

인사하고 영상 스크립트 제공하고...

2024-08-17 오전 11:10:58 - AI Copilot: 🗨️: 안녕하세요! AIPilotSmarteasy에서 온 교수 Na입니다. 텍스트 분석과 이해를 도와드리겠습니다. 오늘은 Google Gemini 모델의 파인 튜닝에 대한 흥미로운 텍스트를 가져오셨네요! 이 텍스트는 Google Gemini 모델을 사용자가 원하는 방식으로 응답하도록 훈련시키는 방법을 단계별로 자세히 설명하는 유튜브 비디오의 스크립트로 보입니다.

****핵심 내용 분석:****

* ****Google Gemini 모델 소개:**** 저렴한 비용으로 100개 이상의 언어를 지원하는 대규모 언어 모델입니다.

* ****파인 튜닝의 필요성:**** 기본 모델은 사용자의 특정 요구 사항에 맞춰져 있지 않으므로, 원하는 결과를 얻으려면 파인 튜닝이 필요합니다.

* ****파인 튜닝 과정:**** Google Cloud 설정, 데이터 준비, 모델 훈련, 모델 평가, 애플리케이션과의 통합 등 단계별로 설명합니다.

* ****코드 예제 제공:**** Python 코드를 사용하여 각 단계를 자세히 보여줍니다.

* ****장점 강조:**** 저렴한 비용, 사용자 정의 데이터를 사용한 모델 튜닝, 애플리케이션과의 손쉬운 통합 등 Google Gemini 모델의 이점을 강조합니다.

****전반적으로 이 텍스트는 Google Gemini 모델을 처음 접하는 사람들에게 유용한 정보를 제공합니다.**** 특히, 파인 튜닝을 통해 모델을 자신의 필요에 맞게 조정하는 방법을 배우고자 하는 개발자들에게 유용합니다.

이 텍스트에 대한 더 자세한 분석을 원하시면 알려주세요! 예를 들어 특정 코드 부분이나 모델 학습 과정에 대한 질문이 있으면 언제든지 물어보세요.

2024-08-17 오전 11:10:58 - Human Pilot: 전체 글은 몇 부분으로 나눌 수 있을까요?

2024-08-17 오전 11:10:58 - AI Copilot: 🗨️: 좋은 질문입니다! 이전에 분석했던 Google Gemini fine-tuning에 대한 긴 텍스트를 몇 부분으로 나누어 분석하면 더 효과적일 것 같네요. 내용에 따라 다음과 같이 크게 세 부분으로 나눌 수 있습니다.

1. ****도입부:**** Google Gemini 모델과 파인 튜닝에 대한 간략한 소개, 그리고 이 비디오 튜토리얼에서 다룰 내용을 설명하는 부분입니다.
2. ****파인 튜닝 단계 설명:**** Google Cloud 설정부터 모델 훈련, 평가, 애플리케이션 통합까지 각 단계를 Python 코드 예제와 함께 자세히 설명하는 부분입니다.
3. ****결론 및 추가 정보:**** 파인 튜닝의 장점을 다시 한번 강조하고, 앞으로 이어질 비디오 시리즈에 대한 기대감을 높이는 부분입니다.

각 부분을 더 세분화하여 분석할 수도 있습니다. 예를 들어, 두 번째 부분인 '파인 튜닝 단계 설명'은 Google Cloud 설정, 데이터 준비, 모델 훈련, 모델 평가, 애플리케이션과의 통합 등 각 단계별로 다시 나눌 수 있습니다.

어떤 부분부터 자세히 살펴볼까요? 😊

2024-08-17 오전 11:10:58 - Human Pilot: 각 부분에 대해 분석해 주세요.

2024-08-17 오전 11:10:58 - AI Copilot: 🗨️: 좋습니다! 이전에 나눈 Google Gemini fine-tuning 텍스트의 세 부분에 대해 각각 자세히 분석해 보겠습니다.

