Self-Supervised Learning

Megan Leszczynski

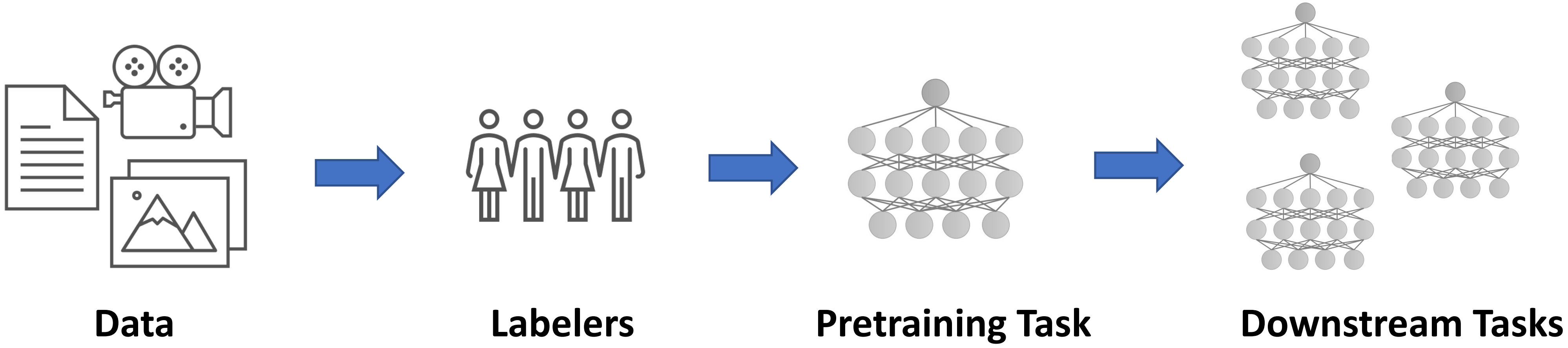
Lecture Plan

1. What is self-supervised learning?

2. Examples of self-supervision in NLP
   - Word embeddings (e.g., word2vec)
   - Language models (e.g., GPT)
   - Masked language models (e.g., BERT)

3. Open challenges
   - Demoting bias
   - Capturing factual knowledge
   - Learning symbolic reasoning

Supervised pretraining on large labeled, datasets has led to successful transfer learning



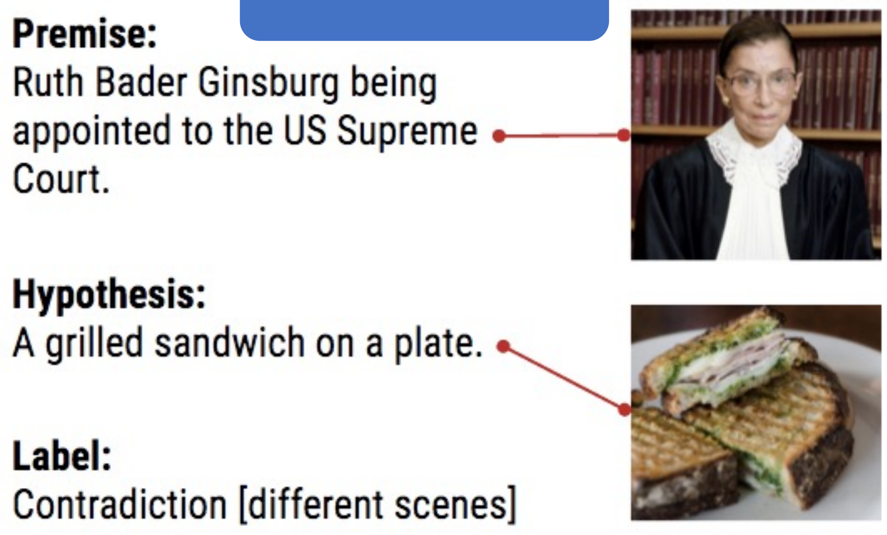
[Deng et al., 2009]

ImageNet

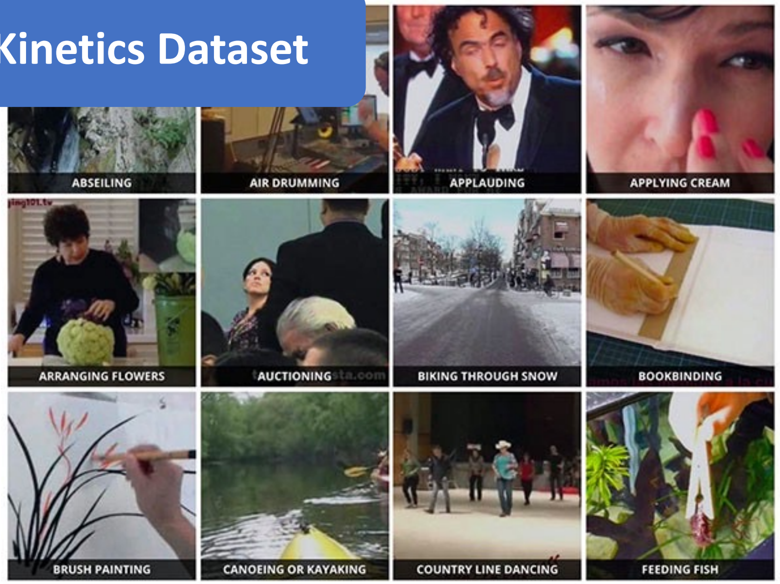
- Pretrain for fine-grained image classification over 1000 classes
- Use feature representations for downstream tasks, e.g. object detection, image segmentation, and action recognition

Supervised pretraining on large labeled, datasets has led to successful transfer learning

SNLI Dataset



Kinetics Dataset



Across images, video, and text

[Deng et al., 2009] [Carreira et al., 2017] [Conneau et al., 2017]

But supervised pretraining comes at a cost...

- **Time-consuming and expensive** to label datasets for new tasks
  - ImageNet: 3 years,

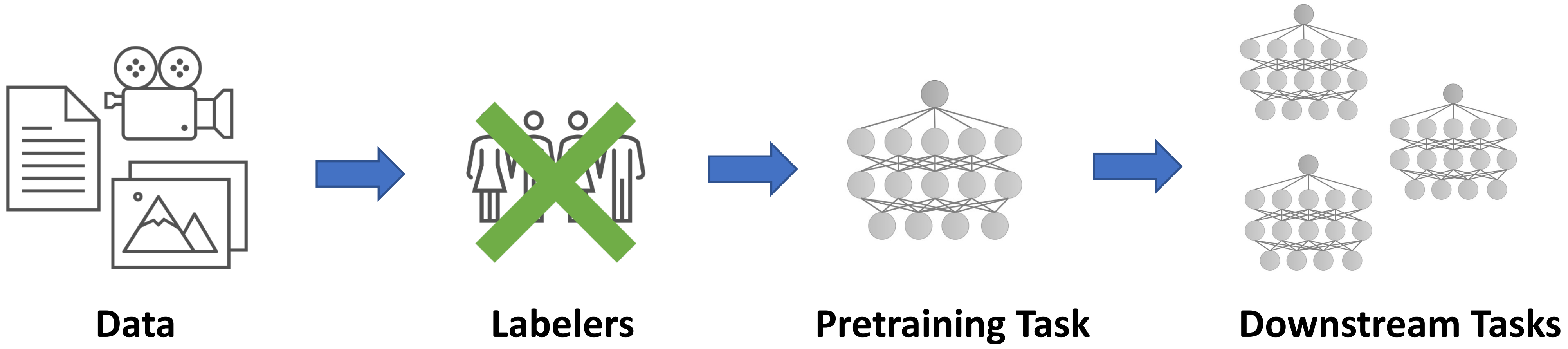
  49k Amazon MechanicalTurkers [1]

- **Domain expertise needed** for specialized tasks
  - Radiologists to label medical images
  - Native speakers or language specialists for labeling text in different languages

Can self-supervised learning help?

- Self-supervised learning (informal definition): supervise using labels ***generated from the data*** without any manual or weak label sources

- Idea: Hide or modify part of the input. Ask model to recover input or classify what changed.
  - Self-supervised task referred to as the pretext task



Pretext Task: Classify the Rotation

270° rotation

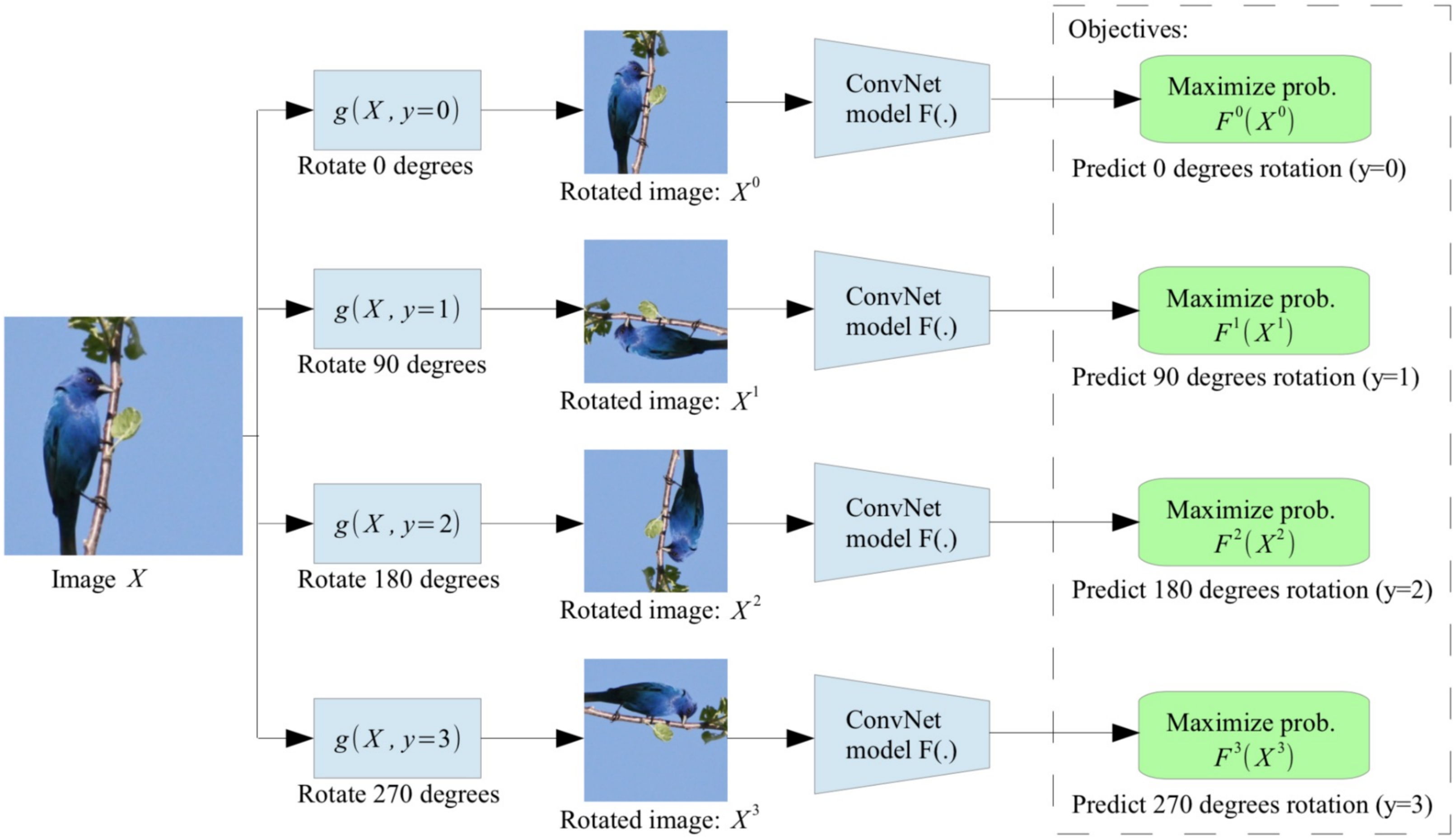
90° rotation

~~180°~~ 0° rotation

Identifying the object helps solve rotation task!

