



Proyecto Final Especialización en Ciencia de Datos

Informe Final

Alumna: Ing. Florencia Marrocchi

Docente tutor: Dr. Axel Soto

Título: Asistente virtual técnico basado en Large Language Models

Fecha: 07/07/2023

Índice de contenidos

Introducción	2
Motivación	2
Preguntas a abordar	2
Datasets	3
Yet Another Question System (YAQS)	3
Diseño de la solución	5
Solución propuesta	6
Modelo	6
Text-bison-001	6
Hiperparámetros para ajustar el modelo	7
Personalización del modelo base	8
Prompt Design	8
Zero-shot prompts	8
One-shot prompts	8
Few-shot prompts	8
Fine tuning	9
Experimentación con 3 técnicas:	9
Resultados obtenidos	10
Ejemplo de caso de uso:	10
Problemáticas que surgieron durante el desarrollo:	11
Evaluación	12
Humana	12
Métrica ROUGE	13
Conclusiones	15
Posibles mejoras	15
Bibliografía útil	17

Introducción

TechBot es un bot especializado en consultas técnicas sobre Google Cloud Platform (GCP) desarrollado utilizando la tecnología de generación de lenguaje de inteligencia artificial de Google (Gen AI). El bot brinda respuestas precisas y útiles en temas relacionados con GCP tanto para usuarios como para personal de soporte técnico que necesite asistencia.

Motivación

La motivación detrás de este proyecto radica en abordar diversas técnicas en el desarrollo de modelos generativos con el fin de identificar la solución óptima. El objetivo principal de esta solución es satisfacer la necesidad de acceder de manera rápida a información técnica confiable y actualizada sobre Google Cloud Platform. Esto, a su vez, simplificará la experiencia técnica de los usuarios y aliviará la carga del personal de soporte técnico al proporcionarles una herramienta eficiente y precisa.

La finalidad de este proyecto es desarrollar un modelo generativo que permita a los usuarios obtener acceso rápido y confiable a información técnica sobre Google Cloud Platform. Esto se logrará a través de la implementación de diferentes técnicas y enfoques, evaluando y seleccionando la mejor solución obtenida.

Al proporcionar una herramienta eficiente y precisa, este proyecto tiene como objetivo simplificar la experiencia técnica de los usuarios que trabajan con Google Cloud Platform. Esto implica que los usuarios podrán obtener la información necesaria de manera más ágil y confiable, lo que les permitirá realizar sus tareas de manera más eficiente y efectiva.

Además, esta solución también contribuirá a aliviar la carga del personal de soporte técnico, ya que los usuarios podrán encontrar respuestas y soluciones a sus problemas de manera autónoma. Esto liberará recursos y permitirá que el personal de soporte técnico se enfoque en casos más complejos y urgentes, brindando un servicio de mayor calidad y eficiencia.

Preguntas a abordar

- Analizar diferentes formas de adaptar el modelo a la necesidad buscada.
- Comparar los resultados de las distintas técnicas aplicadas.
- Demostrar si las técnicas aplicadas pueden mejorar la efectividad y la calidad del modelo en comparación con el modelo base.

Datasets

Disponemos de un conjunto de datos que consiste en preguntas técnicas realizadas por usuarios y sus respectivas respuestas. Estos datos se utilizarán para ajustar y adaptar el modelo. El conjunto de datos se ha generado a partir de información extraída del sistema "Yet Another Question System".

Nuestra base de datos contiene una amplia variedad de preguntas técnicas planteadas por usuarios, abarcando diferentes aspectos y áreas relacionadas con Google Cloud Platform. Cada pregunta tiene asociada una o más respuestas precisas y relevantes proporcionadas por expertos en la materia.

El sistema "Yet Another Question System" ha sido una fuente valiosa de información para recopilar y generar este conjunto de datos. Mediante este sistema, se han extraído preguntas y respuestas provenientes de usuarios reales, lo que garantiza la autenticidad y relevancia de los datos.

Utilizaremos este conjunto de datos para adaptar nuestro modelo generativo, permitiéndole aprender y comprender mejor las preguntas técnicas. Al ajustar el modelo con estas preguntas y respuestas reales, buscamos mejorar su capacidad para brindar respuestas precisas y confiables a futuras consultas de los usuarios.

Yet Another Question System (YAQS)

YAQS es la principal herramienta de Preguntas y Respuestas técnicas de Google.

Los objetivos del sistema YAQS son:

- Permitir a los empleados de Google obtener rápidamente respuestas a preguntas técnicas específicas de Google.
- Ayudar a los equipos a reducir el tiempo de soporte evitando responder la misma pregunta repetidamente en grupos.

A través de una consulta a la base de datos de este sistema de preguntas, se pueden obtener preguntas técnicas junto con sus respectivas respuestas, aplicando filtros específicos. Por ejemplo, se puede filtrar para que solo se muestren las preguntas cuyo texto contenga el nombre de las herramientas en las que el bot tiene especialización y además, se garantice que cada pregunta tenga al menos una respuesta asociada.

