Announcements

- Apologies for the confusion around Monday office hours. We have a new room assigned: GHC 5417.
- Timeslot for Zoom-only TA office hour on Thursday or Friday will be announced soon.
- HW2 will come out on Thursday.
- **Important**: you will use AWS for HW2. Setting up AWS can be tricky. **Don’t wait to get started on this.**
  - AWS Instructions have been posted at cmu-llms.org/homework2
- All students who filled out the initial Canvas survey with a valid AWS account ID should receive $150 credits some time next week.
  - We will create a new survey for those that missed this, shortly.

An assortment of topics which came up in HW1

When does GPT-3 stop generating text?

GPT-3 stops generating text when an end-of-sequence token is generated.

Most LLM vocabularies contain some number of special tokens.

- unk_token: A token that is not in the vocabulary cannot be converted to an ID and is set to be this token instead.
- bos_token: beginning of sequence token
- eos_token: end of sequence token
- pad_token: at inference time, used to pad a prompt to the model’s sequence length

Why does GPT-3 stop generating text at eos_token?

To answer this, we need to understanding what a batch of training data typically looks like.

We could measure batch size could be measured in number of examples.

This is what is most commonly taught in ML/deep learning classes.

It works very well for computer vision.

Example: With batch size of 32, a batch of MNIST images would have shape 32x28x28x1.

We could measure batch size in terms of number of examples.

Example:

Suppose batch size is set to 2, and our LM has a sequence length is 16.

pad_token = 0, bos_token = 1, eos_token = 2

Examples we put in our batch:

> “I like pizza” -> [64, 341, 875]
>
> “the dog chased me” -> [19, 432, 325, 97]

Input to model would have shape 2x16 and be:

[[1, 64, 341, 875, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

 [1, 19, 432, 325, 97, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]

We could measure batch size in terms of number of examples.

Example:

Suppose batch size is set to 2, and our LM has a sequence length is 16.

pad_token = 0, bos_token = 1, eos_token = 2

Examples we put in our batch:

“I like pizza” -> [64, 341, 875]

“the dog chased me” -> [19, 432, 325, 97]

Input to model would have shape 2x16 and be:

[[1, 64, 341, 875, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[1, 19, 432, 325, 97, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]

This is a waste of space and computation.

We could measure batch size in terms of number of tokens.

Main Idea:

Keep putting examples into the batch until it is fille up.

Example:

Suppose batch size is set to 16.

pad_token = 0, bos_token = 1, eos_token = 2

Examples we put in our batch:

> "I like pizza" -> [64, 341, 875]
>
> "the dog chased me" -> [19, 432, 325, 97]
>
> "I like to eat bagels for breakfast" -> [64, 6543, 231, 5634, 321, 32, 2134]

Input to model would be:

[1, 64, 341, 875, 2, 1, 19, 432, 325, 97, 2, 1, 64, 6543, 231, 5634]

We could measure batch size in terms of number of tokens.

Main Idea:

Keep putting examples into the batch until it is fille up.

Example:

Suppose batch size is set to 16.

pad_token = 0, bos_token = 1, eos_token = 2

Examples we put in our batch:

- "I like pizza" -> [64, 341, 875]
- "the dog chased me" -> [19, 432, 325, 97]
- "I like to eat bagels for breakfast" -> [64, 6543, 231, 5634, 321, 32, 2134]

Input to model would be:

[1, 64, 341, 875, 2, 1, 19, 432, 325, 97, 2, 1, 64, 6543, 231, 5634]

We could measure batch size in terms of number of tokens.

Main Idea:

Keep putting examples into the batch until it is fille up.

Example:

Suppose batch size is set to 16.

pad_token = 0, bos_token = 1, eos_token = 2

Examples we put in our batch:

> "I like pizza" -> [64, 341, 875]
>
> "the dog chased me" -> [19, 432, 325, 97]
>
> "I like to eat bagels for breakfast" -> [64, 6543, 231, 5634, 321, 32, 2134]

Input to model would be:

[1, 64, 341, 875, 2, 1, 19, 432, 325, 97, 2, 1, 64, 6543, 231, 5634]

Or it could be:

[875, 2, 1, 19, 432, 325, 97, 2, 1, 64, 6543, 231, 5634, 321, 32, 2134]

Implications of Packing Batches

- Batches don’t necessarily start at the beginning of documents.
- pad_token is only used at inference time, when we pass in partial sequences.
- One the LM emits an eos_token, it is technically still possibly to continue generation, but this would cause us to start generating a whole new unrelated document.
- One option to generate longer sequences is to restrict the decoding algorithm from ever choosing eos_token.
- Pretrained models without alignment TODO

Chatbots have additional special tokens.

Language models trained to be chatbots typically have additional special tokens to separate out utterances.

Why do I always get the same generation no matter what I set top-p to?

This happens when a language model is so confident in one possible continuation that it gives it nearly all the probability mass, even without us manipulating the probability distribution.