1. 도입부 분석:

* **주제 소개:** Google Gemini 모델과 fine-tuning이라는 주제를 명확하게 제시하며 시작합니다. 특히 "this is amazing"이라는 표현을 사용하여 청중의 흥미와 기대감을 높입니다.

* **파인 튜닝의 목적:** 모델을 훈련시켜 사용자가 예상하는 방식으로 응답하도록 만드는 것이라고 명확히 설명합니다.

* **간단한 예시 사용:** 숫자 55를 입력했을 때, 일반적인 설명 대신 56이라는 결과를 얻고 싶은 예시를 통해 이해를 돕습니다.

* **튜토리얼 내용 소개:** Google Gemini fine-tuning 설정 방법, 데이터 준비 및 훈련, 평가, 애플리케이션 통합 방법을 배울 수 있다고 안내합니다.

2. 파인 튜닝 단계 설명 분석:

* **단계별 설명:** Google Cloud 설정, 데이터 준비, 모델 훈련, 모델 평가, 애플리케이션과의 통합 등 각 단계를 논리적인 순서로 설명합니다.

* **코드 예제 활용:** 각 단계를 설명하면서 Python 코드 예제를 함께 제시하여 실제 적용 방법을 명확하게 보여줍니다.

* **세부 정보 제공:** 모델 이름, 파일 이름, 매개변수 설정 방법 등 구체적인 정보를 제공하여 사용자가 튜토리얼을 따라 하기 쉽도록 돕습니다.

* **요약 제공:** 각 단계 설명 후 핵심 내용을 요약하여 복습을 돕고, 다음 단계로 자연스럽게 연결합니다.

3. 결론 및 추가 정보 분석:

* **장점 요약:** 저렴한 비용으로 Google Gemini 모델을 자신의 데이터로 fine-tuning하여 애플리케이션에 활용할 수 있다는 점을 다시 강조합니다.

* **추가 정보 제공:** 유튜브 채널 구독, 알림 설정, 좋아요 클릭을 유도하여 사용자와의 소통을 지속하려는 모습을 보입니다.

* **추후 콘텐츠 예고:** 유사한 내용의 비디오를 계속 제작할 것이라고 언급하여 사용자의 기대감을 높입니다.

전반적으로 이 텍스트는 Google Gemini 모델 파인 튜닝에 대한 유익하고 따라 하기 쉬운 튜토리얼입니다. 각 부분이 유기적으로 연결되어 있으며, 명확한 설명과 코드 예제를 통해 사용자의 이해를 돕습니다. 😊

이 텍스트 분석에 대해 더 궁금한 점이 있으시면 무엇이든 물어보세요!

2024-08-17 오전 11:10:58 - Human Pilot: 각 부분을 처음부터 단계별로 하나씩 분석해 주세요.

2024-08-17 오전 11:10:58 - AI Copilot: 🙌: 좋습니다! Google Gemini fine-tuning 텍스트를 처음부터 단계별로 하나씩 꼼꼼하게 분석해 보겠습니다. 마치 탐험가가 새로운 지역을 탐험하듯이, 텍스트 속으로 함께 여행을 떠나보는 건 어떨까요? 😊

1. 도입부 분석:

1.1. 흥미 유발:

* 텍스트는 "this is amazing"이라는 문장으로 시작합니다. 이는 단순한 정보 전달을 넘어, 청중의 호기심을 자극하고 긍정적인 분위기를 조성하는 효과적인 방법입니다. 마치 친구에게 흥미로운 이

야기를 시작하기 전에 "이거 진짜 대박인데!"라고 말하는 것과 같은 효과를 냅니다.

****1.2. 주제 소개:****

* 이어서 "now we are going to see about Google Gemini fine-tuning"이라는 문장을 통해 본격적으로 주제를 소개합니다. Google Gemini라는 최첨단 기술을 활용하는 "fine-tuning"이라는 개념을 제시함으로써, 청중에게 전문적인 지식을 전달하고자 하는 화자의 의도를 엿볼 수 있습니다.

