google / gemma-2-2b-jpn-it
like 203
Follow
Google
31.2k

Text Generation
Transformers
Safetensors
japanese
gemma2
conversational
text-generation-inference
License: gemma

Model card
Files and versions
Community 38
Train
Deploy
Use this model

Gated model You have been granted access to this model

Edit model card

Gemma 2 JPN model card

Resources and Technical Documentation:

- Responsible Generative AI Toolkit
- Gemma 2 JPN on Kaggle
- Gemma 2 JPN on Hugging Face

Terms of Use: Terms

Authors: Google

Model Information

Summary description and brief definition of inputs and outputs.

Description

Gemma is a series of best-in-class open models and draws inspiration and technological lineage from the Gemini family of models. They are text-to-text, decoder-only large language models with open weights. Gemma models are well-suited for a variety of text generation tasks, including question answering, summarization, and reasoning.

Gemma 2 JPN is a Gemma 2 2B model fine-tuned on Japanese text. It supports the Japanese language with the same level of performance as Gemma 2 27B models of English only queries on Gemma 2.

Usage

Below we share some code snippets on how to get quickly started with running the model. First, install the Transformers library with:

```
pip install -U transformers
```

Then, copy the snippet from the section that is relevant for your use.

Running with the pipeline API

```
import torch
from transformers import pipeline

pipe = pipeline(
    "text-generation",
    model="google/gemma-2-2b-jpn-it",
    model_kwargs={"torch_dtype": torch.bfloat16},
    device="cuda",  # replace with "mps" to run on a Mac device
)

messages = [
    {"role": "user", "content": "マンハッタン計画について日本を含めて教えてください。"},
]

outputs = pipe(messages, return_full_text=False, max_new_tokens=256)
assistant_response = outputs[0]["generated_text"].strip()
print(assistant_response)
```

▶ Example output

It can also be used for translation, as follows:

```
translation_input_text = f"Translate the following pose from Japanese to English:\n\n{assistant_response}"
messages = [
    {"role": "user", "content": translation_input_text},
]

outputs = pipe(messages, return_full_text=False, max_new_tokens=1024)
translated_response = outputs[0]["generated_text"].strip()
print(translated_response)
```

▶ Example output

Running the model on a single / multi GPU

```
# pip install accelerate
from transformers import AutoTokenizer, AutoModelForCausalLM
import torch

tokenizer = AutoTokenizer.from_pretrained("google/gemma-2-2b-jpn-it")
model = AutoModelForCausalLM.from_pretrained(
    "google/gemma-2-2b-jpn-it",
    device_map="auto",
    torch_dtype=torch.bfloat16,
)

messages = [
    {"role": "user", "content": "マンハッタン計画について日本を含めて教えてください。"},
]
inputs = tokenizer.apply_chat_template(messages, return_tensors="pt", add_generation_prompt=True, return_dict=True).to(model.device)

outputs = model.generate(**inputs, max_new_tokens=256)
generated_text = tokenizer.batch_decode(outputs)
input_text = inputs['input_ids'].to(model.device)[0]
print(generated_text[len(input_text):])
```

Running the model on a GPU using different precisions

The native weights of this model were exported in bfloat16 precision.

You can also use float16 if you skip the dtype, but no precision increase will occur (model weights will just be upcasted to float32). See examples below.

- Upcasting to torch.float32

```
# pip install accelerate
from transformers import AutoTokenizer, AutoModelForCausalLM

tokenizer = AutoTokenizer.from_pretrained("google/gemma-2-2b-jpn-it")
model = AutoModelForCausalLM.from_pretrained(
    "google/gemma-2-2b-jpn-it",
    device_map="auto",
)

messages = [
    {"role": "user", "content": "マンハッタン計画について日本を含めて教えてください。"},
]
inputs = tokenizer.apply_chat_template(messages, return_tensors="pt", add_generation_prompt=True, return_dict=True).to(model.device)

outputs = model.generate(**inputs, max_new_tokens=256)
generated_text = tokenizer.batch_decode(outputs)
input_text = inputs['input_ids'].to(model.device)[0]
print(generated_text[len(input_text):])
```

Inputs and outputs

- **Input:** Text string, such as a question, a prompt, or a document to be summarized.
- **Output:** Generated Japanese-language text in response to the input, such as an answer to a question, or a summary of a document.

Model Data

Data used for model training and how the data was processed.

Training Dataset

These models were trained on a dataset of text data that includes a wide variety of sources, totaling 8 trillion tokens. Here are the key components:

- Web Documents: A diverse collection of web text ensures the model is exposed to a broad range of linguistic styles, topics, and vocabulary. Primarily English language content.
- Code: Exposing the model to code helps it to learn the syntax and patterns of programming languages, which improves its ability to generate code or understand code-related questions.
- Mathematics: Training on mathematical text helps the model learn logical reasoning, symbolic representation, and to address mathematical queries.
- Instruction data set: large-scale and high-quality Japanese and multilingual instruction data.