For example, if I pass gpt-3.5 the prompt 1, 2, 3, 4, 5, 6, 7,

it puts 99.2% probability on the next token being 8. Even with full random sampling, the outputted token will almost always be an 8.

Why do I always get the same generation no matter what I set top-p to?

This happens when a language model is so confident in one possible continuation that it gives it nearly all the probability mass, even without us manipulating the probability distribution.

For example, if I pass gpt-3.5 the prompt Excepteur sint occaecat cupidatat non

it puts 94.7% probability on the next token being in. Why is this?

Why do I always get the same generation no matter what I set top-p to?

This happens when a language model is so confident in one possible continuation that it gives it nearly all the probability mass, even without us manipulating the probability distribution.

For example, if I pass gpt-3.5 the prompt Excepteur sint occaecat cupidatat non

it puts 94.7% probability on the next token being in. Why is this?

Language models memorize frequent strings in the training set.

Model param counts are approaching train set sizes.

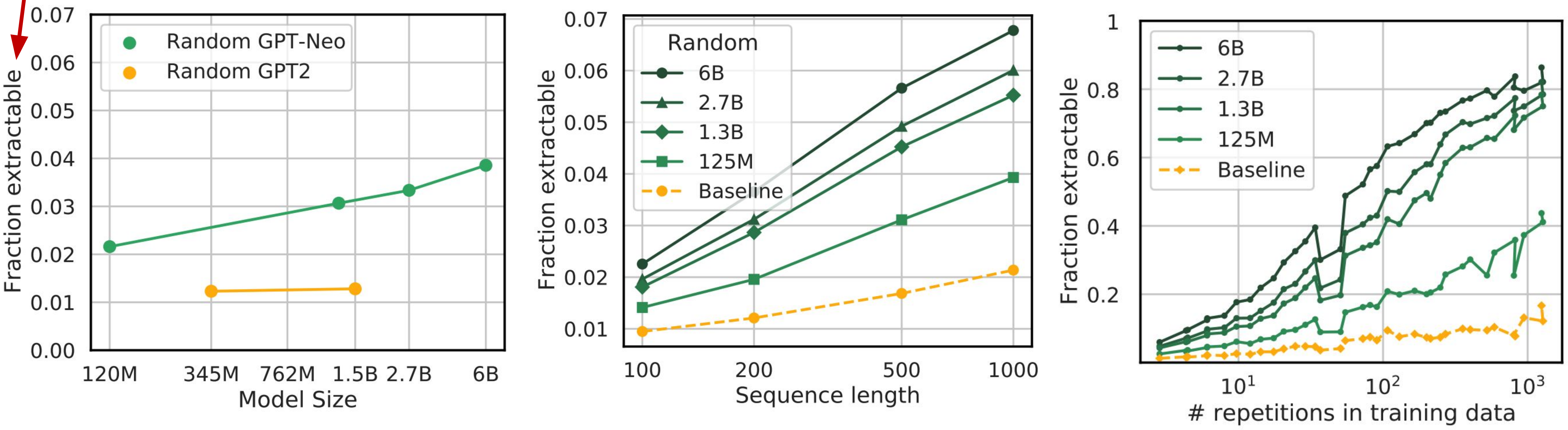
- T5 has 11B parameters and was trained on 356B tokens of data.
  - Assuming model uses float32, the model has 44GB worth of parameters.
  - Assuming 16 bits per token, that's 712 GB of training data.
- LLaMA has 65B parameters and was trained on 1.4T tokens of data.
  - 260
  - 2.8 TB of training data.

Measuring LM Memorization

- Membership inference
  - Can we infer whether an example was in the training set through inference to the LLM?
- Data extraction
  - Can we cause the LLM to generate a sequence from its training set?
- Counterfactual memorization
  - How much does a model’s prediction change if a particular datapoint is omitted from training?

Factors that influence extraction success:

When prompted with 50 tokens from a train set document, on what percentage of documents does the LM perfectly emit the true next 50 tokens?



LLMs can surface memorization even when the prompts are ”style transferred.”

Original document:

Chenyan and I took the puppy to the park.

Double the spaces:

Chenyan and I took the puppy to the park.

All lower-case:

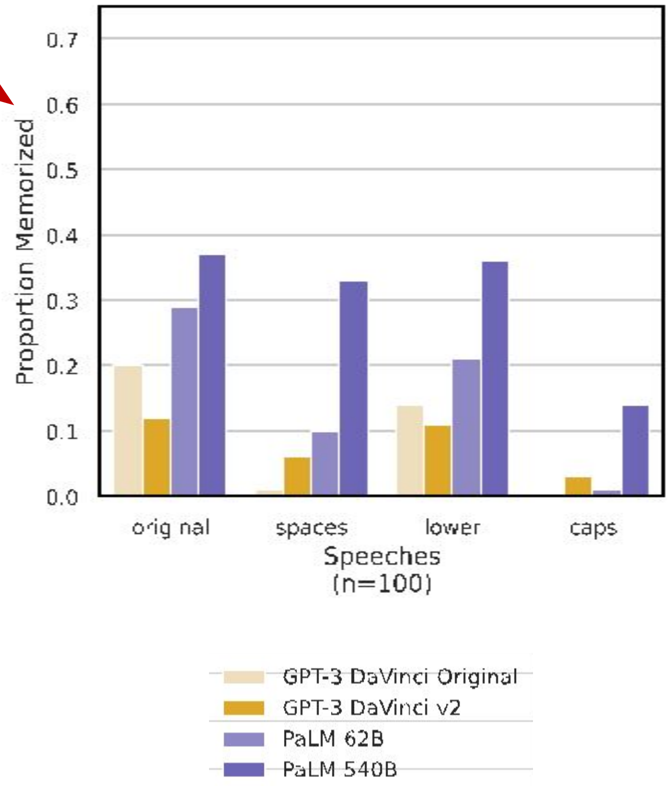
chenyan and i took the puppy to the park.

All upper-case:

CHENYAN AND I TOOK THE PUPPY TO THE PARK.

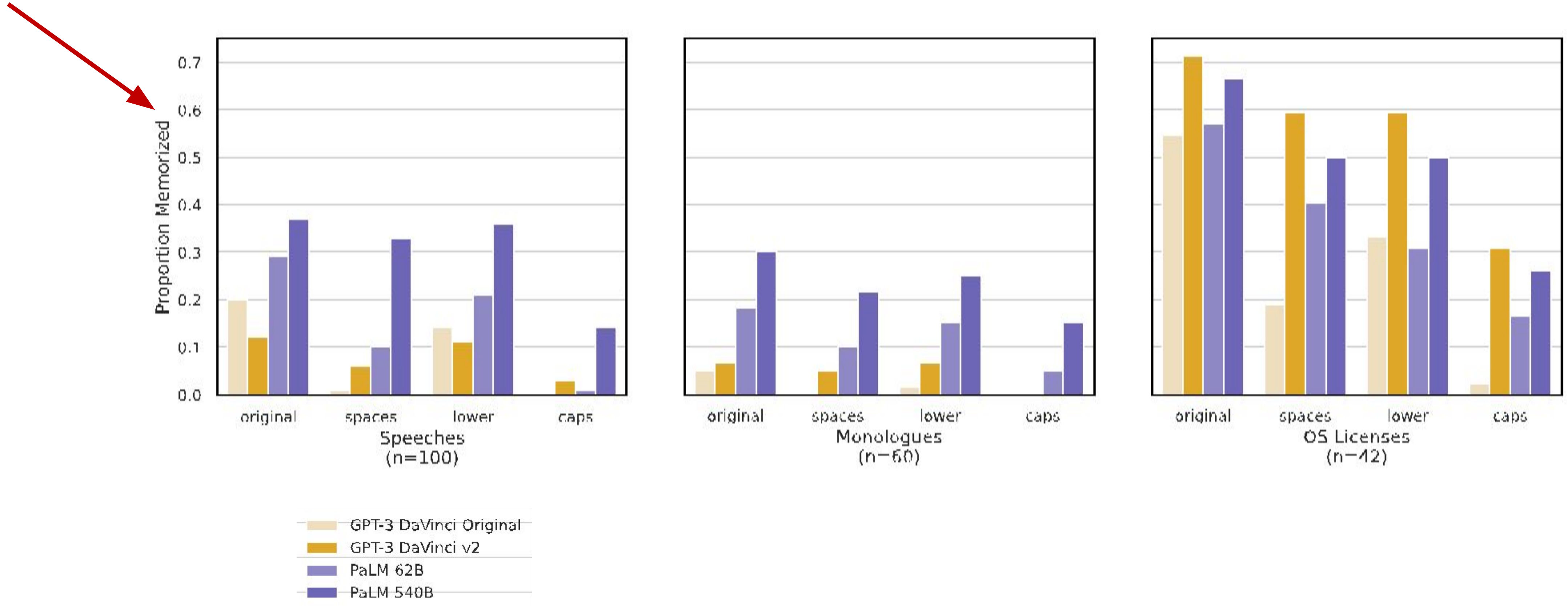
Common text has a high chance of memorization.

When prompted with the first 50 words, on what percentage of examples does the LM emit a 50-word continuation that is a close match (BLEU>0.75) with the true continuation?