****1.3. 파인 튜닝 의미 설명:****

* "which means we are going to train the model and make sure that it responds as expected"라는 문장은 "fine-tuning"의 의미를 명확하게 설명합니다. 즉, 모델을 학습시켜 사용자가 기대하는 방식으로 응답하도록 만드는 과정임을 알 수 있습니다.

****1.4. 구체적인 예시 제시:****

* "before fine-tuning when you give a number like 55 this will automatically generate some text randomly which is not what I want"라는 문장은 fine-tuning이 필요한 이유를 설명하기 위해 구체적인 예시를 제시합니다. 숫자 55를 입력했을 때, 모델이 임의의 텍스트를 생성하는 것은 사용자가 원하는 결과가 아니라는 점을 보여줍니다.

****1.5. 원하는 결과 제시:****

* "I want whenever I type 55 it's supposed to be 56 and it should be incremental without any other text"라는 문장은 사용자가 원하는 결과를 명확하게 제시합니다. 즉, 55를 입력하면 56이라는 숫자가 순차적으로 증가하며 출력되기를 기대하는 것입니다.

****1.6. 문제 제기 및 해결 방안 제시:****

* "so how can we do this how can we train the model to respond just like this"라는 질문을 통해 앞서 제시된 문제에 대한 해결 방안을 모색합니다. 이는 청중에게 궁금증을 유발하고, 다음 내용에 대한 기대감을 높이는 역할을 합니다.

****1.7. 튜토리얼 내용 요약:****

* "by the end of this video you will learn how you can set up Google Gemini fine tuning and how you can prepare the data and fine-tune with that data then how you can evaluate and finally how you can integrate that with your own application"라는 문장은 이 튜토리얼을 통해 배우게 될 내용을 요약해서 보여줍니다.

****1.8. 튜토리얼 내용 구체화:****

* "that's exactly what we're going to see today let's get started"라는 문장은 앞서 요약된 튜토리얼 내용을 다시 한번 강조하며, 본격적인 시작을 알립니다.

이처럼 도입부는 Google Gemini fine-tuning이라는 주제를 소개하고, 독자의 흥미를 유발하며 튜

토리얼 내용에 대한 기대감을 높이는 중요한 역할을 합니다. 다음 단계도 계속해서 분석해볼까요?
😊

2024-08-17 오전 11:10:58 - Human Pilot: 다음 단계

2024-08-17 오전 11:10:58 - AI Copilot: 🤖: 좋아요! 이제 Google Gemini fine-tuning 텍스트 탐험의 두 번째 지역으로 떠나볼까요? 두 번째 부분은 본격적으로 Google Gemini fine-tuning을 어떻게 수행하는지 단계별로 설명하는 부분입니다. 마치 친절한 가이드가 안내하는 지도처럼, 텍스트는 독자들을 한 단계씩 안내합니다.

****2. 파인 튜닝 단계 설명 분석:****

****2.1. Google Cloud 설정:****

* "hi everyone I'm really excited to show you about Google Gemini fine tuning"이라는 활기찬 인사와 함께 시작합니다. 마치 강연장에 들어온 청중들을 향해 열정적으로 인사를 건네는 연사의 모습이 떠오르는군요!

* "there's a huge price drop for Gemini 1.5 flash model and it's only 7 cents for a million token that is really cheap and with context catching it's much more cheaper"라는 문장은 Google Gemini 1.5 flash 모델의 저렴한 가격을 강조합니다. 특히 "huge price drop", "really cheap", "much more cheaper"와 같은 표현을 통해 가격 경쟁력을 강조하며 청중을 설득하고 있습니다.

* "this is really cheap compared to GPT 4o mini and you are able to train this model for free and you are able to integrate that with your own application with reduced cost with 100 plus additional languages and it's available for all developers"라는 문장은 경쟁 모델인 GPT 4o mini와 비교하여 Google Gemini 모델의 장점을 부각합니다. 무료 훈련, 저렴한 비용, 100개 이상의 언어 지원, 모든 개발자에게 제공 등 Google Gemini 모델의 매력적인 특징을 나열하며 청중의 흥미를 당기고 있습니다.

