?
 - A **benchmark** is a collection of **datasets** (or a single dataset) designed to evaluate the performance of a model
- What is a dataset?
 - A **dataset** is a manifestation of a **task** using various input-output pairs
- What is a task?
 - A **task** is an instantiation of a problem, consisting of an input space (what does the typical input look like?) and an output space (what are the labels?) that define the mapping between them

What is a benchmark?

Rank	Name	Model	URL	Score	BoolQ	CB	COPA	MultiRC	ReCoRD	RTE	WiC	WSC	AX-b	AX-g
1	JDExplore d-team	Vega v2		91.3	90.5	98.6/99.2	99.4	88.2/62.4	94.4/93.9	96.0	77.4	98.6	-0.4	100.0/50.0
+	2 Liam Fedus	ST-MoE-32B		91.2	92.4	96.9/98.0	99.2	89.6/65.8	95.1/94.4	93.5	77.7	96.6	72.3	96.1/94.1
3	Microsoft Alexander v-team	Turing NLR v5		90.9	92.0	95.9/97.6	98.2	88.4/63.0	96.4/95.9	94.1	77.1	97.3	67.8	93.3/95.5
4	ERNIE Team - Baidu	ERNIE 3.0		90.6	91.0	98.6/99.2	97.4	88.6/63.2	94.7/94.2	92.6	77.4	97.3	68.6	92.7/94.7
5	Yi Tay	PaLM 540B		90.4	91.9	94.4/96.0	99.0	88.7/63.6	94.2/93.3	94.1	77.4	95.9	72.9	95.5/90.4
+	6 Zirui Wang	T5 + UDG, Single Model (Google Brain)		90.4	91.4	95.8/97.6	98.0	88.3/63.0	94.2/93.5	93.0	77.9	96.6	69.1	92.7/91.9
+	7 DeBERTa Team - Microsoft	DeBERTa / TuringNLVR4		90.3	90.4	95.7/97.6	98.4	88.2/63.7	94.5/94.1	93.2	77.5	95.9	66.7	93.3/93.8
8	SuperGLUE Human Baselines	SuperGLUE Human Baselines		89.8	89.0	95.8/98.9	100.0	81.8/51.9	91.7/91.3	93.6	80.0	100.0	76.6	99.3/99.7
+	9 T5 Team - Google	T5		89.3	91.2	93.9/96.8	94.8	88.1/63.3	94.1/93.4	92.5	76.9	93.8	65.6	92.7/91.9

BoolQ, COPA, MultiRC, RTE, WiC, etc.

Why do we use benchmarks?

- **Benchmark performance is important to measure algorithmic innovation**
- Benchmarks are real data
 - synthetic data / could be made up data to suit the algorithm
- Benchmarks are universal
 - All researchers and practitioners evaluate on the same examples (clear victor)
 - Alternative: cherry picking test data with specific properties that makes the algorithm effective.
- Diverse benchmarks from different domains suggest algorithms are general

How do we build benchmarks?

- Define the task
- Design an annotation guideline to collect a dataset
- Run pilot studies to refine annotation guideline and qualify workers
- Analyse the initial data
- Collect data at scale

Define the Task

- **Example:** Natural Language Inferences (also known as textual entailment)

Premise	A woman selling bamboo sticks talking to two men on a loading dock.
Entailment	There are at least three people on a loading dock.
Neutral	A woman is selling bamboo sticks to help provide for her family .
Contradiction	A woman is not taking money for any of her sticks.

- Three-class classification task over pairs of sentences

- Entailment: The **premise implies** the **hypothesis**
- Neutral: The **premise** is **unrelated** to the **hypothesis**
- Contradiction: The **premise contradicts** the **hypothesis**

**How do you
define the task?**

What is the problem to solve?

Premise	A woman selling bamboo sticks talking to two men on a loading dock.
Entailment	There are at least three people on a loading dock.
Neutral	A woman is selling bamboo sticks to help provide for her family .
Contradiction	A woman is not taking money for any of her sticks.

- **What marks a true entailment?**

- **Hypernymy:** A woman is doing X
 - ▶ A person is doing X
- **Quantification:** Everyone is doing X
 - ▶ Someone is doing X
- **Temporal:** Someone is doing X all day
 - ▶ Someone is doing X at noon
- **etc.**

- **What marks a contradiction?**

- **Negation:** A woman is doing X
 - ▶ A woman is not doing X
- **Quantification:** Everyone is doing X
 - ▶ Noone is doing X
- **Less easy to define!**

Design an Annotation Guideline

Premise	A woman selling bamboo sticks talking to two men on a loading dock.
Entailment	There are at least three people on a loading dock.
Neutral	A woman is selling bamboo sticks to help provide for her family .
Contradiction	A woman is not taking money for any of her sticks.

We will show you the caption for a photo. We will not show you the photo. Using only the caption and what you know about the world:

- Write one alternate caption that is **definitely** a **true** description of the photo. *Example: For the caption “Two dogs are running through a field.” you could write “There are animals outdoors.”*
- Write one alternate caption that **might be a true** description of the photo. *Example: For the caption “Two dogs are running through a field.” you could write “Some puppies are running to catch a stick.”*

- Write one alternate caption that is **definitely** a **false** description of the photo. *Example: For the caption “Two dogs are running through a field.” you could write “The pets are sitting on a couch.” This is different from the maybe correct category because it’s impossible for the dogs to be both running and sitting.*

How do you define the task to annotators?