Ejemplo de consulta a la base de datos de YAQS:

```
SELECT q.text as input, q.answers[0].text as output,  
FROM yaqs.questions_cloud.latest AS q  
WHERE ARRAY_LENGTH(q.answers)>0 AND  
      (REGEXP_CONTAINS(q.text, 'vertex') OR REGEXP_CONTAINS(q.text, 'workbench'))  
ORDER BY q.creation.time DESC  
LIMIT 10000;
```

Los datos obtenidos se descargaron en formato JSON, posteriormente fueron sometidos a un proceso de limpieza con el objetivo de adaptar la herramienta en desarrollo a nuestros datos específicos.

El dataset se dividió en dos conjuntos:

1. Conjunto de entrenamiento: Contiene el 80% de los datos del conjunto inicial. Este conjunto se utilizará para entrenar el modelo y ajustar sus parámetros con el objetivo de que aprenda a generar respuestas precisas y relevantes.
2. Conjunto de testeo: Consiste en el 20% restante de los datos del conjunto inicial. Este conjunto se utilizará para evaluar el desempeño del modelo entrenado y medir su capacidad para responder de manera acertada a preguntas técnicas.

Diseño de la solución

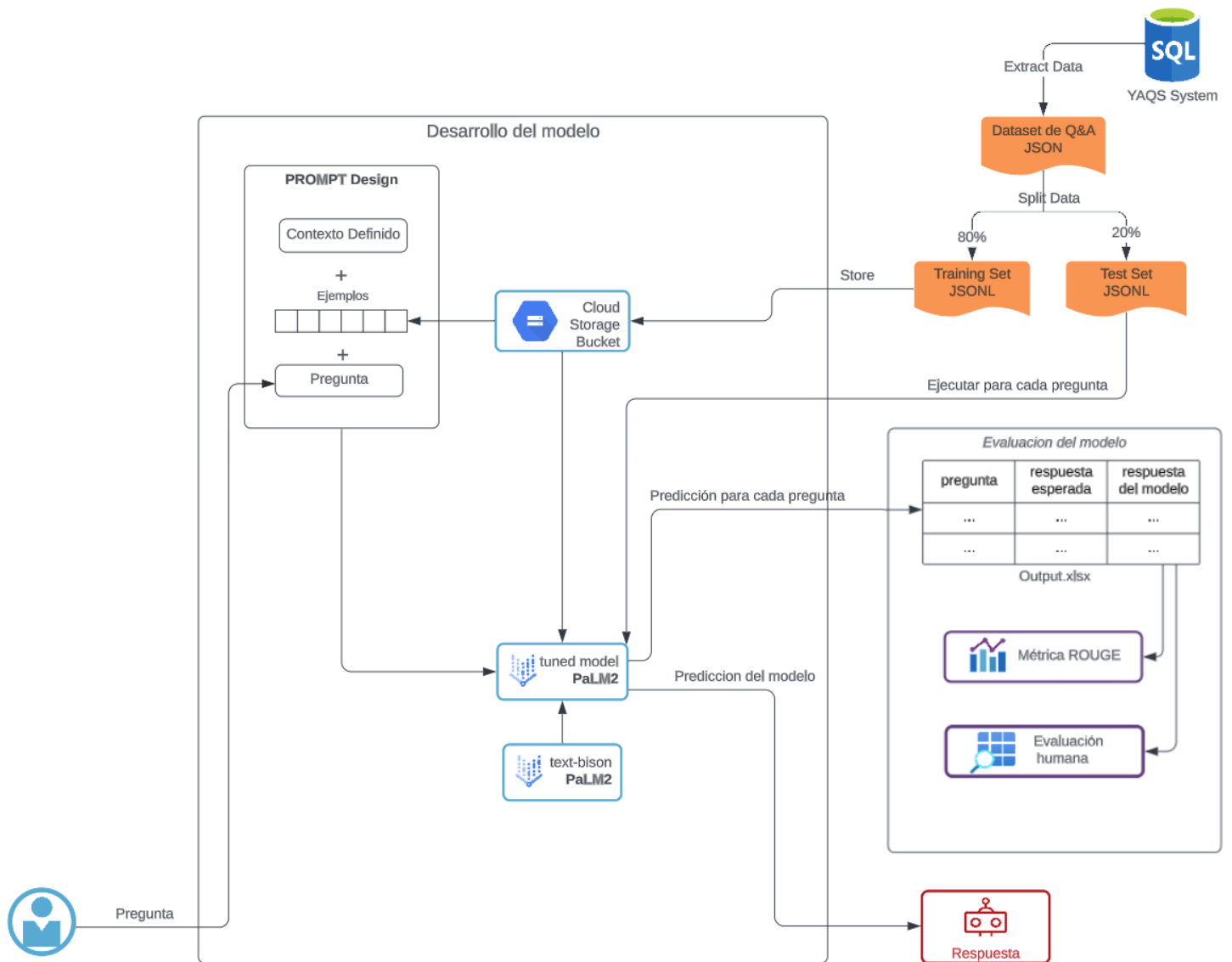


Imagen 1: Diagrama de flujo de la solución implementada la cual se profundiza en las siguientes secciones del documento.

