Google Hackathon Report
Team 10: Prompt to Slides
November 4, 2023
Aniruddh Ingle, Sylvanne Braganza, Sherry Li, Hritik Bhansali, Cece Zhang

Project Overview

Our project aims to transform text-based content into engaging presentations with the help of Google's powerful PaLM2 API. By harnessing the capabilities of this cutting-edge natural language processing tool, we have streamlined the process of creating dynamic presentations from textual information. Whether it is converting lengthy reports into informative slides or transforming written content into compelling spoken presentations, our team has enhanced the efficiency and effectiveness of communication. With the PaLM2 API, we automated the structuring and summarizing of information, allowing users to effortlessly transform their ideas and data into engaging spoken presentations. This project is a game-changer in simplifying the presentation creation process, making it an invaluable tool for professionals, educators, and anyone looking to communicate their ideas effectively.

MakerSuite for Prompt Engineering

In our project, we leveraged Google's MakerSuite for prompt engineering. Initially, we explored different prompt types, including text prompts, data prompts, and chat prompts. After experimenting with these options, we found that the data prompt was the most suitable choice for our project aims due to its ability to tailor the input and output structure according to our project's specific requirements (text to Markdown-formatted script).

Data prompts work by taking custom instructions provided by users. In our case, we asked it to generate a presentation in Markdown script format tailored to a specific age group. Then we give the model specific input and output examples to work with. These prompts give the model examples to replicate. Google calls these "few-shot prompts." They served as training data for the model. Over time, we fine-tuned these examples based on the specific content structure and style we wanted for our presentations. This iterative process allowed the model to learn and adapt to our requirements. We proceeded step by step.

Our initial goal was to use this data prompt to efficiently break blocks of text into bullet points, simplifying the content structuring process. Once we achieved this, we took it a step further by instructing the data prompt to format the extracted bullet points as a Markdown script for a two-slide presentation, consisting of one title slide and one content slide. Finally, we tailored our examples to generate multi-slide presentation Markdown outputs, providing even greater flexibility and automation in the presentation creation process.

Our input texts were initially taken from Wikipedia articles; our outputs were initially written based on a template HTML Markdown code (Appendix A) that we manually filled in. In MakerSuite, we were then able to have the model use our examples (Appendix B) to generate more examples.

We also experimented with the underlying LLM at this stage. LLM responses exhibit both deterministic and random characteristics. When you input a prompt to an LLM, it generates a probability distribution over potential tokens (words) likely to appear next. This initial stage is entirely deterministic; the LLM consistently produces the same distribution when given the same prompt. However, in the subsequent stage, the LLM transforms these distributions into actual text responses using various decoding methods. A straightforward decoding approach might select the most probable token at each step, ensuring determinism. Alternatively, you can opt for a response generation method that involves random sampling from the model's output distribution, introducing a degree of randomness. This stochastic behavior can be controlled by adjusting the temperature setting. A temperature of 0 results in deterministic token selection, while higher temperatures introduce more randomness, yielding unexpected and surprising model responses. (LLM Concepts Guide, 2023)

In our prompt generation stage, we changed this temperature setting to achieve a broad range of example prompts that still formatted the output as our desired Markdown code script. MakerSuite then allowed us to add the newly generated examples that we approved as more examples to train the model.

Once we were happy with the model’s responses to our test inputs, we easily exported it to Python code (Appendix C) in Google Colab, and called the same model using the PaLM API.

Document Search with Embeddings

This technique allows for searching documents by comparing their embedded representations, or vectors of numbers that capture the semantic meaning of text.

The steps are as follows. First, we must generate embedded representations for all of the text in the document(s). This can be done using a pre-trained embedding model, a few of which are provided by Google. Next, we must generate an embedded representation of the query text. This is done using the same embedding model as used in the first step. Now, we can compare the embedded representation of the query text to the embedded representations of the input document(s). This is done using similarity metrics, such as distance determinations. Finally, we must return the results with the highest scores.

We based our code on the example provided by Google here:
https://developers.generativeai.google/examples/doc_search_emb.