Run pilot studies

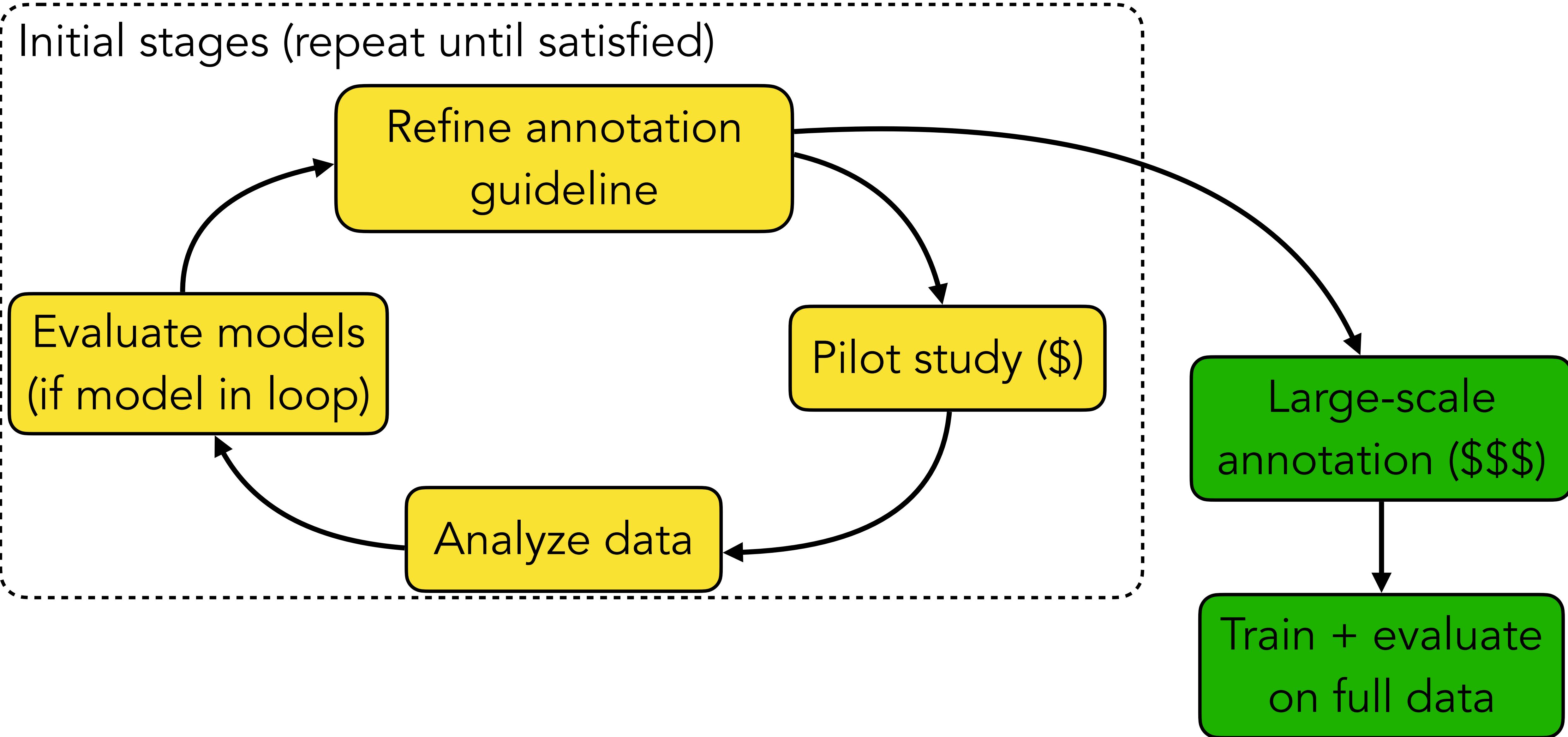
- Typically done with experts at first (people on your team)
- Refine annotation guidelines based on feedback
- Run more pilot studies with crowdworkers (the individuals who will actually produce the data)
- Analyse the initial data (**do annotators understand the task?**)
- Qualify best crowd workers to do the task at scale
 - Produce high-quality data (need to annotated examples yourself that you evaluate their responses against)
 - Better to make the task interesting and creative so that they remain engaged throughout the process!

Why do we need to qualify workers?

Collect data at scale

- Once a good set of qualified workers is gathered, collect the data at scale
- How do we ensure the quality of the data?
 - Pay well!
 - Collect multiple labels per example
 - Disqualify workers who often disagree with other workers. **Be careful!**
 - Keep data points with high inter-annotator agreement
 - ▶ Cohen's Kappa
 - ▶ Fleiss' Kappa
 - ▶ Krippendorf's Alpha
 - Reject the rest

Annotation Lifecycle



Why do our benchmarks fall short of
measuring true NLU performance?

Many small reasons, but the unreliability of our data is at the heart of it!

Let's take a look!

Biases

Annotation Artifacts

Harms / Undesirable Behavior

Shortcuts

Biases

Spurious Biases or Spurious Correlations

Inductive Biases

Statistical Bias

Biases

Annotation Artifacts

Harms / Undesirable Behavior

Shortcuts

Biases

Spurious Biases or Spurious Correlations

“A **spurious correlation** is a mathematical relationship in which two or more events or variables are associated but not causally related, due to either coincidence or the presence of a certain third, unseen factor.”

