EEP 596: LLMs: From Transformers to GPT || Lecture 12

Dr. Karthik Mohan

Univ. of Washington, Seattle

February 13, 2024

Deep Learning References

Deep Learning

Great reference for the theory and fundamentals of deep learning: Book by Goodfellow and Bengio et al Bengio et al

Deep Learning History

Embeddings

SBERT and its usefulness

SBert Details

Instacart Search Relevance

Instacart Auto-Complete

Attention

Illustration of attention mechanism

Generative AI References

Prompt Engineering

Prompt Design and Engineering: Introduction and Advanced Methods

Retrieval Augmented Generation (RAG)

Toolformer
RAG Toolformer explained

Misc GenAI references

Time-Aware Language Models as Temporal Knowledge Bases

Previous Lecture

- Automated Prompt Engineering
- Use of Tools

Topics for rest of February

Different themes in LLMs!

- Working with images - Image APIs
- Evaluation of LLMs
- LLM Attacks
- Privacy and Cost considerations for LLMs in production

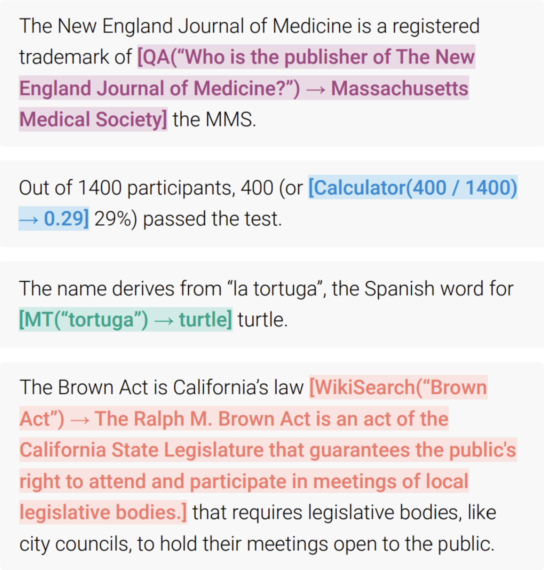
Today's lecture

- Toolformer: Recent paper from Meta AI on use of Tools with LLMs
- Working with Images and Image APIs

Toolformer Motivation

- **Not up-to-date:** Don't have access to up-to-date information on questions.
- **Hallucination:** The idea of LLM concocting its own facts for certain queries or inputs
- **Mathematical Skills:** LLMs are not built to do math, although they can get by on simpler Math questions.
- **Time awareness:** Many facts come with an expiration date (e.g. name the president? which basketball team does LeBron James play for?)

Toolformer Applications



The New England Journal of Medicine is a registered trademark of **[QA("Who is the publisher of The New England Journal of Medicine?") → Massachusetts Medical Society]** the MMS.

Out of 1400 participants, 400 (or **[Calculator(400 / 1400) → 0.29]** 29%) passed the test.

The name derives from "la tortuga", the Spanish word for **[MT("tortuga") → turtle]** turtle.

The Brown Act is California's law **[WikiSearch("Brown Act") → The Ralph M. Brown Act is an act of the California State Legislature that guarantees the public's right to attend and participate in meetings of local legislative bodies.]** that requires legislative bodies, like city councils, to hold their meetings open to the public.

Manual vs Automated use of Tools

1. **Manual use of Tools:** When we combine LLMs with a google search API manually - This is a manual use of the google search tool along with an LLM prompt.
2. **Automated use of Tools:** What Toolformer does is it trains LLMs to use tools automatically, when it feels there is a need.

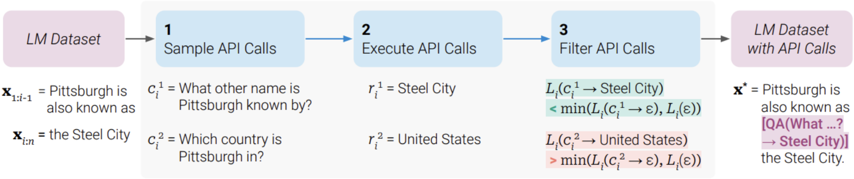
Example

Out of 1400 participants, 400 ....
Here at the token 400, the LLM triggers a call to the Calculator tool to calculate 400/1400 and yield the result 0.29 or 29% as the response.
The automated use of the *Calculator Tool by the LLM*, yields the following response:
Out of 1400 participants, 400 (or 29 %) passed the test.