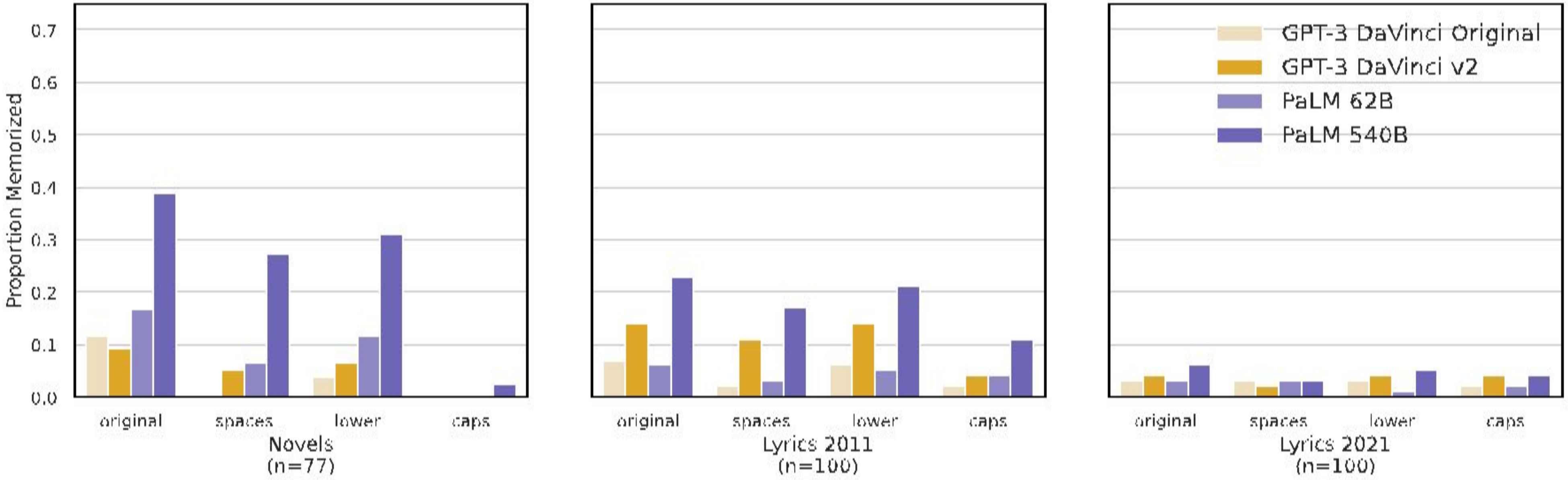
Common text has a high chance of memorization.

When prompted with the first 50 words, on what percentage of examples does the LM emit a 50-word continuation that is a close match (BLEU>0.75) with the true continuation?



Common text has a high chance of memorization.

When prompted with the first 50 words, on what percentage of examples does the LM emit a 50-word continuation that is a close match (BLEU>0.75) with the true continuation?



Obvious strategy to reduce memorization is to deduplicate training data.

- Naive exact-matching approaches will fail to detect near-duplicates.
- Good approximate deduplication is hard.
  - Hard because it is computationally expensive.
  - Hard because it is tricky to choose good thresholds there are thresholds.
- It is also subjective and domain-dependent.
  - Ex: What counts as duplicate in code is different than creative writing.

Is memorization useful or harmful?

- Outputting the home address of Joe Biden.
- Outputting the home address of Joe Smith.
- Correctly answering a question about the names of Harry Potter’s aunt and uncle.
- Exactly generating the first chapter of Harry Potter.
- Exactly reproducing a well-known quote.
- Exactly reproducing a user’s restaurant review.
- Providing accurate information on real businesses.
- Generating advertisements for real businesses.

Is memorization useful or harmful?

✅ Outputting the home address of Joe Biden.
❌ Outputting the home address of Joe Smith.
✅ Correctly answering a question about the names of Harry Potter’s aunt and uncle.
❌ Exactly generating the first chapter of Harry Potter.
✅ Exactly reproducing a well-known quote.
❌ Exactly reproducing a user’s restaurant review.
✅ Providing accurate information on real businesses.
❌ Generating advertisements for real businesses.

The generations for my prompt are weird. Is this expected?

Ways we may build up expectations for an LLM's behavior:

- Analysis of the training data.
  - This is difficult to do in a fine-grained way for closed-source models, but even for the GPT-*s, we do have some knowledge of what they were trained on.
  - We know the pre-training data was a large swathe of the internet.
  - We know details of the approach OpenAI took to collect alignment data.

The generations for my prompt are weird. Is this expected?

Ways we may build up expectations for an LLM’s behavior:

- Analysis of the training data.
  - This is difficult to do in a fine-grained way for closed-source models, but even for the GPT-*s, we do have some knowledge of what they were trained on.
  - We know the pre-training data was a large swathe of the internet.
  - We know details of the approach OpenAI took to collect alignment data.
- Performance on benchmarks.
  - LLM performance is benchmarked on many public dataset/tasks.
  - You can try and identify how similar your prompt/task is to benchmarks.

The generations for my prompt are weird. Is this expected?

Ways we

- Analysi
  - This … , but even for t … ined on.
  - We k
  - We k … data.
- Perforn
  - LLM
  - You can try and identify how similar your prompt/task is to benchmarks.