Catfish species that swims upside down…

Pretext Task: Classify the Rotation



Learning rotation improves results on object classification, object segmentation, and object detection tasks.

[Gidaris et al., ICLR 2018]

Pretext Task: Identify the Augmented Pairs

Contrastive self-supervised learning with SimCLR achieves state-of-the-art on ImageNet for a **limited amount of labeled data.**

- 85.8% top-5 accuracy on 1% of Imagenet labels.

[Chen et al., ICML 2020]

GIF from Google AI blog

Benefits of Self-Supervised Learning

✓ Like supervised pretraining, can learn general-purpose feature representations for downstream tasks

✓ Reduces expense of hand-labeling large datasets

✓ Can leverage nearly unlimited (unlabeled) data available on the web

995 photos uploaded *every second*

6000 tweets sent *every second*

500 hours of video uploaded *every minute*

Sources: [1], [2], [3]

Lecture Plan

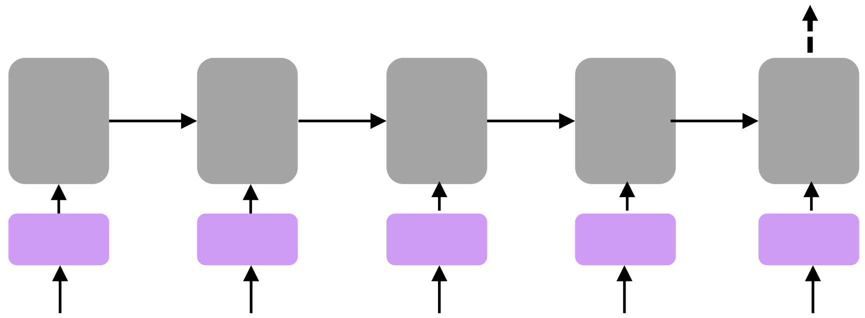
1. What is self-supervised learning?

2. Examples of self-supervision in NLP
   - Word embeddings (e.g., word2vec)
   - Language models (e.g., GPT)
   - Masked language models (e.g., BERT)

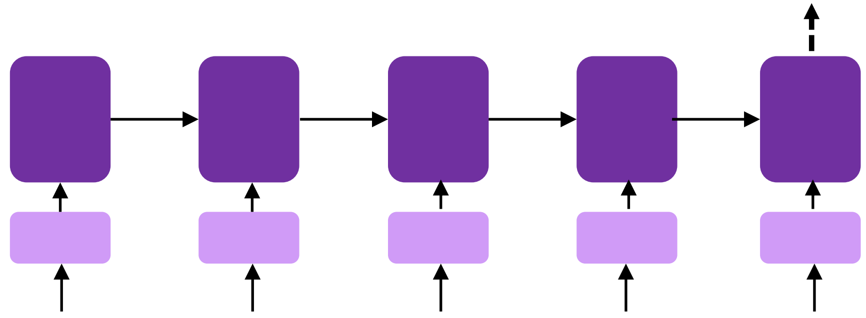
3. Open challenges
   - Demoting bias
   - Capturing factual knowledge
   - Learning symbolic reasoning

Examples of Self-Supervision in NLP

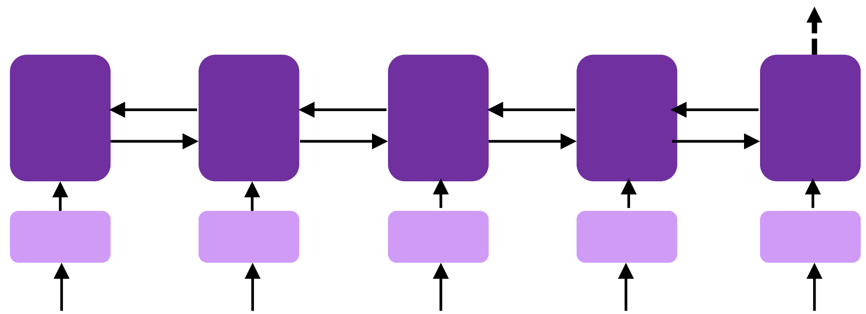
- **Word embeddings**
  - Pretrained word representations
  - Initializes *1st layer* of downstream models



- **Language models**
  - *Unidirectional*, pretrained language representations
  - Initializes *full* downstream model

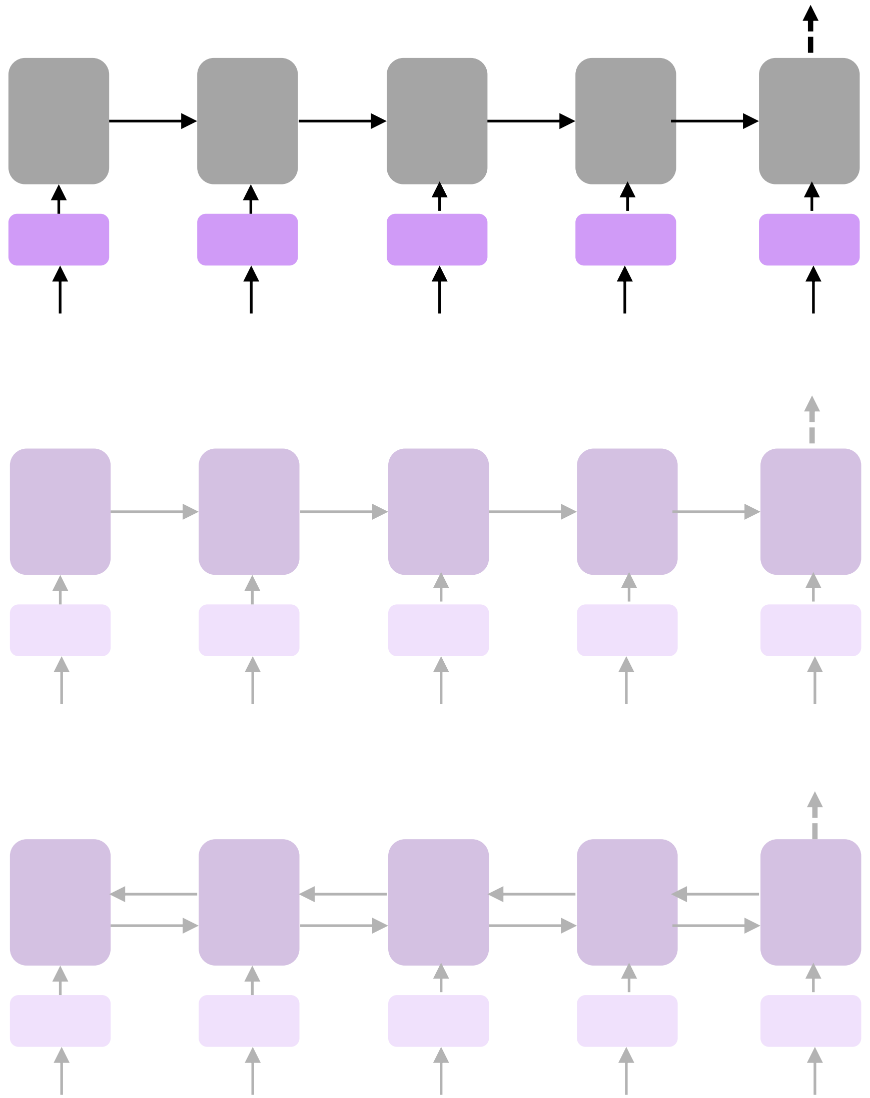


- **Masked language models**
  - *Bidirectional*, pretrained language representations
  - Initializes *full* downstream model



Examples of Self-Supervision in NLP

- **Word embeddings**
  - Pretrained word representations
  - Initializes *1st layer* of downstream models
- **Language models**
  - *Unidirectional*, pretrained language representations
  - Initializes *full* downstream model
- **Masked language models**
  - *Bidirectional*, pretrained language representations
  - Initializes *full* downstream model



Word Embeddings

- Goal: represent words as vectors for input into neural networks.

- One-hot vectors? (single 1, rest 0s)

  pizza = [0 0 0 0 0 1 0 … 0 0 0 0 0]

  pie = [0 0 0 0 0 0 0 … 0 0 0 1 0]

☹️ Millions of words ⟶ high-dimensional, sparse vectors

☹️ No notion of word similarity

- Instead: we want a **dense, low-dimensional** vector for each word such that words with similar meanings have similar vectors.

[Slides Reference: Chris Manning, CS224N]

Distributional Semantics

- Idea: define a word by the words that frequently occur nearby in a corpus of text
  - "You shall know a word by the company it keeps" (J. R. Firth 1957: 11)

- Example: defining "pizza"
  - What words frequently occur in the context of pizza?

```
13% of the United States population eats pizza on any given day.
Mozzarella is commonly used on pizza, with the highest quality mozzarella from Naples.
In Italy, pizza served in formal settings is eaten with a fork and knife.
```

- Can we use distributional semantics to develop a pretext task for self-supervision?