This code allowed us to use the PaLM API to create embeddings so that we could perform document search. In our code, we can generate embeddings, build an embeddings database, and then document search with a Q&A system. The embedded representations are compared using the dot product of the vectors, ranging from -1 to 1.

We then incorporated our engineered prompts into our document search with embeddings code.

One challenge encountered in our project was that the embedding system we used had a limitation where it only worked with a maximum of 10,000 bytes of data. Consequently, this meant that we could not include more than approximately 5,000 words in the input document. To address this limitation, we implemented a line in our code which processed text stored in the variable cleaned_text. Initially, it encodes this text into bytes using the UTF-8 encoding, which is a common practice to ensure that text can be represented as a sequence of bytes. Following the encoding step, the code then extracts the first 9900 bytes from the encoded byte sequence using slicing. This ensured that our text remains within the acceptable size range. Finally, the sliced byte sequence is decoded back into a text string using the UTF-8 encoding, with the 'ignore' option specified for error handling. If any decoding issues arise due to text truncation, this option instructs Python to skip those problematic characters and continue decoding the rest of the text, thus preserving as much of the original content as possible.

Another challenge we encountered in our code was related to the extraction of text from PDF documents. For some reason, possibly due to the encoding of the PDF files, occasionally the extracted text had no spacing. As a result, when the LLM processed the text, it interpreted it as a single, uninterrupted string, rendering it virtually unusable for further analysis. To address this issue, we implemented a line of code that ensured the insertion of spaces between words, thus enabling the LLM to interpret and analyze the text correctly, enhancing the overall functionality of our system.

There are several advantages to using document search with embedded text over keyword-based, traditional search methods for our project. The main reason is that it is more robust to synonyms, misspellings, which might crop up in our users’ queries or in the conversion from the input PDF document(s) to string(s). Another advantage is that it can be used to search for documents in a variety of languages, since it does just depend on semantic meanings.

Google API

Our final step was connecting our input and output PDF files to our Google Drive account for ease of use and seamless access.

We were able to do this using Google APIs Explorer. The Google Drive API allows our model to access resources from Google Drive.

How to Use

We have included a ReadMe file in our submission.

Briefly, a user can upload a PDF document into a folder in their Google Drive. They can then adjust the query to specify topic and age or education level. Finally, they run our python script to generate a PDF presentation of slides, which will be saved back into their Google Drive.

Future Directions

We plan to continue to build on this project for the remainder of the mini.

As previously discussed, we are presently utilizing the PaLM API to generate embeddings that enable us to perform searches within PDF documents stored in a Google Drive using our queries. However, there are several ideas we would like to work on in the foreseeable future:

1. Multilingual Testing Expansion

So far, our testing has predominantly focused on prompts in the English language. However, as document search with embedded text is adaptable and capable of searching in various languages, we intend to extend our testing to encompass languages beyond English, including Chinese and Hindi, to build on this project.

2. Diversifying Document Types and Sources

We intend to broaden our scope by exploring additional types and lengths of documents that can be searched using our queries. Currently, our testing is limited to searching through PDF documents. However, we are planning to expand our search capabilities by including Word documents and websites in our testing, aiming to enhance the versatility and adaptability of our query.

3. Challenges and Future Improvements

So far, our project has encountered two main challenges: document length limitations and slide formatting issues. Due to the constraints of the makersuite, we are restricted to using documents with a word count of less than 9,900 characters, which may limit our ability to expand the types of documents we can work with in the future. As for slide formatting issues, we aim to continuously optimize the content of each presentation slide, ensuring that each slide has an appropriate content length and that the organization of pages for each topic is more precise.

Citations

“American Civil War.” Wikipedia, Wikimedia Foundation, 4 November 2023, https://en.wikipedia.org/wiki/American_Civil_War

“Document Search with Embeddings : Palm API : Generative AI for Developers.” *Generative AI for Developers*, 1 Sept. 2023, developers.generativeai.google/examples/doc_search_emb.

“Fact Sheets about Genomics.” Genome.Gov, www.genome.gov/about-genomics/fact-sheets. Accessed 4 Nov. 2023.