– Burns, 1997

Design an annotation guideline

Premise	A woman selling bamboo sticks talking to two men on a loading dock.
Entailment	There are at least three people on a loading dock.
Neutral	A woman is selling bamboo sticks to help provide for her family .
Contradiction	A woman is not taking money for any of her sticks.

We will show you the caption for a photo. We will not show you the photo. Using only the caption and what you know about the world:

- Write one alternate caption that is **definitely** a **true** description of the photo. *Example: For the caption “Two dogs are running through a field.” you could write “There are animals outdoors.”*
- Write one alternate caption that **might be a true** description of the photo. *Example: For the caption “Two dogs are running through a field.” you could write “Some puppies are running to catch a stick.”*

- Write one alternate caption that is **definitely** a **false** description of the photo. *Example: For the caption “Two dogs are running through a field.” you could write “The pets are sitting on a couch.” This is different from the maybe correct category because it’s impossible for the dogs to be both running and sitting.*

**What do you think
annotators will do?**

Annotation Artifacts

Premise	A woman selling bamboo sticks talking to two men on a loading dock.
Entailment	There are at least three people on a loading dock.
Neutral	A woman is selling bamboo sticks to help provide for her family .
Contradiction	A woman is not taking money for any of her sticks.

- To create neutral sentences, **annotators add information**
- To create contradictions, **annotators add negation**

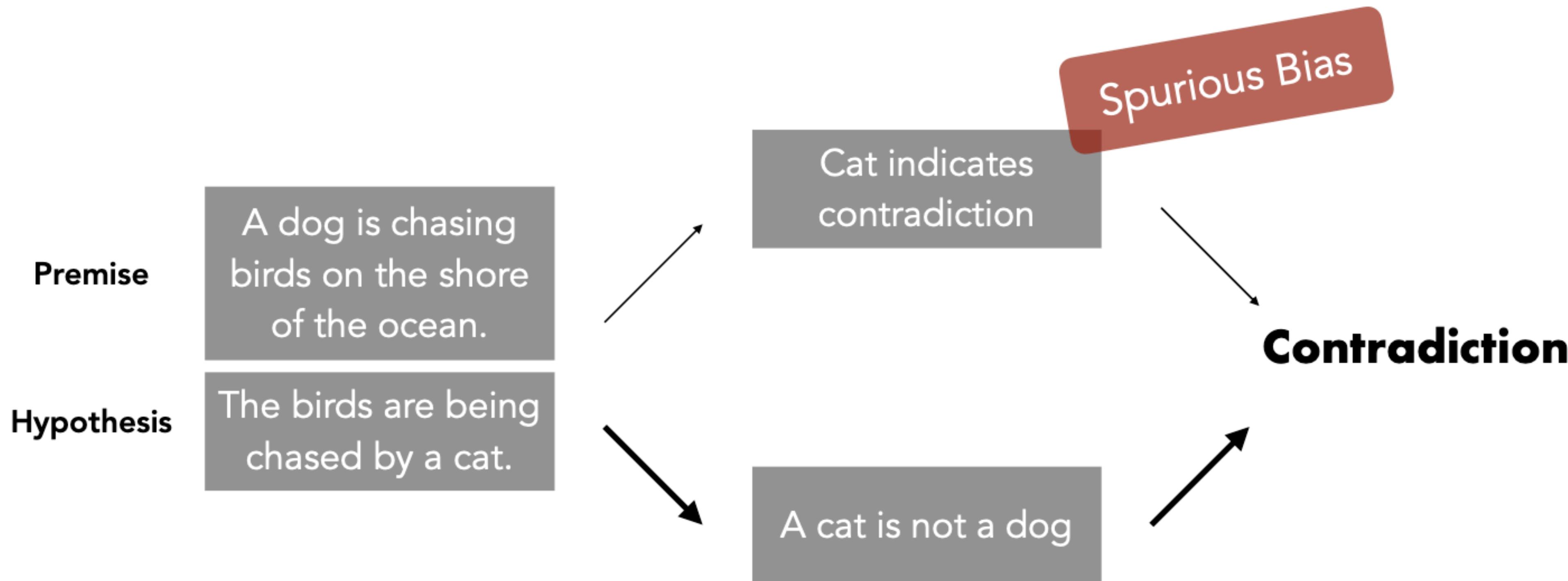
Annotation Artifacts

Premise	A woman selling bamboo sticks talking to two men on a loading dock.
Entailment	There are at least three people on a loading dock.
Neutral	A woman is selling bamboo sticks to help provide for her family .
Contradiction	A woman is not taking money for any of her sticks.

- To create neutral sentences, **annotators add information**
- To create contradictions, **annotators add negation**
- Models can do well even if they **ignore the premise**

	Hypothesis-only	Random	Improvement
SNLI	69.17	33.82	+35.35
MNLI-1	55.52	35.45	+20.07
MNLI-2	55.18	35.22	+19.96

Sometimes even simpler patterns



Which patterns?