Approach

- Toolformer is based on a pre-trained GPT-J model with 6.7 billion parameters.
- Toolformer was trained using a self-supervised learning approach that involves sampling and filtering API calls to augment an existing dataset of text.
- Toolformer takes a sentence from pre-train data set and augments it with tokens that indicate a call to an API as follows:
  $< API > QA(query) < /API >$ for question answering or
  $< API > Calculator(400/1400) < /API >$ for using a calculator.

Toolformer Data Augmentation



LM Dataset

1 Sample API Calls

2 Execute API Calls

3 Filter API Calls

LM Dataset with API Calls

$\mathbf{x}_{1:i-1}$ = Pittsburgh is also known as

$\mathbf{x}_{i:n}$ = the Steel City

c_i^1 = What other name is Pittsburgh known by?

c_i^2 = Which country is Pittsburgh in?

r_i^1 = Steel City

r_i^2 = United States

$L_i(c_i^1 \rightarrow$ Steel City$)$ $< \min(L_i(c_i^1 \rightarrow \varepsilon), L_i(\varepsilon))$

$L_i(c_i^2 \rightarrow$ United States$)$ $> \min(L_i(c_i^2 \rightarrow \varepsilon), L_i(\varepsilon))$

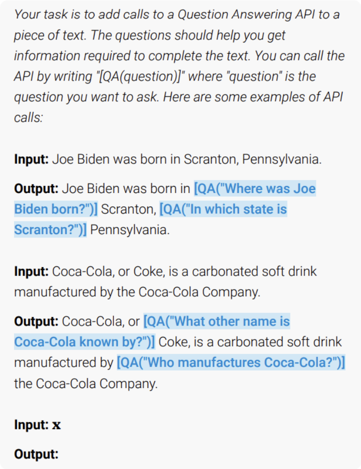
$\mathbf{x}^*$ = Pittsburgh is also known as **[QA(What …? → Steel City)]** the Steel City.

Figure 2: Key steps in our approach, illustrated for a *question answering* tool: Given an input text $\mathbf{x}$, we first sample a position i and corresponding API call candidates $c_i^1, c_i^2, \ldots, c_i^k$. We then execute these API calls and filter out all calls which do not reduce the loss L_i over the next tokens. All remaining API calls are interleaved with the original text, resulting in a new text $\mathbf{x}^*$

Approach

- Use in-context learning to sample API calls using an LLM
- Filter the API sample calls to the ones that give good results
- Add the filtered set to a new data corpus for fine-tuning
- **Example:** "Pittsburgh is also known as the Steel city" is converted into "Pitssburgh is also known as [QA(What other name is Pitssburgh known by?] the steel city."
- In the previous example, an LLM prompt is used to generate candidate API calls that get embedded into the original sentence.
- For each API/tool, generate enough API-augmented sentences to add to the new LM dataset for fine-tuning.
- **LLM generates dataset for a tool-based LLM!** Essentially, use an existing LLM (e.g. GPT) to annotate existing pre-training data corpus with API calls that "make sense" and add it to a "new pre-training dataset". Train an LLM from scratch on the new pre-training data set!

Toolformer Data Augmentation



Your task is to add calls to a Question Answering API to a piece of text. The questions should help you get information required to complete the text. You can call the API by writing "[QA(question)]" where "question" is the question you want to ask. Here are some examples of API calls:

Input: Joe Biden was born in Scranton, Pennsylvania.

Output: Joe Biden was born in **[QA("Where was Joe Biden born?")]** Scranton, **[QA("In which state is Scranton?")]** Pennsylvania.

Input: Coca-Cola, or Coke, is a carbonated soft drink manufactured by the Coca-Cola Company.

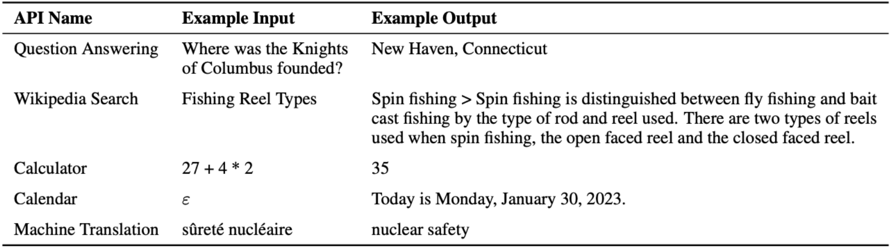
Output: Coca-Cola, or **[QA("What other name is Coca-Cola known by?")]** Coke, is a carbonated soft drink manufactured by **[QA("Who manufactures Coca-Cola?")]** the Coca-Cola Company.

