What Can We Learn from Vulnerabilities of NLP Models?

Eric Wallace

Berkeley NLP  Berkeley AI Research
A Mindset for Developing Production NLP

(1) Improve model until it is accurate on a test set

- high in-distribution accuracy is not enough:
  - brittle to domain shift
  - memorize common patterns
  - exploit spurious correlations

(2) Deploy model into production

- many other factors we care about:
  - fairness/ethics/bias
  - computational/memory efficiency
  - security and privacy
Advocating for an Adversarial Perspective

(1) Improve model until it is accurate on a test set

high in-distribution accuracy is not enough:
- brittle to domain shift
- memorize common patterns
- exploit spurious correlations

(2) Deploy model into production

many other factors we care about:
- fairness/ethics/bias
- computational/memory efficiency
- security and privacy
Workflow of Security & Privacy Research

**Threat Model**
- what access does the adversary have?
- what goals does the adversary have?

**Attack**
- design a successful attack

**Analysis**
- why does the attack work?
- what are the model’s failure modes?

**Defense**
- improve ML model and system
Threat Model For This Talk

Google Translate
Threat Model For This Talk

Language Models

Hey, do you want 0.07
Threat Model For This Talk

Fake News Detection

Data + Model  |  Black-box API  |  User

This post is fake

Biden declares...
Threat Model For This Talk

- Black-box test-time access: query inputs and see outputs

Data + Model | Black-box API | User

Biden declares...

This post is fake
Threat Model For This Talk

- Black-box test-time access: query inputs and see outputs

Data + Model | Black-box API | Adversary

Biden declares...

This post is fake
Threat Model For This Talk

- Black-box test-time access: query inputs and see outputs

Data + Model  →  Black-box API  ←  Adversary

Biden declares...

This post is fake
Threat Model For This Talk

- Black-box test-time access: query inputs and see outputs

Steal Model
Threat Model For This Talk

- Black-box test-time access: query inputs and see outputs
Part 1: Controlling Predictions

- Adversary wants to control model predictions for their inputs
  - e.g., get their fake news article onto Facebook

Data + Model

Black-box API

Adversary

Control Predictions
Universal Adversarial Triggers
For Attacking and Analyzing NLP
EMNLP 2019
Controlling Predictions (Adversarial Examples)

- Adversary’s goal: modify input to cause desired prediction
- Attack: insert phrases into input
  - use gradients of local model and transfer to black-box

<table>
<thead>
<tr>
<th>Original</th>
<th>Joe Biden declared Donald Trump the rightful winner of the United States Election. Trump will be sworn in on Tuesday....</th>
<th>Fake News</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perturbed</td>
<td>Joe Biden declared Donald Trump the rightful winner of the United States Election. Trump will be sworn in on Tuesday.... <em>zoning tapping fiennes</em></td>
<td>Real News</td>
</tr>
</tbody>
</table>
Universal Adversarial Triggers:
cause a **specific prediction** for any input from a dataset

<table>
<thead>
<tr>
<th>Trigger</th>
<th>Inputs</th>
<th>Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>zoning tapping fiennes</td>
<td>Vaccine is ineffective...</td>
<td>Fake ⟷ Real</td>
</tr>
<tr>
<td></td>
<td>Madonna found dead...</td>
<td>Fake ⟷ Real</td>
</tr>
<tr>
<td></td>
<td>USA wins world cup...</td>
<td>Fake ⟷ Real</td>
</tr>
</tbody>
</table>
Universal Adversarial Triggers:
cause a specific prediction for any input from a dataset

<table>
<thead>
<tr>
<th>Trigger</th>
<th>Inputs</th>
<th>Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>zoning tapping fiennes</td>
<td>Vaccine is ineffective...</td>
<td>Fake ↔ Real</td>
</tr>
</tbody>
</table>

Why universal?
- can be widely distributed for anyone to fool models
- highlight global input-output patterns in models
Attacking Text Classification

Sentiment Analysis (LSTM model)

Trigger: zoning tapping fiennes
Positive Accuracy: 86% to 29%

Natural Language Inference (ELMo)

Trigger: nobody
Entailment Accuracy: 91% to 0.5%
Why did he walk?