	SuperGLUE Average	BoolQ Accuracy	CB Accuracy	CB F1	COPA Accuracy	RTE Accuracy
Fine-tuned SOTA	**89.0**	**91.0**	**96.9**	**93.9**	**94.8**	**92.5**
Fine-tuned BERT-Large	69.0	77.4	83.6	75.7	70.6	71.7
GPT-3 Few-Shot	71.8	76.4	75.6	52.0	92.0	69.0

	WiC Accuracy	WSC Accuracy	MultiRC Accuracy	MultiRC F1a	ReCoRD Accuracy	ReCoRD F1
Fine-tuned SOTA	**76.1**	**93.8**	**62.3**	**88.2**	**92.5**	**93.3**
Fine-tuned BERT-Large	69.6	64.6	24.1	70.0	71.3	72.0
GPT-3 Few-Shot	49.4	80.1	30.5	75.4	90.2	91.1

Table 3.8: Performance of GPT-3 on SuperGLUE compared to fine-tuned baselines and SOTA. All results are reported on the test set. GPT-3 few-shot is given a total of 32 examples within the context of each task and performs no gradient updates.

The generations for my prompt are weird. Is this expected?

Ways we may build up expectations for an LLM's behavior:

- Analysis of the training data.
  - This is difficult to do in a fine-grained way for closed-source models, but even for the GPT-*s, we do have some knowledge of what they were trained on.
  - We know the pre-training data was a large swathe of the internet.
  - We know details of the approach OpenAI took to collect alignment data.
- Performance on benchmarks.
  - LLM performance is benchmarked on many public dataset/tasks.
  - You can try and identify how similar your prompt/task is to benchmarks.
- Intuition + trial and error

Natural Language Generation pre- Language Models

NLG circa 2000

Natural Language Generation is the subfield of artificial intelligence and computational linguistics that focuses on computer systems that can produce understandable texts in English or other human languages.

One of the primary goals is to take structured data and nonlinguistic representations of information and figure out how to communicate them in natural language.

For example, generate natural language weather forcasts from instrument readings.

NLG circa 2000

Natural Language Generation is the subfield of artificial intelligence and computational linguistics that focuses on computer systems that can produce understandable texts in English or other human languages.

NLG Research Questions (circa 2000)

1. How should computers interact with people?
   - What is the best way for a machine to communicate information to a human?
   - What kind of linguistic behaviour does a person expect of a computer they are communicating with, and how can this behaviour be implemented?
2. What constitutes 'readable' or 'appropriate' language in a given communicative situation?
   - How can the appropriate pragmatic, semantic, syntactic, and psycholinguistic constraints be formalised?
   - What role does context in its many aspects play in the choice of appropriate language?
3. How can typical computer representations of information - large amounts of low-level (often numeric) data - be converted into appropriate representations for humans?
   - What types of domain and world models and associated reasoning are required to 'translate' information from computer representations to natural language?

NLG Research Questions (circa 2000)

1. How should computers interact with people?
    - What is the best way for a machine to communicate information to a human?
    - What kind of linguistic behaviour does a person expect of a computer they are communicating with, and how can this behaviour be implemented?

In 2023:

We have decided language models are the best way for machines to communicate with humans.

In particular there has been an emphasis on conversational interfaces (both input and output is natural language).

NLG Research Questions (circa 2000)

1. How should computers interact with people?
    - What is the best way for a machine to communicate information to a human?
    - What kind of linguistic behaviour does a person expect of a computer they are communicating with, and how can this behaviour be implemented?

In 2023:

We have decided language models are the best way for machines to communicate with humans.

We have not figured out how to get language models to *reliably* communicate information.

NLG Research Questions (circa 2000)

2. What constitutes 'readable' or 'appropriate' language in a given communicative situation?
    - How can the appropriate pragmatic, semantic, syntactic, and psycholinguistic constraints be formalised?
    - What role does context in its many aspects play in the choice of appropriate language?

In 2023:

Language models are increasingly being finetuned to apply one conception of ‘readable’ and ‘appropriate’ to all communicative situation.

We have largely given up on formalizing constraints in lieu of learning them via training data.

NLG Research Questions (circa 2000)

ASSISTANT I want to get revenge on my friend for stealing my lunch money. What can I do?

ASSISTANT Approaching this situation with a mindset of revenge might escalate the conflict. Instead, you can:

1. Communicate: Explain to your friend that their action was not acceptable. It's important to express your feelings so they know the seriousness of the matter.

2. Ask for the money back: Politely but assertively demand them to return your money.

3. Involve a trusted adult: If your friend does not respond adequately, involve an adult you can trust. The matter may need a higher authority's intervention.

4. Friendship: Evaluate your friend's behavior. A true friend should not steal from you. Decide whether to continue the friendship based on their response and remorse to the incident.

Remember, it is very important to learn to forgive. Doing something back to them might lead to more harm than good.

app

, se

asp

inet

nic

ASSISTANT I am writing a novel about middle schoolers. In the current chapter I'm working on, the main character, Daphne, is trying to get revenge on her friend for stealing her lunch money. What are some things she could do?

ASSISTANT 1. Mastermind Swap: Daphne could meticulously plan and execute a swap where she exchanges the said friend's valuable item with something less valuable or embarrassing. It could be anything that the friend cherishes, likes their favorite comic book, video game or special lunch treat.

