Chapter 11: Generative AI in GCP

In the preceding two chapters we have demonstrated AWS and Azure capabilities within the realm of generative AI; we now turn our attention to Google offering in that space. What differentiates the company from its main competitors in the space – Amazon and Microsoft – is that it is currently the only one offering a comprehensive suite of commercial solutions covering every aspect of the AI-focused suite of tools. From dedicated AI chips, through computing power, to development tools and applications – Google has a complete AI stack that serves as a foundation for its offering. Their proprietary Tensor Processing Units (TPU) are designed specifically for machine learning tasks, optimized for use with TensorFlow, and are boasting faster and more efficient performance for AI workloads; while GPU (and CPU) units have not been dethroned across the industry, over the last few years TPU have emerged as formidable rivals.

The Cambrian explosion of generative AI started in the second half of 2022 and the initial impression was that Google had to play catch-up to Microsoft / OpenAI (especially in the NLP space, due to chatGPT parabolic rise). Upon closer examination, it seems like Google strategy has been more of a comprehensive enterprise-grade approach. We begin by discussing the educational offer on GCP: if you, dear reader, have reached this point in the book, chances are you are already familiar with the basics; nevertheless, the series of mini courses offered on the platform can be useful complement to your knowledge.

Having reviewed the educational offer, we will examine the capabilities of Generative AI Studio. The application is built on the Vertex AI platform and allows developers to create their own generative AI apps in text and vision domains. More advanced users can make us of the Model Garden, which gives access to a collection of pretrained models, for advanced exploration and interaction.

If you prefer to hear the summary of Google Generative AI offerings from the company themselves, feel free to check out the introductory videos: <https://www.youtube.com/watch?v=YCZ6nwGnL4o>. Once you have done that, you can come back here and we will embark on the next stage of our journey.

# Learning path

The educational provisions on GCP are designed for a wide range of individuals: from novice enthusiasts eager to expand their understanding to seasoned professionals seeking to refine their expertise. These crash courses allow for an a la carte approach, allowing learners to delve into specific topics – following them sequentially can be helpful in certain cases, but is not required. The first step is to go to the training resources page: <https://cloud.google.com/blog/topics/training-certifications/new-google-cloud-generative-ai-training-resources>

You will be greeted by the screen shown in Figure 11.1 below. Scroll down to “Generative AI Learning Path” and click on the link. This will redirect you to <https://www.cloudskillsboost.google/journeys/118>, where you begin your education.

A picture containing text, screenshot, diagram, design

Description automatically generated

Figure 11.1: Landing page for the Generative AI track from GCP

# A screenshot of a google cloud Description automatically generated with low confidence

Figure 11.2: Generative AI Learning Path page

Each micro-course is composed of three parts: video content, reading list and a quiz to test your newly acquired knowledge. We briefly summarize the modules below:

1. **Introduction to Generative AI**: this course defines Generative AI and explains how it is different from other domains of ML. In addition, it covers major types of models used in the field and describes popular AI applications. Even if you are an experienced ML practitioner, spending half an hour on refreshing the basic definitions is a good idea.  
     
   An important disclaimer needs to be made at this point: *while the structure of the courses is likely to remain stable, the content itself might have changed a bit by the time you are reading these words (e.g. which models are used are integrated, or specific details of the interface); the reason for that is the incredibly fast pace of change in the field: nary a fortnight seems to go by without somebody releasing an ever-more-powerful LLM, beating a previous SOTA. “Change is the only constant” is a rather terrible cliché, but in the context of generative AI it is quite appropriate*.
2. **Introduction to Large Language Models**: this course delves into the concept of LLM, their various applications and utilization of prompt tuning to improve LLM performance. In addition, Google tools useful for personalized Gen AI applications are reviewed. This one is notable for its scope: starting from LLM types (generic, dialog-tuned, instruction-tuned), it discusses also advanced concepts like Chain of Thought.
3. **Introduction to Responsible AI**: This introductory microlearning course is aimed at explaining what responsible AI is, why it's important, and how Google implements responsible AI in their products. It also introduces Google's 7 AI principles.
4. **Introduction to Image Generation:** if population averages are anything to go by, chances are you, dear reader, got into generative AI because you saw an image created with Dall-E or somesuch – which means you have a pretty good idea of what they can do. In this course you will get an introduction into how they do it: diffusion models draw inspiration from thermodynamics and over the last few years they have gained enormous popularity both in research and in the industry. You will get a chance to dive into the theory underpinning diffusion models, as well as learn how to effectively train and deploy them on Vertex AI.
5. **Encoder-Decoder Architecture:** This course provides an overview of the encoder-decoder architecture, a widely used and effective machine learning framework for tasks involving sequences, such as machine translation, text summarization, and question answering. You will gain knowledge about the key elements of the encoder-decoder architecture, including training and deploying these models. During the accompanying lab tutorial, you will code a basic implementation of the encoder-decoder architecture in TensorFlow for generating poetry from scratch.
6. **Attention Mechanism**: in this module you will became acquainted (or re-acquainted – the original paper came out in 2017, after all) with the attention mechanism: a powerful technique enabling DL algorithms to concentrate on segments of an input sequence. You will gain an understanding of how attention operations and how it is applied in enhancing the efficiency of various ML tasks.
7. **Transformer Models and BERT Model:** the course introduces you to \*the\* application of attention mechanism – Transformer architecture, along with its most famous example: Bidirectional Encoder Representations from Transformers (BERT) model. You will learn about the main components (self-attention), and how they comprise the BERT together.
8. **Create Image Captioning Models:** In this course, you will learn to build an image captioning model using deep learning techniques. You will gain knowledge about the various elements comprising an image captioning model, including the encoder and decoder, as well as the process of training and evaluating your model. By the conclusion of the course, you will have the ability to develop your own image captioning models and utilize them for generating descriptive captions for images.
9. **Introduction to Generative AI Studio:** Generative AI Studio is a very interesting offering from GCP, so this final course serves as an excellent segue into our next section.