계속해서 다음 단계 분석을 이어갈까요? 😊

2024-08-17 오전 11:10:58 - Human Pilot: 다음 단계

2024-08-17 오전 11:10:58 - AI Copilot: 🤖: 좋아요! Google Gemini fine-tuning 텍스트 탐험을 계속 이어가 봅시다. 이전 단계에서 Google Gemini 모델의 장점과 가격 경쟁력을 살펴봤으니, 이제 실제로 Google Cloud에서 어떻게 설정하는지 알아봐야겠죠?

****2.2. Google Cloud 계정 생성 및 OAuth 설정:****

* "first you need to create an account in Google Cloud then go to OAuth authentication then create credentials oAuth client ID then on this choose desktop app giving a desktop app name and then click create" 이 부분은 Google Cloud에서 fine-tuning을 시작하기 위한 첫 번째 단계, 즉 계정 생성과 OAuth 설정에 대한 간략한 설명입니다. 마치 여행 가이드가 여행 전 필요한 준비물을 꼼꼼하게 알려주는 것처럼, 텍스트는 Google Cloud 사용을 위한 필수적인 절차를 안내합니다.

****2.3 Client ID 및 Secret 다운로드:****

* "this will automatically create the client ID and client Secrets which you can just download and place it in your same folder and I have renamed that as client secret. Json" 이 문장은 생성된 Client ID와 Client Secret을 다운로드하여 저장하는 방법을 설명합니다. 특히 파일 이름을 "client secret. Json"으로 변경했다는 구체적인 정보를 제공함으로써, 독자가 튜토리얼을 따라 하기

쉽도록 돕습니다.

2.4. Google Cloud 인증:

* "next we need to authenticate just going to type this command I'll provide all the information in the description below Google Cloud oath application default login and then the client ID file that is the client secret. Json file which you have just downloaded and the Scopes are these with generative language tuning after this click enter" 이 부분은 Google Cloud에 인증하기 위한 과정을 설명합니다. 필요한 명령어와 함께, client secret. Json 파일을 사용하고, "generative language tuning" 범위를 지정해야 한다는 중요한 정보를 제공합니다. 마치 여행 가이드가 여행지에서 필요한 언어를 알려주는 것처럼, 텍스트는 Google Cloud와 소통하기 위한 필수적인 코드를 제공합니다.

2.5. 인증 과정:

* "this will automatically ask you to authenticate so click the authenticate and click continue and allow access" 이 문장은 인증 과정을 자세히 설명합니다. 인증 요청이 나타나면 "authenticate" 버튼을 클릭하고, "continue"를 눌러 접근 권한을 허용해야 합니다.

2.6 OAuth 동의 화면 설정:

* "sometime you might also need to set up your oat consent screen in Google Cloud" 이 문장은 경우에 따라 OAuth 동의 화면을 설정해야 할 수도 있다는 추가 정보를 제공합니다.

이처럼 텍스트는 Google Cloud 설정 과정을 단계별로 자세히 설명하고, 필요한 정보를 명확하게 제공하여 독자들이 튜토리얼을 따라 하기 쉽도록 돕습니다. 다음 단계도 흥미진진한 내용이 기다리고 있을 것 같네요. 계속해서 살펴볼까요? 😊

2024-08-17 오전 11:10:58 - Human Pilot: 다음 단계

2024-08-17 오전 11:10:58 - AI Copilot: 🗨️: 네, Google Cloud 설정을 마쳤으니 이제 본격적으로 Google Gemini 모델을 다루기 위한 준비를 해야겠죠?

2.7. 필요한 패키지 설치:

* "now in your terminal pip install Google generative Ai and then Rich and then click enter" 이 부분에서는 Google Gemini 모델을 사용하기 위해 필요한 Python 패키지를 설치하는 과정을 설명합니다. "pip install" 명령어를 사용하여 "Google generative Ai"와 "Rich" 패키지를 설치하도록 안내합니다. 마치 여행 전에 필요한 여행 가이드북을 준비하는 것처럼, 텍스트는 Google Gemini 모델을 사용하기 위한 필수적인 도구를 설치하도록 안내합니다.

