Technology Review: OpenAI Codex

By Michael Haines – mhaines2@illinois.edu

Introduction

Historically, natural language processing has been one of the few remaining tasks at which humans perform better than machines. Languages have evolved to serve humans, and our unique human qualities have shaped characteristics of these languages. Machines are good at discerning true and false; but language is full of nuance, which has historically been tricky for machines to process. But as humans invent machines that are capable of performing more efficient computations, we find a larger set of previously unsolvable problems are now solvable with new programming approaches such as machine learning. Natural language processing is one example where significant strides have been made in the machine learning space in the past few years. Notably in 2020, the OpenAI research lab released a beta of GPT-3, a "neural network machine learning model that can take input text as an input and transform it into what it predicts the most useful result will be."[1] This was the most advanced development in the field of natural language processing to date at its time and has since been extended to support several additional use cases. One particularly interesting application of GPT-3 has been its use in translating natural language instructions into functional code. In this technology review, I will discuss OpenAI Codex, a set of neural network models derived from GPT-3 that are used for translating natural language queries to code[2].

Codex Models and Performance Evaluation

OpenAI unveiled Codex to the world in 2021[3] as "a GPT language model finetuned on publicly available code from GitHub"[4]. To train Codex, researchers at OpenAI collected 179GB worth of Python files from the open source, ignoring code that seemed likely to have been generated, or exhibiting other odd characteristics implying that a non-human entity wrote the code. The performance of the model was evaluated based on whether or not the generated code sample passed the defined unit tests. The dataset that was used to evaluate the performance of the models is published for the world to use.[5]

Using Codex

Codex offers two available models called `code-davinci-002` and `code-cushman-001` which serve very similar use cases, with the latter slightly optimized for runtime performance[6]. Per the

[1] https://www.techtarget.com/searchenterpriseai/definition/GPT-3
[2] https://blogs.infosys.com/digital-experience/emerging-technologies/introduction-to-open-ai-codex.html
[3] https://www.infoq.com/news/2021/08/openai-codex/
[4] https://arxiv.org/pdf/2107.03374.pdf - page 1
[5] https://github.com/openai/human-eval
[6] https://beta.openai.com/docs/models/codex-series-private-beta

documentation, the models are most capable when generating Python code, but are proficient in several other languages as well. Per the documentation[7], Codex can be used to:

- Generate code from comments or a docstring
- Auto-complete code in context
- Identify useful libraries or APIs to accomplish a programming task
- Add comments to code
- Refactor code for runtime efficiency

Unfortunately, both of the models are in private beta so the following YouTube video is the best sandbox demonstration[8]. Although Codex is not available for most, this is not the case for Microsoft, who paid $1 billion in 2019 to partner with OpenAI and for rights to use GPT-3[9]. Their new product GitHub Co-pilot, which launched for general availability in June 2022, was built on top of Codex and provides "an AI pair programmer that suggests code in your editor."[10]

Limitations

Per the white paper published by OpenAI,[11] Codex does not perform well in a few areas, including:

- Training efficiency
- Translating particularly long and complex/specific instructions
- Producing code that is consistently free of syntax errors or bugs

In addition, there are some amusing limitations mentioned in the white paper around docstring generation, which are used to document the code. Given that the training data has less strict requirements (i.e. code has to be runnable; docstrings don't have the same requirements), sometimes the generated docstring will contain text like, "'I just found this function online' and 'This test is not correctly written and it's not my solution'".[12]

Separately, there are potential legal implications. A few questions were ambiguous when co-pilot first came out, for instance:

- Is code generated by the AI my intellectual property, or the entity that built the model that generated the code?
- Under what license does the generated code fall?
- If the generated code infringes on existing intellectual property, is the liability on me or the entity that built the model that generated the code?

[7] https://beta.openai.com/docs/guides/code/introduction
[8] https://www.youtube.com/watch?v=Zm9B-DvwOgw&list=PLOXw6I10VTv_FhQbbvYh1FvbiaPf43Ve2&index=3
[9] https://onezero.medium.com/openai-sold-its-soul-for-1-billion-cf35ff9e8cd4
[10] https://github.blog/2022-06-21-github-copilot-is-generally-available-to-all-developers/
[11] https://arxiv.org/pdf/2107.03374.pdf - page 10
[12] https://arxiv.org/pdf/2107.03374.pdf - page 9

Since its release, Microsoft has clarified that the developer owns any code generated by Copilot, suggesting that the developer is liable for any infringement that might follow.[13] This seems ripe for contest in a court of law, and will be interesting to observe in the coming years.

Conclusion

Natural language processing is one of the many new frontiers enabled by our recent developments in computational efficiency and machine learning. OpenAI Codex is one interesting application of this technology, and code generation through natural language interfaces has has the potential to disrupt the entire software development process. However, it will be some time before humans are replaced by machines in developing the machines, as there are many technical and legal hurdles that need to be crossed. Until then, I am excited for the end of the private beta of OpenAI Codex so that I can start using it in my own projects!

[13] https://www.techtarget.com/searchsoftwarequality/news/252526359/Developers-warned-GitHub-Copilot-code-may-be-licensed