“Great Depression.”, Wikimedia Foundation, 31 October 2023,https://en.wikipedia.org/wiki/Great_Depression

“LLM Concepts Guide : Palm API : Generative AI for Developers.” *Generative AI for Developers*, 27 Sept. 2023, developers.generativeai.google/guide/concepts#prompt-types.

“Renaissance.” Wikipedia, Wikimedia Foundation, 29 Oct. 2023, en.wikipedia.org/wiki/Renaissance.

“World War II.” Wikipedia, Wikimedia Foundation, 4 November 2023, https://en.wikipedia.org/wiki/World_War_II.

Appendices

Appendix A

MarkDown Output Template Code:

```
import markdown

# Title and subtitle for the first slide
 title = ""
subtitle = ""

# Slide content for the second slide
slide_title = ""
slide_content = ""

# Slide content for the third slide
slide_title2 = ""
slide_heading2 = ""
slide_content2 = ""

# Slide content for the fourth slide
slide_title3 = ""
slide_heading3 = ""
slide_content3 = "l"

# Slide content for the fifth slide slide_title4 = ""
slide_heading4 = ""
slide_content4 = ""

# Slide content for the sixth slide
slide_title5 = ""
slide_heading5 = ""
slide_content5 = ""

# Create the markdown content
#title: {title}
#subtitle: {subtitle}
markdown_content = f"""---

# {title}
## {subtitle}
---
# {slide_title_1}
## {slide_heading_1}

{slide_content_1}
```

```
"""
---
# {slide_title_2}
## {slide_heading_2}

{slide_content_2}
"""
---
# {slide_title_3}
## {slide_heading_3}

{slide_content_3}
"""


# Convert the markdown content to HTML
html_content = markdown.markdown(markdown_content)

# Print or save the markdown content
print(markdown_content)

with open("output.html", "w") as f:
    f.write(html_content)
```

Appendix B

Prompt Examples:

	Input	Output

1	The Renaissance is a period in history and a cultural movement marking the transition from the Middle Ages to modernity, covering the 15th and 16th centuries and characterized by an effort to revive and surpass the ideas and achievements of classical antiquity; it occurred after the crisis of the Late Middle Ages and was associated with great social change in most fields and disciplines, including art, architecture, politics, literature, exploration, and science. In addition to the standard periodization, proponents of a "long Renaissance" may put its beginning in the 14th century and its end in the 17th century. The traditional view focuses more on the Renaissance's early modern aspects and argues that it was a break from the past, but many historians today focus more on its medieval aspects and argue that it was an extension of the Middle Ages. The beginnings of the period—the early Renaissance of the 15th century and the Italian Proto-Renaissance from around 1250 or 1300—overlap considerably with the Late Middle Ages, conventionally dated to, and the Middle Ages themselves were a long period filled with gradual changes, like the modern age; as a transitional period between both, the Renaissance has close similarities to both, especially the late and early sub-periods of either. The Renaissance's intellectual basis was its version of humanism, derived from the concept of Roman humanitas and the rediscovery of classical Greek philosophy. Early examples were the development of perspective in oil painting and the revived knowledge of how to make concrete. Although the invention of metal movable type sped the dissemination of ideas from the later 15th century, the changes of the Renaissance were not uniform across Europe: the first traces appear in Italy as early as the late 13th century, in particular	import markdown
		
		# Title and subtitle for the first slide
		title = "The Renaissance: A Cultural Revolution"
		subtitle = "Exploring the Origins, Characteristics, and Impact"
		
		# Slide content for the second slide
		 slide_title = "Introduction"
		slide_heading = "Defining the Renaissance"
		slide_content = "Main Ideas.\n\n- Transition from the Middle Ages to modernity\n- 15th and 16th centuries\n- Revival of classical antiquity\n- Variations in defining the Renaissance\n- The concept of a long Renaissance"
		
		# Slide content for the third slide
		slide_title2 = "Renaissance Features"
		slide_heading2 = "Characteristics of the Renaissance"
		slide_content2 = "The main characteristics are:\n\n- Humanism and classical Greek philosophy\n- Perspective in painting and concrete knowledge\n- Resurgence of learning and natural reality in art\n- Diplomacy and observation in politics and science\n- Intellectual and social scientific progress"
		