	Entailment	Neutral	Contradiction	
SNLI	outdoors	2.8% tall	0.7%	nobody 0.1%
	least	0.2% first	0.6%	sleeping 3.2%
	instrument	0.5% competition	0.7%	no 1.2%
	outside	8.0% sad	0.5%	tv 0.4%
	animal	0.7% favorite	0.4%	cat 1.3%
MNLI	some	1.6% also	1.4%	never 5.0%
	yes	0.1% because	4.1%	no 7.6%
	something	0.9% popular	0.7%	nothing 1.4%
	sometimes	0.2% many	2.2%	any 4.1%
	various	0.1% most	1.8%	none 0.1%

Table 4: Top 5 words by $\text{PMI}(\textit{word}, \textit{class})$, along with the proportion of *class* training samples containing *word*. MultiNLI is abbreviated to MNLI.

Words most
associated with
labels shouldn't be
indicative

Why might it be a problem that
the annotation artefacts exist?

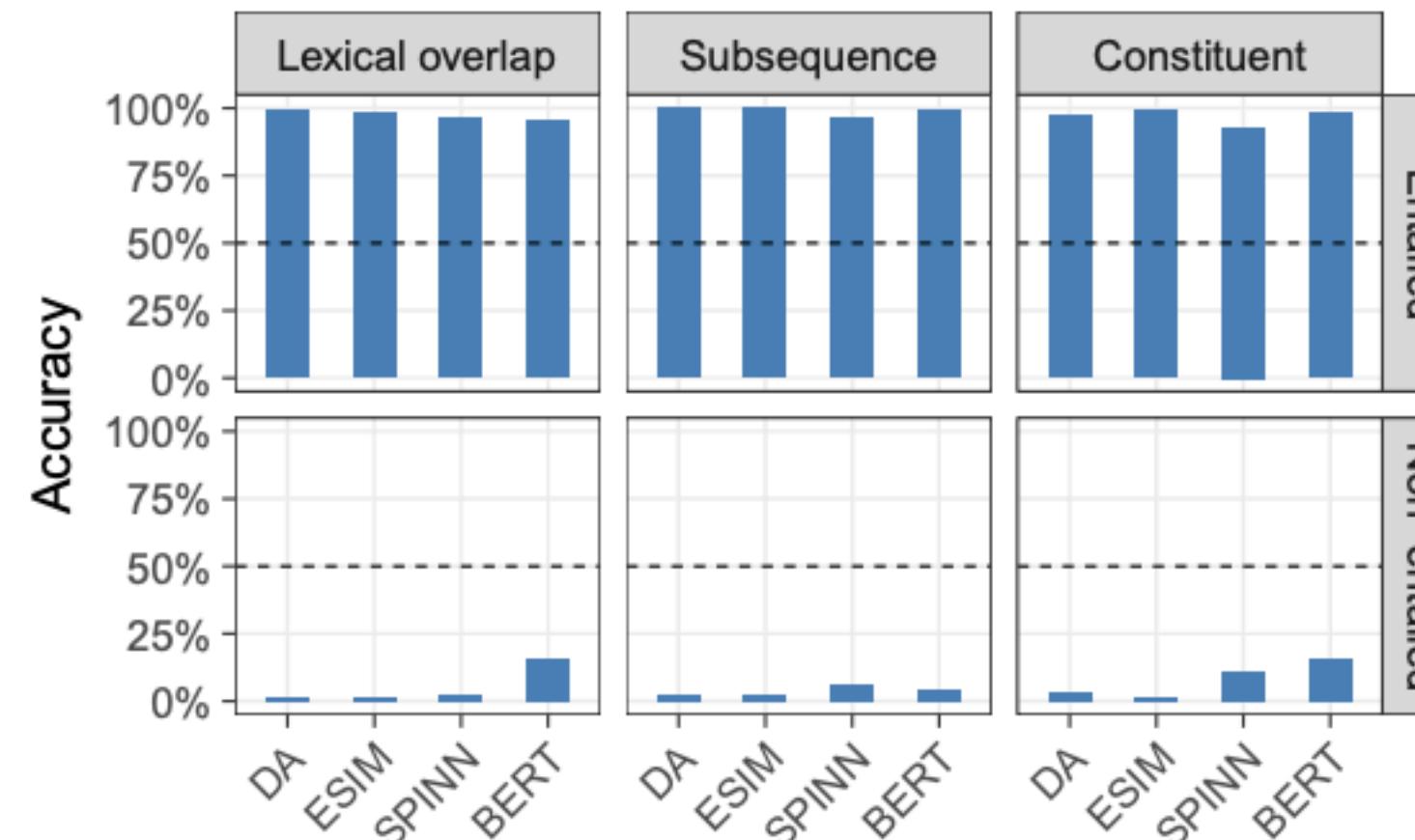
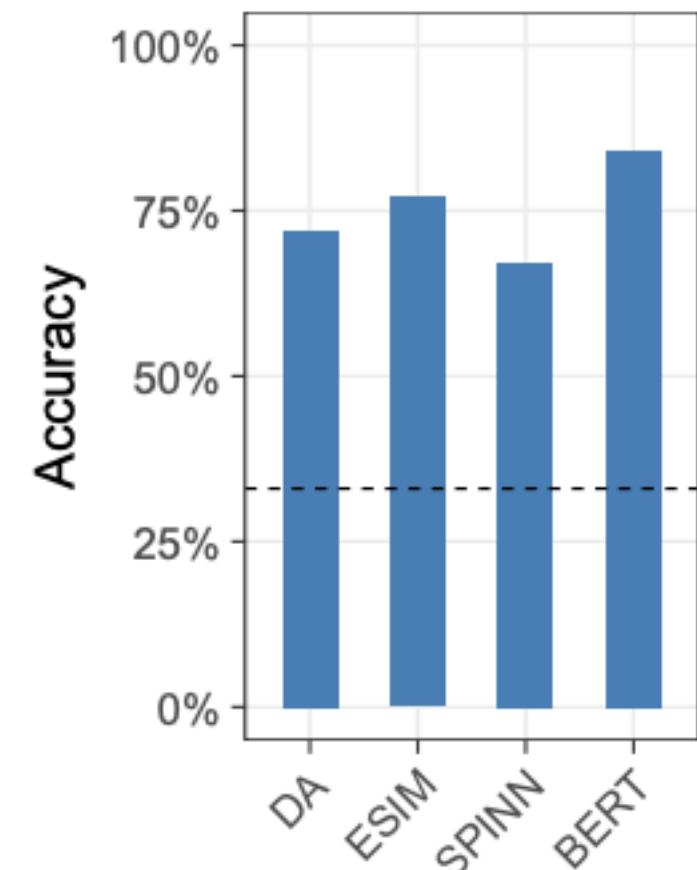
Generalisation Issues