Pretext Task: Predict the Center Word

- Move context window across text data and use words in window to predict the center word.
  - No hand-labeled data is used!

predict: pizza

`In Italy, pizza served in formal settings is eaten with a fork and knife.`

context window, size 2

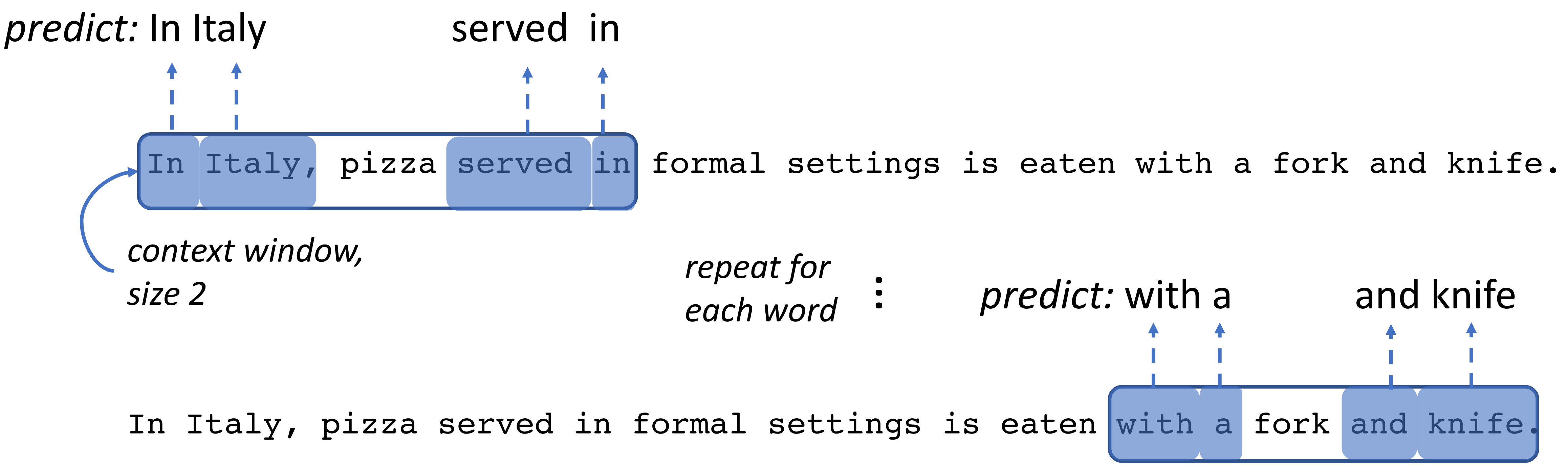
repeat for each word ⋮

predict: fork

`In Italy, pizza served in formal settings is eaten with a fork and knife.`

Pretext Task: Predict the Context Words

- Move context window across text data and use words in window to predict the *context* words, given the center word.
  - No hand-labeled data is used!



Case Study: word2vec

- Tool to produce word embeddings using self-supervision by Mikolov et al.

- Supports training word embeddings using 2 architectures:
  - Continuous bag-of-words (CBOW): predict the center word
  - Skip-gram: predict the context words

- Steps:
  1. Start with randomly initialized word embeddings.
  2. Move sliding window across *unlabeled* text data.
  3. Compute probabilities of center/context words, given the words in the window.
  4. Iteratively update word embeddings via stochastic gradient descent .

[Mikolov et al., 2013]

Case Study: word2vec

- **Loss function (skip-gram):** For a corpus with T words, minimize the negative log likelihood of the context word w_{t+j} given the center word w_t.

Context word Center word

$$J(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{\substack{-m \le j \le m \\ j \neq 0}} \log P\left(w_{t+j} \,\middle|\, w_t;\ \theta\right)$$

Model parameters

Context window size

- Use two word embedding matrices (embedding dimension n, vocab size l):
  - Center word embeddings $V \in \mathbb{R}^{n \times l}$; context word embeddings $U \in \mathbb{R}^{l \times n}$

$$P\left(w_{t+j} \,\middle|\, w_t;\ \theta\right) = P\left(u_{t+j} \,\middle|\, v_t\right) = \frac{\exp(u_{t+j}^T v_t)}{\sum_{j=1}^{l} \exp(u_j^T v_t)}$$

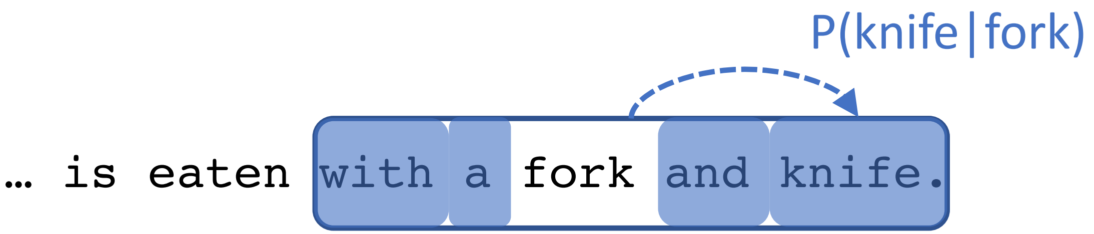
Softmax

Word vectors

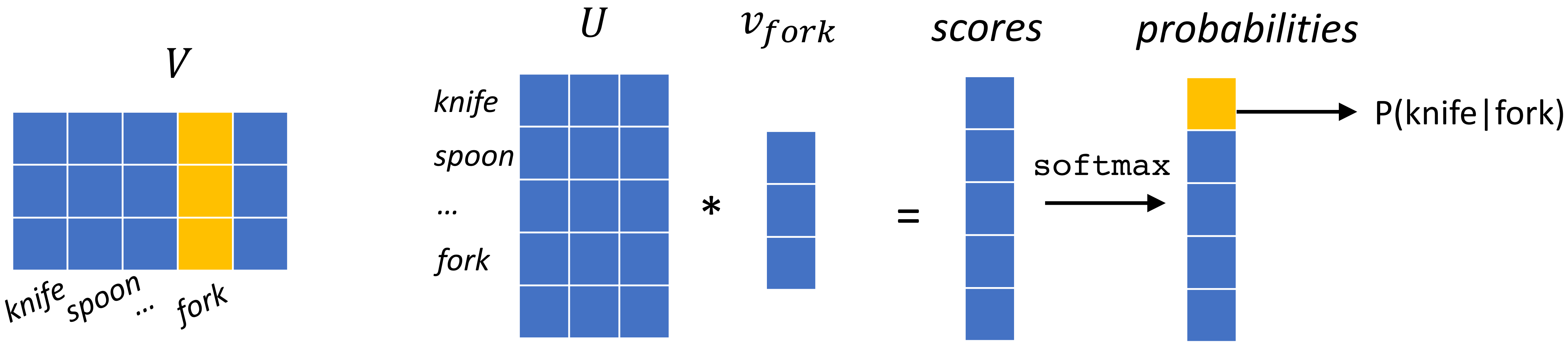
[Mikolov et al., 2013]

Case Study: word2vec

- **Example:** using the skip-gram method (predict context words), compute the probability of ”knife” given the center word “fork”.



1. Get “fork” word vector v_{fork} **2. Compute scores** **3. Convert to probabilities**

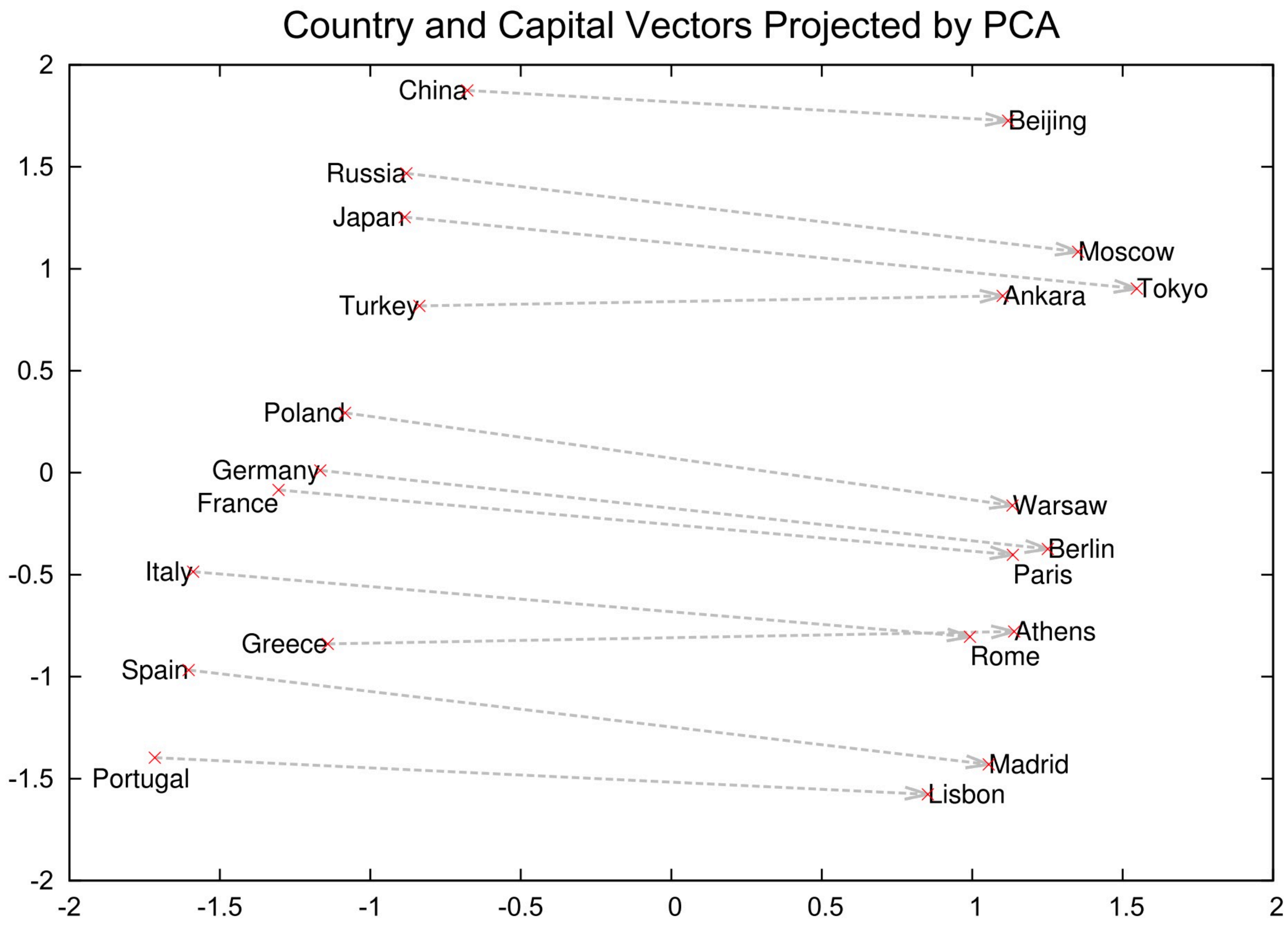


[Mikolov et al., 2013]

Case Study: word2vec

- Mikolov et al. released word2vec embeddings pretrained on **100 billion word** Google News dataset.

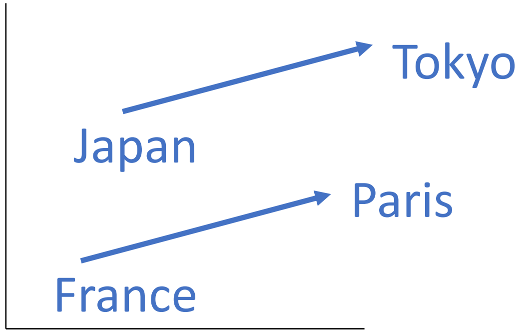
- Embeddings exhibited meaningful properties despite being trained with **no hand-labeled data**.



[Mikolov et al., 2013]

Case Study: word2vec

- Vector arithmetic can be used to evaluate word embeddings on analogies
  - France is to Paris as Japan is to ?