		# Slide content for the fourth slide
		slide_title3 = "Renaissance in Italy"
		slide_heading3 = "Origins and Centers in Italy"
		slide_content3 = "In Italy:\n\n- Florence as the starting point\n- Political structure and Medici patronage\n- Venice, Genoa, Milan, Rome, and Naples\n- The role of the Medici family\n- Greek scholars' migration after Constantinople's fall"
		
		# Slide content for the fifth slide

	with the writings of Dante and the paintings of Giotto. As a cultural movement, the Renaissance encompassed innovative flowering of Latin and vernacular literatures, beginning with the 14th-century resurgence of learning based on classical sources, which contemporaries credited to Petrarch; the development of linear perspective and other techniques of rendering a more natural reality in painting; and gradual but widespread educational reform. In politics, the Renaissance contributed to the development of the customs and conventions of diplomacy, and in science to an increased reliance on observation and inductive reasoning. Although the Renaissance saw revolutions in many intellectual and social scientific pursuits, as well as the introduction of modern banking and the field of accounting, it is perhaps best known for its artistic developments and the contributions of such polymaths as Leonardo da Vinci and Michelangelo, who inspired the term "Renaissance man". The Renaissance began in Florence, one of the many states of Italy. Various theories have been proposed to account for its origins and characteristics, focusing on a variety of factors, including Florence's social and civic peculiarities at the time: its political structure, the patronage of its dominant family, the Medici, and the migration of Greek scholars and their texts to Italy following the fall of Constantinople to the Ottoman Turks. Other major centers were Venice, Genoa, Milan, Rome during the Renaissance Papacy, and Naples. From Italy, the Renaissance spread throughout Europe and also to American, African, and Asian territories ruled by the European colonial powers of the time or where Christian missionaries were active.	

```
slide_title4 = "Spread of the Renaissance"
slide_heading4 = "Dissemination Across Europe and Beyond"
slide_content4 = "\n\n- How the Renaissance spread in Europe\n- Its influence on American territories\n- Renaissance's impact on African regions\n- Role of Christian missionaries\n- The global reach of the Renaissance"

# Slide content for the sixth slide
slide_title5 = "Legacy of the Renaissance"
slide_heading5 = "Enduring Impact"
slide_content5 = "The main impacts:\n\n- Contributions to art and artists like da Vinci and Michelangelo\n- Intellectual and scientific advancements\n- Influence on modern diplomacy\n- Economic and financial changes\n- The concept of the Renaissance Man and its significance"

# Create the markdown content
#title: {title}
#subtitle: {subtitle}
markdown_content = f"""---


# {title}
## {subtitle}


—

# {slide_title}
## {slide_heading}

{slide_content}


---

# {slide_title2}
## {slide_heading2}

{slide_content2}


 ---

# {slide_title3}
```

		## {slide_heading3}
		
		{slide_content3}
		
		 --- 	
		
		# {slide_title4}
		## {slide_heading4}
		
		{slide_content4}
		
		 --- 	
		
		# {slide_title5}
		## {slide_heading5}
		
		{slide_content5}
		
		"""
		
		# Convert the markdown content to HTML (optional) html_content = markdown.markdown(markdown_content)
		
		# Print or save the markdown content
		print(markdown_content)
		
		with open("output.html", "w") as f:
		f.write(html_content)

2	World War II (WWII or WW2) or the Second World War was a global conflict that lasted from 1939 to 1945. The vast majority of the world's countries, including all the great powers, fought as part of two opposing military alliances: the Allies and the Axis. Many participants threw their economic, industrial, and scientific capabilities behind this total war, blurring the distinction between civilian and military resources. Aircraft played a major role, enabling the strategic bombing of population centres and delivery of the only two nuclear weapons ever used in war. It was by far the deadliest conflict in history, resulting in 70 to 85 million fatalities, mostly civilians. Millions died due to genocides, including the Holocaust, as well as starvation,	import markdown
		
		# Title and subtitle for the first slide
		 title = "World War II: The Global Conflict of 1939-1945"
		subtitle = "Causes, Alliances, and Impact on Humanity"
		
