How Much Can We Trust Large Language Models?

Fatemehsadat Mireshghallah
Adversarial ML Workshop @KDD 2022
What are Language Models?

- A language model is a probability distribution over sequences of words
- Model what words a given word/context normally appears with
- Used in medical, legal, financial, etc. domains

The students opened their _______.

books
laptops
exams
Different Types of Language Models

- Statistical Models:
  - N-grams: build tables based on n-gram frequency

Uni-gram: The students opened their books
Bi-gram: The students opened their books.
Tri-gram: The students opened their books.
Different Types of Language Models

- **Statistical Models:**
  - N-grams

- **Neural Models:**
  - Recurrent Neural Networks

The students opened their LSTM books.
Different Types of Language Models

- **Statistical Models:**
  - N-grams

- **Neural Models:**
  - Recurrent Neural Networks
  - Transformer-based Models

The students opened their books
Different Types of Language Models

- **Statistical Models:**
  - N-grams

- **Neural Models:**
  - Recurrent Neural Networks
  - Transformer-based Models
Different Types of Language Models

- **Statistical Models:**
  - N-grams

- **Neural Models:**
  - Recurrent Neural Networks
  - Transformer-based Models
    - Auto-regressive, causal language models (left to right)
Different Types of Language Models

- Statistical Models:
  - N-grams

- Neural Models:
  - Recurrent Neural Networks
  - Transformer-based Models
    - Auto-regressive, causal language models (left to right)
    - Non-autoregressive, Masked language models (bidirectional)
Large Language Models (LLMs)

- Transformer-based language models are often referred to as ‘Large LMs’ due to their parameter count (ranging from 100 million to billions of parameters)

- Deployed with Pre-train and Fine-tune paradigm
Pre-train and Fine-tune

- Pretrain on large ‘public’ (usually web-scraped) data, with causal or masked LM objective.
Pre-train and Fine-tune

- Pretrain on large ‘public’ (usually web-scraped) data, with causal or masked LM objective.

- Fine-tune for down-stream domain adaptation, the objective depends on the task.
Large Language Models: The Good and the Bad …

- Large language models are very good at generating text and learning representations. However:
  - They are extremely large models: high capacity for memorization
  - They are trained on huge, unvetted, scraped data: high potential for harmful/hateful/private content
Talk outline

1. Safety Issues with Large Language Models
2. Measuring Leakage in Pre-training Large Language Models
3. Measuring Leakage in Fine-tuning Methods
4. Open Problems and Future Directions
Problem 1: Large Models are Leaky!

Long live the revolution.
Our next meeting will be
at the docks at midnight
on June 28

Aha, found them!

When you train predictive models
on input from your users, it can
leak information in unexpected ways.

xkcd.com/2169/
Problem 1: Large Models are Leaky!

Prompt: East Stroudsburg Stroudsburg...

Large Language Model (GPT-2)

Memorized Text:
- Corp. Name: **** Corp. Seabank Centre
- Person’s Name: Peter W****
- Email: ****@****. com
- Phone Number: +****7 5****
Problem 2: Large Models (and Even Humans) are Sneaky!

Both humans and ML models can classify sensitive attributes about author given raw text.
Problem 2: Large Models (and Even Humans) are Sneaky!

Representations learned from text can reflect sensitive attributes.

Wang et al. Dynamically Disentangling Social Bias from Task-Oriented Representations with Adversarial Attack. NAACL 2021
Problem 2: Large Models (and Even Humans) are Sneaky!

Representations learned from text can reflect sensitive attributes.

Wang et al. Dynamically Disentangling Social Bias from Task-Oriented Representations with Adversarial Attack. NAACL 2021
Problem 3: Large Models are Creepy!

What was the Muslim girl known for?

For being fat and old.

Being from North Africa, I assume that one.

What was the Muslim boy known for?

There is actually a story where he was the father of a guy who wanted to murder the Jews with his shotgun.

Being born in Sweden.
Talk outline

1. Safety Issues with Large Language Models
2. Measuring Leakage in Pre-training Large Language Models
3. Measuring Leakage in Fine-tuning Methods
4. Open Problems and Future Directions
What is information leakage in an ML model?

- ‘Leakage’ is being able to learn information about the training data, which cannot be learned from other models/data (from the same distribution)
Measuring Leakage: Membership Inference Attacks

- Can an adversary infer whether a particular data point “x” is part of its training set?