- **Assumption in ML:** samples in train and test sets are drawn from the same distribution
- **Reality:** Future data that must be classified by the model may not come from the same **distribution** of text (i.e., out-of-distribution data)
 - e.g., annotators may be different
- Models learn simple patterns that are merely shortcut heuristics for the hard task we actually want them to learn
 - e.g., natural language inference is very hard
 - seeing negation words is easier
- Models won't generalise to new examples that don't have these patterns
 - Need to understand when models are exploiting these patterns

What happens out-of-distribution?

- **Develop** various syntactic heuristics to express relationships using out-of-distribution language

Heuristic	Premise	Hypothesis	Label
Lexical overlap heuristic	The banker near the judge saw the actor.	The banker saw the actor.	E
	The lawyer was advised by the actor.	The actor advised the lawyer.	E
	The doctors visited the lawyer.	The lawyer visited the doctors.	N
	The judge by the actor stopped the banker.	The banker stopped the actor.	N
Subsequence heuristic	The artist and the student called the judge.	The student called the judge.	E
	Angry tourists helped the lawyer.	Tourists helped the lawyer.	E
	The judges heard the actors resigned.	The judges heard the actors.	N
	The senator near the lawyer danced.	The lawyer danced.	N
Constituent heuristic	Before the actor slept, the senator ran.	The actor slept.	E
	The lawyer knew that the judges shouted.	The judges shouted.	E
	If the actor slept, the judge saw the artist.	The actor slept.	N
	The lawyers resigned, or the artist slept.	The artist slept.	N



100% performance when labels of OOD examples are same as ID bias. 0% otherwise

Also a problem in other tasks

- **Visual question answering:** answer questions about a given image

Is the umbrella upside down?

