

devfest



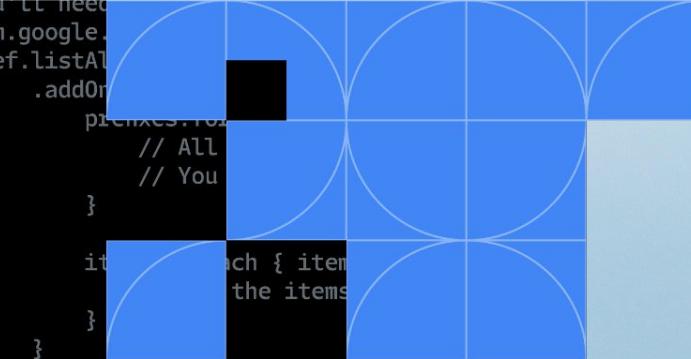
LLMs in Your PaLM

ChengCheng Tan



Google Developer Student Clubs

Earlham



What's a LLM?

1. Natural Language Processing [NLP]
2. Transformers
3. Pre-trained vs Fine-tuned Models

How to Use Google PaLM API?

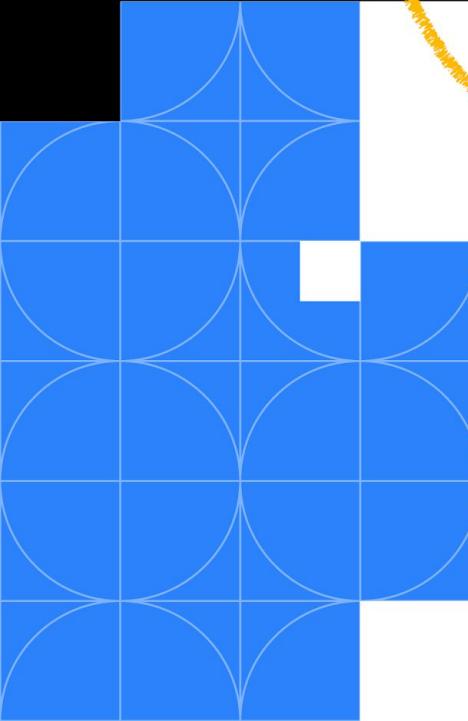
1. MakerSuite
2. Text Endpoint
3. Chat Endpoint
4. Embeddings Endpoint

```
text  
  'Section Title',  
  style: TextStyle(  
    color: Colors.blue[200],  
  ),  
,
```

devfest

```
s.star,  
r: Colors.blue[500],
```

```
Text('23'),
```



01

What's a LLM?

Natural Language
Processing [NLP]:
**Computers can
speak & understand
human languages**



Pre-1990s:

Rule-based Expert Systems



1990s-2000s: **Statistics & Probabilities**



You shall know a
word by the
company it keeps

J.R. Firth, Linguist

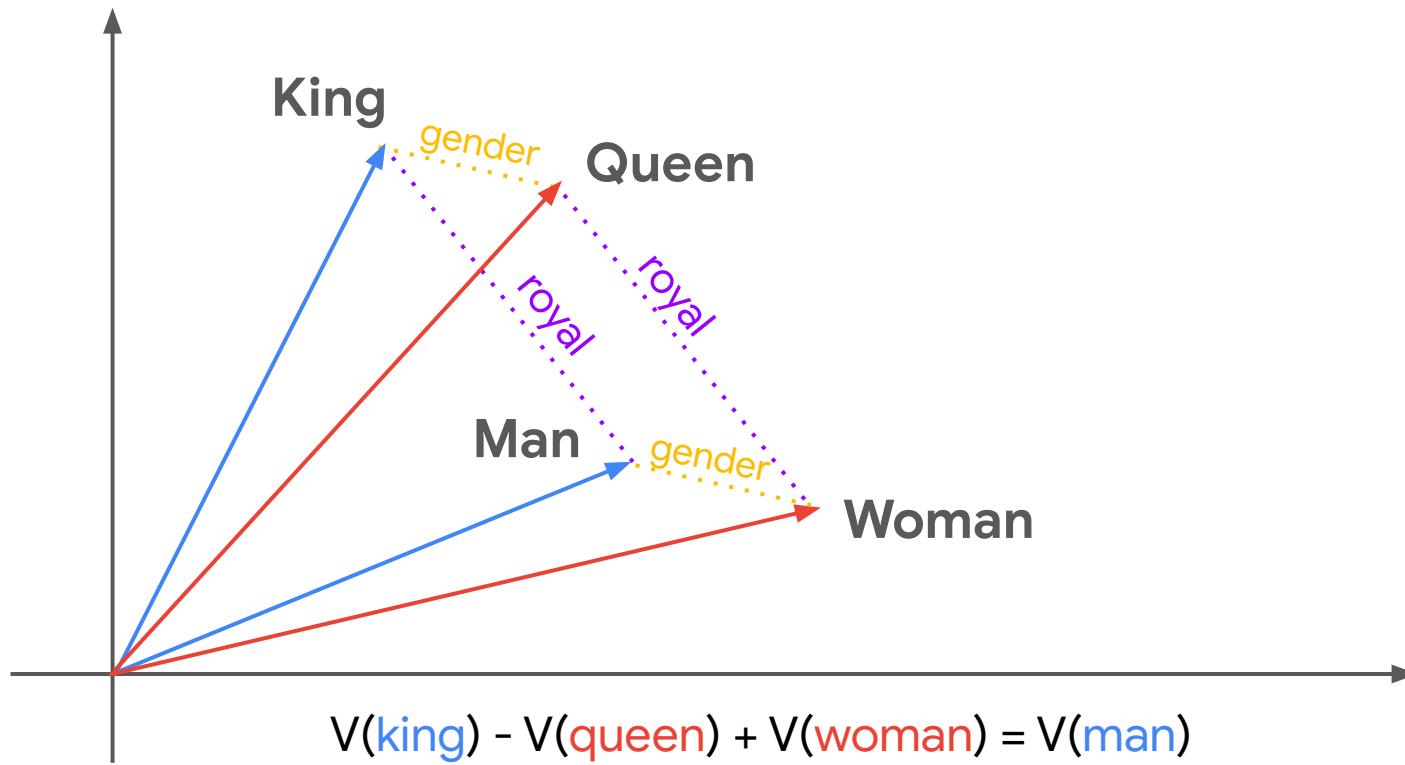


2010s:

Rise of Deep Learning and Neural Networks



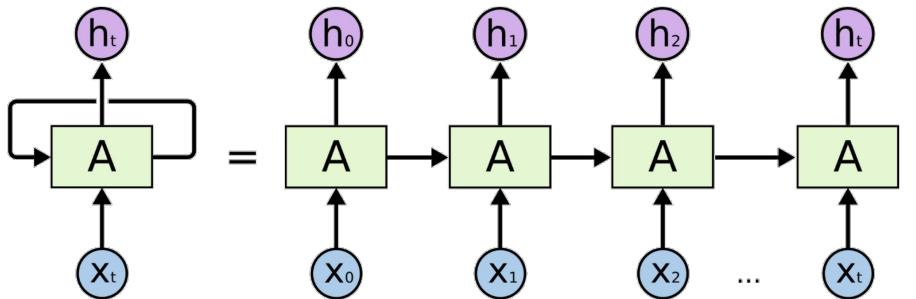
2013: Word2Vec Embeddings



2013: Word2Vec Embeddings

Analogies	Word Pair 1		Word Pair 2	
Man-Woman	king	queen	man	woman
Capital city	Athens	Greece	Oslo	Norway
City-in-state	Chicago	Illinois	Sacramento	California
Opposite	possibly	impossibly	ethical	unethical
Comparative	great	greater	tough	tougher
Nationality adjective	Switzerland	Swiss	Canada	Canadian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars

2010s: Neural Networks RNN, GRU, LSTM



Early Neural Networks

- Slow & forgetful



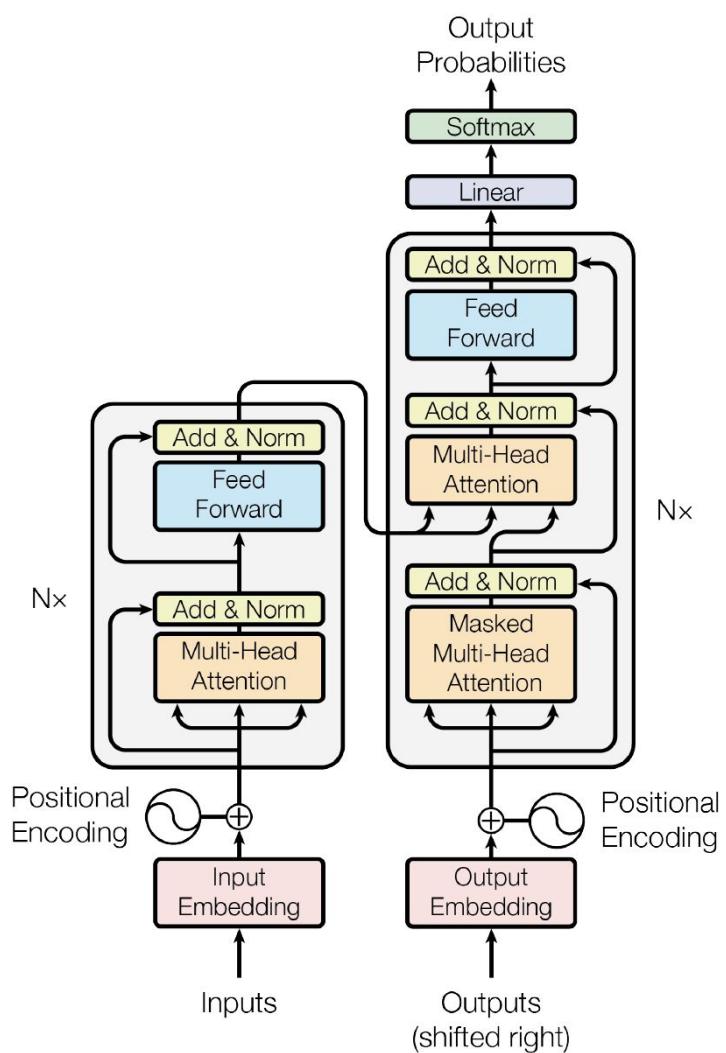
2017: Transformers

- Self-attention Mechanism
- Data hungry
- Parallel processing GPU-optimized



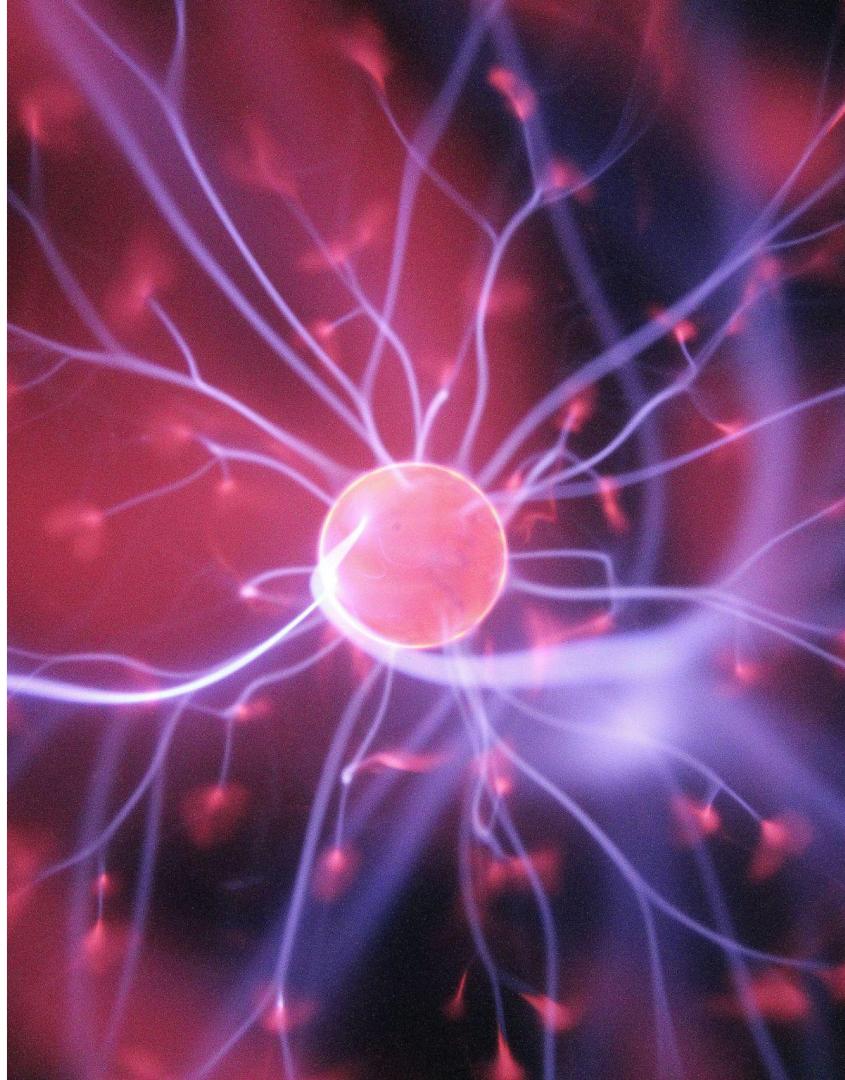
Transformer Architecture

Encoder + Decoder



Rise of LLMs

>1 billion neurons



Trained for
next word prediction



Pre-trained Base

Generalist

vs

Fine-tuned Models

Specialists



Fine-tuned for
Instructions +
Conversations

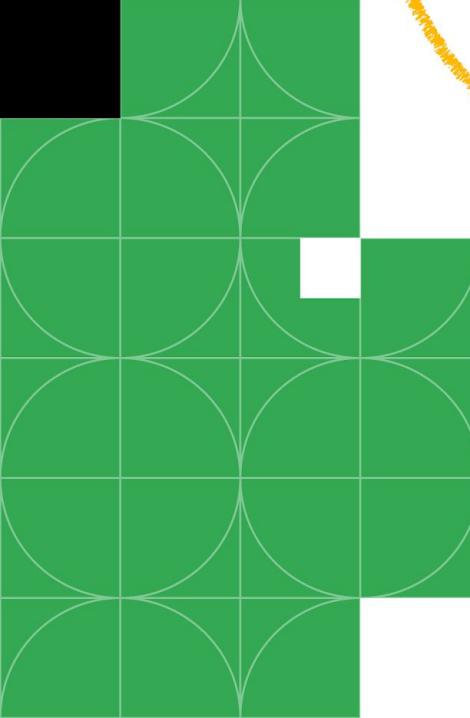


```
text  
  'Section Title',  
  style: TextStyle(  
    color: Colors.green[200],  
  ),  
,
```

devfest

```
s.star,  
r: Colors.green[500],
```

```
Text('23'),
```



Google Developer Groups

02

How to Use Google PaLM?

Google's LLM Tech

T5, BERT, Gopher, Chinchilla, GLaM... LLMs

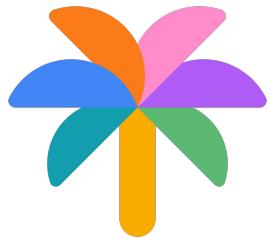
LaMDA: Language Model for Dialogue Applications

PaLM2: Pathways Language Model like GPT-3

Bard: Chatbot like Bing

MakerSuite: Prototyping with PaLM

Gemini: Generalized Multimodal Intelligence Network



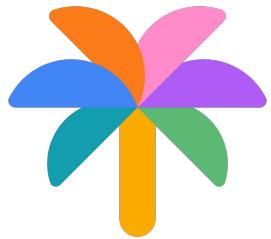
PaLM

Pathways Language Model

Comparable to GPT-3

g.co/palm



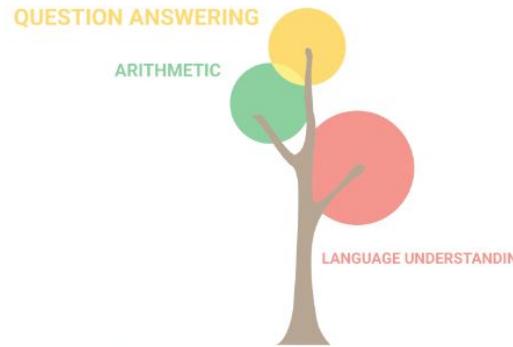


PaLM

- Emergent Abilities
- General Knowledge
- Natural Interactions



Emergent Abilities



8 billion parameters

General Knowledge

- Code
- Writing
- Problem Solving
- Recommend
- Data
- Agents



Natural Interactions

Prompts

- Input
- Context
- Examples:
Few-shot vs 0-shot

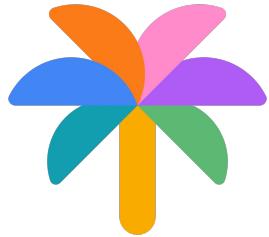


Natural Interactions

Best Practices

- Give instructions to customize its behavior
- Make each instruction clear and concise





PaLM Powered Suite

- Consumer Apps
i.e. Bard
- Developer Tools
i.e. MakerSuite



MakerSuite

makersuite.google.com

Prototyping with PaLM

The screenshot shows the MakerSuite application window with the URL makersuite.google.com/app/home in the address bar. The main content area is titled "Prototype with generative AI". It features three cards for different prompt types: "Text prompt", "Data prompt", and "Chat prompt", each with a "Create" button. Below these cards, a section titled "Explore the full capabilities of the PaLM API." contains two sections: "Model tuning" (with a "Create a tuned model" button) and "Start developing" (with a "Create an API key" button). On the left side of the interface, there is a vertical sidebar with icons for file operations like Open, Save, and Print, along with a "+" icon for creating new files.

Prototype with generative AI

Pick a prompt type to get started working with the PaLM API. [Learn more](#)

Text prompt
A freeform way to experiment with language models

Data prompt
A table that uses rows and columns to organize prompts

Chat prompt
A template for back-and-forth chatbot conversations

Explore the full capabilities of the PaLM API.