- Success of attacker is a metric to quantify information leakage of the model about its individual training data
Membership Inference Attack Categories

- Shadow Model-based attacks [Shokri et al. 2017]
  - Train many shadow models with different data partitioning, train an attack model on the logits of the shadow models.
Membership Inference Attack Categories

- Shadow Model-based attacks [Shokri et al. 2017]
  - Train many shadow models with different data partitioning, train an attack model on the logits of the shadow models.
  - Inefficient!
Membership Inference Attack Categories

- **Shadow Model-based attacks** [Shokri et al. 2017]
- **Population loss-based threshold attacks** [Yeom et al. 2018]:
  - Set a hard threshold on loss for inferring membership, based on a held out test set
Membership Inference Attack Categories

- Shadow Model-based attacks [Shokri et al. 2017]
- Population loss-based threshold attacks [Yeom et al. 2018]:
  - Set a hard threshold on loss for inferring membership, based on a held out test set
  - Efficient, but inaccurate: does not consider sample difficulty!
Membership Inference Attack Categories

- Shadow Model-based attacks [Shokri et al. 2017]
- Population loss-based threshold attacks [Yeom et al. 2018]
- Reference-based attacks [Carlini et al., 2021]
  - Train reference models that have a large agreement with the target model on all data, except the target data
Membership Inference Attack Categories

- Shadow Model-based attacks [Shokri et al. 2017]
- Population loss-based threshold attacks [Yeom et al. 2018]
- Reference-based attacks [Carlini et al., 2021]

  - Train reference models that have a large agreement with the target model on all data, except the target data
  - Use likelihood ratio: \( LR(s) = \frac{p(s; \theta_R)}{p(s; \theta)} \)
Membership Inference Attack Categories

- Shadow Model-based attacks [Shokri et al. 2017]
- Population loss-based threshold attacks [Yeom et al. 2018]
- Reference-based attacks [Carlini et al., 2021]

  - Train reference models that have a large agreement with the target model on all data, except the target data
  
  - Use likelihood ratio: \[ LR(s) = \frac{p(s; \theta_R)}{p(s; \theta)} \]
  
  - By thresholding the LR, we infer membership:

\[ if \ LR(s) < t \rightarrow s \in D \]
Existing Attacks

Shokri et al. (2017); Nasr et al. (2019); Sablayrolles et al. (2019); Leino & Fredrikson (2020); Li & Zhang (2020); Choquette-Choo et al. (2021); Melis et al. (2019); Zou et al. (2020); Hayes et al. (2019); Carlini et al. (2019); Song & Shmatikov (2019); Carlini et al. (2020); Song & Raghunathan (2020); Shah et al. (2021); Long et al. (2018); Song & Mittal (2021); Salem et al. (2018); Murakonda et al. (2021); Homer et al. (2008); Sankararaman et al. (2009); Backes et al. (2016); Dwork et al. (2015); Yeom et al. (2018); Del Grosso et al. (2021); Chang & Shokri (2020); Song et al. (2019); Shokri et al. (2021); Erlingsson et al. (2019); Humphries et al. (2020); Jayaraman & Evans (2019); Rahman et al. (2018); Jagielski et al. (2020); Nasr et al. (2021); Malek et al. (2021) ...
How Do We Measure Leakage in Language Models?

- Autoregressive (causal) Models:
  - Exposure metric [Carlini et al. 2019]: How easy is it to extract artificially inserted sentences from a model.
How Do We Measure Leakage in Language Models?

- **Autoregressive (causal) Models:**
  - Exposure metric [Carlini et al. 2019]: How easy is it to extract artificially inserted sentences from a model
  - Sample Extraction Attack on GPT-2 [Carlini et al. 2021]:
    - Generate 500k samples from the model
How Do We Measure Leakage in Large Language Models?

- **Autoregressive (causal) Models:**
  - Exposure metric [Carlini et al. 2019]: How easy is it to extract artificially inserted sentences from a model
  - Sample Extraction Attack on GPT-2 [Carlini et al. 2021]:
    - Generate 500k samples from the model
    - Sift through them using a reference based MIA to find actual training samples: over 60% precision
How Do We Measure Leakage in Masked Language Models?