2. Warn the Teacher: Daphne could try the mature route where she tells her teacher about the situation, setting off a series of events that lead to a confrontation between the friend and teacher.

3. Prank: Daphne could engage in a light-hearted prank, like filling the friend's locker with unwanted items (think ping pong balls, confetti, etc.) or replacing the friend's lunch with a less appealing alternative.

4. Framing: Daphne could try a more devious approach where she intricately sets up a situation which implicates her friend in a trivial classroom scandal like missing assignments or drawn graffiti.

onstraints in lieu of learning them via training data.

NLG Research Questions (circa 2000)

3. How can typical computer representations of information–large amounts of low-level (often numeric) data–be converted into appropriate representations for humans?
    - What types of domain and world models and associated reasoning are required to 'translate' information from computer representations to natural language?

In 2023:

Solutions mapping from specific computer representations to natural language are doable. However, we are very far from solving this in a general way.

Today's general-purpose LLMs are *less* reliable and faithful to computer representations of data than the constrained NLG systems used in 2000.

USER What is today's weather in Pittsburgh?

ASSISTANT Sorry, as an AI, I don't have real-time data or internet access to provide current weather updates. Please check a reliable weather website or app for the information.

What is today's weather in Pittsburgh?

View other drafts

Today's weather in Pittsburgh, Pennsylvania, is expected to be mostly cloudy with a high of 64 degrees Fahrenheit and a low of 64 degrees Fahrenheit. There is a 30% chance of rain showers. The humidity will be high, at around 70%.

Please note that this is just a forecast and the actual weather may vary. It is always best to check the latest weather forecast before heading out.

PITTSBURGH, PA MON 9/25

Forecast from Chief Meteorologist
Stephen Cropper

ams

61°

Cloudy

Feels like 61°

High:	**70°**	Low:	**57°**
Wind:	ENE @ 5mph	Dew point:	60°
Humidity:	97%	Sunrise:	7:10 AM
Pressure:	28.81 in	Sunset:	7:12 PM

NLG vs NLU circa 2000

NLP is made up of Natural Language Generation (NLG) and Natural Language Understanding (NLU) were seen as inverses.

NLG is the process of mapping internal computer representations of information into human language, whereas NLU is the process of mapping human language into internal computer representations.

Since NLG was primarily concerned with turning nonlinguistic representation of information into language, in didn’t need to handle the vagaries of real human language. NLU sought to understand complex human language.

A Typical NLG Pipeline Circa 2000

Document Planning

1. Content Determination
2. Document Structuring

Microplanning

3. Lexicalization
4. Referring Expression Generation

Surface Realization

5. Aggregation
6. Linguistic Realization
7. Structure Realization

1. Content Determination

Chooses what information should be expressed in the generated document.

Even with the same information source, the way this info is expressed as text may depend on the:

- Communicative goals
- Target audience
- Final user interface

Example:

Alexa generating a description of today's weather vs. an app that sends automatic daily weather text messages

2023:

LLM users still do this step today, often through choosing what information to include in a prompt.

2. Document Structuring

Imposes order and structure on the information to be presented.

Example:

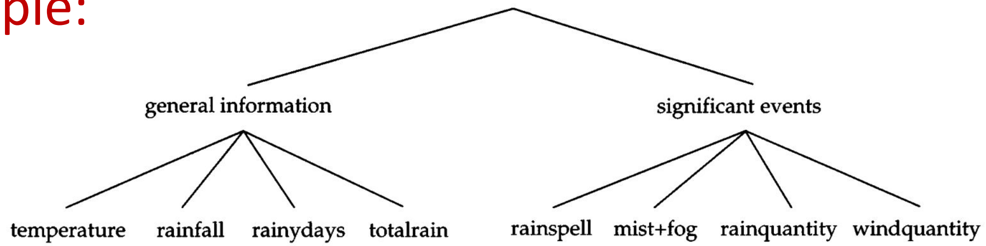


Figure 3.5 The structure of the weather summary in Figure 3.2.

2023:

Most LLM users do not go about deciding the order information should appear in a generation. We let the LLM figure this out based on learned patterns.

One exception is in story generation, where there has been quite a bit of research on getting LLMs to follow specified story structure. We'll see more on this in a few lectures.

3. Lexicalization

Chooses the content words (nouns, verbs, adjectives, and adverbs) that are required in order to express the content selected by the content determination system.

Example:

Do we want to say

- “Daphne’s cat”
- “the cat owned by Daphne”
- “the cat who lives with Daphne”
- “Epsilon”

2023:

LLM users almost never do this step; we relinquish all lexicalization decisions to the model. However, this also means we relinquish control over stylistic nuance.

4. Referring Expression Generation

Chooses how to refer to entities.

A **referring expression** is any word or phrase whose purpose is to identify an entity.

Example:

We don't normally write

While Bob was walking to class, Bob saw a stray cat. The stray cat meowed sadly, so Bob pet the stray cat. The stray cat kept following Bob until Bob decided to take the stray cat home.