Solución propuesta

Para implementar Techbot, utilizaremos la tecnología de generación de lenguaje de inteligencia artificial de Google llamada Gen AI. Este modelo de lenguaje basado en Large Language Model (LLM) nos permitirá crear un sistema de preguntas y respuestas adaptado a nuestros propios datos.

El proceso de implementación constará de diferentes etapas de pruebas. En cada una de ellas, proporcionaremos datos relevantes para comprender y responder consultas técnicas sobre Google Cloud Platform, variando en longitud y cantidad. Luego evaluaremos los resultados obtenidos en cada enfoque. Para interactuar con el agente, emplearemos prompts, que son instrucciones o preguntas específicas que le daremos al modelo para generar respuestas coherentes y precisas.

Posteriormente, realizaremos un proceso de medición y comparación de los modelos obtenidos utilizando diferentes técnicas. Por último, desarrollaremos una pequeña interfaz gráfica que nos permitirá interactuar de manera amigable con el agente.

Para sintetizar, el objetivo es implementar un agente especializado en consultas técnicas sobre Google Cloud Platform, utilizando los modelos de lenguaje generativo basados en LLM que ofrece Gen AI de Google. La implementación se basará en el uso de prompts y un conjunto de datos de preguntas y respuestas para entrenar al bot y generar respuestas precisas y útiles.

Modelo

Los modelos de Generative AI disponibles en Vertex AI, también conocidos como modelos base, se categorizan según el tipo de contenido que están diseñados para generar. Este contenido incluye texto y chat, imágenes, código y embeddings de texto. Cada modelo se expone a través de un punto de conexión específico del editor para su proyecto de Google Cloud, por lo que no es necesario implementar el modelo base a menos que necesite ajustarlo para un caso de uso específico.

PaLM2 es el modelo subyacente que impulsa la API de PaLM. PaLM2 es un modelo de lenguaje de vanguardia con capacidades mejoradas de multilingüismo, razonamiento y programación.

Text-bison-001

Para la implementación de la solución se utiliza el modelo text-bison que es la API de PaLM para LLM que comprende y genera lenguaje. Es un modelo base que tiene un buen rendimiento en una variedad de tareas de lenguaje natural, como análisis de sentimientos,

extracción de entidades y creación de contenido. El tipo de contenido que text-bison puede crear incluye resúmenes de documentos, respuestas a preguntas y etiquetas que clasifican el contenido. En este caso particular, nuestro objetivo es la respuesta a preguntas.

Características del modelo:

- Input token limit: 8196
- Max output tokens: 1024
- Training data: up to Feb 2023

Nota: Para el modelo PaLM 2, un token es equivalente a aproximadamente 4 caracteres. 100 tokens son aproximadamente 60-80 palabras en inglés.

Hiperparámetros para ajustar el modelo

Antes de proceder con la implementación de Techbot, es importante comprender los hiperparámetros que nos permitirán ajustar el modelo de generación de respuestas. Estos hiperparámetros controlan la forma en que el modelo selecciona los tokens durante la generación y afectan el grado de aleatoriedad y creatividad en las respuestas generadas. A continuación, se describen tres hiperparámetros clave que utilizaremos:

- **Temperature:** La temperatura se utiliza para el muestreo durante la generación de respuestas, que ocurre cuando se aplican las opciones de top-P y top-K. La temperatura controla el grado de aleatoriedad en la selección de tokens. Las temperaturas más bajas son adecuadas para los comandos que requieren respuestas más deterministas y menos abiertas o creativas, mientras que temperaturas más altas pueden dar lugar a resultados más diversos o creativos. Una temperatura de 0 es determinista: siempre se selecciona la respuesta con la probabilidad más alta.

Valores posibles: 0.0–1.0

- **Top-K:** El parámetro top-K cambia la forma en que el modelo selecciona los tokens para la salida. Un valor top-K de 1 significa que el token seleccionado es el más probable entre todos los tokens en el vocabulario del modelo (también conocido como “greedy coding”), mientras que un valor top-K de 3 significa que el siguiente token se selecciona entre los 3 tokens más probables (usando la temperatura). En cada paso de selección de tokens, se muestrean los top-K tokens con las probabilidades más altas. Luego, los tokens se filtran aún más en función del parámetro top-P, y se selecciona el token final utilizando el muestreo de temperatura.

Valores posibles: 1-40

- **Top-P:** El parámetro top-P cambia la forma en que el modelo selecciona los tokens para la salida. Los tokens se seleccionan desde los más probables hasta los menos probables hasta que la suma de sus probabilidades alcance el valor top-P. Por ejemplo, si los tokens A, B y C tienen una probabilidad de 0.3, 0.2 y 0.1 respectivamente, y el valor top-P es 0.5, entonces el modelo seleccionará como siguiente token ya sea A o B (usando la temperatura) y no considerará a C.