Model tuning
Improve the model's responses by using more examples than a standard prompt. [Create a tuned model](#)

Start developing
Call the PaLM API directly from your code. [Create an API key](#)

Relaunch to update

Text Prompt

Summarize a paragraph

Summarize a paragraph | M

makersuite.google.com/app/prompts/simple-summarizer

Save a copy Get code Relaunch to update

Summarize a paragraph

Insert test input

Sample prompts Run settings Reset

Summarize this paragraph and detail some relevant context.

Text: "I am by birth a Genevese, and my family is one of the most distinguished ancestors had been for many years counsellors and syndics, and my father had situations with honour and reputation. He was respected by all who knew him for indefatigable attention to public business. He passed his younger days perpetually in the affairs of his country; a variety of circumstances had prevented his marrying early, nor was it until the decline of life that he became a husband and the father of a family."

Summary: In this text, the narrator is describing his background and upbringing. He tells us that he is a native of Geneva, Switzerland, and that his family is one of the most distinguished in the republic. He then goes on to describe his father, who was a respected public servant.

Text: "The thing the Time Traveller held in his hand was a glittering metallic framework, scarcely larger than a small clock, and very delicately made. There was ivory in it, and some transparent crystalline substance. And now I must be explicit, for this that follows—unless his explanation is to be accepted—is an absolutely unaccountable thing. He took one of the small octagonal tables that were scattered about the room, and set it in front of the fire, with two legs on the hearthrug."

Summary: The Time Traveller holds a glittering metallic device in his hand. It is small and delicate, and made of ivory and some transparent crystalline substance. The Time Traveller sets the device on a table in front of the fire.

Context: The Time Traveller is a scientist who has invented a time machine. He has invited his friend, the narrator, to witness a demonstration of the machine.

Add stop sequence

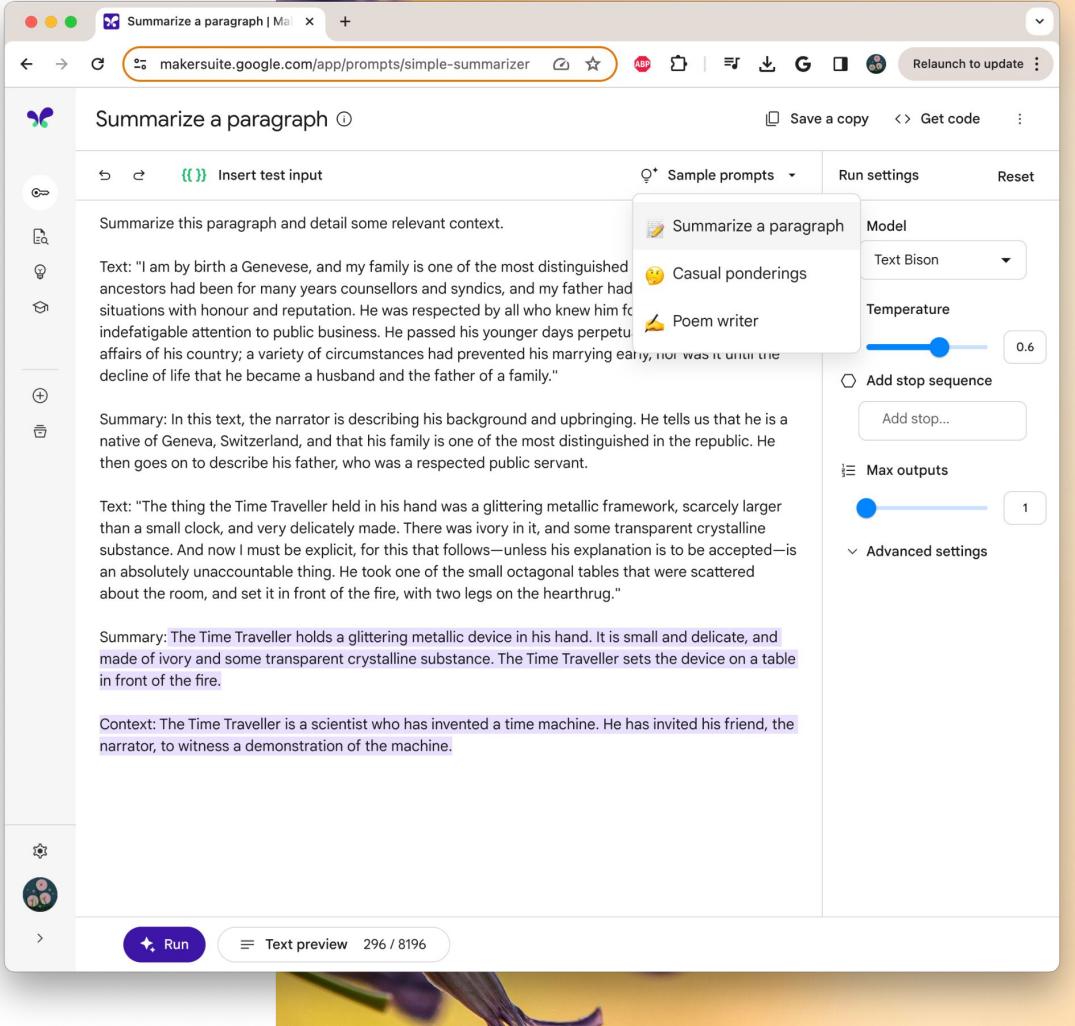
Model Text Bison Temperature 0.6

Add stop...

Max outputs 1

Advanced settings

Run Text preview 296 / 8196



Data Prompt

Analogy factory

- Structure format
 - Examples
- ### Input & Output

The screenshot shows the 'Analogy factory' page in the MakerSuite application. The interface includes a sidebar with icons for lock, search, lightbulb, and more. The main area displays a table of examples:

INPUT Word	OUTPUT Analogy
Busy	like a bee
Living	in a bubble
Talking	to a brick wall
Sly	like a fox
Life	is but a walking shadow

Below the table, there's a section for testing a prompt:

INPUT Word	OUTPUT Analogy
Drop	like a stone

Buttons at the bottom include 'Run' and 'Text preview 55 / 8196'. On the right side, there are settings for 'Model' (Text Bison), 'Temperature' (0.35), 'Add stop sequence', 'Max outputs' (1), and 'Advanced settings'.