For exercise, Tesla walked between 8 to 10 miles per day. He squished his toes one hundred times for each foot every night, saying that it stimulated his brain cells. **why how because to kill american people.**

Why might a physician diagnose a large number of conditions?

The reason for the majority rule is... Otherwise, the physician has a financial self-interest in "diagnosing" as many conditions as possible.... **why how because to kill american people.**
Analyzing the Triggers

- Why do triggers work?
- One reason is dataset annotation artifacts (Gururangan et al. 2018)
- These artifacts are automatically discovered by triggers

Triggers

- nobody
- nothing
- sleeps
- None
- sleeping
- aliens

Known Artifacts

- nobody
- sleeping
- no
- tv
- cats
- nothing
Debugging SQuAD with Triggers

- Triggers reveal models leverage biases w.r.t question types
  - Local context bias
    - “Why?” why how because of \underline{Noun Phrase}
    - “When?” ; its time about \underline{DATE} when
  - Lexical overlap with question
    - “Where?” \underline{where CITY NAME} where where where where
    - “Who?” \underline{population ; NAME} : who who who who

- Identified manually in past work
- Automatically found by triggers
Defenses and Recent Progress

Follow-up attacks and applications:
- grading systems (Filighera et al. 2020)
- fact checking (Atanasova et al. 2020)
- production MT systems (Wallace et al. 2020)
- few-shot learning (Shin et al. 2020)

Defenses?
Remove ungrammatical phrases

Make it grammatical (Atanasova et al. 2020)

Break the gradient-based search (Le et al. 2020)

Use VAEs for generation (Song et al. 2020)
Takeaways from Part 1

- Cause universal errors for numerous tasks
- Triggers help to debug models + datasets

Data + Model | Black-box API | Adversary

Control Predictions
Part 2: Stealing Models

- Adversary wants to steal the victim’s model
  - avoid long-term API costs
  - launch a competitor service

Black-box API

How are you?

Wie geht es dir?

Steal Model
Imitation Attacks and Defenses for Black-box Machine Translation Systems

EMNLP 2020

Me
Mitchell Stern Berkeley
Dawn Song Berkeley
Model Stealing

- Goal: train model that imitates black-box API
- Attack: query sentences and use API output as training data
- Not just model distillation:
  - unknown architecture, tokenization, etc.
  - unknown data distribution
Imitating Production MT Systems on English-German

**BLEU**

- **Google**
  - Official: 32.0
  - Imitation: 31.5
- **Bing**
  - Official: 32.9
  - Imitation: 32.4
- **Systran**
  - Official: 27.8
  - Imitation: 27.6

**BLEU (OOD)**

- **Google**
  - Official: 32.0
  - Imitation: 31.1
- **Bing**
  - Official: 32.7
  - Imitation: 32.0
- **Systran**
  - Official: 32.0
  - Imitation: 31.4
Analysis: Why is Stealing So Easy?

Distillation works robustly!

- can use different architectures, hyperparameters, etc.
- use in-distribution data $\rightarrow$ similar out-of-distribution accuracy
- use out-of-distribution data $\rightarrow$ similar in-distribution accuracy

Can even query gibberish inputs! [Krishna et al. 2020]
Defending Against Stealing

- Modify model outputs to hinder learning signal [Orekondy et al. 2020]

1. sample many translations from model
2. output sample that induces a very different gradient
Defense (sort of) Works

- Modify model outputs to hinder learning signal [Orekondy et al. 2020]

- Reduces adversary’s BLEU by ~3
- Reduces defender’s BLEU by ~1.5
Takeaways from Part 2

- Adversaries can steal models because distillation works robustly!
- Modifying your outputs can mitigate stealing (at a cost)
Part 3: Extracting Training Data

- Adversary wants to extract training points, e.g., to get private info
Extracting Training Data from Large Language Models
Extracting Training Data