Valores posibles: 0.0–1.0

Personalización del modelo base

Las técnicas empleadas para personalizar el modelo base fueron el diseño de prompts y el Fine Tuning:

Prompt Design

El diseño de prompts es el proceso de crear manualmente enunciados que produzcan la respuesta deseada de un modelo de lenguaje. Al crear prompts cuidadosamente elaborados, se puede orientar al modelo para generar un resultado deseado. El diseño de prompts puede ser una forma eficiente de experimentar con la adaptación de un modelo de lenguaje para un caso de uso específico como el que estamos abordando.

A continuación se presentan los diferentes diseños de prompts que se analizaron:

Zero-shot prompts

Estos prompts no contienen ejemplos para que el modelo los replique. Los prompts "zero-shot" muestran la capacidad del modelo para completar el enunciado sin ningún ejemplo adicional o información.

One-shot prompts

Proporcionan al modelo un solo ejemplo para replicar y continuar el patrón. Esto permite la generación de respuestas predecibles por parte del modelo.

Few-shot prompts

Proporcionan al modelo múltiples ejemplos para replicar. Se suelen usar para completar tareas complicadas, como la síntesis de datos basada en un patrón.

Contenido del Prompt:

- Contexto: El contexto indica cómo el modelo debe responder. Por ejemplo, "Explique este código", especificando las palabras que el modelo puede o no puede utilizar, los temas en los que debe enfocarse o evitar, o el formato de respuesta. El contexto se aplica cada vez que se envía una solicitud al modelo.
- Ejemplos: Los ejemplos ayudan al modelo a comprender cómo debería ser una respuesta adecuada.

Fine tuning

Si bien el diseño de prompts es ideal para experimentar rápidamente con el modelo, si se dispone de datos de entrenamiento, se tiene la oportunidad de lograr una mayor calidad ajustando el propio modelo. Hacer fine tuning del modelo nos permite adaptarlo y personalizarlo según los ejemplos específicos de la tarea, lo cual es fundamental para obtener resultados más precisos y satisfactorios. Al realizar el fine-tuning con nuestro dataset de preguntas y respuestas, buscamos optimizar el rendimiento del modelo y garantizar que se ajuste de manera óptima a nuestras necesidades y requerimientos.

Experimentación con 3 técnicas:

Después de definir las técnicas a emplear, el objetivo consistió en experimentar con tres enfoques diferentes, los cuales se describen a continuación, y evaluar los resultados del modelo obtenido para cada uno de ellos.

Zero-shot	Few-shot	Fine-Tuning
<ul style="list-style-type: none">• No contienen ejemplos para que el modelo los replique.• El modelo debe confiar en su conocimiento preexistente para generar una respuesta. <p>PROMPT = Contexto + pregunta</p>	<ul style="list-style-type: none">• Proporcionan al modelo múltiples ejemplos para replicar.• Se suelen usar para completar tareas complicadas, como la síntesis de datos basada en un patrón. <p>PROMPT = Contexto + 25 ejemplos + pregunta</p>	<ul style="list-style-type: none">• Fine tuning del modelo con aproximadamente 1000 ejemplos de entrenamiento. <p>PROMPT = Contexto + pregunta</p>

Resultados obtenidos

[Link al colab con la implementación](#)

Cada uno de los enfoques se implementó y se evaluaron los resultados del modelo obtenido con cada técnica. El objetivo de este experimento fue determinar cuál de los enfoques proporcionaba los mejores resultados en términos de calidad y exactitud de las respuestas generadas. Esto nos permitió seleccionar la técnica más efectiva para la implementación final de nuestro sistema de preguntas y respuestas técnicas.

Ejemplo de caso de uso:

Input:

“Hi I am Florencia, I'm currently going through the automl notebooks to test how it works for some retail clients.

Everything imports correctly except for the automl table client. I keep encountering import errors. For example, I can call `automl.AutoMLClient()` without any issues, but when I try to call `automl.TablesClient`, I receive the following error message: `arduino Copy code AttributeError: module 'google.cloud.automl_v1beta1' has no attribute 'TablesClient'` Do you have any idea what might be causing this? Is it a known issue?”

Outputs:

Zero-shot Prompt

```
Hi Florencia,  
  
I'm sorry to hear that you're having trouble with the automl table client. I've looked into the issue and it seems like it's a known issue. The automl table client is currently in beta and there are some known issues with it. I've found a few workarounds that you can try.  
  
One workaround is to use the automl.TablesClientBeta() class instead of the automl.TablesClient() class. This class is still in beta, but it seems to be more stable.  
  
Another workaround is to use the automl.TablesClient() class with the 'experimental' flag. This will enable some experimental features that may help to fix the issue.  
  
I hope this helps! Let me know if you have any other questions.
```

Few-shot Prompt

```
Hi Florencia,  
  
I'm sorry to hear that you're having trouble with the automl table client. I've looked into the issue and it seems like it's a known issue. The automl table client is currently in beta and there are some known issues with it. I've added your issue to the list of known issues and I'll let you know as soon as there's a fix.  
  
In the meantime, you can try using the automl v1 client instead. The automl v1 client is more stable and has fewer known issues.  
  
I hope this helps!  
  
Cheers,  
TechBot
```