- # Chat Prompt
- ## Chat with an alien
- Context
 - Character roleplay
 - Examples
 - Sample behavior
 - Messages
 - Conversation history

The screenshot shows a web-based application window titled "Chat with an alien | MakerSuite". The URL in the address bar is "makersuite.google.com/app/prompts/chat-with-an-alien". The interface includes a sidebar with icons for key, search, idea, and file operations. The main area displays a conversation between a User and a Model. The User's input is "Be an alien that lives on one of Jupiter's moons". The Model's response is:
I am doing well, thank you for asking. I am currently enjoying the beautiful view of Jupiter from my home on Europa. The atmosphere is very thin here, so I can see the stars very clearly. The surface of Europa is also very interesting. It is covered in ice, but there are cracks in the ice that reveal a liquid ocean below. Scientists believe that this ocean may contain life, so I am very excited to learn more about it.
How are you doing today?
The user's input: The model's response:
Despite the challenges, visiting Europa is a once-in-a-lifetime experience. The views of Jupiter and the other moons are simply stunning. And the chance to explore a world that is so different.

At the top right, there are buttons for "Save a copy", "Get code", "Run settings", and "Reset". A dropdown menu labeled "Sample prompts" is open, showing options like "Imagine a world", "Chat with an alien", and "Talk to deep sea diver". There are also sliders for "Model" (set to "Chat Bison") and "Temperature" (set to 0.9), and a checkbox for "Add stop sequence" with a "Add stop..." button. On the right, there are sections for "Max outputs" (set to 1) and "Advanced settings". At the bottom, there is a text input field "Write something the user could type" and a "Text preview" button.

Chat Prompt

Chat with an alien

- Get code

The screenshot shows a web browser window titled "Chat with an alien | Makersuite". The URL is "makersuite.google.com/app/prompts/chat-with-an-alien". The main interface has a sidebar with icons for file operations like Open, Save, and Print. The main area is titled "Chat with an alien" and includes sections for "Sample prompts", "Run settings", "Model" (set to "Chat Bison"), and "Test your prompt". A "Get code" button is prominent, and a modal window is displayed over the interface. The modal title is "Get code" and contains a message: "Create your API key before using the code in your project". Below this, it says "You can call this prompt from the PaLM API by copying the following code into your project". It provides code examples in cURL, JavaScript, JSON, and Python. The Python tab is selected, showing the following code:

```
24 messages = [
25     "I'd like to visit, what should I do?",
26     "To visit Europa, you will need to take a spaceship. The journey will take several years",
27     "Is there any food on Europa?",
28     "There is no food on Europa that we know of. The surface of Europa is covered in ice,
29 ]
30 messages.append("NEXT REQUEST")
31 response = palm.chat(
32     **defaults,
33     context=context,
34     examples=examples,
35     messages=messages
36 )
37 print(response.last) # Response of the AI to your most recent request
```

Below the code, there are sections for "User" (The user's input), "Model" (The model's response), and a text input field labeled "Write something the user could type". At the bottom, there is a "Text preview" section showing "652 / 4096".

REST APIs

Client libraries for

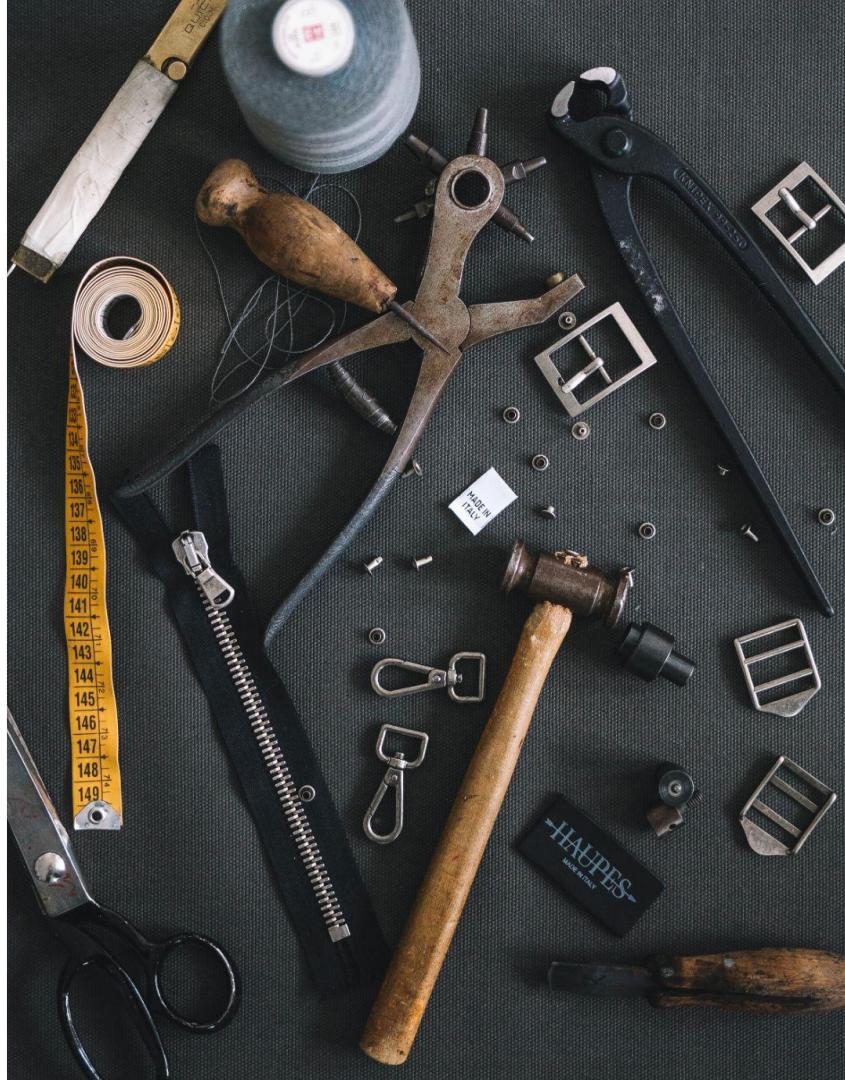
- Python
- Node.js
- Java
- Swift
- cURL