Input: x

Output:

Figure 3: An exemplary prompt $P(\mathbf{x})$ used to generate API calls for the question answering tool.

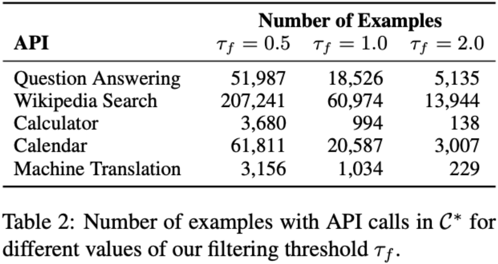
Toolformer Sampling API calls



API Name	Example Input	Example Output
Question Answering	Where was the Knights of Columbus founded?	New Haven, Connecticut
Wikipedia Search	Fishing Reel Types	Spin fishing > Spin fishing is distinguished between fly fishing and bait cast fishing by the type of rod and reel used. There are two types of reels used when spin fishing, the open faced reel and the closed faced reel.
Calculator	27 + 4 * 2	35
Calendar	ε	Today is Monday, January 30, 2023.
Machine Translation	sûreté nucléaire	nuclear safety

Table 1: Examples of inputs and outputs for all APIs used.

Toolformer



API	Number of Examples		
	$\tau_f = 0.5$	$\tau_f = 1.0$	$\tau_f = 2.0$
Question Answering	51,987	18,526	5,135
Wikipedia Search	207,241	60,974	13,944
Calculator	3,680	994	138
Calendar	61,811	20,587	3,007
Machine Translation	3,156	1,034	229

Table 2: Number of examples with API calls in $\mathcal{C}^*$ for different values of our filtering threshold τ_f.

Fine-tuning or "Re-training"

Toolformer training

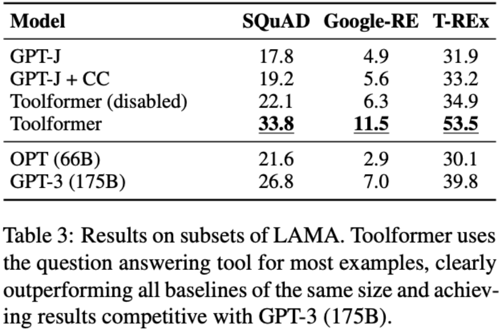
- Obtain C^* from C by data-augmenting examples for each API type with the relevant API calls.
- Train using LM loss on C^*

Inference Time

Inference

At inference time, the decoding process is interrputed by the $\rightarrow$, where the LLM calls the appropriate API (Calculator or Machine Translation, etc) to fetch the appropriate response. After the inserting the retrieved response, the LLM continues generating words until done!

Toolformer Results



Model	SQuAD	Google-RE	T-REx
GPT-J	17.8	4.9	31.9
GPT-J + CC	19.2	5.6	33.2
Toolformer (disabled)	22.1	6.3	34.9
Toolformer	**33.8**	**11.5**	**53.5**
OPT (66B)	21.6	2.9	30.1
GPT-3 (175B)	26.8	7.0	39.8

Table 3: Results on subsets of LAMA. Toolformer uses the question answering tool for most examples, clearly outperforming all baselines of the same size and achieving results competitive with GPT-3 (175B).

Toolformer Results

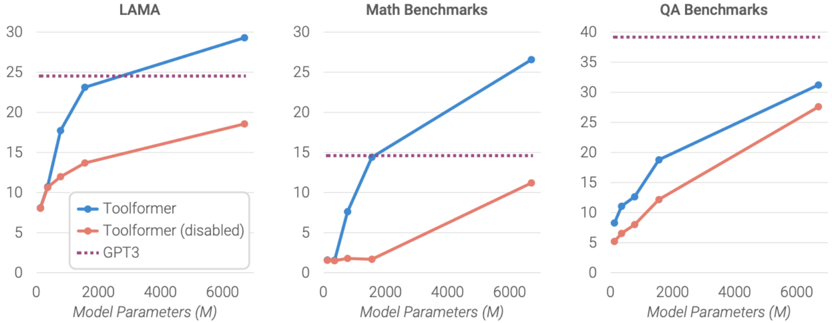


Figure 4: Average performance on LAMA, our math benchmarks and our QA benchmarks for GPT-2 models of different sizes and GPT-J finetuned with our approach, both with and without API calls. While API calls are not helpful to the smallest models, larger models learn how to make good use of them. Even for bigger models, the gap between model predictions with and without API calls remains high.

Limitations

- No explicit chaining of APIs
- Tools are not used in an interactive manner
- Wording of input makes an impact on whether API gets called or not!