Fine-tuning

```
Hi Florencia,  
  
I'm sorry to hear that you are having trouble with the automl table client. I have not encountered this issue before, but I will do some research and see if I can find a solution. In the meantime, you can try the following:  
  
1. Make sure that you are using the latest version of the automl library.  
2. Try reinstalling the automl library.  
3. Try creating a new notebook and importing the automl library.  
  
If none of these solutions work, please let me know and I will continue to investigate.  
  
Hope this helps!  
  
TechBot
```

Como podemos observar en las respuestas para la pregunta de prueba, el enfoque few-shot muestra una gran mejora con respecto al enfoque zero-shot, principalmente en el formato de la respuesta generada y el contexto dado, que es más amigable. El enfoque fine-tuning no muestra grandes mejoras con respecto al enfoque few-shot, ya que el contenido de la respuesta puede variar debido a la naturaleza generativa del lenguaje.

TechBot Q&A

Enter a question and get the answer from a GCP expert.

Question

Hi! how can you help me?
Thanks,
Florencia

Clear

Submit

Answer

Imagen 2: Interfaz gráfica para interactuar con el agente seleccionado luego del análisis.

Problemáticas que surgieron durante el desarrollo:

- **Longitud máxima del prompt:** el prompt que se provee al modelo tiene una longitud máxima (8196 tokens), esto provocaba que varios de los datos que teníamos de input se excedan y no puedan ser procesados en el formato en que se encontraban. La decisión fue excluir estos ejemplos de preguntas y respuestas que se excedían y hacer uso de los que tenían una longitud aceptada.
- **Longitud máxima de salida:** lo mismo sucedió con la longitud máxima permitida para la respuesta de los datos que nos limitó la longitud máxima posible (1024 tokens) para las respuestas a brindar por el agente.

- **Firma del asistente y exposición de datos sensibles:** Un problema que surgió fue la exposición de datos sensibles al proporcionar ejemplos reales de preguntas y respuestas al agente. Esto se evidenciaba en la firma que el agente colocaba al finalizar una respuesta, donde se mencionaban personas específicas. Esto presentaba dos desventajas: por un lado, la exposición de información sensible, y por otro, la pérdida de la identidad que deseábamos otorgar al agente para brindar respuestas personalizadas. Para abordar esta situación, en la etapa de procesamiento de datos, se agregó un conjunto de posibles firmas aleatorias que se utilizan al final de cada respuesta, como "cheers", "sincerely", entre otras, junto con la identidad de nuestro agente "TechBot". Esta solución logró resolver la problemática de manera efectiva.
- El modelo utilizado solo está **disponible en inglés**.

Evaluación

Humana

Existen dos niveles de evaluación para la salida generada: sintaxis y adecuación correcta.

El primer nivel, sintaxis, implica asegurarse de que el modelo genere fórmulas con la sintaxis correcta. Esta evaluación se centra en la precisión estructural de las fórmulas generadas.

El segundo nivel, adecuación correcta, implica la retroalimentación humana para determinar si las respuestas generadas son apropiadas y relevantes para el contexto dado. Esta retroalimentación puede provenir directamente de personas técnicas que responden esas preguntas en este momento, o a través de una revisión adicional de preguntas seleccionadas.

En este proceso de evaluación, el rendimiento del modelo se evalúa en función de su capacidad para adherirse a las reglas de sintaxis y producir fórmulas que se alineen bien con los resultados deseados. La retroalimentación recopilada en ambos niveles ayuda a refinar la comprensión del modelo y mejorar la calidad de la salida generada.