Parameter Settings

Tokens

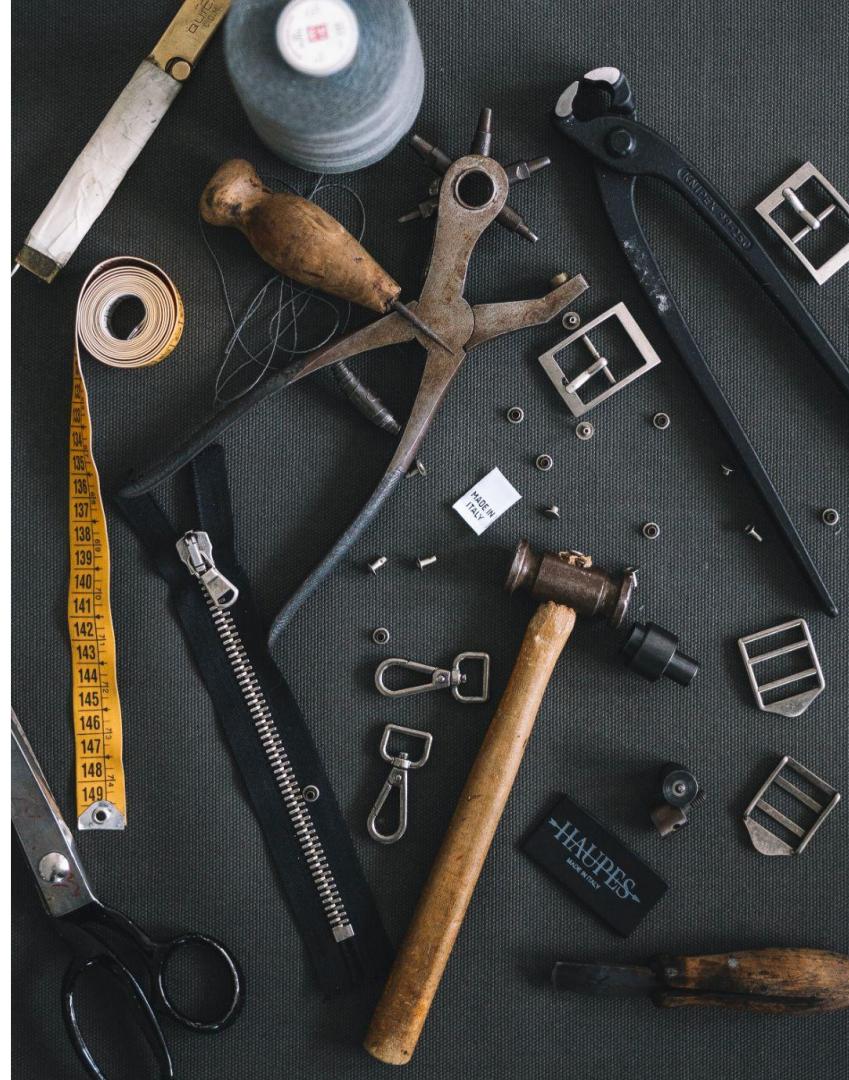
- Words, subwords, bytes
- Different LLMs have different tokenizers
- Measure training data, context window, output
- Base for charges



Parameter Settings

Fact vs Fun?

- Temperature
Between 0 to 1.0
- Top K
Fix number candidates
- Top P
Probability distribution
- Max Output



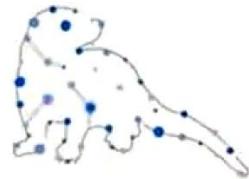
Parameter Settings

Models



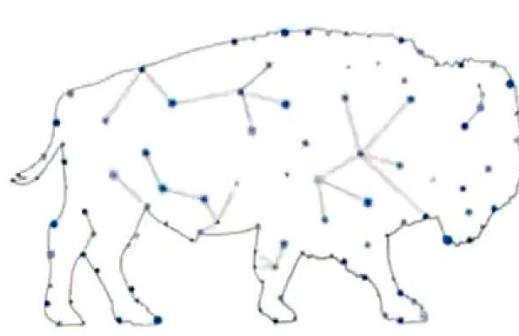
Gecko

Mobiles
Embedded



Otter

Laptops
Desktops



Bison

Cloud-based apps
Large-scale



Unicorn

Cloud-based apps
Complex

API Keys

Treat it like a password

API keys

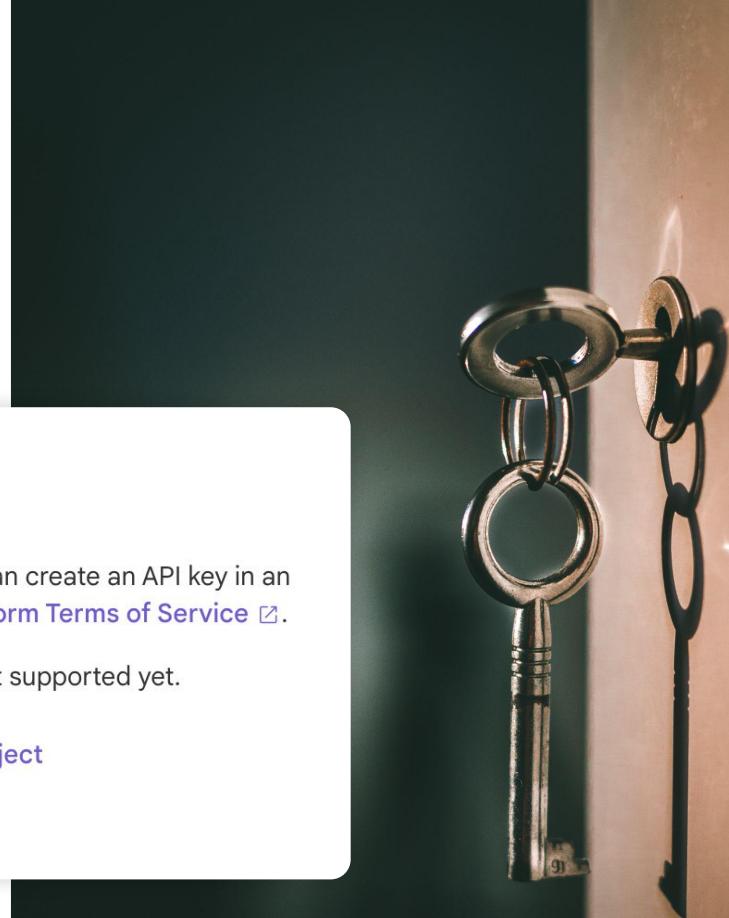
MakerSuite creates a new Google Cloud project for each new API key. You also can create an API key in an existing Google Cloud project. All projects are subject to the [Google Cloud Platform Terms of Service](#).

