Week 12: Lecture B

LLMs and Fuzzing

Wednesday, April 3, 2024
How are semester projects going?

Smoothly?

Obstacles?
# The Next Few Weeks

## Part 4: New Frontiers in Fuzzing

<table>
<thead>
<tr>
<th>Monday Meeting</th>
<th>Wednesday Meeting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apr. 01</td>
<td>Apr. 03</td>
</tr>
<tr>
<td><strong>Fuzzing OS Kernels</strong></td>
<td><strong>LLM-guided Fuzzing</strong></td>
</tr>
<tr>
<td>▶️ Readings:</td>
<td>▶️ Readings:</td>
</tr>
<tr>
<td>Apr. 08</td>
<td>Apr. 10</td>
</tr>
<tr>
<td><strong>Fuzzing Compilers</strong> (guest lecture by John Regehr)</td>
<td><strong>Fuzzing Hardware</strong></td>
</tr>
<tr>
<td>▶️ Readings:</td>
<td>▶️ Readings:</td>
</tr>
<tr>
<td>Apr. 15</td>
<td>Apr. 17</td>
</tr>
<tr>
<td><strong>Fuzzing Multi-language Software</strong></td>
<td><strong>Final Presentations I</strong></td>
</tr>
<tr>
<td>▶️ Readings:</td>
<td></td>
</tr>
<tr>
<td>Apr. 22</td>
<td>Apr. 24</td>
</tr>
<tr>
<td><strong>Final Presentations II</strong></td>
<td><strong>No Class (Reading Day)</strong></td>
</tr>
</tbody>
</table>

Stefan Nagy
Recap: Project Schedule

- **Apr. 17th & 22nd:** final presentations
  - 45–20 **5-minute** slide deck and discussion
  - What you did, and why, and what results

- We have 26 teams...
  - So, 13 teams per two days
  - **5 minute presentation each**
  - One-minute audience Q&A
  - Keep the details tight!

- What’s most important:
  - High-level technique
  - Challenges and workarounds
  - Key results (bugs found, other successes, etc.)
Questions?
LLMs: Large Language Models

Slides courtesy of Ana Marasović’s lecture “Ingredients of Generative AI”
Standard Supervised Deep Learning

text + labels + neural network → (not) spam
Pretrain-then-Finetune (2018-2022)

Stage 1:
Pretrain a model

\[ \text{text} \]

Objective: generate next word
\((does \ not \ require \ that \ people \ label \ the \ next \ word)\)

Stage 2:
Finetune the model

\[ \text{text + labels} \]

Objective: standard supervised training
Next word prediction: Language modeling

[BOS] → Sylvester
Next word prediction: Language modeling

[BOS] Sylvester → Stallone
Next word prediction: Language modeling

[BOS] Sylvester Stallone → has
Next word prediction: Language modeling

[BOS] Sylvester Stallone has → made
the number of tokens in the vocabulary

the size of the vector representation for a current token

vector(current token) \times output matrix

= “+” softmax

i-th dimension \sim \text{the “probability” [not really] that the next token is the i-th token in the vocabulary}

select the token with the high(est) “probability” as a token to display (generate)

Read about other sampling strategies here: https://huggingface.co/blog/how-to-generate
Next word prediction: Language modeling

We know which word actually occurred in the text next.

Loss: how far the calculated “probability” of that word is far from the highest possible probability (1.0).

We change values in matrices we are multiplying each token vector with in a way that minimizes the loss.

We do this many, many times during pretraining (this stage can last for months).

Through this process a model implicitly captures features of language without explicitly told to do so.

These features are transferable to other language task.
Stage 1: Pretrain a model

Objective: generate next word
(does not require that people label the next word)

Stage 2: Finetune the model

Objective: standard supervised training
Next word prediction: Language modeling

This can be applied to code too

g = sns.scatterplot (data → )

=
Pre-2022 pretraining:
- Next word prediction
- Self-supervised learning

2022 addition:
- Instruction finetuning
- Supervised learning

2023 addition:
- Human feedback
- Reinforcement learning
How do we represent text?