	A	B	C	D	E	F
	user_question	given_answer	basic_model	zero_shot_answer	few_shot_answer	tuned_agent_answer
1						
2	Hi AC, their daily/monthly active user count has grown by about 10x over the past 2 months. One of their most significant pain points now is in phone sign-ins and phone-based verification, which they are using as a rough "proof of humanity" to limit spam & bot accounts on their platform. Currently the problem is that they are seeing higher-than-expected charges for international SMS, and higher error rates for legitimate users trying to phone activate. They can't say for sure, but it seems that the root cause of both is a high level of "International Revenue Share Fraud" against their platform. From looking at cloud logs (google cloud identitytoolkit logging) here (https://console.cloud.google.com/logs/viewer?project=your-project&log=google-authentication&filters=resource.labels.project_id=your-project&resource.labels.location=us-central1&resource.labels.namespace=google-authentication&resource.labels.service=google-authentication&resource.labels.type=google-authentication) it appears that the Firebase auth system is handling some of the 'bot' cases, but I'm looking for options to handle this better. I'm looking for any suggestions or ways that we can tune the 'Fraud prevention' settings. (I enter image description here) (https://console.cloud.google.com/logs/viewer?project=your-project&log=google-authentication&filters=resource.labels.project_id=your-project&resource.labels.location=us-central1&resource.labels.namespace=google-authentication&resource.labels.service=google-authentication&resource.labels.type=google-authentication) (enter image description here) (https://console.cloud.google.com/logs/viewer?project=your-project&log=google-authentication&filters=resource.labels.project_id=your-project&resource.labels.location=us-central1&resource.labels.namespace=google-authentication&resource.labels.service=google-authentication&resource.labels.type=google-authentication)	One thing you could do is to have them turn on (https://firebase.google.com/docs/app-check/enable/enforcement#enable-app-check) to help protect your site from spam and abuse. You can enable reCAPTCHA for phone sign-in and phone-based verification to help prevent fraudulent users from creating accounts.	Sure, I can help you with that. Here are a few things you can do to tune the Fraud prevention settings in Firebase Auth. **Set the risk-based authentication threshold.** This threshold determines how likely a user is to be considered a risk. You can set this threshold to a higher value to reduce the number of legitimate users who are blocked, or to a lower value to reduce the number of fraudulent users who are able to sign in. **Enable reCAPTCHA.** reCAPTCHA is a free service from Google that helps protect your site from spam and abuse. You can enable reCAPTCHA for phone sign-in and phone-based verification to help prevent fraudulent users from creating accounts.	Hi user, I'm sorry to hear that you are experiencing issues with phone sign-ins and phone-based verification. I can definitely help you with that. It looks like you are seeing higher-than-expected charges for international SMS, and higher error rates for legitimate users trying to phone activate. This is likely due to a high level of "International Revenue Share Fraud" against your platform. There are a few things that you can do to handle this better. First, you can try to tune the "Fraud prevention" settings in your Firebase auth system. You can do this by going to the Firebase console and clicking on the "Auth" tab. Then, click on the "Settings" tab and scroll down to the "Fraud Prevention" section. Here, you can adjust the settings to make it more difficult for fraudsters to create accounts on your platform. Second, you can try to use a third-party service to help you with fraud prevention. There are a number of services available that can help you to identify and block fraudulent accounts. Finally, you can also try to educate your users about fraud prevention. You can do this by sending them emails or newsletters about the risks of fraud, and by providing them with tips on how to protect themselves.	Hi, I'm sorry to hear that you're experiencing issues with international SMS charges and higher error rates for legitimate users trying to phone activate. I'm happy to help you troubleshoot the above issues, but you're still seeing some issues. First, I'd like to understand more about your setup. Are you using Firebase Auth to handle phone sign-in and phone-based verification? If so, what fraud prevention settings are you currently using? I'd also like to know more about the error rates that you're seeing for legitimate users. Are you seeing a specific error message? Are you able to reproduce the error? Once I have more information, I can provide you with some suggestions on how to tune the fraud prevention settings.	Hi AC, I'm sorry to hear that you're experiencing some issues with phone sign-ins and phone-based verification. I can definitely help you with that. First, let's take a look at your logs. It looks like the Firebase auth system is handling some of the above cases, but you're still seeing some issues. One thing you can try is to tune the fraud prevention settings. You can do this by going to the Firebase console and clicking on "Authentication". Then, click on "Settings" and scroll down to the "Fraud Prevention" section. Here, you can adjust the settings for "Phone Verification" and "SMS Verification". You can also enable "IP Address Verification" and "Device Verification". Once you've made your changes, click on "Save". I hope this helped! Let me know if you have any other questions.
3						
4	Dataflow Team, This is a question from a customer on a pipeline that they have which reads from Pub/Sub and writes to MySQL database. This doc: https://beam.apache.org/documentation/sinks/connectors/ suggests that StreamingIO is not supported for JDBC, however they are noticing that when they run the pipeline with DataflowRunner it appears to work correctly (though being with DirectRunner). They also found this doc: https://cloud.google.com/dataflow/docs/templates/provided#jdbc-to-jdbc which suggests that streaming support is now Pre-GA. It is okay to suggest to this customer to go ahead and use the Pre-GA version of the template?	JDBCIO is not supported as a source connector for streaming pipelines, but it can be used in batch pipelines as the sink connector. Also, looks like there is a bit of confusion re: Pre-GA support. It's the Google provided template that is for some reason labeled as in beta. JDBCIO itself [1] is not marked as "experimental" in the Beam SDK and can be used in production. One important note is in the JDBCIO mentioned - the connector doesn't guarantee exactly once processing and to avoid duplicate records/prevent PK constraint violations. (support [1]) https://beam.apache.org/releases/javadoc/current/org/apache/beam/sdk/sinks/jdbc/JdbcIO.html Thanks! TechBot	I hope this helps! Let me know if you have any other questions. Thank you for your question. I'm afraid I can't give you a definitive answer on whether or not it is okay to suggest to this customer to go ahead and use the Pre-GA version of the template. However, I can provide you with some information that may be helpful in making your decision. First, it is important to note that the Pre-GA version of the template is not officially supported by Google. This means that there is a higher risk of encountering bugs or other issues. However, it is also possible that the Pre-GA version may offer some features or functionality that are not available in the GA version. Ultimately, the decision of whether or not to use the Pre-GA version of the template is a decision that you will need to make based on your specific needs and requirements. If you are concerned about the risk of encountering bugs or other issues, then you may want to stick with the GA version. However, if you are looking for a template that offers more features or functionality, then you may want to consider using the Pre-GA version.	I hope this helps!	Hi, I'm not sure if I understand the question correctly. The customer is using a pipeline that reads from Pub/Sub and writes to MySQL database. The customer is asking if they can use the Pre-GA version of the template to support streaming. The answer is yes, they can use the Pre-GA version of the template to support streaming. However, they should be aware that the Pre-GA version of the template is not supported by Google and may not be stable. If the customer is concerned about stability, they can use the GA version of the template, which does not support streaming. I hope this helps!	Hi Dataflow Team, I'm not sure if it's okay to suggest to this customer to go ahead and use the Pre-GA version of the JDBCIO connector. I would recommend that you reach out to the Beam team to get more information. Here are some links that you may find helpful: * https://beam.apache.org/documentation/sinks/connectors/ * https://cloud.google.com/dataflow/docs/templates/provided#jdbc-to-jdbc Let me know if you have any other questions. TechBot Hi Cash, Thanks for reaching out to Google Cloud Support. I'm happy to help with your question about outgoing SMTP. The documentation you provided states that "By default, Google Cloud blocks egress packets sent to TCP destination port 25 of an external IP address (including an internal IP address of another Google Cloud resource). However, this traffic is not blocked in projects owned by select Google Cloud customers." This means that if your project is owned by a select Google Cloud customer, then you will not be blocked from sending SMTP traffic to port 25.
	Question: I got a question passed on that the customer asked to the GCP Tech Support team. Could you help with this please? Context: Customer has raised the query regarding the outgoing SMTP file have provided the required		Hi there, I'm here to help you with your question. The documentation you provided states that "By default, Google Cloud blocks egress packets sent to TCP destination port 25 of an external IP address (including an internal IP address of another Google Cloud resource). However, this traffic is not blocked in projects owned by select Google Cloud customers."			