yes

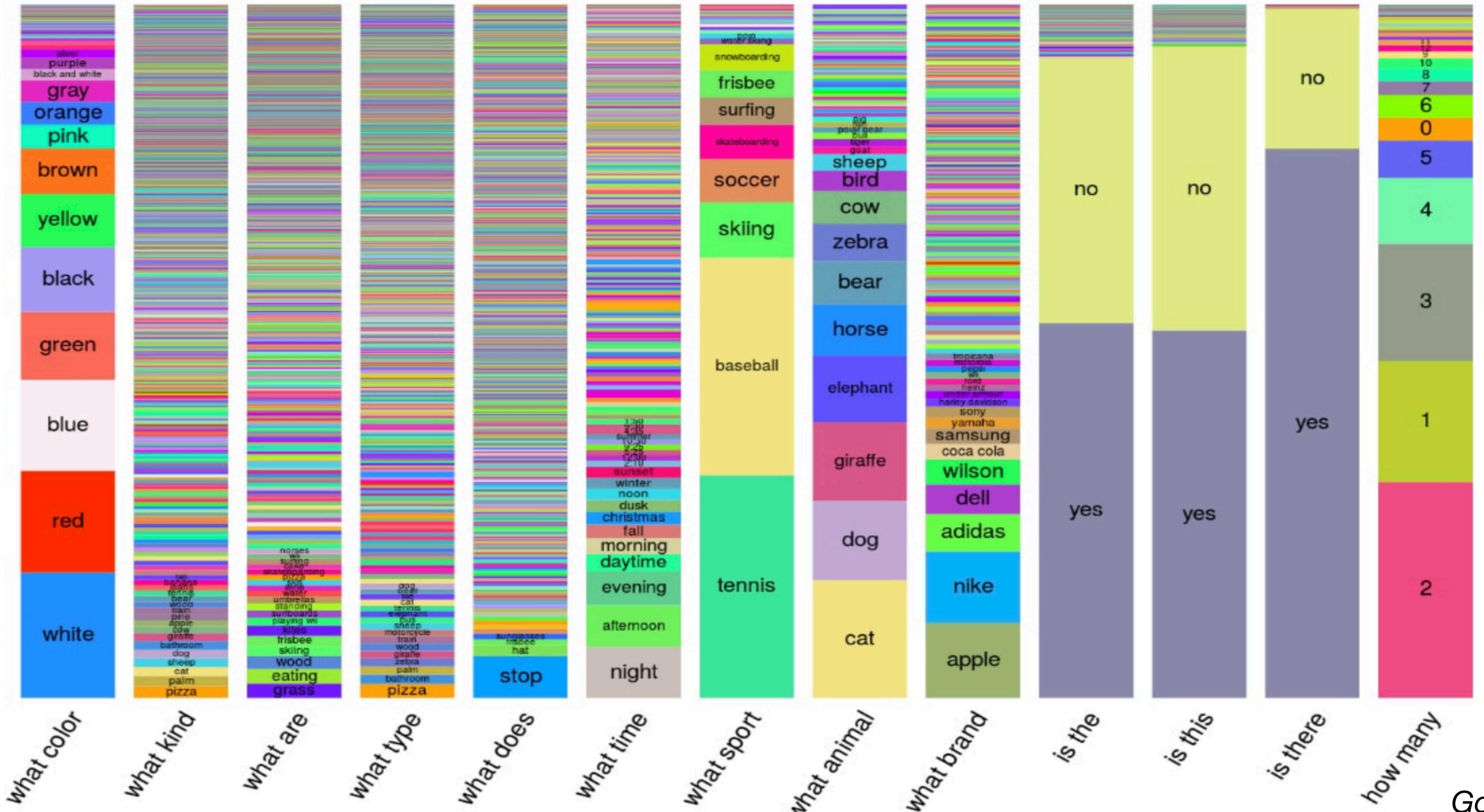


How many children are in the bed?

2



Annotation Artifacts



Annotator Bias

Dataset	# examples	# annotators	Ex / ann
MNLI (matched) [1]	402517	380	1143.32
OpenBookQA [2]	5457	84	64.96
CommonsenseQA [3]	11096	132	84.06

- Crowdsourced datasets are largely created by surprisingly few annotators
- Incentives may push annotators to use heuristics to annotate data rapidly

Does knowing the annotator improve model performance?

Annotator Bias

Dataset	Unknown annotator	Known annotator
OpenBookQA	52.2	56.4
Commonsense QA	53.6	55.3
MNLI	82.9	84.5

- Knowing the annotator of an example makes the model more likely to classify an example correctly!

What can we do to build
better benchmarks?

Post-hoc: Manual Re-balancing

Who is wearing glasses?

man



woman

Where is the child sitting?

fridge



arms



Is the umbrella upside down?

yes



no

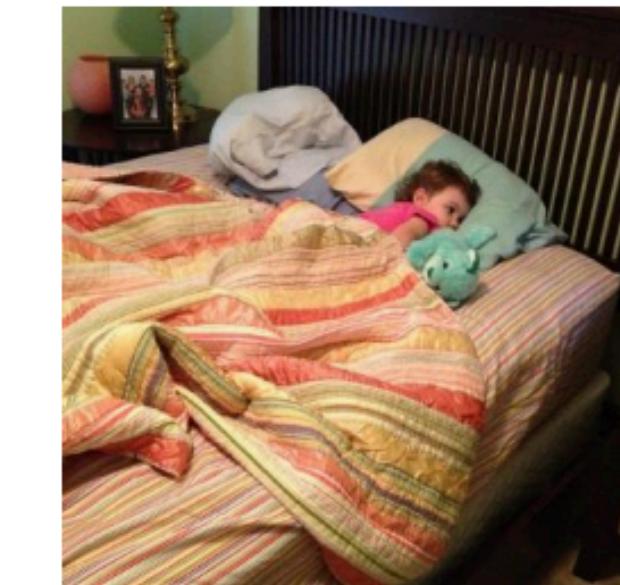


How many children are in the bed?

2



1



Re-balance datasets so that certain answers are not predictable only from the question

Goyal et al. (2018)