		# Slide content for the second slide
		slide_title = "World War II: The Global Conflict"
		slide_heading = "Key Aspects of World War II"
		slide_content = "Here are the Key Aspects.\n\n- Duration: 1939 to 1945\n- Two Major Alliances: Allies and Axis\n- Economic, Industrial, and Scientific

	massacres, and disease. In the wake of Axis defeat, Germany, Austria and Japan were occupied, and war crimes tribunals were conducted against German and Japanese leaders. The causes of the war are debated; contributing factors included the Spanish Civil War, Second Sino-Japanese War, Soviet–Japanese border conflicts, the rise of fascism in Europe, and tensions in the aftermath of World War I. World War II is generally considered to have begun on 1 September 1939, when Nazi Germany, under Adolf Hitler, invaded Poland. The United Kingdom and France declared war on Germany on 3 September. Under the Molotov–Ribbentrop Pact of August 1939, Germany and the Soviet Union had partitioned Poland and marked out their "spheres of influence" across Finland, Estonia, Latvia, Lithuania and Romania. From late 1939 to early 1941, in a series of campaigns and treaties, Germany conquered or controlled much of continental Europe, in a military alliance with Italy, Japan and other countries called the Axis. Following the onset of campaigns in North Africa and East Africa, and the fall of France in mid-1940, the war continued primarily between the European Axis powers and British Empire, with war in the Balkans, the aerial Battle of Britain, the Blitz of the UK, and the Battle of the Atlantic. In June 1941, Germany led the European Axis powers in an invasion of the Soviet Union, opening the Eastern Front, the largest land theatre of war in history.	Mobilization\n- Significance of Aircraft in Warfare\n- Devastating Impact: Millions of Fatalities and Genocides"
		
		# Slide content for the third slide
		slide_title2 = "Causes and Outbreak of World War II" slide_heading2 = "Factors Leading to World War II" slide_content2 = "Debated Causes: The origins of World War II are subject to debate, with multiple contributing factors.\n\n- Spanish Civil War: The Spanish Civil War was one of the factors that contributed to the outbreak of the war.\n- Second Sino-Japanese War: Conflict in Asia, particularly the Second Sino-Japanese War, played a role in escalating tensions.\n- Rise of Fascism in Europe: The rise of fascist regimes in Europe, such as Nazi Germany, significantly impacted the outbreak of the war.\n- Post-World War I Tensions: Lingering tensions from the aftermath of World War I also contributed to the onset of World War II."
		
		# Create the markdown content
		#title: {title}
		#subtitle: {subtitle}
		markdown_content = f"""---
		
		# {title}
		## {subtitle}
		
		—
		
		# {slide_title}
		## {slide_heading}
		
		{slide_content}
		
		--- 	
		
		# {slide_title2}
		## {slide_heading2}

		{slide_content2}
		
		"""
		
		# Convert the markdown content to HTML
		(optional) html_content =
		markdown.markdown(markdown_content)
		
		# Print or save the markdown content
		print(markdown_content)
		
		with open("output.html", "w") as f:
		f.write(html_content)

3	The human genome project was a 13-year collaborative effort, led by the United States government through the National Institutes of Health (NIH), which produced the first complete sequence of the human genome. Completed in April 2003, the project was a landmark of scientific achievement and a major contribution to human knowledge. It was a joint effort of researchers from the United States, the United Kingdom, Japan, France, Germany, China, and other countries. The Human Genome Project was formally launched on October 1, 1990, and was completed in April 2003. The project's goal was to determine the complete sequence of the 3 billion DNA base pairs that make up human DNA, and to identify all of the genes within the genome.	import markdown
		
		# Title and subtitle for the first slide
		title = "The Human Genome Project: A
		Landmark of Scientific Achievement"
		subtitle = "A 13-Year Collaborative Effort"
		
		 # Slide content for the second slide
		slide_title = "The Human Genome Project"
		slide_heading = "Key Aspects"
		slide_content = "Here are the Key
		Aspects.\n\n- A 13-year collaborative
		effort\n- Led by the United States
		government\n- Developed the first complete
		sequence of the human genome\n- Major
		contribution to human knowledge\n- A joint
		effort of researchers from the United States,
		the United Kingdom, Japan, France,
		Germany, China, and other countries"
		
		# Slide content for the third slide
		slide_title2 = "The Human Genome
		Project's Goals"
		slide_heading2 = "To Determine the
		Complete Sequence of Human DNA"
		slide_content2 = "The Human Genome
		Project's goals were to: Determine the
		complete sequence of human DNA.\n\n-
		Identify all of the genes within the
		genome.\n- Store the information in a
		publicly accessible database.\n- Develop
		tools for analyzing the data."
		