2.8. 패키지 설명:

* "Google generative AI is the main package rich is just for printing in clear format after this click enter" 이 문장은 설치한 패키지에 대한 간략한 설명입니다. "Google generative AI"는 Google Gemini 모델을 사용하기 위한 핵심 패키지이며, "Rich"는 출력 결과를 보기 좋게 정리해 주는 역할을 합니다.

2.9. Fine-tuning 전 모델 테스트:

* "next let's create a file called before.py to see how it look like before fine tuning" 이 문장부터는 fine-tuning을 하기 전에 모델이 어떻게 동작하는지 테스트하는 과정을 설명합니다. "before.py"라는 파일을 생성하여 fine-tuning 전의 모델 동작을 확인하도록 안내합니다. 마치 여행 전에 목적지에 대한 정보를 미리 알아보는 것처럼, 텍스트는 fine-tuning 전 모델의 기본적인 동작을 파악하도록 유도합니다.

자, 이제 Google Gemini 모델을 사용하기 위한 준비를 마쳤습니다. 다음 단계에서는 fine-tuning 전에 모델이 어떻게 동작하는지 확인하고, 본격적인 fine-tuning 과정으로 들어가겠죠? 😊

2024-08-17 오전 11:10:58 - Human Pilot: fine-tuning 과정은 전체를 하나로 분석해 주세요.

2024-08-17 오전 11:10:58 - AI Copilot: 🤖: 좋습니다! Google Gemini fine-tuning 과정 전체를 하나로 묶어서 자세히 살펴보고겠습니다. 이 부분은 마치 요리 레시피처럼, Google Gemini 모델을 원하는 방식으로 튜닝하기 위한 재료와 조리 과정을 상세하게 설명하고 있습니다.

2.10. Fine-tuning 전 모델 동작 확인:

* "inside the file I just call this gen a generator model function and provide the model name that that is Gemini 1.5 fl001 and pause these queries 55 1 2 3 4 55 4 and then these all values and let's see how it's going to look like before fine tuning" 이 문장은 "before.py" 파일 내부에 작성할 코드를 설명합니다. "gen a generator model" 함수를 호출하여 "Gemini 1.5 fl001" 모델을 사용하고, 55, 1 2 3 4 55 4 등의 쿼리를 입력하여 fine-tuning 전 모델의 응답을 확인하는 과정입니다.

* "so in your terminal Python before.py and then click enter and now as you can see here we asked 55 and the result is 55 is a number with more information about 55 but this is not what we want we just want the response to be just 56" 이 부분은 터미널에서 "before.py" 파일을 실행한 결과를 보여줍니다. 55를 입력했을 때, 모델은 55에 대한 추가 정보를 제공하지만, 사용자가 원하는 것은 단순히 56이라는 숫자입니다.

* "and similarly when I entered 1 2 3 4 55 my expectation is 1 2 3 4 5 6 rather than explaining what the number is about so same as I see here I don't want to explain the same number again what it is I just want it to respond with the next number like this" 이 문장은 1 2 3 4 55를 입력했을 때도 마찬가지로 1 2 3 4 5 6이라는 결과를 기대하지만, 모델은 숫자에 대한 설명을 제공한다는 점을 지적합니다. 즉, fine-tuning 전 모델은 사용자의 의도와 다르게 동작한다는 것을 보여줍니다.

2.11. Fine-tuning 데이터 준비 및 훈련:

* "so we have completed the step of setting up now step number two preparing data and fine tuning let's create a file called app.py and let's open it inside the file import Google do generate AI as gen AI then import random time and then import print" 이 부분은 fine-tuning을 위한 두 번째 단계인 데이터 준비 및 훈련 과정을 설명합니다. "app.py"라는 파일을 생성하고, 필요한 패키지를 import합니다.