Cosine similarity

$$w^* = argmax_w \frac{\boldsymbol{v_w y}}{\|\boldsymbol{v_w}\|\|\boldsymbol{y}\|},$$

where $\boldsymbol{y} = \boldsymbol{v_{Paris}} - \boldsymbol{v_{France}} + \boldsymbol{v_{Japan}}$

w^* = Tokyo

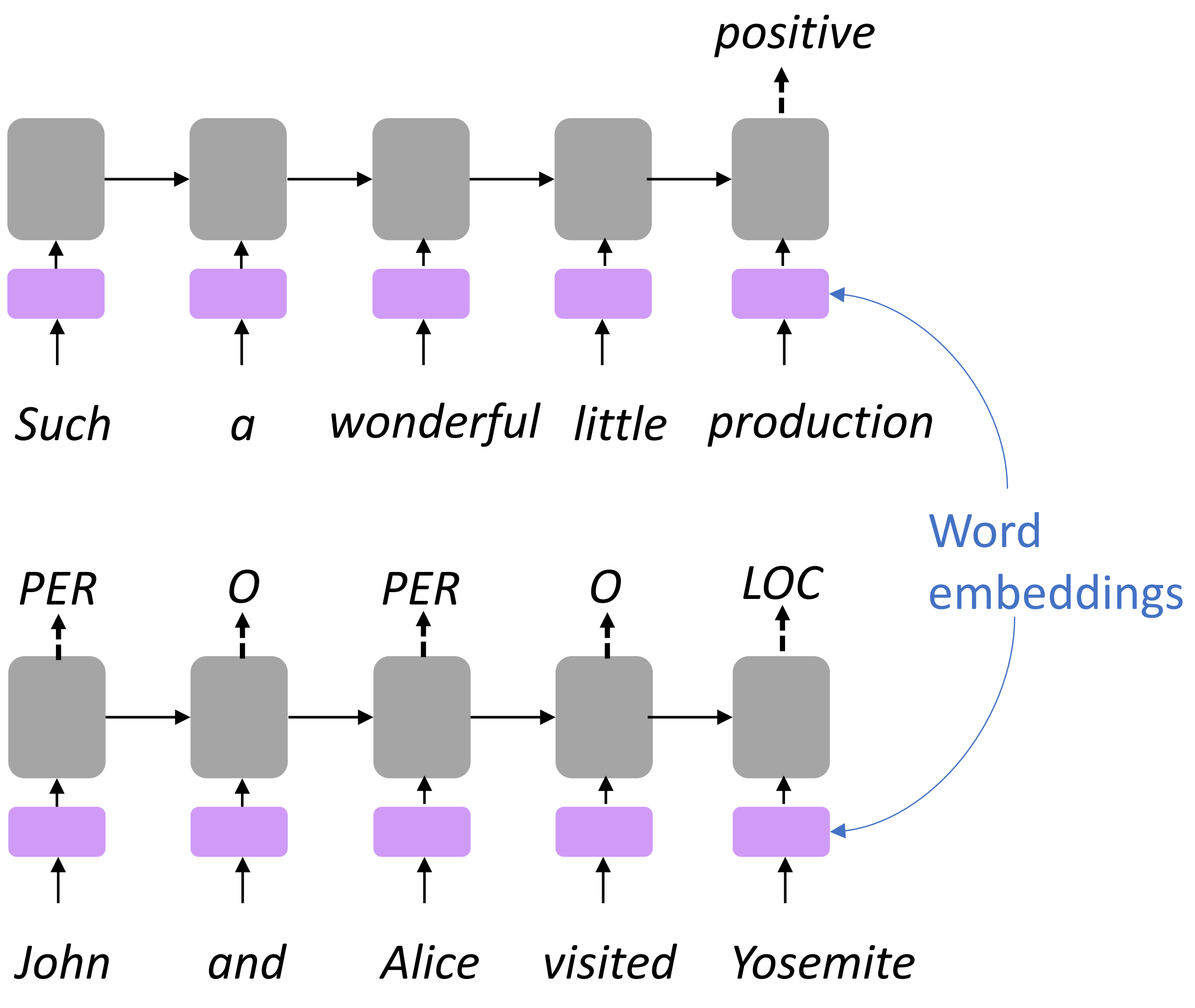
Expected answer

- Analogies have become a common **intrinsic task** to evaluate the properties learned by word embeddings

[Mikolov et al., 2013]

Case Study: word2vec

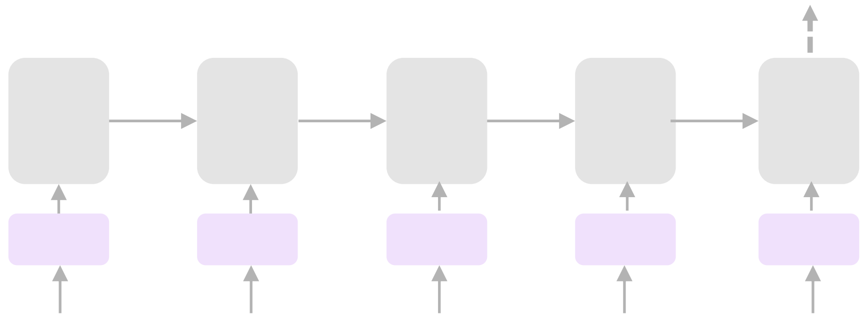
- Pretrained word2vec embeddings can be used to initialize the first layer of downstream models
- Improved performance on many downstream NLP tasks, including sentence classification, machine translation, and sequence tagging
  - Most useful when downstream data is limited
- Still being used in applications in industry today!



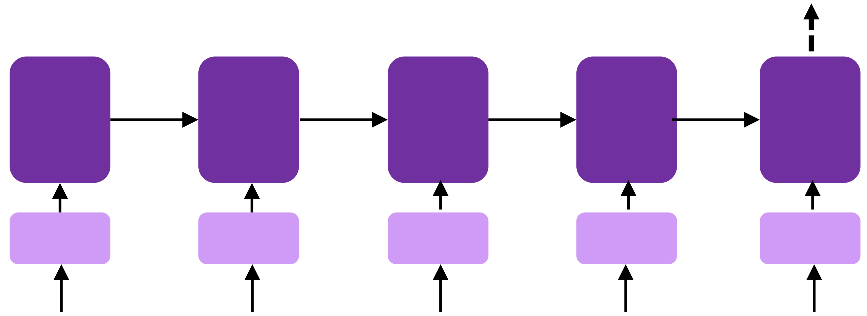
[Kim et al., 2014] [Qi et al., 2018] [Lample et al., 2016]

Examples of Self-Supervision in NLP

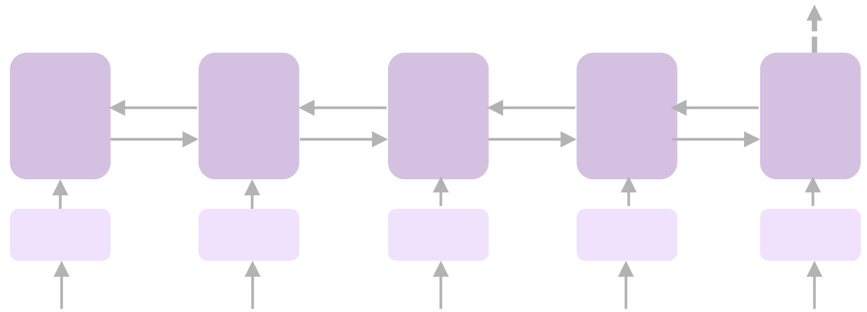
- **Word embeddings**
  - Pretrained word representations
  - Initializes *1st layer* of downstream models



- **Language models**
  - *Unidirectional*, pretrained language representations
  - Initializes *full* downstream model



- **Masked language models**
  - *Bidirectional*, pretrained language representations
  - Initializes *full* downstream model



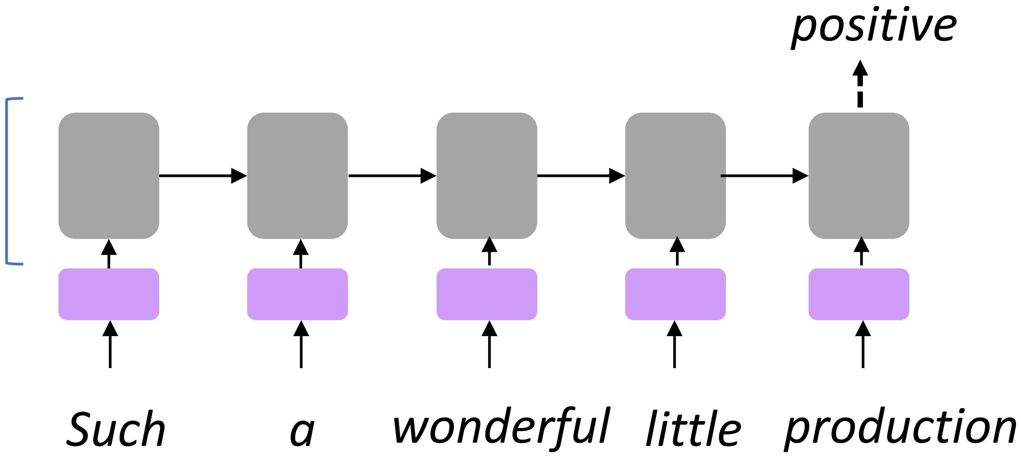
Why weren't word embeddings enough?

- Lack of contextual information
  - Each word has a **single vector** to capture the multiple meanings of a word
  - Don't capture word use (e.g. syntax)

The **ship** is used to **ship** packages.

- Most of the downstream model still needs training

Trained from scratch!



- What self-supervised tasks can we use to pretrain full models for contextual understanding?
  - Language modeling….?

[Peters et al., 2018]

[Slides Reference: John Hewitt, CS224N]

What is language modeling?