“a visually stunning rumination on love”

Figure: A Visual Guide to Using BERT for the First Time by Jay Alammar
Byte-Pair Encoding for Tokenization

**Token learner:**
raw train corpus ⇒ vocabulary (a set of tokens)
- 2 characters that are most frequently adjacent (“A”, “B”)
- Adds a new merged symbol (“AB”)
- Continues doing this until $k$ merges are done
- Respects word boundaries

**Token segmenter:**
raw sentences ⇒ tokens in the vocabulary

Original paper: [Sennrich et al., 2016]
Read more: Byte-Pair Encoding for Tokenization

Text Classification on GLUE
Embedding tokens with fixed embeddings

For each token in our vocab we have a high-dimensional representation associated with it.

Each row in the embedding matrix is an embedding/vector of the corresponding token.

row 1 = vector of the 1st token in the vocab
row 2 = vector of the 2nd token in the vocab
row 3 = vector of the 3rd token in the vocab
row 4 = vector of the 4th token in the vocab
How do we represent text?

We retrieve rows 101, 1037, 17453, 14726, 19379, 12758, 2006, 2293, 102 of the embedding matrix.

Those 9 high-dimensional vectors are input to our model.

Figure: A Visual Guide to Using BERT for the First Time by Jay Alammar
What follows is a lot of matrix-vector computations

At the end, each input token vector is transformed into a new vector

We call this new vector “(hidden) representation”

A number in matrices we use to multiply each token vector with is called a parameter

Large in “large language models” means that the total number of parameters is large (a few billion or more)

Figure: A Visual Guide to Using BERT for the First Time by Jay Alammar
Pre "Generative AI"

Model #1 Input:
- 101
- 1037
- 17453
- 14726
- 19379
- 12758
- 2006
- 2293
- 102

Model #1 Output:
- [CLS]
- a
- visually
- stunning
- rum
- #nation
- on
- love
- [SEP]
Pre “Generative AI”

The values in this matrix must be learned for each new task from scratch during the finetuning stage.

vector(input sentence) \times \text{output matrix} = \text{"+" softmax }

\rightarrow i\text{-th dimension } \sim \text{the "probability" [not really] of the } i\text{-th label }

\downarrow predict the highest "probability" label
Stefan Nagy

"+" softmax

i-th dimension \sim \text{the "probability" [not really] that the next token is the i-th token in the vocabulary}

the size of the vector representation for a current token

the number of tokens in the vocabulary

vector(current token) \times \text{output matrix} = \text{the logits vector}

This is already learned!

select the token with the high(est) "probability" as a token to display (generate)

Read about other sampling strategies here: https://huggingface.co/blog/how-to-generate
We want models that can do all sorts of tasks for us \textit{(general-purpose)}

We develop models to generate the label tokens (e.g. “positive”, “negative”)

We add an instruction to induce a task-specific behavior, e.g.:

- “TL;DR” or “summarize: “ for summarization
- “In this task, you are given an article. Your task is to summarize the article in a sentence.”
Trams have operated continuously in Melbourne since 1885 (the horse tram line in Fairfield opened in 1884, but was at best an irregular service). Since then they have become a distinctive part of Melbourne's character and feature in tourism and travel advertising. Melbourne's cable tram system opened in 1885, and expanded to one of the largest in the world, with double track. The first electric tram line opened in 1889, but closed only a few years later in 1896. In 1906 electric tram systems were opened in St Kilda and Essendon, marking the start of continuous operation of Melbourne's electric trams.

Question: If I wanted to take a horse tram in 1884, could I look up the next tram on a schedule?

Answer: The model generates the answer: “No”
A layperson or a domain expert that interacts with an NLP model:

- Have no access to model parameters
- Have no knowledge of how to change the model parameters
- But they are able to provide a few examples of their task
During the 1930s, Jehovah's Witnesses in Germany were sent to concentration camps by the thousands, due to their refusal to salute the Nazi flag, which the government considered to be a crime. Jehovah's Witnesses believe that the obligation imposed by the law of God is superior to that of laws enacted by government. Their religious beliefs include a literal version of Exodus, Chapter 20, verses 4 and 5, which says: "Thou shalt not make unto thee any graven image, or any likeness of anything that is in heaven above, or that is in the earth beneath, or that is in the water under the earth; thou shalt not bow down thyself to them nor serve them." They consider that the flag is an 'image' within this command. For this reason, they refused to salute the flag.

