

Aryan Chaurasia XAI models for LIT					
			•	•	
Mentors Ryan Mullins	Organization Responsible AI and Human Centred Technology	Technologies python, nlp, Jax, Transformers			
Topics nlp, Explainable	e Artificial Intelligence				
This project is about adding new AI models for LIT demos. The whole idea is to include different kinds of models as examples, so people from various backgrounds can easily see and understand model demos and learn to use LIT for their own custom models. I will be focusing on two models: a multilingual question answering model based on the TyDiQA dataset, and another model that generates images from text (the Dalle Mini model).					

Brief

For GSoC 2020, my project was about adding new models to LIT (Language Interpretability Tool). LIT is a tool created by the Google PAIR team specifically to "interactively analyze NLP models for model understanding". I was specifically interested in this project because of my interest in NLP and also getting into software development.

From my viewpoint, working on adding new models wouldn't have required higher experience in software development. At the same time, it would allow me to learn a lot about the development practices and make open source contributions. It gave me an opportunity to make contributions to production-level applications and also get to work and learn under a very experienced and talented mentor.

Token of appreciation

My personal goal, other than completing the project and making open-sourced contributions, was to be able to work with someone who has tons of experience in the field and also receive mentorship which I believe was very valuable.

Before going into the project in detail, I would like to thank my mentor Ryan Mullins who is an excellent mentor and provides great feedback and also code reviews that were both helpful and constructive. He has a wealth of experience and knowledge to share and was always willing to help me learn and grow. Also, he was very patient and supportive even though I think he definitely had tons of other things to do. And he still created a positive and supportive learning environment. Thank you for being an excellent mentor.

Main Tasks

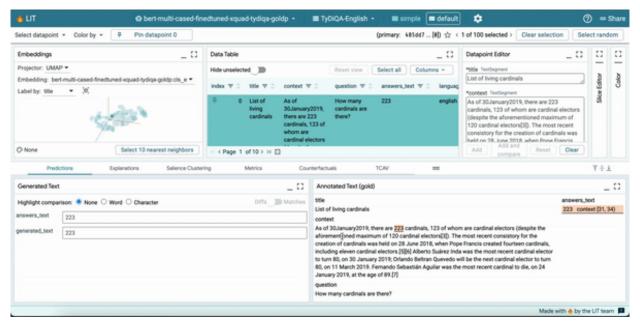
- <u>Multilingual Flax model based TyDi QA dataset</u>: This was the first task for LIT which involved adding a question-answering model based on the TyDi QA dataset.
- <u>Text-to-Image Generation Demo</u>: The second task was about adding text to image generation model, and we decided to go with Dalle-mini, which was just released around the same time and had gained huge popularity.

Pull Requests

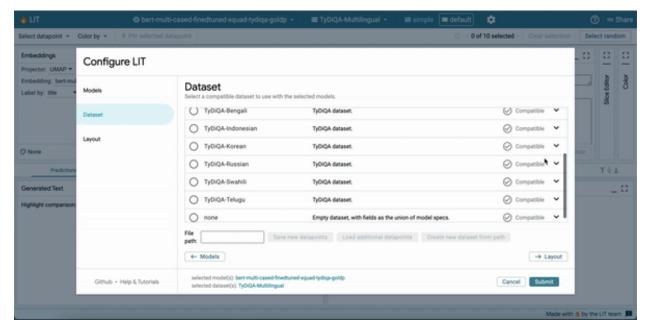
- Multilingual Demo for the TyDi QA task
- Text-to-Image Generation Demo with Dall-E Mini

Project Highlights

- We had very well-organized weekly meetings, with an agenda and notes for the week being noted. Not only did it help to track progress, but also I was able to discuss any other issues which I faced during the week.
- I started with a question-answering model based on the Tydi QA dataset. After a few initial rounds of code reviews and feedback, the demo was looking good.
- A few issues faced while working on the Tydi QA model were mostly on Exact Match metrics which had to be implemented for the model, and proper rendering of the new data type that this model was using (Multi-Segment Annotations) on the front end. Again, all thanks to the mentor with the help of his expertise, this issue was solved.

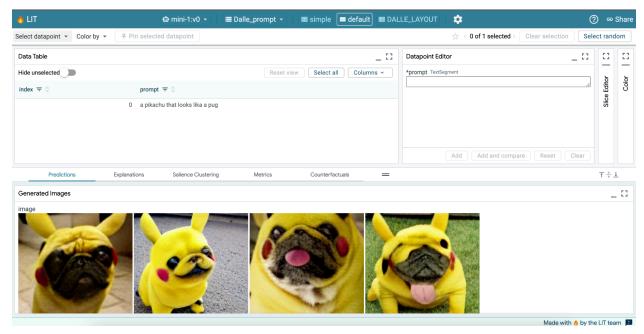


Tydi QA Model on LIT



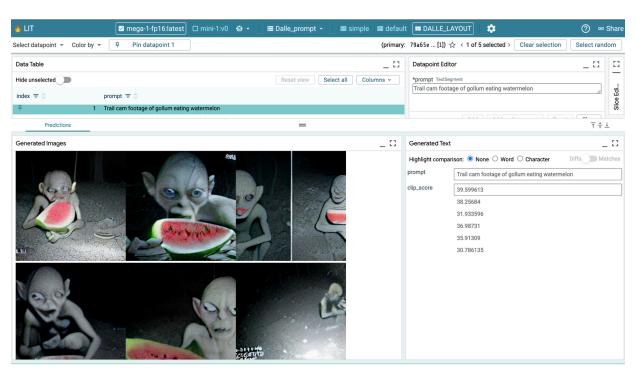
Example of other languages loaded into the LIT Module from the dataset.

- On the other hand, the Dalle-mini model had a much simpler layout.
- Due to not having integrated graphics, it took me a lot of time to work with the Dalle-mini model since it was mainly using the CPU. To overcome this, I first tried implementing the pipeline on Google Colab using GPU runtime, which made the workflow a lot simpler.



Initial Example of Dalle-mini on LIT module

- The model looked very simple, so we thought of adding a saliency map for the text prompt. However, as of now, the Dalle-mini model doesn't output hidden states, due to this, including a saliency map in the LIT model is not possible.
- Soon I came across the <u>CLIP model</u> by Open Ai which takes in an image and a prompt and tells how well they match by generating a score. CLIP was also used along with the Dalle-mini in a few examples. I thought it was important to include and something like this was also mentioned in a meeting that my mentor had arranged with other LIT team members.
- After incorporating CLIP into my Dall-mini model and a few rounds of code reviews, my second task was also completed.



Dalle-mini demo with CLIP score in LIT module

Conclusion

I believe overall it was one of the best decisions I made when I applied to GSoC. Not only did I make some new friends with other contributors, but I also got to know some very talented people who are way more experienced with domain-specific knowledge. Just getting even time to spend with them was so valuable.