Toolformer: Language Models Can Teach Themselves to Use Tools

Authors:

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì†,
Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer,
Nicola Cancedda, Thomas Scialom

Overview of related works

1. **PALM by Chowdhery et al. (2022):** Explores large language models' capabilities, particularly highlighting the challenges with real-time information and factual accuracy that Toolformer addresses.
2. **GPT-3 by Brown et al. (2020):** Establishes baseline capabilities of large-scale language models in zero-shot settings, against which Toolformer’s improvements are measured.
3. **REALM by Guu et al. (2020):** Introduces retrieval-augmented training which helps language models access external information, a conceptual precursor to Toolformer’s API calls.
4. **TALM by Parisi et al. (2022):** Similar to Toolformer, uses a self-supervised approach for integrating tool usage in language models, but focuses more on task-specific training.
5. **LaMDA by Thoppilan et al. (2022):** Examines dialogue models using external tools for enhancing conversational abilities, sharing common goals with Toolformer’s approach to improve information relevance and accuracy.
6. **ATLAS by Izacard et al. (2022):** A retrieval-augmented model for improving question-answering tasks, relevant to Toolformer’s usage of QA systems to enhance model performance.

Methodology

Data Collection and Dataset Usage

- Selection of Pre-existing Datasets: Utilized the pre-existing GPT-J model dataset and the CCNet subset for language modeling and API call integration.
- Augmentation with API Calls: Transformed the original datasets by embedding API calls into the texts to train the Toolformer model to autonomously use these APIs.
- API Diversity: Integrated various API tools including a calculator, a question answering system, a search engine, a translation system, and a calendar to enhance the model's capabilities.

Analysis of Techniques

- Self-supervised Learning Approach: Employed a self-supervised learning strategy where the model learns to use the APIs without human-labeled data, using only a few demonstrations for each API.
- API Call Sampling and Execution: Developed methods for sampling potential API calls from texts and executing them to see if they add value to the text context.
- Loss Calculation for API Calls: Calculated losses to determine the usefulness of each API call, keeping only those that significantly reduce predictive loss in the augmented dataset.

Methodology

Experimental Setup

- Baseline Model Configuration: Used GPT-J model as the baseline to compare Toolformer improvement.
- Controlled Experiment Conditions: Ensured the same environmental and input conditions during experiments to measure the true impact of Toolformer enhancements.
- Zero-shot and Few-shot Evaluation: Assessed Toolformer's performance in zero-shot and few-shot scenarios to evaluate its ability to generalize across different tasks without explicit prior training on those tasks.

Finetuning the Language Model

- Integration of API Calls into Training: After establishing which API calls were useful, these were integrated back into the dataset, creating a new, enriched dataset.
- Model Retraining: Retrained the Toolformer model on this new dataset to adapt its predictions based on the additional context provided by the API calls.
- Preservation of Original Capabilities: Ensured that the introduction of API usage did not detract from the model's original language modeling capabilities.

Methodology

Testing and Validation

- Automated Testing: Set up automated scripts to continuously evaluate the model's performance on a variety of downstream tasks.
- Manual Review: Conducted manual reviews of the model's output to ensure that API calls were contextually appropriate and factually correct.
- Performance Metrics: Utilized standard language modeling metrics such as perplexity, as well as task-specific accuracy measures.

Novel Contributions

1. **Self-Supervised Learning for Tool Use:** Toolformer introduces a self-supervised method that allows language models to learn how to use external tools like calculators and search engines without human annotations.
2. **Dynamic API Integration Framework**: The framework allows the model to autonomously decide when to call an API, which API to use, and how to integrate the results into text generation.
3. **API Call Sampling and Execution:** A mechanism for sampling potential API calls within texts, executing them, and using a loss function to determine their utility, keeping only beneficial interactions.
4. **Augmented Language Model Dataset:** Creation of a novel dataset type where text is enhanced with actionable API calls, training LMs to interact with external data effectively.
5. **Preservation of Core Modeling Capabilities:** Toolformer is engineered to enhance the model's functionality with external tools while maintaining its fundamental language modeling abilities.
6. **Perplexity-based Filtering for API Calls:** Introduces a filtering method based on reducing prediction loss, ensuring the model learns from the most effective API interactions.

Results and Important Findings

1. I**mproved Zero-Shot Performance:** Toolformer demonstrated significantly better zero-shot performance on various NLP tasks compared to models without tool integration, performing close to or surpassing larger models like GPT-3.
2. **Effective Use of External Tools**: The model successfully used external APIs—like calculators, search engines, and translation services—to enhance its responses and decision-making capabilities in task-solving.
3. **Maintained Language Abilities:** Despite the integration of external tools, Toolformer retained its core language modeling performance, showing no loss in perplexity compared to a standard language model without tool usage.
4. **Reduction in Model Size Requirement:** The research showed that Toolformer could achieve competitive results with significantly smaller model sizes, reducing the computational and resource overhead traditionally required for high-performing language models.
5. **Dynamic API Call Management:** Toolformer effectively managed API calls by determining the most beneficial moments to invoke external tools, which optimized both performance and resource use.

Results from the paper

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).

Model	ASDiv	SVAMP	MAWPS
GPT-J	7.5	5.2	9.9
GPT-J + CC	9.6	5.0	9.3
Toolformer (disabled)	14.8	6.3	15.0
Toolformer	**40.4**	**29.4**	**44.0**
OPT (66B)	6.0	4.9	7.9
GPT-3 (175B)	14.0	10.0	19.8

Table 4: Results for various benchmarks requiring mathematical reasoning. Toolformer makes use of the calculator tool for most examples, clearly outperforming even OPT (66B) and GPT-3 (175B).

Model	Es	De	Hi	Vi	Zh	Ar
GPT-J	15.2	**16.5**	1.3	8.2	**18.2**	**8.2**
GPT-J + CC	15.7	14.9	0.5	8.3	13.7	4.6
Toolformer (disabled)	19.8	11.9	1.2	10.1	15.0	3.1
Toolformer	**20.6**	13.5	**1.4**	**10.6**	16.8	3.7
OPT (66B)	0.3	0.1	1.1	0.2	0.7	0.1
GPT-3 (175B)	3.4	1.1	0.1	1.7	17.7	0.1
GPT-J (All En)	24.3	27.0	23.9	23.3	23.1	23.6
GPT-3 (All En)	24.7	27.2	26.1	24.9	23.6	24.0

Table 6: Results on MLQA for Spanish (Es), German (De), Hindi (Hi), Vietnamese (Vi), Chinese (Zh) and Arabic (Ar). While using the machine translation tool to translate questions is helpful across all languages, further pretraining on CCNet deteriorates performance; consequently, Toolformer does not consistently outperform GPT-J. The final two rows correspond to models that are given contexts and questions in English.

Model	WebQS	NQ	TriviaQA
GPT-J	18.5	12.8	43.9
GPT-J + CC	18.4	12.2	45.6
Toolformer (disabled)	18.9	12.6	46.7
Toolformer	**26.3**	**17.7**	**48.8**
OPT (66B)	18.6	11.4	45.7
GPT-3 (175B)	29.0	22.6	65.9

Table 5: Results for various question answering dataset. Using the Wikipedia search tool for most examples, Toolformer clearly outperforms baselines of the same size, but falls short of GPT-3 (175B).

Thank You!