Question: Is it likely that most of these Jehovah's Witnesses survived the war (having the same likelihood of survival as other German civilians) only to later see Soviet flags in their country, or American soldiers proudly saluting the stars and stripes? 
Answer: NO

Francesco Rognoni was another composer who specified the trombone in a set of divisions (variations) on the well-known song "Suzanne ung jour" (London Pro Musica, REP15). Rognoni was a master violin and gamba player whose treatise "Selva di Varie passaggi secondo l'uso moderno" (Milan 1620 and facsimile reprint by Arnaldo Forni Editore 2001) details improvisation of diminutions and Suzanne is given as one example. Although most diminutions are written for organ, string instruments or cornett, Suzanne is "per violone over Trombone alla bastarda". With virtuosic semiquaver passages across the range of the instrument, it reflects Praetorius' comments about the large range of the tenor and bass trombones, and good players of the Quartposaune (bass trombone in F) could play fast runs and leaps like a viola bastarda or cornetto. The term "bastarda" describes a technique that made variations on all the different voices of a part song, rather than just the melody or the bass: "considered legitimate because it was not polyphonic".

Question: Would you likely find the term "bastarda" regularly used in an academic paper on musical theory? 
Answer: DON'T KNOW

Trams have operated continuously in Melbourne since 1885 (the horse tram line in Fairfield opened in 1884, but was at best an irregular service). Since then they have become a distinctive part of Melbourne's character and feature in tourism and travel advertising. Melbourne's cable tram system opened in 1885, and expanded to one of the largest in the world, with of double track. The first electric tram line opened in 1889, but closed only a few years later in 1896. In 1906 electric tram systems were opened in St Kilda and Essendon, marking the start of continuous operation of Melbourne's electric trams.

Question: If I wanted to take a horse tram in 1884, could I look up the next tram on a schedule?
Answer:
Francesco Rognoni was another composer who specified the trombone in a set of divisions (variations) on the well-known song "Suzanne ung jour" (London Pro Musica, REP15). Rognoni was a master violin and gamba player whose treatise "Selva di Varie passaggi secondo l'uso moderno" (Milan 1620 and facsimile reprint by Arnaldo Forni Editore 2001) details improvisation of diminutions and Suzanne is given as one example. Although most diminutions are written for organ, string instruments or cornett, Suzanne is "per violone over Trombone alla bastarda". With virtuosic semiquaver passages across the range of the instrument, it reflects Praetorius' comments about the large range of the tenor and bass trombones, and good players of the Quartposaune (bass trombone in F) could play fast runs and leaps like a viola bastarda or cornetto. The term "bastarda" describes a technique that made variations on all the different voices of a part song, rather than just the melody or the bass: "considered legitimate because it was not polyphonic".

**Question:** Would you likely find the term "bastarda" regularly used in an academic paper on musical theory?

**Answer:** Let's think step by step. From the passage it is unclear whether 'bastarda' was a technique that was impactful and important which are reasons why one could expect to see it regularly in an academic paper on musical theory. **So the answer is DON'T KNOW.**
Instruction Finetuning

Train/finetune a model with the next word prediction objective:

1. To follow instructions
2. With chain-of-thoughts (self-consistency) prompts to elicit reasoning skills
3. With concatenated examples to induce in-context learning

With labeled data of 1800 tasks
Open challenge: Optimize context length & context construction

We want to fit an instruction & a few demonstrations with their explanations & a new instance we want predictions...