Note: The PaLM API is currently in public preview. Production applications are not supported yet.

 [Create API key in new project](#)

or

[Create API key in existing project](#)



Setup

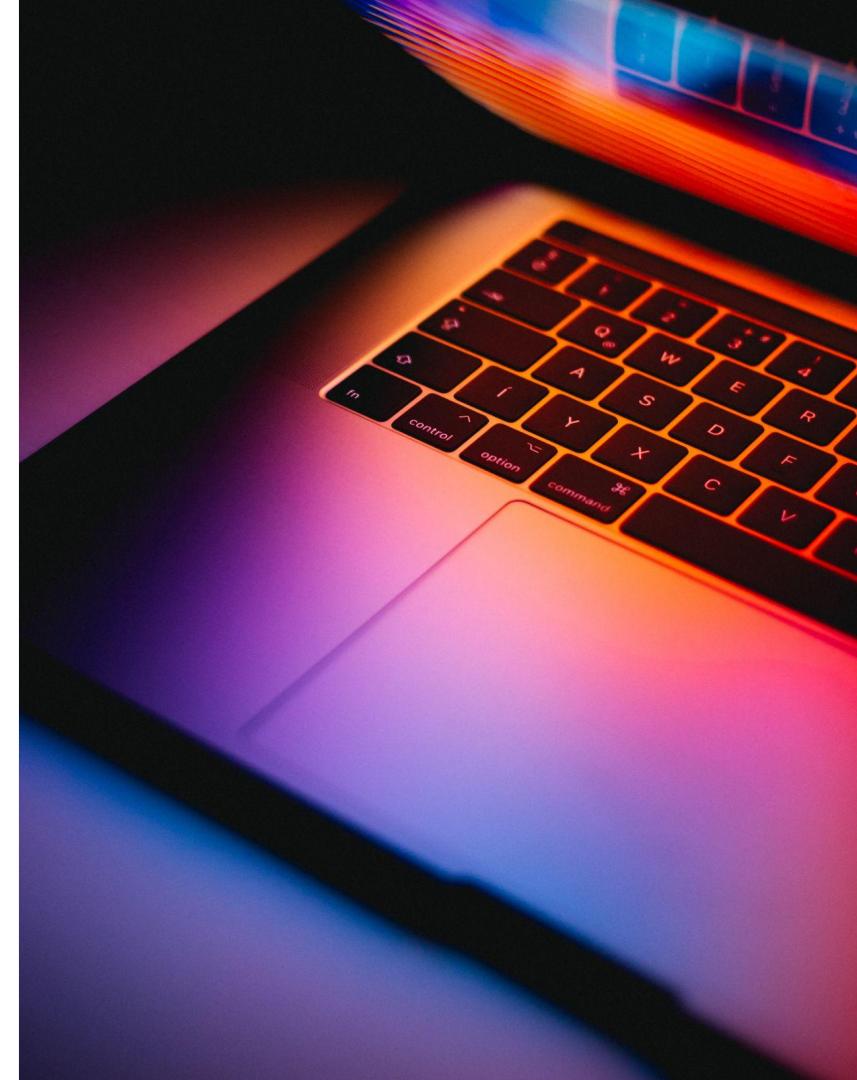
Install & import libraries

```
$ pip install google-generativeai  
  
import google.generativeai as palm  
palm.configure(api_key="<YOUR API KEY>")
```



3 Endpoints

- **Text**
Short interactions
- **Chat**
Interactive applications
- **Embeddings**
Develop new applications



Text Endpoint

Best for short interactions

```
response = palm.generate_text(  
    model="models/text-bison-001",  
    prompt="Write a story about a magic backpack")  
print(response.result)  
# output: Once upon a time, there was a young boy...
```

Chat Endpoint

Best for interactive applications

```
response = palm.chat(  
    model="models/chat-bison-001",  
    context="Be an alien from one of Jupiter's moons",  
    examples=["How's it going?", "I'm well, thank you."],  
    messages=["I'd like to visit, what should I do?"])  
  
print(response.last)
```

Chat Endpoint

Best for interactive applications

```
response = response.reply("Is there any food?")
print(response.last)

# output: There is no food on Europa that is safe...
```