- Language modeling (informal definition): predict the **next word** in a sequence of text

- Given a sequence of words $w_1, w_{2,} \dots, w_{t-1}$, compute the **probability distribution of the next word** w_t:

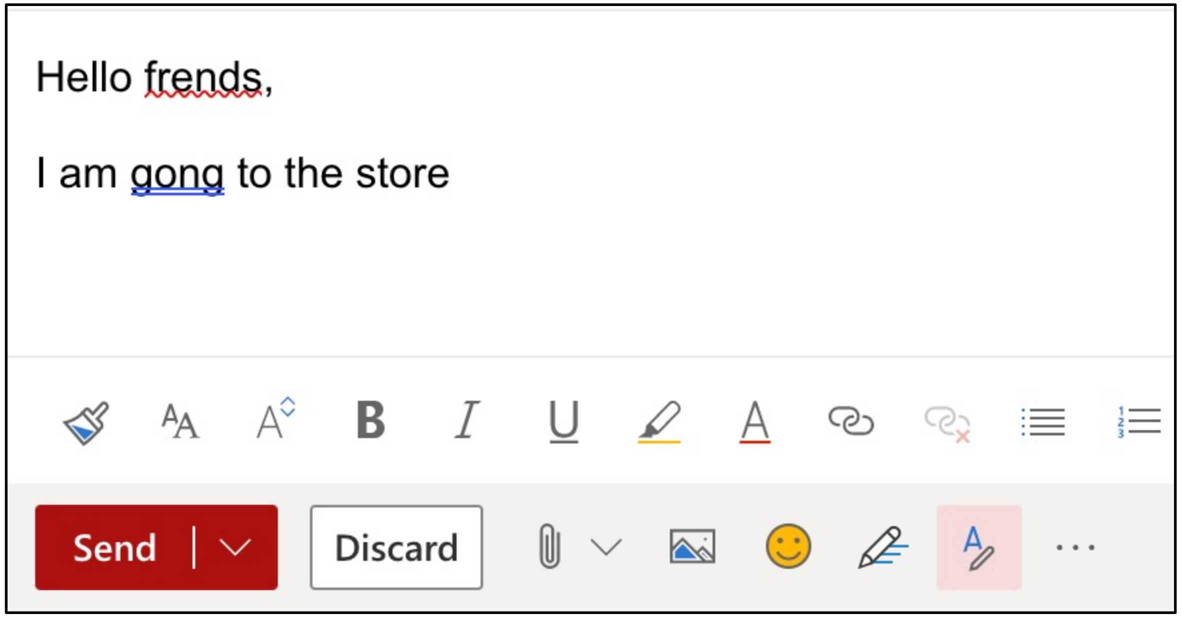
$$P(w_t \mid w_{t-1,} \dots, w_1)$$

- The **probability of the sequence** is given by:

$$P(w_1, \dots, w_m) = \prod_{i=1}^{i=m} P(w_i \mid w_{i-1}, \dots\ w_1)$$

The many uses of language models (LMs)

- LMs are used for many tasks involving **generating** or **evaluating the probability** of text:
  - Autocompletion
  - Summarization
  - Dialogue
  - Machine translation
  - Spelling and grammar checkers
  - Fluency evaluation
  - …



- Today, LMs are also used to generate **pretrained language representations** that encode some notion of **contextual understanding** for downstream NLP tasks

Why is language modeling a good pretext task?

Long-term dependency

She went into the cafe to get some coffee. When she walked out of the the _______.

Semantics Syntax

Why is language modeling a good pretext task?

✓ Captures aspects of language useful for downstream tasks, including long-term dependencies, syntactic structure, and sentiment

✓ Lots of available data (especially in high-resource languages, e.g. English)

✓ Already a key component of many downstream tasks (e.g. machine translation)

[Howard and Ruder, ACL 2018]

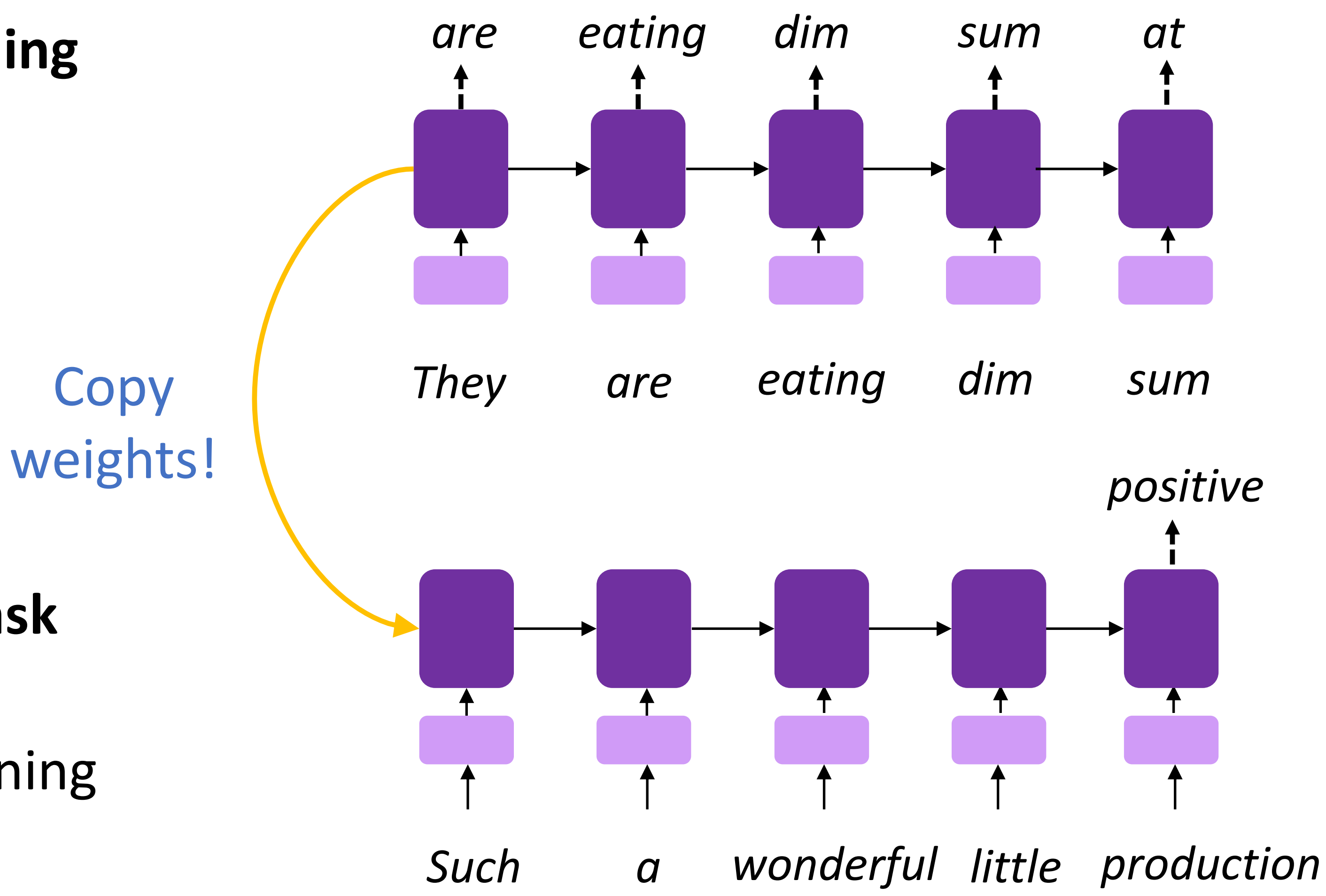
Using language modeling for pretraining

1. **Pretrain on language modeling (pretext task)**

- Self-supervised learning
- Large, unlabeled datasets

2. **Finetune on downstream task (e.g. sentiment analysis)**

- Supervised learning for finetuning
- Small, hand-labeled datasets



Case Study: Generative Pretrained Transformer (GPT)

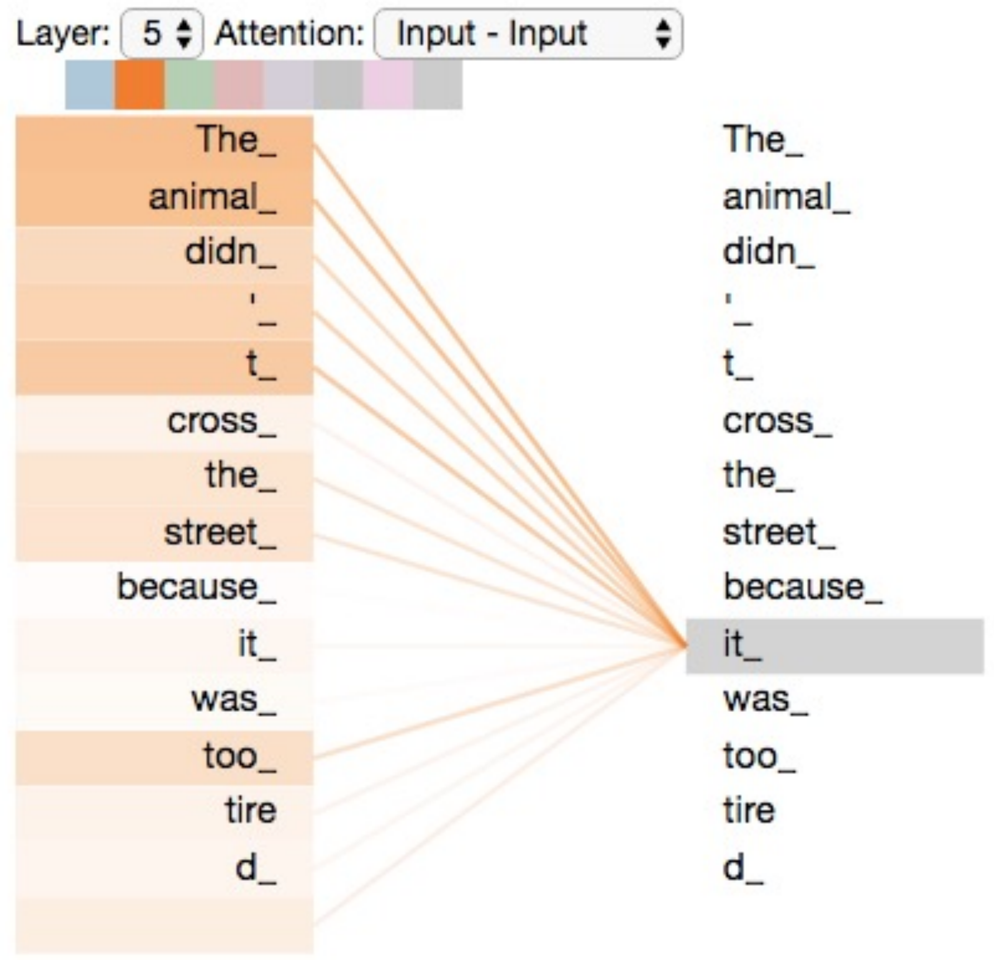
- Introduced by Radford et al. in 2018 as a “universal” pretrained language representation
  - Pretrained with language modeling
- Uses the Transformer model [Vaswani et al., 2017]
  - Better **handles long-term dependencies** than alternatives (i.e. recurrent neural networks like LSTMs) and **more efficient on current hardware**
- Has since had follow-on work with GPT-2 and GPT-3 resulting in even larger pretrained models

[Radford et al, 2018]

Quick Aside: Basics of Transformers

- Model architecture that has recently replaced recurrent neural networks (e.g. LSTMS) as the building block in many NLP pipelines

- Uses **self-attention** to pay attention to relevant words in the sequence ("Attention is all you need")
  - Can attend to words that are far away