		# Create the markdown content
		#title: {title}
		#subtitle: {subtitle}

		markdown_content = f"""---
		
		# {title}
		## {subtitle}
		
		—
		
		# {slide_title}
		## {slide_heading}
		
		{slide_content}
		
		--- 	
		
		# {slide_title2}
		## {slide_heading2}
		
		{slide_content2}
		
		"""
		
		# Convert the markdown content to HTML (optional) html_content = markdown.markdown(markdown_content)
		
		# Print or save the markdown content
		print(markdown_content)
		
		with open("output.html", "w") as f:
		f.write(html_content)

4	The Great Depression was a severe worldwide economic depression that began in the United States in the 1930s and lasted until the late 1930s. The Great Depression was the longest, deepest, and most widespread decline in global economic activity since the Napoleonic Wars. It began in the United States, with the stock market crash of October 1929 (known as Black Tuesday), and rapidly spread to other countries. The global gross domestic product (GDP) decreased by an estimated 15% from 1929 and 1932. During the Great Depression, some economies experienced negative growth rates in excess of 10% for multiple years. Political instability,	import markdown
		
		# Title and subtitle for the first slide
		title = "The Great Depression"
		subtitle = "A Global Economic Crisis"
		
		# Slide content for the second slide
		slide_title = "The Great Depression: A Global Economic Crisis"
		slide_heading = "Key Facts and Figures"
		slide_content = "Here are some key facts and figures about the Great Depression.\n\n- Duration: 1929-1939\n- Start: October 1929, U.S. stock market crash\n- Spread to other countries: 1930\n-

	widespread unemployment, falling incomes, reduced investment and trade, and shortages of basic necessities caused hardship for millions of people across the world.	Global GDP decreased: 15% from 1929-1932\n- Negative growth rates: 10% or more in some countries\n- Political instability, unemployment, and hardship"
		
		# Slide content for the third slide
		slide_title2 = "The Global Impact of the Great Depression"
		slide_heading2 = "Economic and Social Consequences"
		slide_content2 = "The Great Depression had a devastating impact on the global economy and society.\n\n- Economic consequences: GDP decreased by 15%, negative growth rates of 10% or more in some countries, unemployment, and shortages of basic necessities\n- Social consequences: Political instability, hardship for millions of people across the world"
		
		# Create the markdown content
		#title: {title}
		#subtitle: {subtitle}
		markdown_content = f"""---
		
		# {title}
		## {subtitle}
		
		—
		
		# {slide_title}
		## {slide_heading}
		
		{slide_content}
		
		--- 	
		
		# {slide_title2}
		## {slide_heading2}
		
		{slide_content2}
		
		"""

		# Convert the markdown content to HTML
		(optional) html_content =
		markdown.markdown(markdown_content)
		
		# Print or save the markdown content
		print(markdown_content)
		
		with open("output.html", "w") as f:
		f.write(html_content)

Appendix C

Engineered Prompt Python Code:

Copy of makersuite_text_prompt.ipynb

Team Contributions

- Document Search with Embeddings/DocuPresenter, Query (Code) and Google API/Retrieving files from Data (Code) - Aniruddh
- Converting MD to PDF, Saving to Drive (Code) - Hritik, Aniruddh
- ReadMe - Sherry
- Overview, Prompt Engineering, Document Search, Google API, How to (Report) and MakerSuite (Code) - Sylvanne
- Appendix, Citations, Future Directions (Report) - Cece
- MakerSuite Prompt Engineering - Sherry, Sylvanne, Hritik, Cece