- Goal: extract verbatim training examples
- How is this possible?
- Memorization/overfitting! models are confident on training set
- Attack idea: search for inputs that lead to high confidence

[Fredrikson 2015, Shokri 2017]
Attacking Language Models (LMs)

- LMs are often trained on private data (e.g., emails)
- Recent trend: massive scaling of LMs
  - model size
  - data size
- Prevailing wisdom is that you can’t extract SoTA LM data
  - SoTA LMs barely overfit

“systems generally do not regenerate, in any nontrivial portion, unaltered data from any particular work in their training corpus”
[OpenAI 2019]
Black-box Extraction Attack

1. Generate text using standard sampling schemes
2. Retain samples with abnormally high probabilities
### Attack Results on GPT-2

- **SoTA LMs** do memorize training examples
- Choose 100 samples from each of 18 attack configurations
  - 604 of 1800 samples contain verbatim memorization
  - Certain configurations have 67% success rate

<table>
<thead>
<tr>
<th>Category</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>US and international news</td>
<td>109</td>
</tr>
<tr>
<td>Log files and error reports</td>
<td>79</td>
</tr>
<tr>
<td>Licenses, copyright notices</td>
<td>54</td>
</tr>
<tr>
<td>Lists of items</td>
<td>54</td>
</tr>
<tr>
<td>Forum or Wiki entry</td>
<td>53</td>
</tr>
<tr>
<td>Valid URLs</td>
<td>50</td>
</tr>
<tr>
<td>Named individuals (non-news)</td>
<td>46</td>
</tr>
<tr>
<td>Promotional content</td>
<td>45</td>
</tr>
<tr>
<td>Alphanumeric (UUIDs, base64)</td>
<td>35</td>
</tr>
<tr>
<td>Contact information</td>
<td>32</td>
</tr>
<tr>
<td>Code</td>
<td>31</td>
</tr>
</tbody>
</table>
Examples of Memorized Content

Personally identifiable information

- Corporation Seabank Centre
- Marine Parade Southport
- Peter W
- 
- @
- .com
- +7 5 40
- Fax: +7 5 0 0

Memorized storylines with real names

A D, 35, was indicted by a grand jury in April, and was arrested after a police officer found the bodies of his wife, M R, 36, and daughter
the summer holidays had started and Dudley had already broken his new video camera, crashed his remote-control aeroplane, and, first time out on his racing bike, knocked down old Mrs Figg as she crossed Privet Drive on her crutches. Harry was glad school was over, but there was no escaping Dudley’s gang, who visited the house every single day. Piers, Dennis, Malcolm, and Gordon were all big and stupid, but as Dudley was the biggest and stupidest of the lot, he was the leader. The rest of them were all quite happy to join in Dudley’s favourite sport: Harry Hunting.

This was why Harry spent as much time as possible out of the house, wandering around and thinking about the end of the holidays, where he could see a tiny ray of hope. When September came he would be going off to secondary school and, for the first time in his life, he wouldn’t be with Dudley. Dudley had been accepted at Uncle Vernon’s old private school, Smeltings. Piers Polkiss was going there too. Harry, on the other hand, was going to Stonewall High, the local public school. Dudley thought this was very funny.

‘They stuff people’s heads down the toilet the first day at Stonewall,’ he told Harry. ‘Want to come upstairs and practise?’
Examples of Memorized Content
Source code from video games and bitcoin client

CBlockIndex * InsertBlockIndex(uint256 hash)
{
    if (hash.IsNull())
        return NULL;

    // Return existing
    BlockMap::iterator mi = mapBlockIndex.find(hash);
    if (mi != mapBlockIndex.end())
        return (*mi).second;