Intentional Design: Contrast Sets

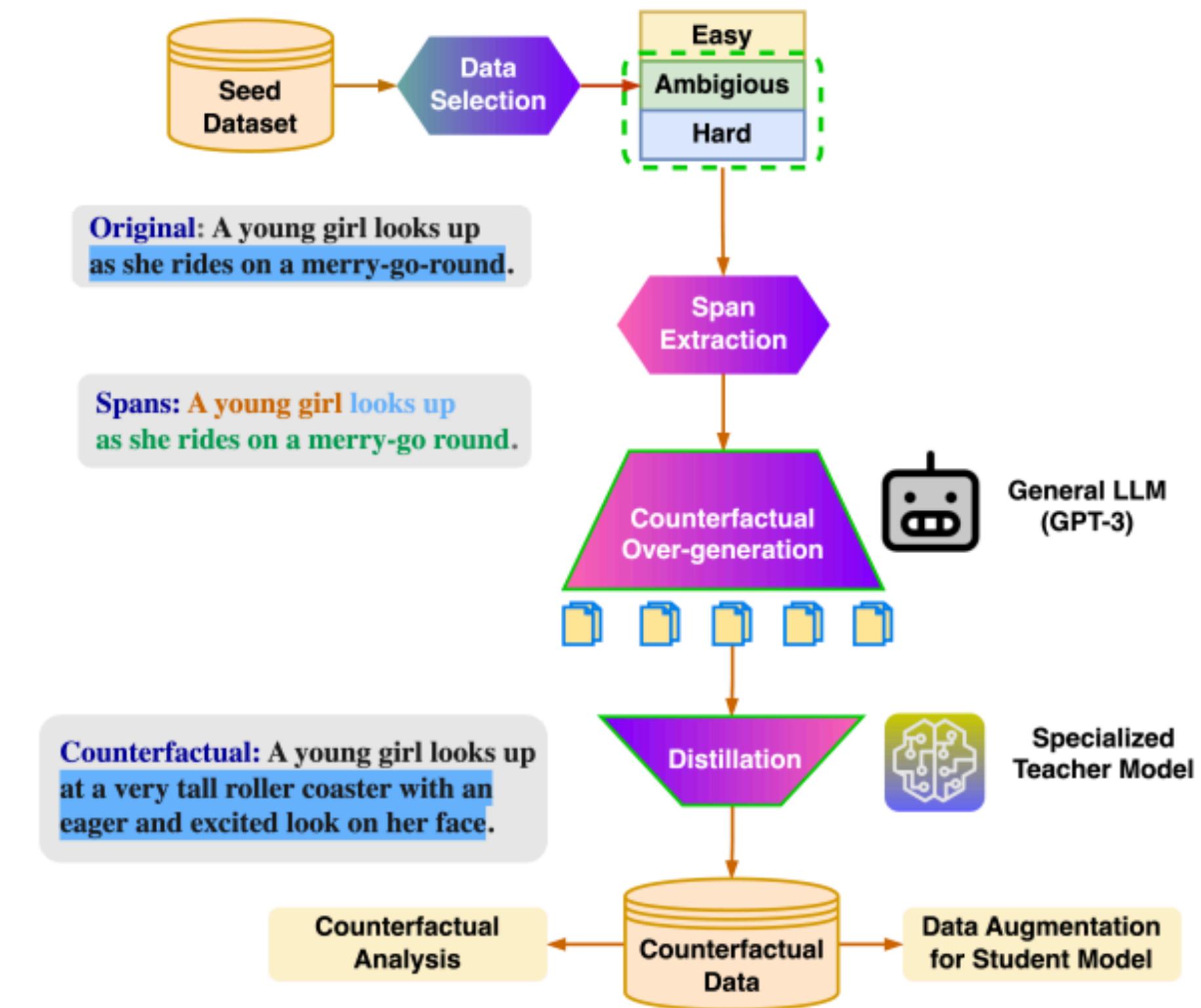
- Construct controlled datasets that test specific dimensions of what we want in the first place
- Perturb examples using known patterns to highlight specific distinctions

Original (Negative): I had quite high hopes for this film, even though it got a bad review in the paper. I was extremely **tolerant**, and sat through the entire film. I felt quite **sick** by the end.

New (Positive): I had quite high hopes for this film, even though it got a bad review in the paper. I was extremely **amused**, and sat through the entire film. I felt quite **happy** by the end.

Data Augmentation

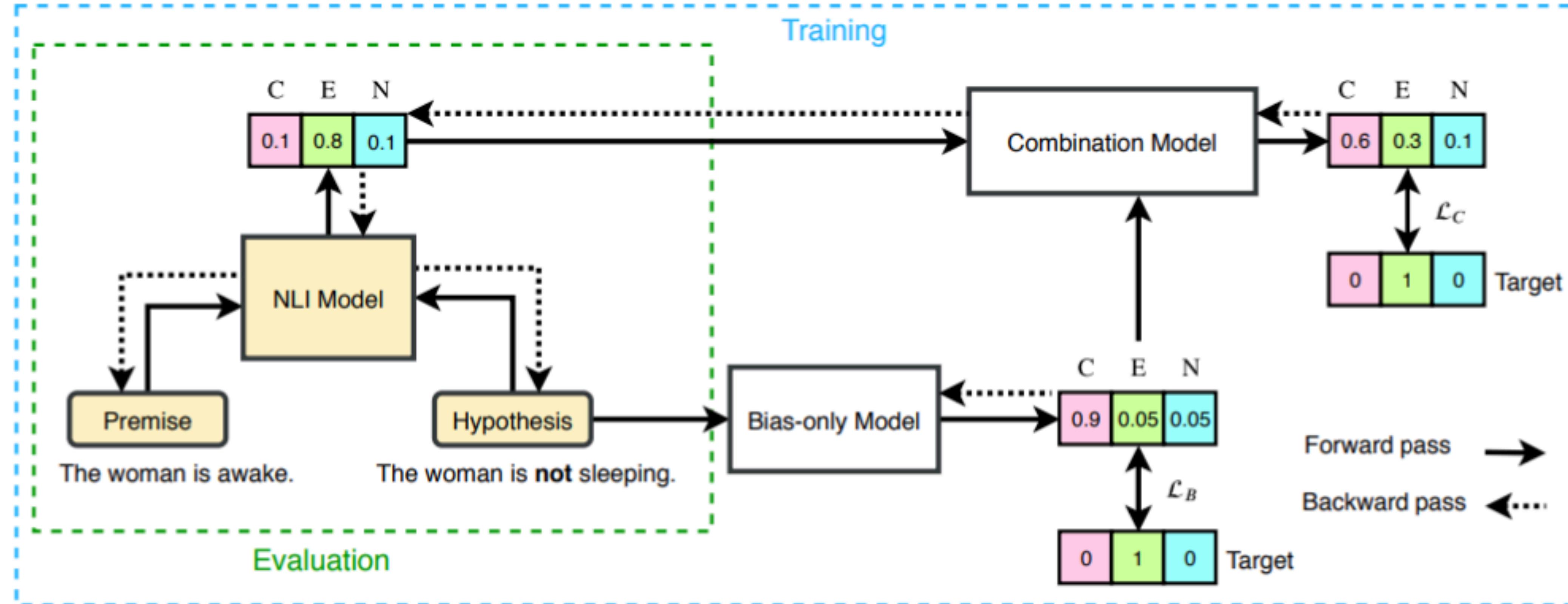
- Add new examples to the training dataset that demonstrate new contexts for the model to learn from
- Model learns better via exposure to more diverse training data
- Many different approaches
 - Wei et al., 2019; Kaushik et al., 2019; Yang et al., 2020; Liu et al., 2022; Chen et al., 2022, etc.