- Masked Language Models: pre-trained with mask reconstruction objective
- Model the conditional \( p([MASK]|s_{\text{\[MASK]\}}; \theta) \) for a given sequence \( s \).
- Used to provide contextual representations for down-stream tasks (e.g. classification)
How Do We Measure Leakage in Masked Language Models?

- Extraction attacks [Lehman et al. 2021]: Fill in the blank and sampling attacks, very low success rate.

Mr. Smith has … Mr. Smith has diabetes.

Probe
How Do We Measure Leakage in Masked Language Models?

- Extraction attacks [Lehman et al. 2021]: Fill in the blank and sampling attacks, very low success rate.
- Membership Inference Attacks [jagannatha2021]: Population-based attack, near random performance
How Do We Measure Leakage in Masked Language Models?

- Extraction attacks [Lehman et al. 2021]: Fill in the blank and sampling attacks, very low success rate.
- Membership Inference Attacks [jagannatha2021]: Population-based attack, near random performance

Does this mean MLMs memorize less than auto-regressive models? Or do we just need stronger attacks?
Reference-based attack for MLMs

- What are stronger attacks?
  1. **Shadow model-based attacks**: these would be extremely hard (even impossible) for large language models, given the scale of pre-training data and compute
Reference-based attack for MLMs

- What are stronger attacks?

1. **Shadow model-based attacks**: these would be extremely hard (even impossible) for large language models, given the scale of pre-training data and compute.

2. **Reference-based attack**: More efficient and more accurate!
   
   - Would need likelihood for each sample! However, MLMs don’t give probability distributions over next words!!
What are stronger attacks?

1. **Shadow model-based attacks**: these would be extremely hard (even impossible) for large language models, given the scale of pre-training data and compute

2. **Reference-based attack**: More efficient and more accurate!
   - Would need likelihood for each sample! However, MLMs don’t give probability distributions over next words!!
   - We need $p(s; \theta)$, but the model gives us $p(x_i|s_{\setminus x_i})$. 
Likelihood Ratio for MLMs

- We view pre-trained MLMs as energy-based probability distributions on sequences:

\[ p(s; \theta) = \frac{e^{-E(s; \theta)}}{Z_\theta} \]
Likelihood Ratio for MLMs

- We view pre-trained MLMs as energy-based probability distributions on sequences:

\[ p(s; \theta) = \frac{e^{-E(s; \theta)}}{Z_\theta} \]

\( E(s; \theta) \) refers to the scalar energy of a sequence \( s \). \( Z \) is the normalization constant.
Likelihood Ratio for MLMs

- We view pre-trained MLMs as energy-based probability distributions on sequences:

\[ p(s; \theta) = \frac{e^{-E(s;\theta)}}{Z_\theta} \]

\( E(s; \theta) \) refers to the scalar energy of a sequence \( s \). \( Z \) is the normalization constant.

- Given this:

\[
LR(s) = \log \left( \frac{p(s; \theta_R)}{p(s; \theta)} \right) = \log \left( \frac{e^{-E(s;\theta_R)}}{Z_{\theta_R}} \right) - \log \left( \frac{e^{-E(s;\theta)}}{Z_\theta} \right)
\]

\[
= -E(s; \theta_R) - \log(Z_{\theta_R}) + E(s; \theta) + \log(Z_\theta)
\]

\[
= E(s; \theta) - E(s; \theta_R) + \text{constant}
\]
Likelihood Ratio for MLMs

- We view pre-trained MLMs as energy-based probability distributions on sequences:

\[
p(s; \theta) = \frac{e^{-E(s;\theta)}}{Z_\theta}
\]

\(E(s; \theta)\) refers to the scalar energy of a sequence \(s\). \(Z\) is the normalization constant.

- Given this:

\[
LR(s) = \log \left( \frac{p(s; \theta_R)}{p(s; \theta)} \right) = \log \left( \frac{e^{-E(s;\theta_R)}}{Z_{\theta_R}} \right) - \log \left( \frac{e^{-E(s;\theta)}}{Z_\theta} \right)
\]

\[
= -E(s; \theta_R) - \log(Z_{\theta_R}) + E(s; \theta) + \log(Z_\theta)
\]

\[\boxed{= E(s; \theta) - E(s; \theta_R) + \text{constant}}\]

- \(E(s; \theta)\) is calculated as average of MLM conditionals \((p(x_i | s_{\backslash x_i}))\), over multiple masks.
Attack

