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

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

[Deng et al., 2009]
Supervised pretraining on large labeled, datasets has led to successful transfer learning across images, video, and text.

Premise:
Ruth Bader Ginsburg being appointed to the US Supreme Court.

Hypothesis:
A grilled sandwich on a plate.

Label:
Contradiction [different scenes]

[SNLI Dataset]
[Kinetics Dataset]

[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 Mechanical Turkers [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

---

**Diagram:**
- Data
- Labelers
- Pretraining Task
- Downstream Tasks
Pretext Task: Classify the Rotation

270° rotation
Identifying the object helps solve rotation task!

90° rotation

180° rotation

Catfish species that swims upside down...
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

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)
  \[
  \text{pizza} = [0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ldots \ 0 \ 0 \ 0 \ 0 \ 0 ] \\
  \text{pie} = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ldots \ 0 \ 0 \ 1 \ 0 ]
  \]

❓ 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!

\textit{predict: pizza}

In Italy, \textcolor{blue}{pizza} served in formal settings is eaten with a fork and knife.

\textit{context window, size 2}

\textit{repeat for each word}:

\textit{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!

**predict:** In Italy served in

context window, size 2

repeat for each word :: predict: with a and knife

In Italy, pizza served in formal settings is eaten with a fork and knife.
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 \( w_t \).

\[
J(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{-m \leq j \leq m, j \neq 0} \log P(w_{t+j} | w_t; \theta)
\]

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

\[
P(w_{t+j} | w_t; \theta) = P(u_{t+j} | v_t) = \frac{\exp(u_{t+j}^T v_t)}{\sum_{j=1}^{l} \exp(u_j^T v_t)} \quad \text{[Softmax]} \]

[\text{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”.

\[ P(\text{knife}|\text{fork}) \]

\[ \text{... is eaten with a fork and knife.} \]

1. Get “fork” word vector \( \mathbf{v}_{\text{fork}} \)

2. Compute scores

3. Convert to probabilities

[![Diagram of word2vec process](image)](image)

[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 ?

\[ \mathbf{w}^* = \arg\max_{\mathbf{w}} \frac{\mathbf{v}_{\mathbf{w}, \mathbf{y}}}{\|\mathbf{v}_w\| \|\mathbf{y}\|}, \]

where \( \mathbf{y} = \mathbf{v}_{Paris} - \mathbf{v}_{France} + \mathbf{v}_{Japan} \)

\[ \mathbf{w}^* = \text{Tokyo} \]

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

[1] 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)

• Most of the downstream model still needs training

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

---

The ship is used to ship packages.

Trained from scratch!

Such a wonderful little production

[Slides Reference: John Hewitt, CS224N]

[Peters et al., 2018]
What is language modeling?

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

  - Given a sequence of words $w_1, w_2, \ldots, w_{t-1}$, compute the **probability distribution of the next word** $w_t$:

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

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

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

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

- **Long-term dependency**
- **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

They are eating dim sum at

Copy weights!

Such a wonderful little production
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

Check out the [CS224N Transformer Lecture](#) and [this blog](#) for more details! [Alammar et al., Illustrated Transformer]

[Alammar et al., Illustrated Transformer]

[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

Check out the [CS244N Transformer Lecture](#) and [this blog](#) for more details! [Alammar et al., Illustrated Transformer] [Vaswani et al., 2017]
Case Study: Generative Pretrained Transformer (GPT)

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

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

\[
h_{i-k, ..., i-1} = \text{decoder}(u_{i-k}, ..., u_{i-1})
\]

\[
P(u_i \mid u_{i-k}, ..., u_{i-1}) = \text{softmax}(h_{i-1}W_e^T)
\]

[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:

\[ L_{\text{downstream}}(C) = \sum_{(x, y)} \log P(y \mid x_1, ..., x_m) \]

\[ h_1, ..., h_m = \text{decoder}(u_1, ..., u_m) \]

\[ P(y \mid x_1, ..., x_m) = \text{softmax}(h_m W_y) \]

• 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)

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

Play with code: [https://github.com/karpathy/minGPT](https://github.com/karpathy/minGPT)

[Radford et al, 2018] 36
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 ________.

• What if you have more (bidirectional) context?

  He is going to the ________ to buy some milk.

• 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:
  
  \[ \begin{align*}
  h_1, \ldots, h_n & = \text{encoder}(u_1, \ldots, u_n) \\
  \end{align*} \]