Embeddings Endpoint

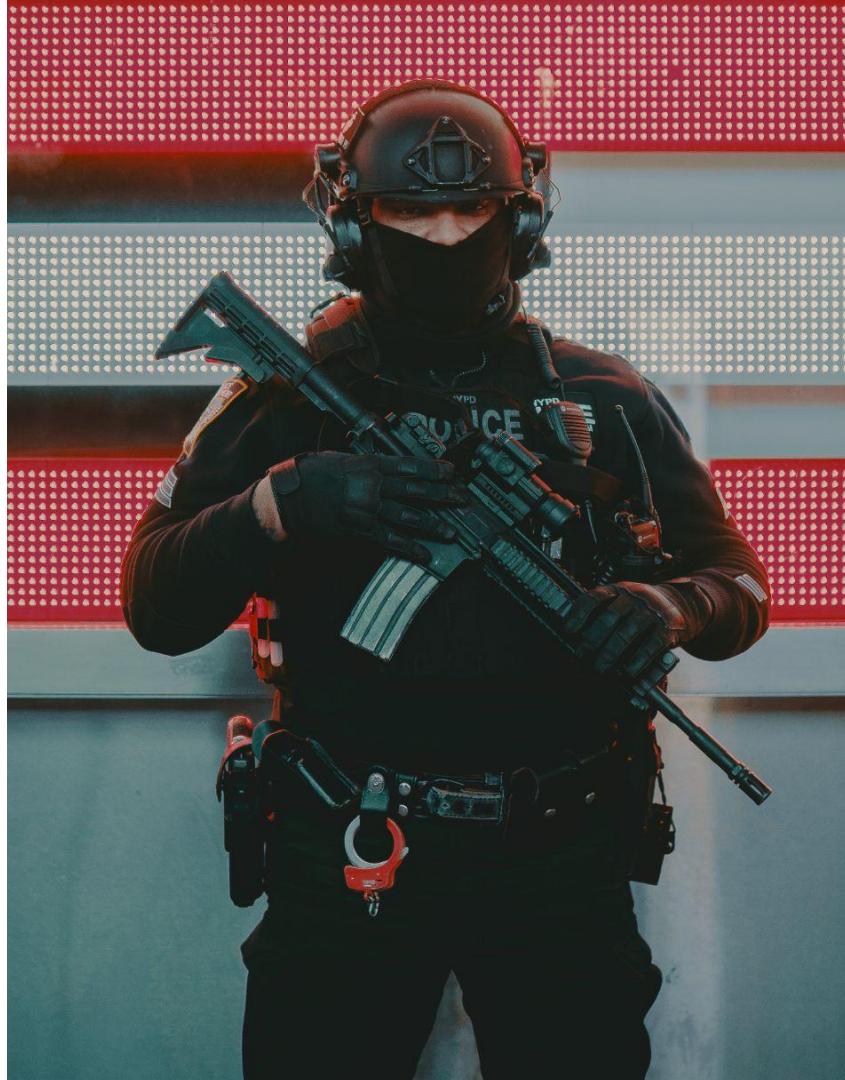
Develop new applications

```
embedding = palm.generate_embeddings(  
    model="models/embedding-gecko-001",  
    text="What are the embeddings for this text?")  
print(embedding)  
# output: [0.019084517, -0.012082022, -0.03974377, ...]
```

Safety Ratings

Harm Categories

- Derogatory
- Toxicity
- Violence
- Sexual
- Medical
- Dangerous



Safety Ratings

Harm Categories

- Derogatory
- Toxicity
- Violence
- Sexual
- Medical
- Dangerous

Harm Probabilities

- NEGLIGIBLE
- LOW
- MEDIUM
- HIGH

Resources

Notebook with all code snippets + links at bit.ly/devfest23_palm

1. **PaLM API**

g.co/palm

2. **Bard Chatbot**

bard.google.com

3. **MakerSuite for Prototyping**

makersuite.google.com

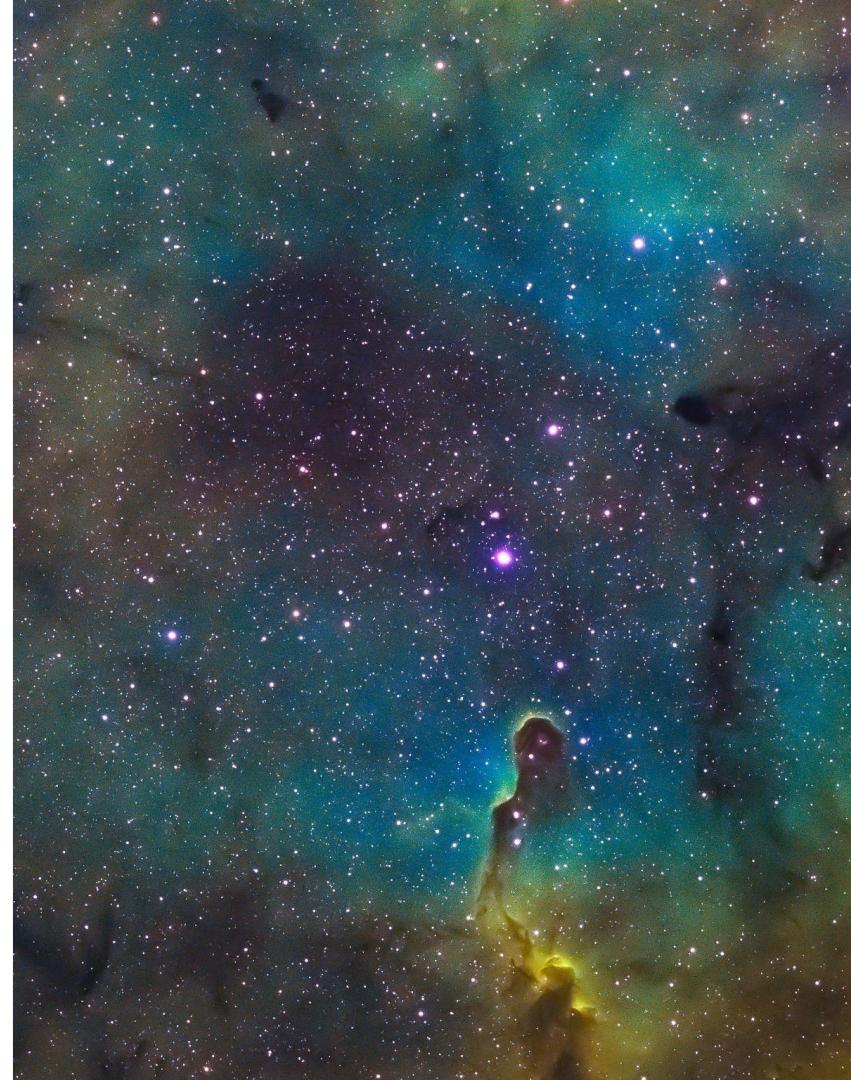
4. **Google Tutorials**

developers.generativeai.google/tutorials

Gemini

Generalized Multimodal Intelligence Network

Comparable to GPT-4



Embrace Safely



devfest



ChengCheng Tan

ccstan99@gmail.com

in cheng2-tan

twitter @cheng2_tan

 Google Developer Groups