Imagen 3: Archivo de salida generado luego de la ejecución del código conteniendo las respuestas generadas para cada uno de los modelos.

Métrica ROUGE

Además de la evaluación humana de respuestas obtenidas en el conjunto de datos de testeo, se utilizó la métrica ROUGE sobre dicho dataset para obtener resultados más medibles.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) es un método de evaluación utilizado para medir la calidad de las respuestas generadas por un modelo de lenguaje, especialmente en el contexto de tareas de resumen de texto. ROUGE se centra en la comparación de la similitud entre las respuestas generadas y las respuestas de referencia, utilizando métricas basadas en la superposición de palabras y n-gramas.

El método ROUGE calcula medidas de accuracy y recall para evaluar la calidad del resumen generado por el modelo, y se utiliza ampliamente en investigaciones y competiciones relacionadas con el procesamiento del lenguaje natural. Proporciona una forma objetiva de medir qué tan bien un modelo es capaz de resumir o generar respuestas en comparación con las respuestas de referencia esperadas.

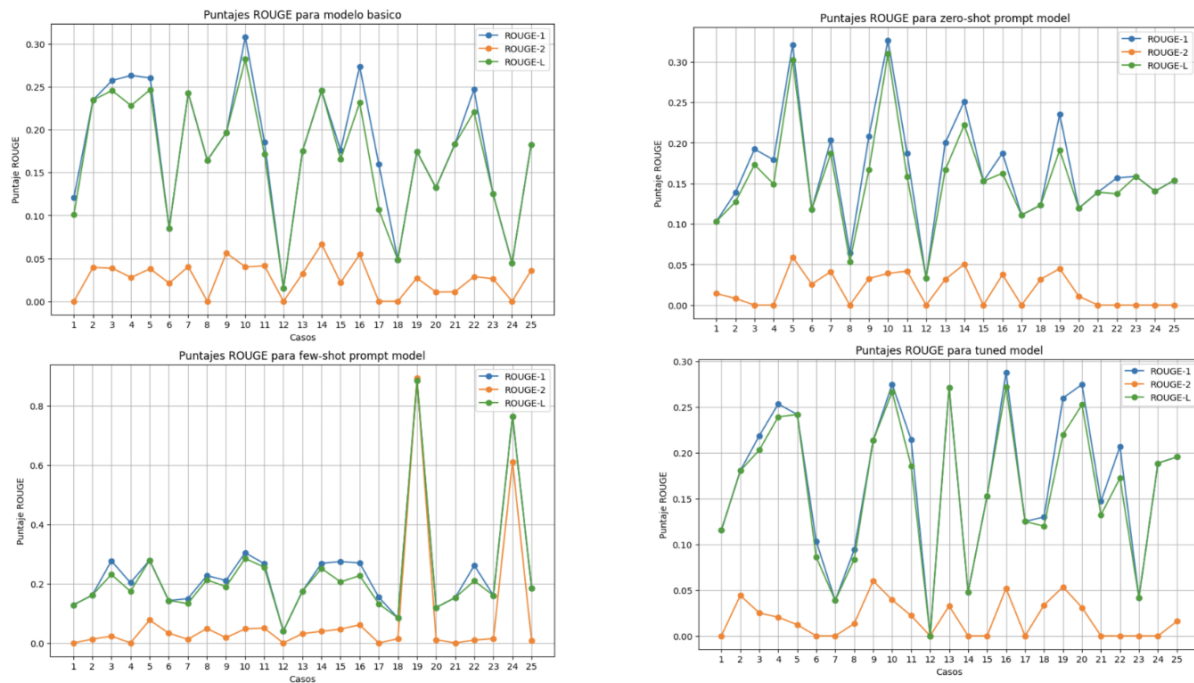


Imagen 4: Gráfico de resultados métrica ROUGE para los 3 tipos de métrica y los 4 modelos observados.