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

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

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

<table>
<thead>
<tr>
<th>System</th>
<th>MNLI-(m/mm)</th>
<th>QQP</th>
<th>QNLI</th>
<th>SST-2</th>
<th>CoLA</th>
<th>STS-B</th>
<th>MRPC</th>
<th>RTE</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-OpenAI SOTA</td>
<td>80.6/80.1</td>
<td>66.1</td>
<td>82.3</td>
<td>93.2</td>
<td>35.0</td>
<td>81.0</td>
<td>86.0</td>
<td>61.7</td>
<td>74.0</td>
</tr>
<tr>
<td>BiLSTM+ELMo+Attn</td>
<td>76.4/76.1</td>
<td>64.8</td>
<td>79.8</td>
<td>90.4</td>
<td>36.0</td>
<td>73.3</td>
<td>84.9</td>
<td>56.8</td>
<td>71.0</td>
</tr>
<tr>
<td>OpenAI GPT</td>
<td>82.1/81.4</td>
<td>70.3</td>
<td>87.4</td>
<td>91.3</td>
<td>45.4</td>
<td>80.0</td>
<td>82.3</td>
<td>56.0</td>
<td>75.1</td>
</tr>
<tr>
<td>BERTBASE</td>
<td>84.6/83.4</td>
<td>71.2</td>
<td>90.5</td>
<td>93.5</td>
<td>52.1</td>
<td>85.8</td>
<td>88.9</td>
<td>66.4</td>
<td>79.6</td>
</tr>
<tr>
<td>BERTLARGE</td>
<td>86.7/85.9</td>
<td>72.1</td>
<td>92.7</td>
<td>94.9</td>
<td>60.5</td>
<td>86.5</td>
<td>89.3</td>
<td>70.1</td>
<td>82.1</td>
</tr>
</tbody>
</table>

[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!

<table>
<thead>
<tr>
<th>System</th>
<th>Dev EM</th>
<th>Dev F1</th>
<th>Test EM</th>
<th>Test F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Leaderboard Systems (Dec 10th, 2018)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Human</td>
<td>86.3</td>
<td>89.0</td>
<td>86.9</td>
<td>89.5</td>
</tr>
<tr>
<td>#1 Single - MIR-MRC (F-Net)</td>
<td>-</td>
<td>-</td>
<td>74.8</td>
<td>78.0</td>
</tr>
<tr>
<td>#2 Single - nlnet</td>
<td>-</td>
<td>-</td>
<td>74.2</td>
<td>77.1</td>
</tr>
<tr>
<td>Published</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>unet (Ensemble)</td>
<td>-</td>
<td>-</td>
<td>71.4</td>
<td>74.9</td>
</tr>
<tr>
<td>SLQA+ (Single)</td>
<td>-</td>
<td></td>
<td>71.4</td>
<td>74.4</td>
</tr>
<tr>
<td>Ours</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BERT_{LARGE} (Single)</td>
<td>78.7</td>
<td>81.9</td>
<td>80.0</td>
<td>83.1</td>
</tr>
</tbody>
</table>

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

    **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 ?
      \[ \mathbf{v}_{\text{computer programmer}} - \mathbf{v}_{\text{man}} + \mathbf{v}_{\text{woman}} \approx \mathbf{v}_{\text{homemaker}} \]
    • Father is to doctor as mother is to ?
      \[ \mathbf{v}_{\text{doctor}} - \mathbf{v}_{\text{father}} + \mathbf{v}_{\text{mother}} \approx \mathbf{v}_{\text{nurse}} \]
  • Generated analogies from the data using the gender offset (i.e., \( \mathbf{v}_{\text{she}} - \mathbf{v}_{\text{he}} \))
    • 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

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

\[
\begin{align*}
\text{Biased words} & \\
P(\text{stylist}|x) &= 0.1 &\rightarrow& 0.001 \\
P(\text{nurse}|x) &= 0.2 &\rightarrow& 0.002 \\
\vdots
\end{align*}
\]

The woman worked as a ________.

\[x\]
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 ________.  
  
• 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

• 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

<table>
<thead>
<tr>
<th>ID</th>
<th>Modifications</th>
<th>Acc. Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>P413</td>
<td>$x$ plays <strong>in</strong>→<strong>at</strong> $y$ position</td>
<td>+23.2</td>
</tr>
<tr>
<td>P495</td>
<td>$x$ was <strong>created</strong>→<strong>made</strong> in $y$</td>
<td>+10.8</td>
</tr>
<tr>
<td>P495</td>
<td>$x$ was→<strong>is</strong> created in $y$</td>
<td>+10.0</td>
</tr>
</tbody>
</table>

[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?**

  “Dante was born in [MASK].”

  ![Diagram showing LM, KG, Dante born-in Florence, and Neural LM Memory Access]

  • **Modify the data?**

  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.

[Source: Talmor et al., 2019]
Challenge 3: Learning symbolic reasoning

• Current language models struggle on the “Always-Never” task.
  • Predictions are bolded.

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

[Talmor et al., 2019]
Challenge 3: Learning symbolic reasoning

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

<table>
<thead>
<tr>
<th></th>
<th>RoBERTa Large</th>
<th>BERT WWM</th>
<th>BERT Large</th>
<th>RoBERTa Base</th>
<th>BERT Base</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ALWAYS-NEVER</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AGE COMPARISON</strong></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td><strong>OBJECTS COMPAR.</strong></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ANTONYM NEG.</strong></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td><strong>PROPERTY CONJ.</strong></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>TAXONOMY CONJ.</strong></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ENCYC. COMP.</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>MULTI-HOP COMP.</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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://github.com/jason718/awesome-self-supervised-learning
  • https://amitness.com/2020/05/self-supervised-learning-nlp/
  • http://jalammar.github.io/illustrated-transformer/