...and even more for Retrieval Augmented Generation (to avoid hallucinations)

Longer the context, the more we can squeeze

[Liu et al., 2023]: models prefer info at the beginning/end of the index than in the middle

[Chip Huyen's April'23 Open challenges in LLM research]

Background: Retrieval Augmented Generation (RAG)

Step 1: Indexing
- Data Loader reads text from Knowledge Base and creates chunks
- Embedding Model converts chunks to embeddings

Step 2: Querying
- Vector DB offers fast lookup of chunks semantically similar to any given string, e.g. a question
- Similar Chunks in Knowledge Base (possibly contain answers)
- Answer (sourced in Knowledge Base)

[LlamalIndex Webinar]
Pre-2022 pretraining:
Next word prediction
Self-supervised learning

2022 addition:
Instruction finetuning
Supervised learning

2023 addition:
Human feedback
Reinforcement learning
Learning from Tay’s introduction

Mar 25, 2016  |  Peter Lee - Corporate Vice President, Microsoft Healthcare

As many of you know by now, on Wednesday we launched a chatbot called Tay. We are deeply sorry for the unintended offensive and hurtful tweets from Tay, which do not represent who we are or what we stand for, nor how we designed Tay. Tay is now offline and we’ll look to bring Tay back only when we are confident we can better anticipate malicious intent that conflicts with our principles and values.

I want to share what we learned and how we’re taking these lessons forward.

For context, Tay was not the first artificial intelligence application we released into the online social world. In China, our Xiaolce chatbot is being used by some 40 million people, delighting with its stories and conversations. The great experience with Xiaolce led us to wonder: Would an AI like this be just as captivating in a radically different cultural environment? Tay – a chatbot created for 18- to 24- year-olds in the U.S. for entertainment purposes – is our first attempt to answer this question.

Read more: https://blogs.microsoft.com/blog/2016/03/25/learning-tays-introduction/
Reinforcement Learning from Human Feedback

Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.

Figure from: https://openai.com/blog/instruction-following/
Reinforcement Learning from Human Feedback

Step 1
Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.

Step 2
Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.

Figure from: https://openai.com/blog/instruction-following/
Reinforcement Learning from Human Feedback

**Step 1**
Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.

**Step 2**
Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.

**Step 3**
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.

Figure from: https://openai.com/blog/instruction-following/
Elicits model vulnerabilities that might lead to undesirable behaviors

Goal: craft a prompt that would trigger the model to generate harmful text:
- upsetting user experiences
- enabling harm by aiding violence
- enabling other unlawful activity for a user with malicious intentions

The outputs from red-teaming
⇒ Train the model to be less likely to cause harm

Read more: https://huggingface.co/blog/red-teaming
LLaMA-2

Figure from: https://ai.meta.com/resources/models-and-libraries/llama/
Open challenge: Learning from human feedback

1. Do we really need “hacky” RL?

Figure from: https://openai.com/blog/instruction-following/
Open challenge:
Learning from human feedback

2. Whose preference is “human” preference, taking into account the differences in cultures, religions, political leanings, etc.?

3. How to balance multiple preferences (helpful, honest, and harmless)?

4. Do we want AIs that can take a stand or a vanilla AI that shies away from any potentially controversial topic?
Pre-2022 pretraining:
Next word prediction
Self-supervised learning

2022 addition:
Instruction finetuning
Supervised learning

2023 addition:
*Human feedback*
Reinforcement learning
What data exactly is used at each stage?

Shayne Longpre
@ShayneRedford

First, PT data selection is mired in mysticism.

1. Documentation Debt: #PALM2 & #GPT4 don't document their data
2. PT is expensive → experiments are sparse
3. So public data choices are largely guided by ⚡ intuition, rumors, and partial info ⚡
Since the data is huge, **analyses are slow**:  

- Comparing vector representations of a given instance with each pretraining instance  
- Finding examples that influence model output [Grosse et al., 2023]

**Data filtering should be done with care:**

✔ Quality filters improve performance [Longpre et al., 2023]  
✘ Toxicity filters trade off ability to reduce risk of toxic generation [Longpre et al., 2023]  
✘ Erasing marginalized voices represented in the data [Dodge et al., 2021]

**Analyzing data that contains e.g. pornography** [Birhane et al., 2021] takes toll on people
What data exactly is used at each stage?

Developer’s dilemma:

1. To make useful models, they need data
2. A developer wants to do right by the people who created the data

How to achieve both of these simultaneously?

Nascent research area
Questions?