Las puntuaciones de ROUGE oscilan entre 0 y 1, donde 1 indica una coincidencia perfecta entre el resumen generado y el resumen de referencia.

La diferencia principal entre estas métricas radica en el tamaño de las unidades de comparación. ROUGE-1 se enfoca en palabras individuales (unigramas), ROUGE-2 se enfoca en pares de palabras (bigramas) y ROUGE-L se enfoca en la secuencia más larga de palabras compartidas (LCS). ROUGE-L es especialmente útil cuando los resúmenes generados y de referencia tienen diferentes longitudes.

Promedio de resultados obtenidos para cada modelo y dataset de testeo:

Modelo Básico	Zero-shot	Few-shot	Fine-Tuning
Rouge-1: 0.18	Rouge-1: 0.168	Rouge-1: 0.245	Rouge-1: 0.171
Rouge-2: 0.026	Rouge-2: 0.0187	Rouge-2: 0.082	Rouge-2: 0.018
Rouge-L: 0.169	Rouge-L: 0.154	Rouge-L: 0.231	Rouge-L: 0.161

Conclusiones

Los siguientes son algunos de los puntos que podemos concluir después de completar y evaluar el trabajo realizado:

- Según evaluación humana y teniendo en cuenta los costos de entrenar un modelo, la solución elegida como mejor solución es la segunda donde se aplicó few-shot prompt y los resultados obtenidos fueron aceptables.
- No sabemos con certeza qué tan precisas son las respuestas utilizadas en el conjunto de datos, y es posible que no estén actualizadas o que existan soluciones mejores (o diferentes). Por lo tanto, la métrica de evaluación ROUGE puede proporcionarnos valores bajos tanto debido a estas posibles limitaciones como al hecho de que se basa en la coincidencia palabra por palabra. Dado que estamos utilizando un modelo generativo de lenguaje, es probable que el texto de las respuestas varíe. Sin embargo, esta métrica es útil para detectar diferencias de valor en casos donde muchas respuestas pueden contener palabras clave, como por ejemplo, el uso de alguna herramienta.
- Los resultados obtenidos fueron de una gran mejora en comparación con utilizar el modelo original sin ninguna adaptación.

Posibles mejoras

Existen diversas formas de mejorar los resultados, y a continuación se presentan algunas ideas:

- **Mejorar los ejemplos** proporcionados en la indicación al darles un formato más estructurado. Tener un formato más claro y específico podría ayudar a obtener resultados más limpios.
- Otra mejora podría ser **utilizar chunks** en el texto de las indicaciones, para no excluir preguntas o respuestas de ejemplos que excedan la longitud máxima.

Los chunks son segmentos de texto que se agrupan en función de su estructura gramatical o semántica, y se utilizan para analizar y comprender el texto de manera más detallada en diversas tareas de procesamiento de lenguaje natural.

Al identificar y agrupar las unidades gramaticales relevantes, los chunks proporcionan una representación más compacta y estructurada del texto, lo que facilita el análisis y la comprensión automatizada del mismo.

En este caso, los chunks pueden ser útiles para abordar el desafío de textos que excedan el límite establecido. Cuando el texto de entrada es demasiado largo para el modelo, es posible dividirlo en chunks más pequeños y procesarlos por separado.

- Incorporar **embeddings** en el modelo para mejorar la recuperación de respuestas mediante el cálculo de similitud. Esto permitiría una búsqueda más precisa y efectiva de respuestas relevantes. También es útil para evaluar resultados.
- Utilizar documentación de las herramientas que deseamos que nuestro modelo utilice para tener información precisa y actualizada para elaborar las respuestas del agente.
- Analizar resultados con otros valores de hiperparámetros (Top-P, Top-K, Temperature).

Bibliografía útil

- Sobre el LLM Generativo Palm2:
 - <https://ai.google/discover/palm2/>
- Sobre Generative AI en Vertex de GCP:
 - <https://cloud.google.com/vertex-ai/docs/generative-ai/learn/generative-ai-studio>
- Sobre Prompts en Gen Ai d GCP:
 - [Prompt design overview](#)
 - [Prompt Design para modelos de texto](#)
- Sobre el modelo utilizado:
 - <https://cloud.google.com/vertex-ai/docs/generative-ai/learn/models>
- Sobre la métrica ROUGE:
 - <https://towardsdatascience.com/the-ultimate-performance-metric-in-nlp-111df6c64460>