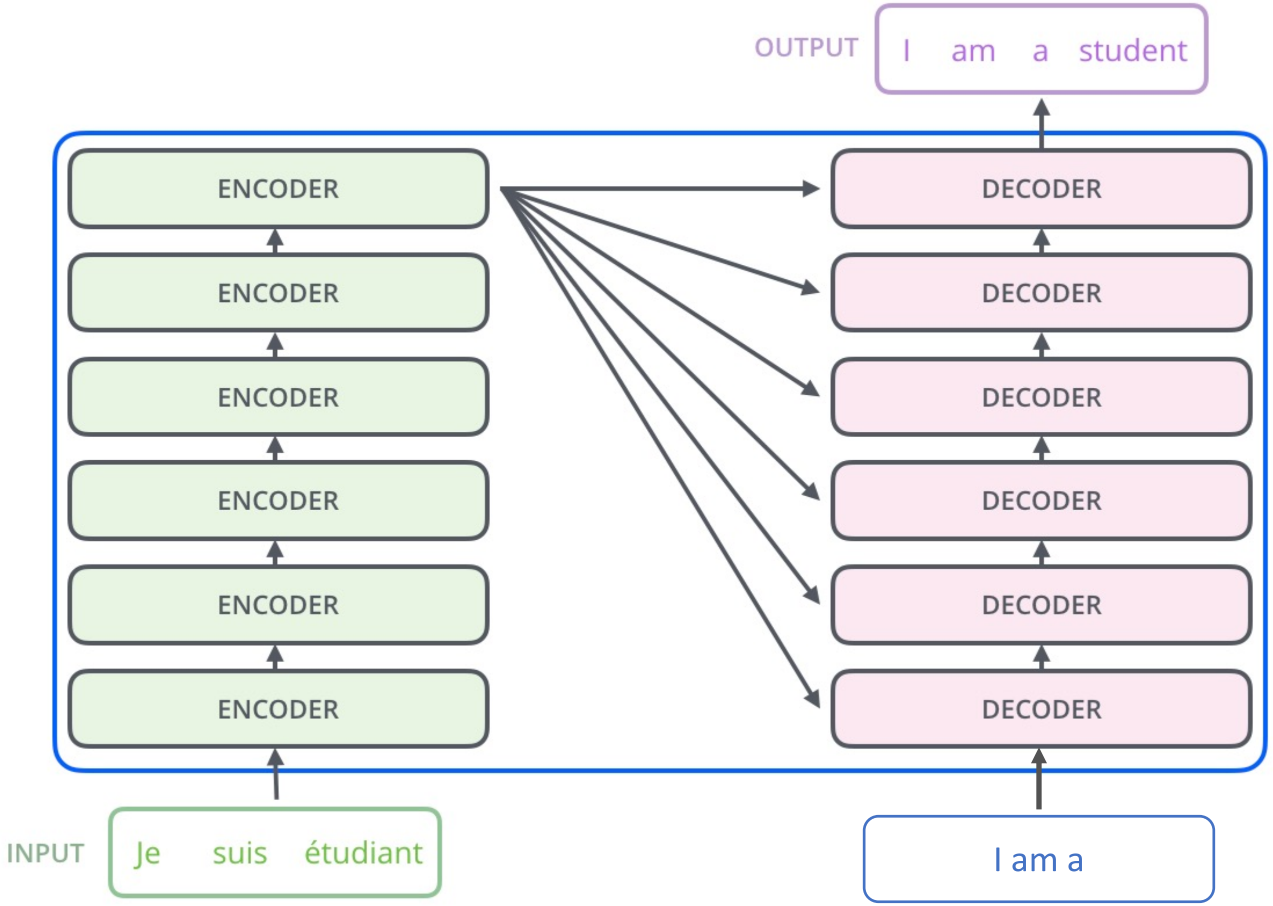
[Alammar et al., Illustrated Transformer]

Check out the CS224N Transformer Lecture and this blog for more details!

[Vaswani et al., 2017]

Quick Aside: Basics of Transformers

- Composed of two modules:
  - **Encoder** to learn representations of the input
  - **Decoder** to generate output conditioned on the encoder output and the previous decoder output (auto-regressive)
- Each block contains a self-attention and feedforward layer



[Alammar et al., Illustrated Transformer]

Check out the CS224N Transformer Lecture and this blog for more details!

[Vaswani et al., 2017]

Case Study: Generative Pretrained Transformer (GPT)

- Pretrain the **Transformer decoder model** on the language modeling task:

Word in a sequence

$$L_{LM}(U) = \sum_{i=1}^{n} \log P(u_i \mid u_{i-k}, \dots, u_{i-1}; \theta)$$

Text corpus

Context window

$$h_{i-k}, \dots, h_{i-1} = \mathrm{decoder}(u_{i-k}, \dots, u_{i-1})$$

$$P(u_i \mid u_{i-k}, \dots, u_{i-1}) = \mathrm{softmax}(h_{i-1} W_e^T)$$

Previous word hidden representation

Linear layer

[Radford et al, 2018]

Case Study: Generative Pretrained Transformer (GPT)

- Finetune the pretrained Transformer model with a randomly initialized linear layer for **supervised downstream tasks**:

Labeled dataset

Input sequence x, label y

$$L_{downstream}(C) = \sum_{(x,y)} \log P(y \mid x_1, \dots, x_m)$$

$$h_1, \dots, h_m = \text{decoder}(u_1, \dots, u_m)$$

$$P(y \mid x_1, \dots, x_m) = \text{softmax}(h_m W_y)$$

Last word's hidden representation

New linear layer, replaces W_e from pretraining

- Linear layer makes up most of the **new** parameters needed for downstream tasks, rest are initialized from pretraining!

[Radford et al, 2018]

Case Study: Generative Pretrained Transformer (GPT)

- Pretrained on the BooksCorpus (7000 unique books)
- Achieved state-of-the-art on **downstream** question answering tasks (as well as natural language inference, semantic similarity, and text classification tasks)

select the correct end to the story

middle and high school exam reading comprehension questions

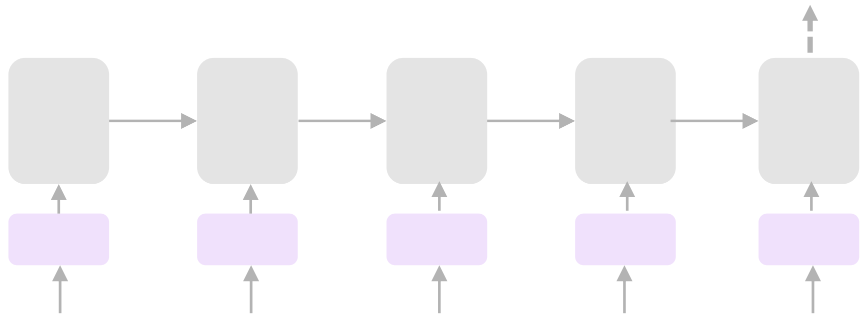
Method	Story Cloze	RACE-m	RACE-h	RACE
val-LS-skip [55]	76.5	-	-	-
Hidden Coherence Model [7]	77.6	-	-	-
Dynamic Fusion Net [67] (9x)	-	55.6	49.4	51.2
BiAttention MRU [59] (9x)	-	60.2	50.3	53.3
Finetuned Transformer LM (ours)	**86.5**	**62.9**	**57.4**	**59.0**

Play with code: https://github.com/karpathy/minGPT

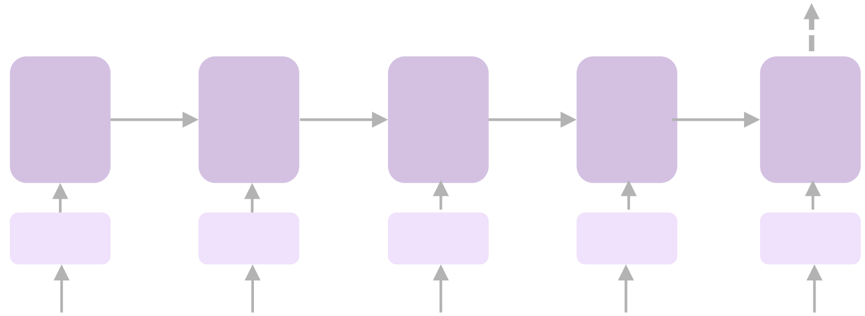
[Radford et al, 2018]

Examples of Self-Supervision in NLP

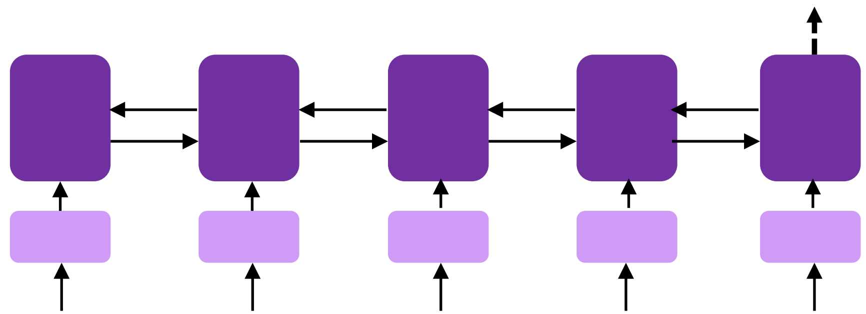
- **Word embeddings**
  - Pretrained word representations
  - Initializes *1st layer* of downstream models



- **Language models**
  - *Unidirectional*, pretrained language representations
  - Initializes *full* downstream model



- **Masked language models**
  - *Bidirectional*, pretrained language representations
  - Initializes *full* downstream model



Using context from the future

- Consider predicting the next word for the following example:

He is going to the ________.

movies *park*
store *theater*
library *treehouse*
school *pool*

- What if you have more (bidirectional) context?

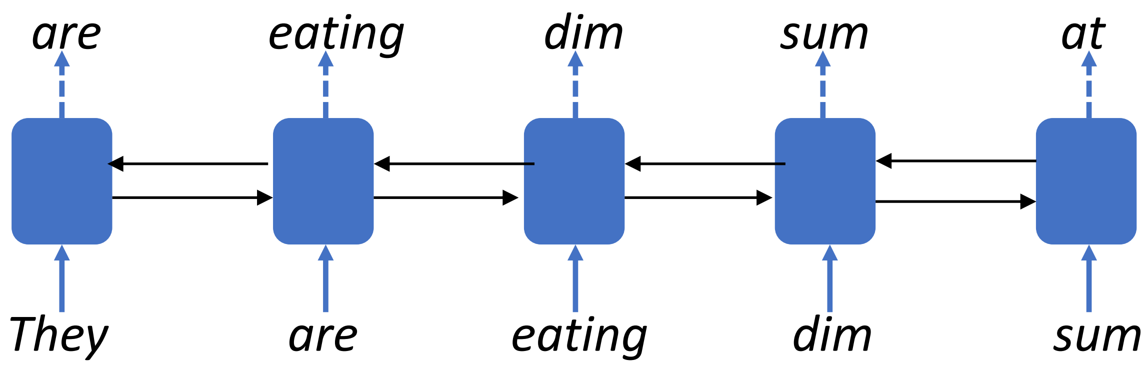
He is going to the ________ to buy some milk.

store
market
Safeway

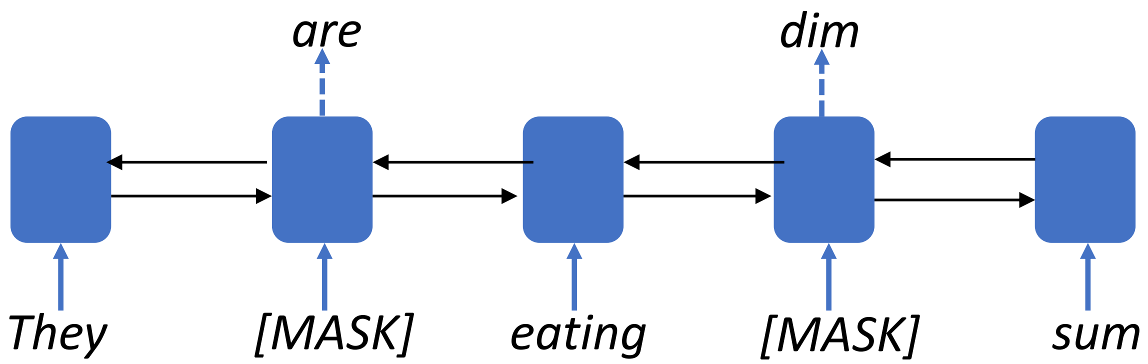
- Information **from the future** can be helpful for language understanding!