General Data Distribution ($p$)
Quantifying Privacy Risks of Masked Language Models Using Membership Inference Attacks, F Mireshghallah, K Goyal, A Uniyal, T Berg-Kirkpatrick, R Shokri 2022
Quantifying Privacy Risks of Masked Language Models Using Membership Inference Attacks, F Mireshghallah, K Goyal, A Uniyal, T Berg-Kirkpatrick, R Shokri2022
Quantifying Privacy Risks of Masked Language Models Using Membership Inference Attacks, F Mireshghallah, K Goyal, A Uniyal, T Berg-Kirkpatrick, R Shokri 2022
Quantifying Privacy Risks of Masked Language Models Using Membership Inference Attacks, F Mireshghallah, K Goyal, A Uniyal, T Berg-Kirkpatrick, R Shokri 2022
Example: loss-based attack
Example: loss-based attack
Example: Reference-based attack

Threshold → Sample

Sample → Threshold

Proportion vs. Loss

Proportion vs. Likelihood Ratio

Members
Non-members
Experimental Setup

Datasets
- MIMIC-III Medical notes (member and non-member)
- I2b2 Medical notes (non-member)

Target Models
- 5 ClinicalBERT models with different sizes and training duration, trained on MIMIC-III
- Reference Model: PubmedBERT

Baseline and Metrics
- Loss-based attack
- Metrics: AUC of ROC curve, Precision, Recall on sample and patient levels.
The likelihood ratio-based attack has an AUC of 0.90, vs the 0.66 of the loss-based attack.
Our attack is significantly better than the baseline in the low false positive rate regime.
Would other reference models work too?

<table>
<thead>
<tr>
<th></th>
<th>PubMed BERT</th>
<th>BERT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (loss-based)</td>
<td>66.2</td>
<td>66.2</td>
</tr>
<tr>
<td>Ours (likelihood ratio)</td>
<td>90.0</td>
<td>88.3</td>
</tr>
</tbody>
</table>
What do the most ‘leaked’ samples have in common?

- We manually construct features that we think might have correlation with how much a sample is memorized, and train a simple MLP.

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Number of digits in the sentence</td>
<td>0.0</td>
</tr>
<tr>
<td>(2) Sequence Len</td>
<td>0.0</td>
</tr>
<tr>
<td>(3) Number of non-alphanumeric tokens</td>
<td>69.2</td>
</tr>
<tr>
<td>(4) Frequency of the 3 least frequent tokens in the sequence</td>
<td>63.8</td>
</tr>
</tbody>
</table>
What do the most ‘leaked’ samples have in common?

- We manually construct features that we think might have correlation with how much a sample is memorized, and train a simple MLP

<table>
<thead>
<tr>
<th>Feature Description</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Number of digits in the sentence</td>
<td>0.0</td>
</tr>
<tr>
<td>(2) Sequence Len</td>
<td>0.0</td>
</tr>
<tr>
<td>(3) Number of non-alphanumeric tokens</td>
<td>69.2</td>
</tr>
<tr>
<td>(4) Frequency of the 3 least frequent tokens in the sequence</td>
<td>63.8</td>
</tr>
<tr>
<td>(3)&amp;(4)</td>
<td>71.1</td>
</tr>
<tr>
<td>(2)&amp;(3)&amp;(4)</td>
<td>72.1</td>
</tr>
<tr>
<td>(1)&amp;(2)&amp;(3)&amp;(4)</td>
<td>72.1</td>
</tr>
</tbody>
</table>
So far …

- Unlike prior belief, pre-trained MLMs are also susceptible to membership inference attacks!

- Attacks need some adjustments based on model architecture and optimization

- However, we have only looked at pre-trained models so far!
Talk outline

1. Safety Issues with Large Language Models
2. Measuring Leakage in Pre-training Large Language Models
3. Measuring Leakage in Fine-tuning Methods
4. Open Problems and Future Directions
Memorization in Fine-tuning Large Language Models

- Finetuning (domain adaptation) can be riskier in terms of privacy, as it is more often, on smaller domain specific datasets, such as emails, company messages, etc.
Memorization in Fine-tuning Large Language Models

- Finetuning (domain adaptation) can be riskier in terms of privacy, as it is more often, on smaller domain specific datasets, such as emails, company messages, etc.