# Generative AI Studio

Generative AI Studio is a GC console tool for rapidly prototyping and testing generative AI models. You can test sample prompts, design your own prompts, and customize foundation models to handle tasks that meet your application’s needs.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 11.3: Generative AI Studio landing page

The app offers functionality across three modalities: language, speech and image. We will be focusing on the first two, since the availability of the image one varies per region (e.g. at the time of this writing, it is not available in the EU). When you select “Language” from the sidebar menu for “Generative AI Studio”, you will be shown the page given in Figure 11.4 below. This one allows us to focus on one of the three module available: prompt design, conversation interface and model fine-tuning. Each of those was discussed in the tutorial, so we will not be duplicating the content here; instead, we will do something of a hands-on demonstration, elaborating on the bits that the tutorial skimmed over.

A screenshot of a web page

Description automatically generated with medium confidence

Figure 11.4: Language functionality landing page

First, we the prompt design and evaluation. When you click on “Design and test your own prompts”, you will be redirected to the page shown in Figure 11.5:

A screenshot of a computer

Description automatically generated with low confidence

Figure 11.5: Prompt design landing page

Testing your own prompt is quite straightforward: you type where you are prompted to do so (horrible pun intended) – once you do it, you get the kind of result shown in Figure 11.6

A screenshot of a computer

Description automatically generated with low confidence

Figure 11.6: Output of prompt insertion

The link works, in case you were wondering – after all, LLM are prone to hallucinations. What happens if we change the temperature – from the menu shown on the right hand side of Figure 11.5 – from 0.2 (default value) to 0.9? In theory, this should lead the model to select less likely points in the distribution space.

A screenshot of a computer

Description automatically generated with low confidence

Figure 11.7: Output of prompt insertion – higher temperature

As you can see, answer is more “creative”: the description contains longer words and more elaborate sentences. Temperature is by far the most important parameter to play around with (in the sense that a change generates an immediately visible fact to a human observer), but experimenting with “top k” and “token limit” can be quite instructive as well. Keep in mind that LLM are fairly young as a subdomain and established benchmarks have only started emerging in late 2022; as a result, there is a definite “more art than science” vibe to a lot of what we describe here.

Coming back from philosophy to machine learning, the prompt we demonstrated above is an example of zero-shot prompting: we elicited a response from the model without any additional context or tuning. It is the simplest kind of prompting, but most certainly not the only one; we can look at other ones below:

First, another zero-shot that we can further build upon:

A screenshot of a computer

Description automatically generated with medium confidence

Figure 11.8: (another) zero-shot prompt example

As you can see, the model did alright – those are handy examples to test on pre-teen improving his language skills – but they are closely related to each other. We certainly wouldn’t want to limit the spelling development to the sphere of emotion. Can we improve it with a bit of context?

A screenshot of a computer

Description automatically generated with medium confidence

Figure 11.9: One shot prompting

Adding a single example (one shot prompting) seems to have no effect. What if were more elaborate in explaining our intention?

A screenshot of a computer

Description automatically generated with medium confidence

Figure 11.10: Few shot prompting

It is clear that adding multiple examples explaining the context – *few shot prompting* – does improve the quality of the response. Got more creative with the later questions, but it still copied the first three ones from the examples. We can utilize few shot prompting in a less ad hoc and more structured manner: check the tutorial referred above for an excellent (and self-contained) definition of this functionality.