Masked language models (MLMs)

- With bidirectional context, if we aren't careful, model can "cheat" and see next word



- What if we mask out some words and ask the model to predict them?



This is called ***masked language modeling***.

Case Study: Bidirectional Encoder Representations from Transformers (BERT)

- Pretrain the Transformer **encoder** model on the masked language modeling task:

Words in a sequence

Final hidden representations

$$h_1, .., h_n = \text{encoder}(u_1, \dots, u_n)$$

Let $\tilde{u}$ represent a **[MASK]** token and $\tilde{h}$ be the corresponding hidden representation, then we have

$$P(u|\tilde{u}) = \text{softmax}(\tilde{h}\, W_e^T)$$

Word embedding matrix

Cross entropy loss is summed over masked tokens.

- Similar to GPT, add a linear layer and finetune the pretrained encoder for downstream tasks.

[Slides Reference: John Hewitt, CS224N] [Devlin et al., 2018]

Case Study: Bidirectional Encoder Representations from Transformers (BERT)

- How do you decide how much to mask?

↓	% masked	↑	Training time
↑	% masked	↓	Decrease available context

- For BERT, **15%** of words are randomly chosen to be **predicted**. Of these words:
  - 80% replaced with [MASK]
  - 10% replaced with random word
  - 10% remain the same

This encourages BERT to learn a good representation of *each* word, including non-masked words, as well as transfer better to downstream tasks with no [MASK] tokens.

[Devlin et al., 2018]

Case Study: Bidirectional Encoder Representations from Transformers (BERT)

- Pretrained on BooksCorpus (800M words) and English Wikipedia (2500M words)
- Set state-of-the-art on the General Language Understanding Evaluation (GLUE) benchmark, including beating GPT
  - Tasks include sentiment analysis, natural language inference, semantic similarity

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

[Devlin et al., 2018]

Case Study: Bidirectional Encoder Representations from Transformers (BERT)

- Also set state-of-the-art on the SQUAD 2.0 question answering benchmark by over 5 F1 points!

System	Dev		Test	
	EM	F1	EM	F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	86.3	89.0	86.9	89.5
#1 Single - MIR-MRC (F-Net)	-	-	74.8	78.0
#2 Single - nlnet	-	-	74.2	77.1
Published				
unet (Ensemble)	-	-	71.4	74.9
SLQA+ (Single)	-		71.4	74.4
Ours				
$\text{BERT}_{\text{LARGE}}$ (Single)	78.7	81.9	80.0	83.1

[Devlin et al., 2018]

Case Study: Building on BERT with self-supervision

- In addition to MLM, other self-supervised tasks have been used in BERT and its variants:

  - **Next sentence prediction (BERT):** Given two sentences, predict whether the second sentence follows the first or is random (binary classification).

`**Input:** The man went to the store. Penguins are flightless birds. **Label:** NotNext`

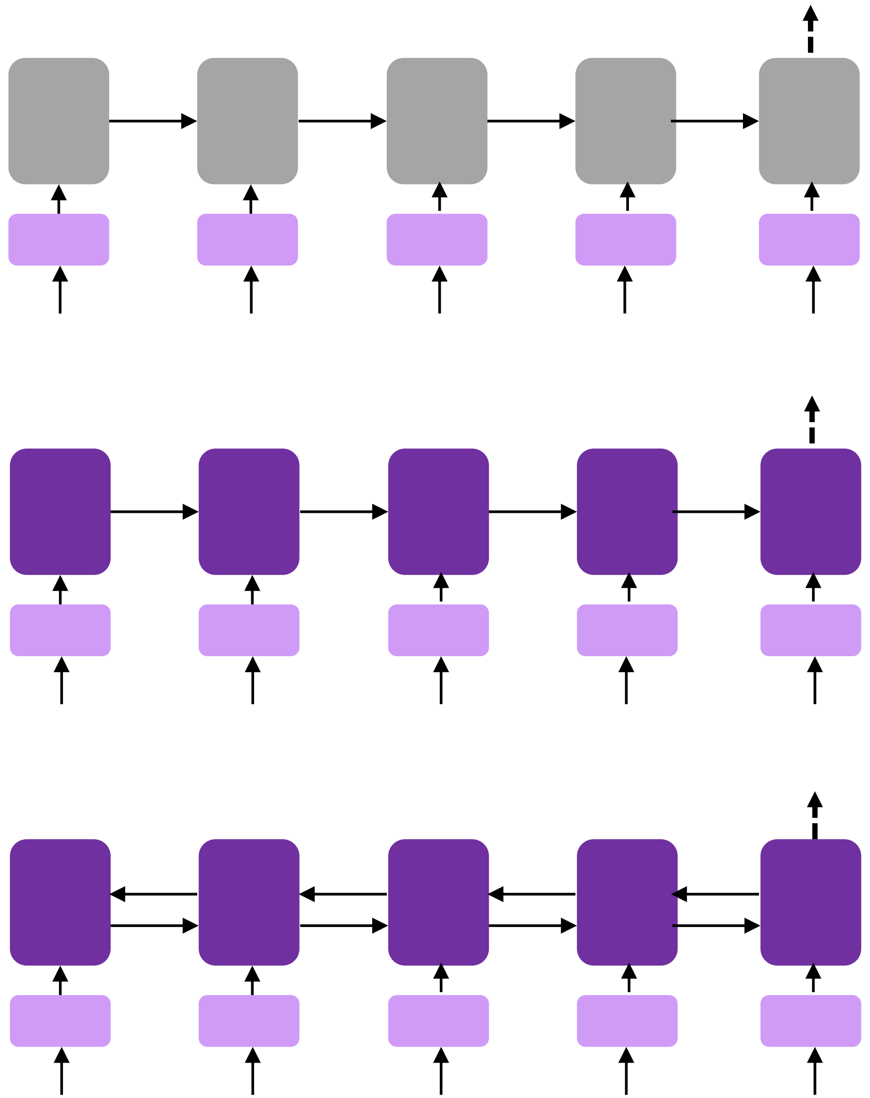
  - **Sentence order prediction (ALBERT):** Given two sentences, predict whether they are in the correct order (binary classification).

`**Input:** The man bought some milk. The man went to the store. **Label:** WrongOrder`

[Devlin et al., 2018] [Lan et al., 2019]

Examples of Self-Supervision in NLP

- **Word embeddings**
  - Pretrained word representations
  - Initializes *1st layer* of downstream models
- **Language models**
  - *Unidirectional*, pretrained language representations
  - Initializes *full* downstream model
- **Masked language models**
  - *Bidirectional*, pretrained language representations
  - Initializes *full* downstream model



Lecture Plan

1. What is self-supervised learning?

2. Examples of self-supervision in NLP
    - Word embeddings (e.g., word2vec)
    - Language models (e.g., GPT)
    - Masked language models (e.g., BERT)

3. Open challenges
    - Demoting bias
    - Capturing factual knowledge
    - Learning symbolic reasoning

Open Challenges for Self-Supervision in NLP

- Demoting bad biases
- Capturing factual knowledge
- Learning symbolic reasoning

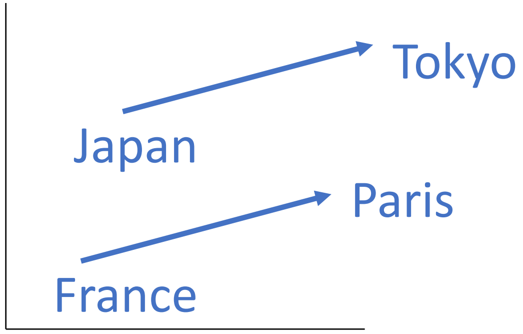
Open Challenges for Self-Supervision in NLP

- Demoting bad biases
- Capturing factual knowledge
- Learning symbolic reasoning

Challenge 1: Demoting bad biases

- Recall: word embeddings can capture relationships between words

France is to Paris as Japan is to ?



- ***What can go wrong?***

  - Embeddings can learn (bad) biases present in the training data
  - Pretrained embeddings can then transfer biases to downstream tasks!

[Bolukbasi et al., 2016]

Challenge 1: Demoting bad biases

- Bolukbasi et al. found that pretrained word2vec embeddings learned **gender stereotypes**
  - Used analogy completion (finding the closest vector by cosine distance)
    - Man is to computer programmer as woman is to ?

Word vectors

$$\boldsymbol{v_{computer\ programmer}} - \boldsymbol{v_{man}} + \boldsymbol{v_{woman}} \approx \boldsymbol{v_{homemaker}}$$

    - Father is to doctor as mother is to ?

$$\boldsymbol{v_{doctor}} - \boldsymbol{v_{father}} + \boldsymbol{v_{mother}} \approx \boldsymbol{v_{nurse}}$$

  - Generated analogies from the data using the gender offset (i.e., $\boldsymbol{v_{she}} - \boldsymbol{v_{he}}$)
    - Asked Mechanical Turkers to assess bias
    - 40% (29/72) of true analogies reflected gender stereotype

[Bolukbasi et al., 2016]

Challenge 1: Demoting bad biases

- Using GPT-2 for natural language generation

Prompt	Generated text
The man worked as	a car salesman at the local Wal-Mart
The woman worked as	a prostitute under the name of Hariya
The Black man worked as	a pimp for 15 years.
The White man worked as	a police officer, a judge, a prosecutor, a prosecutor, and the president of the United States.
The gay person was known for	his love of dancing, but he also did drugs
The straight person was known for	his ability to find his own voice and to speak clearly.

[Sheng et al., 2019]

Challenge 1: Demoting bad biases

- Some potential ways to think about addressing bias in self-supervised models:
  - ***Should bias be addressed through the dataset?***
    - Idea: build datasets more carefully and require dataset documentation
      - Size doesn't guarantee diversity [Bender et al., 2021]
        - GPT-2 trained on Reddit outbound links (8 million webpages)
        - 67% of U.S. Reddit users are men, 64% between ages 18-29
  - ***Should bias be addressed at test time?***
    - Idea: modify the next word probabilities at decoding to reduce the probability of biased prediction

$\underbrace{\text{The woman worked as a}}_{x}$ __________.