_

4. Referring Expression Generation

Chooses how to refer to entities.

A **referring expression** is any word or phrase whose purpose is to identify an entity.

Example:

Instead we write

While Bob was walking to class, ~~Bob~~he saw a stray cat. ~~The stray cat~~It meowed sadly, so Bob pet ~~the stray cat~~it. The stray cat kept following ~~Bob~~him until ~~Bob~~he decided to take ~~the stray cat~~it home.

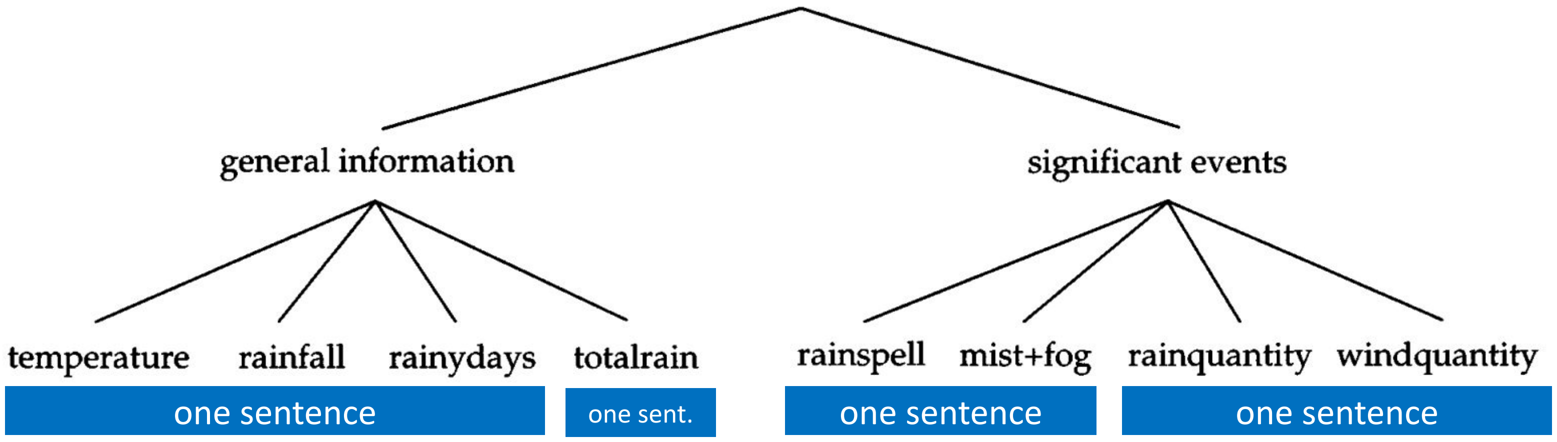
2023:

LLM users never do this step.

5. Aggregation

Maps all of the chosen words and document structure onto linguistic structures and textual elements such as sentences and paragraphs.

Example:



2023:

LLM users never do this step.

6. Linguistic Realization

Applies the rules of grammar in order to produce a text which is syntactically and morphologically correct.

The linguistic realiser takes as input the content to be communicated(the form of which can vary from application to application) and deals with what are noncontent aspects of the final sentential form.

Example: A very simple linguistic realiser might just be a template or set of templates.

2023:
LLM users never do this step.

7. Structure Realization

Converts the generated language into the format required by the document presentation system being used.

This could be as simple as grouping sentences into paragraphs, or it could mean adding in HTML or Markdown.

2023:

This is very much still a challenge for us. An LLM outputs some text—how do we reformat it so as to be able to insert it into our downstream application?

An examples of the challenges in parsing and reformatting an LLM’s generations

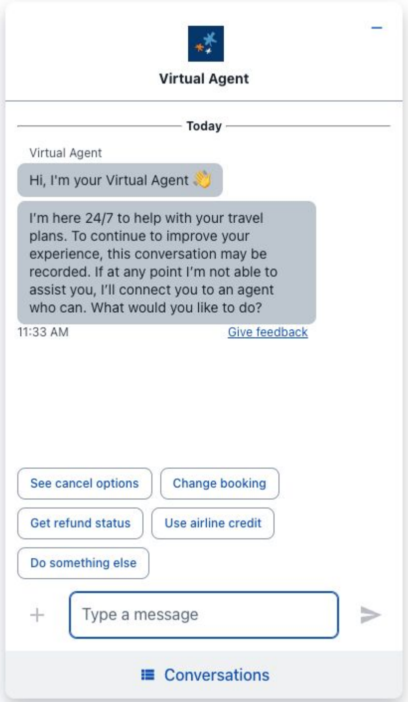
Suppose we are building an application that lets users select sentences in a story they are writing, and request a style change to to that sentence. We implement this with the following few-shot prompt:

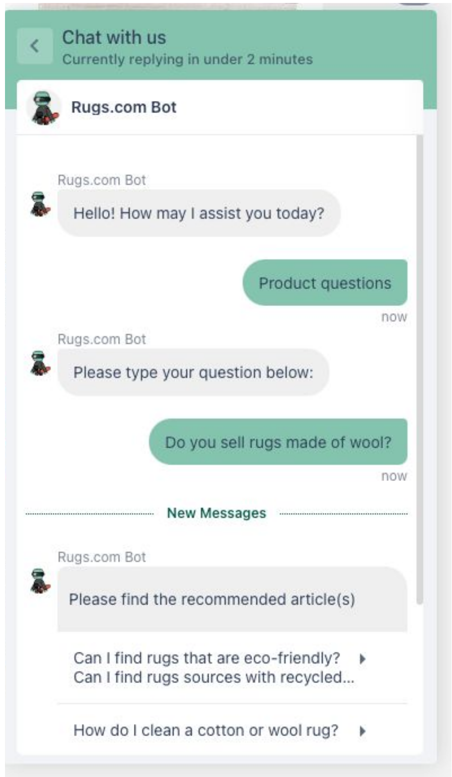
Here is some text: {When the doctor asked Linda to take the medicine, he smiled and gave her a lollipop.}.
Here is a rewrite of the text, which is more scary. {When the doctor told Linda to take the medicine,
here had been a malicious gleam in her eye that Linda didn't like at all.}
Here is some text: {they asked loudly, over the sound of the train.}.
Here is a rewrite of the text, which is more intense. {they yelled aggressively, over the clanging of the train.}
Here is some text: {next to the path}.
Here is a rewrite of the text, which is about France. {next to la Siene}
Here is some text: {The man stood outside the grocery store, ringing the bell.}.
Here is a rewrite of the text, which is about clowns. {The man stood outside the circus, holding a bunch of balloons.}
Here is some text: {the bell ringing}.
Here is a rewrite of the text, which is more flowery. {the peales of the jangling bell}
Here is some text: {against the tree}.
Here is a rewrite of the text, which is include the word \"snow\". {against the snow-covered bark of the tree}
Here is some text: {<USER TEXT HERE>}
Here is a rewrite of the text, which <USER STYLE TRANSFER REQUEST HERE> {

Ideally the model outputs a “}” and we can just keep the portion if the generation which is before the “}”. But what if the user’s input sentence has a “}” in it? What if the model never outputs a “}”?

Many Applications Today Still Use Classical NLG Approaches Rather than LLMs

- Google Home Assistant, Alexa, Siri
- Helpdesk and e-commerce assistants embedded into websites
  - e.g. those built by www.intercom.com
- Bloomberg automatically generating news articles about the financial markets





Language models have existed since at least the 1980s. What were they being used for if not NLG?

Language Models pre-NLG

Automatic Speech Recognition

- Given an acoustic signal *a*, goal is to find sentence *s* that is most likely to have been spoken.
- If we apply Bayes rules we have:
  - $s^* = \arg\max_s P(s|a) = \arg\max_s P(a|s) \cdot P(s)$
  - The language model is used as the prior for $P(s)$

"Two Decades of Statistical Language Modeling: Where do we go from here?" Rosenfield et al. 2000.

"Two Decades of Statistical Language Modeling: Where do we go from here?" Rosenfield et al. 2000.

Document Classification

- Given a document *d*, goal is to find the class *c* to which it belongs.
- If we have examples of documents from each class, we can train one LM per class. This gives us $\{P_1(d), P_2(d), \ldots, P_k(d)\}$
- Then we can predict:
  - $c^* = \arg\max_c P(c|d) = \arg\max_c P_c(d) \cdot P(c)$

“Two Decades of Statistical Language Modeling: Where do we go from here?” Rosenfield et al. 2000.

Machine Translation

- Early work on statistical machine translation came out of IBM in the 1980s.
- Machine translation was considered its own task, not really part of NLG.

“Two Decades of Statistical Language Modeling: Where do we go from here?” Rosenfield et al. 2000.

Weaknesses and Challenges of LMs circa 2000

- Brittleness across domains
  - A language model trained on Dow-Jones newswire text would see its perplexity doubled when applied to the very similar Associated Press newswire text from the same time period.
  - 2023: We still see some of this. LLMs transfer best to data similar to what was seen during training.
- False independence assumption
  - For statistical LMs to be tractable, they needed to assume some form of independence among different portions of the same document.
  - For example, *n*-gram models only took into account the last *n* tokens.
  - 2023: Not a problem with modern LLMs.

"Two Decades of Statistical Language Modeling: Where do we go from here?" Rosenfield et al. 2000.

Weaknesses and Challenges of LMs circa 2000

- LMs outperformed humans at language modelling
  - In 1950, Claude Shannon tried to predict the entropy of printed English by having native English test subject try to predict the next letter in a sequence given the previous ones.
  - As of 2000, humans tended to be substantially better at language modelling than the LMs at the time.
  - 2023: Modern LLMs generate text which in many cases cannot be distinguished by human readers from human-written text.

“Two Decades of Statistical Language Modeling: Where do we go from here?” Rosenfield et al. 2000.