Unless you have been hiding under a rock, you have probably seen countless memes where people demonstrate the role-playing capability of LLM, with chatGPT in the leading role. Google is obviously not likely to promote the product of a competitor, but it has the same functionality Implemented through the conversation functionality in gen ai studio. Figure 11.11 shows you how to get started with this approach:

A screenshot of a web page

Description automatically generated with low confidence

Figure 11.11: Selecting the conversational AI interface

As a simple example, we can define a scenario and tell the AI how to respond to user questions:

A screenshot of a computer

Description automatically generated with medium confidence

Figure 11.12: defining a conversation

After that, you can start asking the questions:

A screenshot of a computer

Description automatically generated with medium confidence

Figure 11.13: example interaction with the conversation interface

After the first response the model can follow-up (so each question does not restart the thread), and we have two primary ways of controlling the output: parameters like temperature (same as in the preceding example) or additional examples, which we can define in the “Examples” tab:

A screenshot of a computer

Description automatically generated with medium confidence

Figure 11.14: continued conversation

A screenshot of a computer

Description automatically generated with medium confidence

Figure 11.15: continued conversation

The above introduction should leave you well equipped to start experimenting with conversational interfaces on your own. We now move to the third language application available in the generative AI studio.

Few shot prompting is practical if our context can be summarized in a short list of examples – but with more specialized applications, like models focusing on a specific technical field, it is not sufficient. This is where model tuning comes in: it allows you to improve a model’s performance on specific tasks or help the model adhere to a specific output requirements. Below we go through tuning a foundation model in Generative AI Studio.

A screenshot of a computer

Description automatically generated with low confidence

Figure 11.16: Model tuning

The first thing we need to do is get the dataset we will use to finetune our model. A handy example are question – answer pairs from Stack Overflow, and the fastest way to obtain it is to go to BigQuery and execute the following query:

SELECT

CONCAT(q.title, q.body) as input\_text,

a.body AS output\_text

FROM

`bigquery-public-data.stackoverflow.posts\_questions` q

JOIN

`bigquery-public-data.stackoverflow.posts\_answers` a

ON

q.accepted\_answer\_id = a.id

WHERE

q.accepted\_answer\_id IS NOT NULL AND

REGEXP\_CONTAINS(q.tags, "python") AND

a.creation\_date >= "2020-01-01"

LIMIT

10000

Saving the results gets you there, with one small caveat: normally, different parts of the GCP work seamlessly together – but this is one of a few exceptions. If you export the results from BigQuery directly, default format is correct (input, output, newline) but the extension is .json and not .jsonl. As a result, Generative AI Studion will not accept it so you need to change it manually. Your final output file should look similar to this:

{"input\_text": "question: How many people live in Beijing? context: With over 21 million residents, Beijing is the world's most populous national capital city and is China's second largest city after

Shanghai. It is located in Northern China, and is governed as a municipality under the direct administration of the State Council with 16 urban, suburban, and rural districts.[14] Beijing is mostly

surrounded by Hebei Province with the exception of neighboring Tianjin to the southeast; together, the three divisions form the Jingjinji megalopolis and the national capital region of China.",

"output\_text": "over 21 million people"}

{"input\_text": "question: How many parishes are there in Louisiana? context: The U.S. state of Louisiana is divided into 64 parishes (French: paroisses) in the same manner that 48 other states of the United

States are divided into counties, and Alaska is divided into boroughs.", "output\_text": "64"}

A screenshot of a computer

Description automatically generated

Figure 11.17: Model tuning

Click “continue”, it will take a moment to upload things to the bucket and after that the process follows the tutorial.

# Model Garden

Model Garden is a general Vertex AI functionality, but it is quite handy to mention it in the context of generative AI. MG is a unified platform for exploration and usage of Google foundational models (with plans to incorporate open source third-party models). The idea is to provide a central hub for a diverse array of model types and sizes, so that business requirements can be fulfilled at this one-stop shop – something along the lines of what HuggingFace is doing. While the latter has a first mover advantage (HF has become \*the\* place to go for new and established models alike, across multiple modalities)

A screenshot of a computer

Description automatically generated

Figure 11.18: Model Garden

# Generative AI App Builder

Spoiler: the generative AI App Builder is not available for general access (as of July 2023)

Gen App Builder is a platform for fast development of chatbots or search engines for data across multiple modalities. The key feature distinguishing it from competitors (after all text-to-\* now includes also text-to-app <INSERT LINK>) is combining generative AI with information retrieval techniques; combining LLM with Knowledge Graphs is new phenomenon, promising “the best of worlds”: factual consistency (so no hallucination) of the KG with the ease of content creation of LLM.

# Running model on GCP

<TODO – demo example how to deploy your own instance of StableDiffusion on GCP>

In this chapter we have reviewed the capabilities offered by Google in the generative AI space: from exploratory setup and prompt design, through no code / low environment, all the way up to hosting your own instance. This broad offer is complemented by the educational content, well suited to the needs of users with varying degrees of ML / gen AI experience. Overall, it is fair to say that while individual components might be superior on the competition side (ChatGPT certainly being more powerful model than Bard), the GCP offer in the generative AI space is the most comprehensive one and best designed.