* "next just printing the list of available models like this and then we need to choose the base model so we going to choose the Gemini 1.5 flash so just printing the base model for our reference next we are going to set the name for our new model which is going to be created and then gen a.create tuned model there we are providing the source model that is the base model name then this is the training data you can also save this data in a separate file and then import it here directly but in my case to keep this tutorial quick and short and for easy for everyone to understand I'm just keeping all the training data in the same file so here you can

see the text input is one if the input is one the L language model should output two if it is three it should be four so similarly I'm teaching the L language model how it should respond based on the input so now you can create your own custom data and place it like this to train the model then providing the ID that's the name of the model the batch size learning rate and the epoke it's better to keep this number higher for better training you can even give 50 or 100" 이 부분은 "app.py" 파일 내부에 작성할 코드와 함께 fine-tuning 과정을 자세히 설명합니다. 사용 가능한 모델 목록을 출력하고, 기본 모델로 "Gemini 1.5 flash"를 선택합니다. 새로운 모델의 이름을 설정하고, "gen a. create tuned model" 함수를 사용하여 fine-tuning을 시작합니다. 훈련 데이터는 코드 내에 직접 작성하거나 별도의 파일로 저장하여 import 할 수 있습니다. 훈련 데이터는 입력값과 원하는 출력값 쌍으로 구성되며, 이를 통해 모델이 사용자의 의도에 맞게 학습됩니다. 모델 ID, 배치 크기, 학습률, 에포크 수 등의 매개변수를 설정하고, 에포크 수를 높게 설정할수록 더 나은 훈련 결과를 얻을 수 있습니다.

2.12. Fine-tuning된 모델 상태 확인:

* "so now after creating this tuned model Next Step checking the Model status by using the get tuned model function next printing the Model status waiting for the training to finish by waiting 30 seconds that's it literally this much amount of code and you are able to train your large language model as a quick summary we can directly name this model here as Gemini 1.5 flash even without all this then give a name for your model then this will train the model and you can get the status of the model training here that's it now I'm going to run this code in your terminal python p and then click enter first you can see the list of available models among that we are using this Gemini 1.5 fl001 tuning model we can also tune this 1.5 Pro 001 model because we can see that this tag create tuned model tag is tagged on both this flash model and also in this pro model so in our instance we are using the flash model and you can see here the fine tuning is happening you can see that the training is now complete" 이 부분은 fine-tuning된 모델의 상태를 확인하는 과정을 설명합니다. "get tuned model" 함수를 사용하여 모델 상태를 가져오고 출력합니다. 30초 동안 기다려 훈련이 완료될 때까지 기다립니다. 간단한 코드로 대규모 언어 모델을 훈련시킬 수 있다는 점을 강조하고, 모델 이름을 직접 지정하여 훈련할 수 있다는 점을 다시 한번 설명합니다. 터미널에서 코드를 실행하여 사용 가능한 모델 목록을 확인하고, "Gemini 1.5 fl001" 모델을 사용하여 fine-tuning을 진행합니다. 훈련 과정이 완료되면 "training is now complete" 메시지가 출력됩니다.

이처럼 Google Gemini fine-tuning 과정은 Google Cloud 설정, 데이터 준비, 모델 훈련, 상태 확인 등 여러 단계를 거치지만, 텍스트는 각 단계를 명확하게 설명하고 코드 예제를 제공하여 독자들이 쉽게 따라 할 수 있도록 안내합니다. 마치 숙련된 요리사의 레시피처럼, 텍스트는 Google Gemini 모델을 fine-tuning하는 데 필요한 모든 정보를 제공합니다. 😊

다음 단계 분석도 흥미로울 것 같네요. 계속해서 살펴볼까요?

2024-08-17 오전 11:10:58 - Human Pilot: 다음은 글 끝까지 남은 부분을 한 번에 분석해 주세요.