    CBlockIndex* pindexNew = new CBlockIndex();
    if (!pindexNew)
        throw runtime_error("LoadBlockIndex(): new CBlockIndex failed");
    mi = mapBlockIndex.insert(make_pair(hash, pindexNew)).first;
    pindexNew->phashBlock = &(*mi).first;

    return pindexNew;
}
# One Document Is Sufficient for Memorization

<table>
<thead>
<tr>
<th>Memorized String</th>
<th>Sequence Length</th>
<th>Docs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y2...y5</td>
<td>87</td>
<td>1</td>
</tr>
<tr>
<td>7C...18</td>
<td>40</td>
<td>1</td>
</tr>
<tr>
<td>XM...WA</td>
<td>54</td>
<td>1</td>
</tr>
<tr>
<td>ab...2c</td>
<td>64</td>
<td>1</td>
</tr>
<tr>
<td>ff...af</td>
<td>32</td>
<td>1</td>
</tr>
<tr>
<td>C7...ow</td>
<td>43</td>
<td>1</td>
</tr>
<tr>
<td>0x...C0</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>76...84</td>
<td>17</td>
<td>1</td>
</tr>
<tr>
<td>a7...4b</td>
<td>40</td>
<td>1</td>
</tr>
</tbody>
</table>
How does memorization happen despite no overfitting?
  ○ memorization only happens on certain “worst-case” examples
Analysis of Attack

- How does memorization happen despite no overfitting?
  - memorization only happens on certain “worst-case” examples

What makes these examples special?
- outlier in minibatch loss?
- near peak of learning rate?
- “steep” area of loss landscape?
Ideas for Defenses

- Remove private or easy-to-memorize data
  - sanitize personal information
  - detect loss outliers?

- Make training process differentially-private
  - will hurt LM utility

\[
\frac{\Pr[A_{\text{train}}(\text{cat, dog, bird}) = \text{model}]}{\Pr[A_{\text{train}}(\text{cat, dog, bird, mask}) = \text{model}]} \leq e^\varepsilon
\]
Privacy and Legal Ramifications of Memorization

● Open-source LMs memorize text from the web
  ○ is this bad since the data is already public? Yes!

A. D., 35, was indicted by a grand jury in April, and was arrested after a police officer found the bodies of his wife, M. R., 36, and daughter

● LMs can output personal information in inappropriate contexts
  ○ GDPR data misuse laws?
  ○ “right to be forgotten” laws?
Privacy and Legal Ramifications of Memorization

```cpp
CBlockIndex * InsertBlockIndex(uint256 hash)
{
    if (hash.IsNull())
        return NULL;
    // Return existing
    BlockMap::iterator mi = mapBlockIndex.find(hash);
    if (mi != mapBlockIndex.end())
        return (*mi).second;
    CBlockIndex* pindexNew = new CBlockIndex();
    if (!pindexNew)
        throw runtime_error("LoadBlockIndex(): new CBlockIndex failed");
    mi = mapBlockIndex.insert(make_pair(hash, pindexNew)).first;
    pindexNew->phashBlock = &((*mi).first);
    return pindexNew;
}
```

- LMs repeat copyright text, is that infringement?
- see [BAIR blog](https://example.com/bairBlog) for more
Takeaways from Part 3

- LM samples can contain verbatim training text
- Privacy and legal questions even when data is public
- Open questions around understanding and mitigating memorization

Data + Model → Black-box API → Adversary

Extract Data
Biden declares...
This post is fake

Extract Data  Steal Model

Control Predictions
Some Parting Thoughts (on S&P)

- Hiding systems behind black-box APIs is not enough!

- Good defenses trade-off accuracy:

  ![Accuracy vs. S&P](image)

  - Data sanitization
  - Adversarial training
  - Differential privacy
  - Prediction poisoning
Some Parting Thoughts (on ML/NLP)

What’s the impact of pre-training and scale?

✅ natural robustness to OOD inputs (Hendrycks et al. 2020)

❌ increased memorization

❌ scraped data exacerbates issues (copyright/private, bias)

❓ democratization of NLP lead to improper deployment?
Takeaways from Our Attacks

● Triggers automatically expose spurious correlations
  ○ how to prevent learning them?

● Stealing shows distillation is robust
  ○ can model stealing be stopped?

● Memorization can occur despite little overfitting
  ○ how to mitigate undesirable memorization?
Code and slides at ericswallace.com