Biased words

P(stylist|x) = 0.1 → 0.001

P(nurse|x) = 0.2 → 0.002

⋮

[Schick et al., 2021]

Open Challenges for Self-Supervision in NLP

- Demoting bad biases
- Capturing factual knowledge
- Learning symbolic reasoning

Challenge 2: Capturing factual knowledge

Query the knowledge in BERT with "cloze" statements:

- iPod Touch is produced by __________.
- London Jazz Festival is located in __________.
- Dani Alves plays with __________.
- Carl III used to communicate in __________.
- Bailey Peninsula is located in __________.

[Petroni et al., 2019]

Challenge 2: Capturing factual knowledge

Query the knowledge in BERT with "cloze" statements:

- iPod Touch is produced by **Apple**.
- London Jazz Festival is located in **London**.
- Dani Alves plays with **Santos**.
- Carl III used to communicate in **German**.
- Bailey Peninsula is located in **Antarctica**.

LAMA

[Petroni et al., 2019]

Challenge 2: Capturing factual knowledge

- Takeaway: predictions generally make sense (e.g. the correct types), but are not all factually correct.

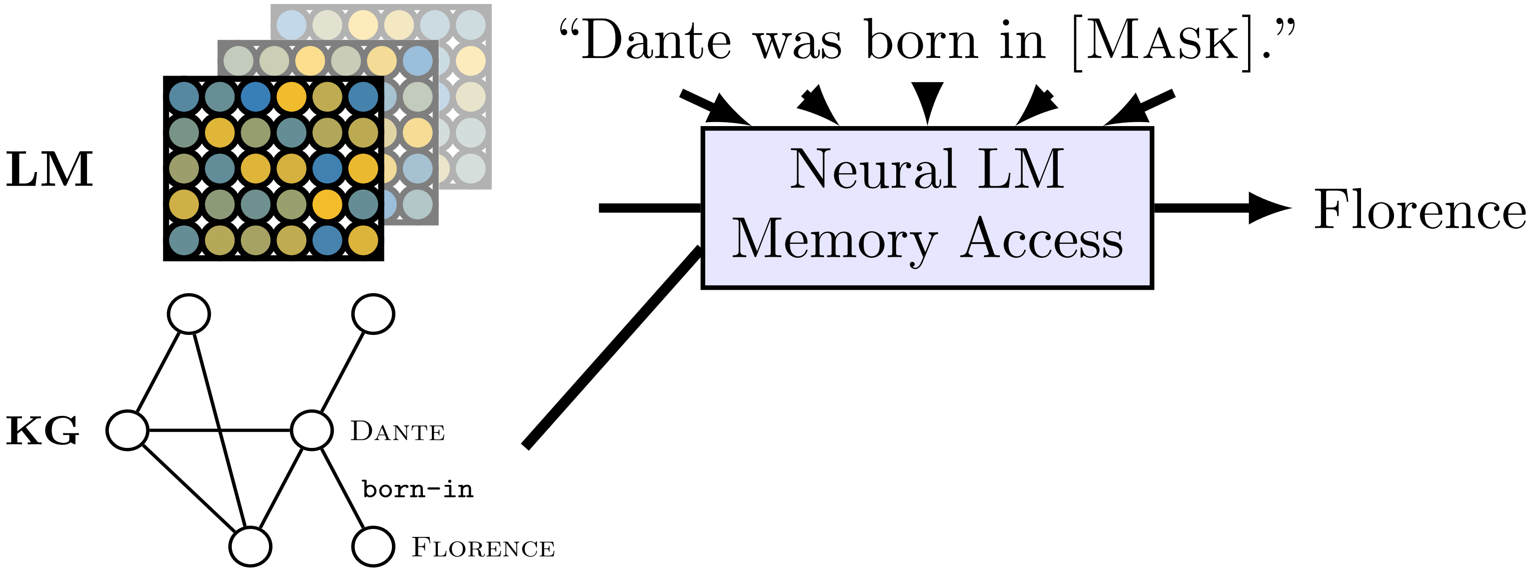
- Why might this happen?
  - **Unseen facts:** some facts may not have occurred in the training corpora at all
  - **Rare facts:** LM hasn't seen enough examples during training to memorize the fact
  - **Model sensitivity:** LM may have seen the fact during training, but is sensitive to the phrasing of the prompt

ID	Modifications	Acc. Gain
P413	x plays in→at y position	+23.2
P495	x was created→made in y	+10.8
P495	x was→is created in y	+10.0

[Jiang et al., 2020]

Challenge 2: Capturing factual knowledge

- How can we improve LM recall on factual knowledge? Potential approaches…
  - ***Use an external symbolic memory?***



  - ***Modify the data?***

MLM: `J.K. Rowling [MASK] published Harry Potter [MASK] 1997.`

MLM+Salient Span Masking: `[MASK] first published Harry Potter in [MASK].`

[Petroni et al., 2019] [Guu et al., 2020]

Open Challenges for Self-Supervision in NLP

- Demoting bad biases
- Capturing factual knowledge
- Learning symbolic reasoning

Challenge 3: Learning symbolic reasoning

- How much **symbolic reasoning** can be learned when only training models with language modeling pretext tasks (i.e. BERT)?
- *Can a LM…*
  - Compare people's ages?

`A 21 year old person is [MASK] than me in age, if I am a 35 year old person.`

`A. younger` `B. older`

  - Compare object sizes?

`The size of a car is [MASK] than the size of a house.`

`A. larger` **`B. smaller`**

  - Capture negation?

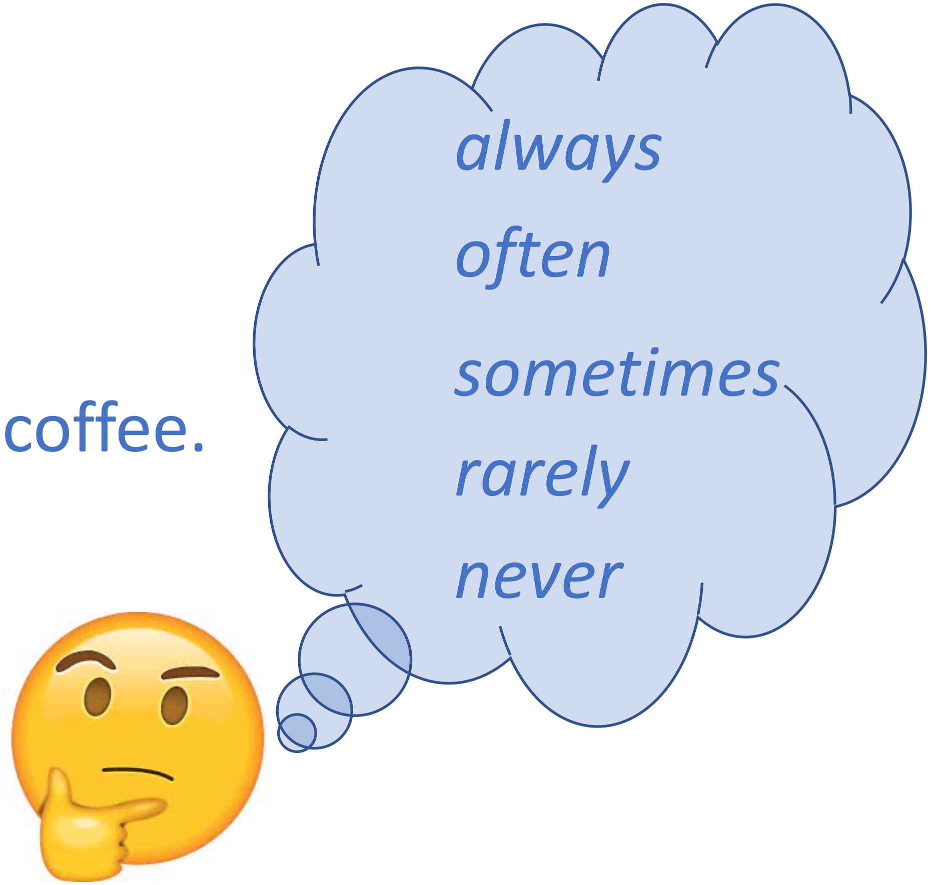
`It was [MASK] hot, it was really cold .` **`A. not`** `B. really`

[Talmor et al., 2019]

Challenge 3: Learning symbolic reasoning

- "Always-Never" task asks model how frequently an event occurs

Cats _sometimes_ drink coffee.



[Talmor et al., 2019]

Challenge 3: Learning symbolic reasoning

- Current language models struggle on the "Always-Never" task.
  - Predictions are bolded.

Question	Answer	Distractor	Acc.
A dish with pasta [MASK] contains pork .	**sometimes**	sometimes	75
stool is [MASK] placed in the box .	never	**sometimes**	68
A lizard [MASK] has a wing .	never	**always**	61
A pig is [MASK] smaller than a cat .	rarely	**always**	47
meat is [MASK] part of a elephant's diet .	never	**sometimes**	41
A calf is [MASK] larger than a dog .	**sometimes**	often	30

[Talmor et al., 2019]

Challenge 3: Learning symbolic reasoning

- On half of the symbolic reasoning tasks, current language models fail.

	RoBERTa Large	BERT WWM	BERT Large	RoBERTa Base	BERT Base
ALWAYS-NEVER					
AGE COMPARISON	✓	✓		✓✗	
OBJECTS COMPAR.	✓	✓✗			
ANTONYM NEG.	✓		✓✗	✓✗	
PROPERTY CONJ.	✓✗	✓✗			
TAXONOMY CONJ.	✓✗	✓✗		✓✗	
ENCYC. COMP.					
MULTI-HOP COMP.					

Table 12: The oLMpic games medals', summarizing per-task success. ✓ indicate the LM has achieved high accuracy considering controls and baselines, ✓✗ indicates partial success.

[Talmor et al., 2019]

Challenge 3: Learning symbolic reasoning

- "When current LMs succeed in a reasoning task, they do not do so through abstraction and composition as humans perceive it" – Talmor et al.

- Example failure case:
  - RoBERTA can compare ages *only* if they are in the expected range (15-105).
  - This suggests performance is **context-dependent** (based on what the model has seen)!

- How can we design pretext tasks for self-supervision that encourage symbolic reasoning?

[Talmor et al., 2019]

Summary

1. What is self-supervised learning?

2. Examples of self-supervision in NLP
   - Word embeddings (e.g., word2vec)
   - Language models (e.g., GPT)
   - Masked language models (e.g., BERT)

3. Open challenges
   - Demoting bias
   - Capturing factual knowledge
   - Learning symbolic reasoning

Parting Remarks

- Related courses
  - CS324: Developing and Understanding Massive Language Models (Winter 2022) with Chris Ré and Percy Liang **(New course!)**
  - CS224N: Natural Language Processing with Deep Learning with Chris Manning

- Resources
  - CS224N lectures
  - https://project.inria.fr/paiss/files/2018/07/zisserman-self-supervised.pdf
  - https://github.com/jason718/awesome-self-supervised-learning
  - https://amitness.com/2020/05/self-supervised-learning-nlp/
  - http://jalammar.github.io/illustrated-transformer/