2024-08-17 오전 11:10:58 - AI Copilot: 🎉: 좋습니다! 이제 Google Gemini fine-tuning 텍스트 탐험의 마지막 지역에 도착했습니다. 이 부분은 fine-tuning된 모델을 평가하고 실제 애플리케이션에 통합하는 방법, 그리고 마무리 인사로 구성되어 있습니다. 마치 여행의 마지막 날, 여행지에서 얻은 경험을 정리하고 기념품을 챙기는 것처럼, 텍스트는 fine-tuning된 모델을 활용하는 방법과 마무리 메시지를 전달합니다.

3. 모델 평가 및 애플리케이션 통합:

* **3.1. Fine-tuning된 모델 목록 확인:** "next we are going to see how you can evaluate the Creator model to do that first you need to know the list of tuned models available so using this code we are going to check the list of tuned models so in your terminal python tuned models.py and here you can see the list of tuned models which I tried before so the latest model is this one so I'm going to copy the name of the model" fine-tuning된 모델을 평가하기 전에, 먼저 사용 가능한 fine-tuning된 모델 목록을 확인해야 합니다. 터미널에서 "python tuned models.py" 명령어를 실행하여 이전에 fine-tuning한 모델 목록을 확인하고, 가장 최근 모델을 선택합니다.

* **3.2. 모델 평가:** "then go to the file and we are going to call the model by using geni generator model and providing the model name here that's it and then to evaluate the process I'm going to ask a few questions the first one is 55 then a number like this then four then quadre then three and similarly we're going to ask questions and find how the response is going to be like so now I'm going to run this code in your terminal python eval py and you can see the response here as per our fine-tuning when the query is 55 the answer is 56 and when the number is like this it's again 56 when there four it's five and similarly other results now fine tuning is successful" 선택한 모델을 호출하여 평가를 진행합니다. 55, 4 등의 쿼리를 입력하여 fine-tuning된 모델의 응답을 확인하고, fine-tuning이 성공적으로 이루어졌는지 평가합니다.

* **3.3. 애플리케이션 통합:** "now the final step how you can integrate that with your own application the latest model the same command as we saw in the evaluation using the Gen generator model function and providing the model name that is a fine-tuned model name as you can see here then providing the query and then generate content to generate the response that's it as simple as that now you are able to tune Gemini model with your own custom data and then use that model in your own application with low cost" 마지막 단계는 fine-tuning된 모델을 실제 애플리케이션에 통합하는 방법을 설명합니다. 평가 단계에서 사용한 것과 동일한 명령어를 사용하여 fine-tuning된 모델을 호출하고, 쿼리를 입력하여 응답을 생성합니다. 저렴한 비용으로 사용자 정의 데이터를 사용하여 Google Gemini 모델을 fine-tuning하고, 이를 자신의 애플리케이션에 활용할 수 있다는 점을 강조합니다.

4. 결론 및 추가 정보:

* **4.1. 튜토리얼 요약 및 마무리:** "I'm really excited about this I'm going to create more videos similar to this so stay tuned I hope you like this video do like share and subscribe and thanks for watching" 화자는 Google Gemini fine-tuning에 대한 튜토리얼을 마무리하며, 유사한 주제의 비디오를 계속 제작할 예정임을 알립니다.

* **4.2. 시청자 참여 유도:** "stay tuned", "do like share and subscribe"와 같은 표현을 사용하여 시청자의 지속적인 참여를 유도합니다.

이처럼 Google Gemini fine-tuning 텍스트는 모델 평가, 애플리케이션 통합, 결론, 추가 정보 제공 등의 단계를 거치며, 독자들에게 Google Gemini 모델을 효과적으로 활용하는 방법을 제시합니다. 마치 여행의 끝에서 다음 여정을 기약하며 작별 인사를 나누는 것처럼, 텍스트는 Google Gemini fine-tuning에 대한 튜토리얼을 마무리합니다. 😊

이 텍스트 분석에 대한 질문이나 궁금한 점이 있으면 언제든지 저에게 물어보세요!