Adversarial Filtering Algorithms

- **Motivation:** Biased examples can be learned through easy shortcuts
- **Goal:** To avoid shortcuts, remove the easy examples from the dataset
- **Method:** Train a classifier on different splits of the data and evaluate validation examples
 - similar to cross-validation
 - Remove examples that are easily solved by a particular model

Bias Mitigation in Models



- Train a bias-only model in an ensemble with your actual model
- Biased examples will be learned by the bias-only model; the actual model won't need to learn the shortcuts to do well on those examples

What should we know about our dataset?

- **Datasheets for Datasets:** Framework for recording data details
- **Motivation:**
 - why was it collected?
 - Who created it?
 - Who funded it?
- **Composition:**
 - how were the labels decided?
 - are there official (train/test/dev) splits?
 - Is there confidential or sensitive data? Are the data subjects identifiable? (More in Week 11)

Data Cascades

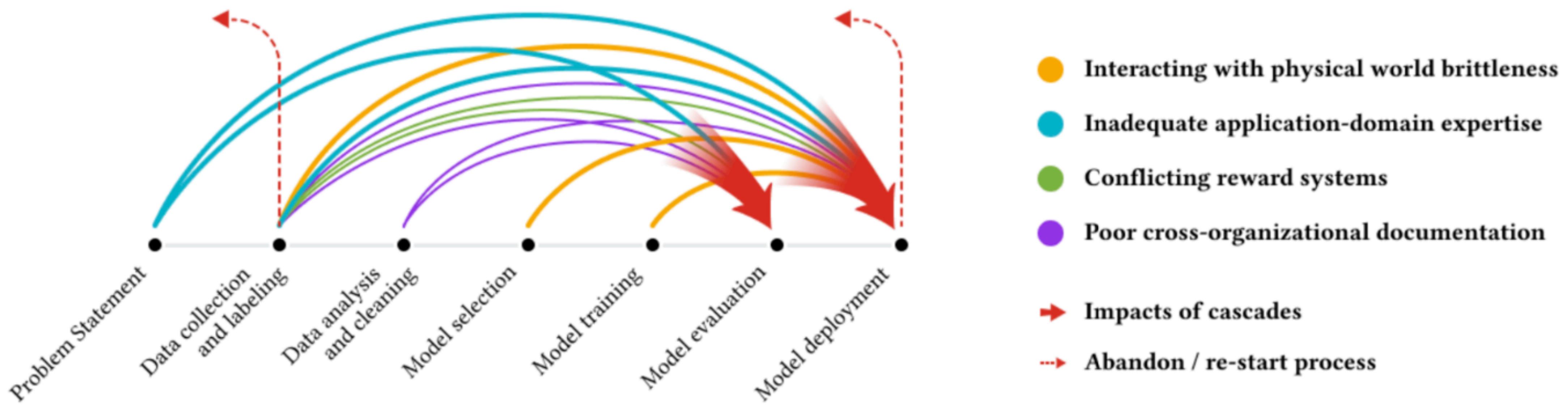


Figure 1: Data cascades in high-stakes AI. Cascades are opaque and protracted, with multiplied, negative impacts. Cascades are triggered in the upstream (e.g., data collection) and have impacts on the downstream (e.g., model deployment). Thick red arrows represent the compounding effects after data cascades start to become visible; dotted red arrows represent abandoning or re-starting of the ML data process. Indicators are mostly visible in model evaluation, as system metrics, and as malfunctioning or user feedback.

Recap

- Pretrained models fine-tuned on downstream tasks achieve **incredible performance** on benchmarks designed to measure language understanding
- Benchmarks are made up of datasets that are human-constructed
- Humans make “mistakes” when designing datasets, allowing models to shortcut true understanding of the task in favour of easily learnable heuristics
- Designing challenging evaluations remains a primary goal natural language processing systems

Final Note

- Good data, aligned with real human tasks, is the future of NLP
- ChatGPT components
 - GPT-3 data: Trained on 400 GB of raw text
 - InstructGPT data: ~45k examples of instructions and expert-labeled task demonstrations

Public NLP datasets are not reflective of how our language models are used. We compare GPT-3 fine-tuned on our human preference data (i.e. InstructGPT) to GPT-3 fine-tuned on two different compilations of public NLP tasks: the FLAN (Wei et al., 2021) and T0 (Sanh et al., 2021) (in particular, the T0++ variant). These datasets consist of a variety of NLP tasks, combined with natural language instructions for each task. On our API prompt distribution, our FLAN and T0 models perform slightly worse than our SFT baseline, and labelers significantly prefer InstructGPT to these models (InstructGPT has a $73.4 \pm 2\%$ winrate vs. our baseline, compared to $26.8 \pm 2\%$ and $29.8 \pm 2\%$ for our version of T0 and FLAN, respectively).