- Three main fine-tuning methods:
  1. Fine-tuning the model in full (all parameters)
  2. Fine-tuning the 'head': head is a dense classifier layer added on top of the transformer architecture to perform the given down-stream task.
Memorization in Fine-tuning Large Language Models

- Finetuning (domain adaptation) can be riskier in terms of privacy, as it is more often, on smaller domain specific datasets, such as emails, company messages, etc.

- Three main fine-tuning methods:
  1. Fine-tuning the model in full (all parameters)
  2. Fine-tuning the 'head': head is a dense classifier layer added on top of the transformer architecture to perform the given down-stream task.
  3. Fine-tuning Adapters
Adapters

- The adapter consists of a small bottleneck module, with less than a million parameters, and is spread between the different layers of the model.
Experimental Setup

Datasets
- Penn Tree Bank
- Wikipedia
- Enron email dataset

Task and Model
- Autoregressive (causal) language modeling
- Pre-trained GPT-2

Metrics
- MIA Recall
- Exposure Metric
Memorization Trends

![Graph showing Memorization Trends with validation PPL on the x-axis and MIA Attack Recall on the y-axis. The graph highlights a trend associated with fitting and memorization.]

(1) Fitting + Memorization
Memorization Trends

(1) Fitting + Memorization

(2) Memorization Only
Memorization Trends

(1) Fitting + Memorization

(2) Memorization Only

(3) Overfitting
Memorization Trends

1. Head fine-tuning has the least desirable utility-privacy trade-off, although it doesn’t have the most number of parameters (38 Million, vs 124 Million of full fine-tuning)
Memorization Trends

1. Head fine-tuning has the least desirable utility-privacy trade-off, although it doesn’t have the most number of parameters (38 Million, vs 124 Million of full fine-tuning)

2. Adapter fine-tuning and full-fine tuning are on the Pareto frontier
Memorization Trends

1. Head fine-tuning has the least desirable utility-privacy trade-off, although it doesn’t have the most number of parameters (38 Million, vs 124 Million of full fine-tuning)

2. Adapter fine-tuning and full-fine tuning are on the Pareto frontier

3. Fine-tuning a pre-trained model leaks less information, than fine-tuning from scratch.
Talk outline

1. Safety Issues with Large Language Models
2. Measuring Leakage in Pre-training Large Language Models
3. Measuring Leakage in Fine-tuning Methods
4. Open Problems and Future Directions
Open Problems and Future Directions

1. We saw that MLMs are susceptible to MIA as well. However:

   • We still don’t have an apples to apples comparison between memorization in MLMs and autoregressive LMs, in similar setups.
Open Problems and Future Directions

1. We saw that MLMs are susceptible to MIA as well. However:
   
   • We still don’t have an apples to apples comparison between memorization in MLMs and autoregressive LMs, in similar setups.
   
   • We still haven’t successfully extracted full training sequences from MLMs, as opposed to autoregressive LMs. Sampling from MLMs is not straightforward.
Open Problems and Future Directions

1. We saw that MLMs are susceptible to MIA as well. However:
   
   • We still don’t have an apples to apples comparison between memorization in MLMs and autoregressive LMs, in similar setups.
   
   • We still haven’t successfully extracted full training sequences from MLMs, as opposed to autoregressive LMs. Sampling from MLMs is not straightforward.

2. We observed that prior work successfully extracts full training samples from GPT-2, however, these attacks are not very efficient/targeted. How can we customize them?
Open Problems and Future Directions

3. There are also some ethical/philosophical/linguistic questions too:

   - In mounting our attacks or applying differential privacy (or other notions of privacy), we are extracting/protecting ‘records’, however, the record definition is arbitrary. Should we protect a sentence? A document? What is really the granularity of private data when we are looking at in language? What is our expectation of a LLM that ‘preserves’ privacy?
Open Problems and Future Directions

3. There are also some ethical/philosophical/linguistic questions too:
   • In mounting our attacks or applying differential privacy (or other notions of privacy), we are extracting/protecting ‘records’, however, the record definition is arbitrary. Should we protect a sentence? A document? What is really the granularity of private data when we are looking at in language? What is our expectation of a LLM that ‘preserves’ privacy?

4. There are many other architectures and optimization processes that have not yet been studied in terms of privacy leakage:
   • Seq2Seq (encoder decoder) models like BART and T5.
   • More recent uses of language models such as prompting are also under explored.
Thank you!

fatemeh@ucsd.edu