The combination of these diverse data sources is crucial for training a powerful language model that can handle a wide variety of different tasks and text formats.

Data Preprocessing

Here are the key data cleaning and filtering methods applied to the training data:

- CSAM Filtering: Rigorous CSAM (Child Sexual Abuse Material) filtering was applied at multiple stages in the data preparation process to ensure the exclusion of harmful and illegal content.
- Sensitive Data Filtering: As part of making Gemma pre-trained models safe and reliable, we used automated techniques to filter out certain personal information and other sensitive data from training sets.
- Additional methods: Filtering based on content quality and safety in line with our policies.

Implementation Information

Details about the model internals.

Hardware

Gemma was trained using the latest generation of Tensor Processing Unit (TPU) hardware (TPUv5p).

Training large language models requires significant computational power. TPUs, designed specifically for matrix operations common in machine learning, offer several advantages in this domain:

- Performance: TPUs are specifically designed to handle the massive computations involved in training LLMs. They can speed up training considerably compared to CPUs.
- Memory: TPUs often come with large amounts of high-bandwidth memory, allowing for the handling of large models and batch sizes during training. This can lead to better model quality.
- Scalability: TPU Pods (large clusters of TPUs) provide a scalable solution for handling the growing complexity of large foundation models. You can distribute training across multiple TPU devices for faster and more efficient processing.
- Cost-effectiveness: In many scenarios, TPUs can provide a more cost-effective solution for training large models compared to CPU-based infrastructure, especially when considering the time and resources saved due to faster training.

These advantages are aligned with Google's commitments to operate sustainably.

Software

Training was done using JAX and ML Pathways.

JAX allows researchers to take advantage of the latest generation of hardware, including TPUs, for faster and more efficient training of large models.

ML Pathways is Google's latest effort to build artificially intelligent systems capable of generalizing across multiple tasks. This is specially suitable for foundation models, including large language models like these ones.

Together, JAX and ML Pathways are used as described in the paper about the Gemini family of models; "the 'single-controller' programming model of Jax and Pathways allows a single Python process to orchestrate the entire training run, dramatically simplifying the development workflow."

Evaluation

To assess the quality of this model, we collected a diverse set of Japanese prompts and evaluated performance using an LLM-as-a-judge approach against GPT-3.5. The rating system is based on a 7-point scale assessments, which are MuchWorseThan, BetterThan, SlightlyBetterThan, AboutTheSame, SlightlyWorseThan, WorseThan, MuchWorseThan associated with the numerical scores (-3, -2, -1, 0, 1, 2, 3), where (-3, -2, -1) is respectively. We also tracked the ability of the model to answer in the correct language. For a Japanese prompt, the model should typically answer in Japanese rather than defaulting to English.

Benchmark	Gemma 2-2IT	Gemma 2-JPN
Preference on GPT-3.5	4.23 ± 0.05	5.03 ± 0.04
Language correctness	86.47%	99.24%

Ethics and Safety

Ethics and safety evaluation approach and results.

Evaluation Approach

Our evaluation methods include structured evaluations and internal red-teaming testing of relevant content policies. Red-teaming was conducted by a number of different teams, each with different goals and human evaluation metrics. These models were evaluated against a number of different categories relevant to ethics and safety, including:

- Text-to-Text Content Safety: Human evaluation on prompts covering safety policies including child sexual abuse and exploitation, harassment, violence and gore, and hate speech.
- Text-to-Text Representational Harms: Benchmark against relevant academic datasets.
- Memorization: Automated evaluation of memorization of training data, including the risk of personally identifiable information exposure.
- Large-scale harm: Tests for "dangerous capabilities," such as chemical, biological, radiological, and nuclear (CBRN) risks.

Usage and Limitations

These models have certain limitations that users should be aware of.

Intended Usage

Open Large Language Models (LLMs) have a wide range of applications across various industries and domains. The following list of potential uses is not comprehensive. The purpose of this list is to provide contextual information about the possible use-cases that the model creators considered as part of model training and development.

- Content Creation and Communication
  - Text Generation: These models can be used to generate creative text formats such as poems, scripts, code, marketing copy, and email drafts.
  - Chatbots and Conversational AI: Power conversational interfaces for customer service, virtual assistants, or interactive applications.
  - Text Summarization: Generate concise summaries of a text corpus, research papers, or reports.
- Research and Education
  - Natural Language Processing (NLP) Research: These models can serve as a foundation for researchers to experiment with NLP techniques, develop algorithms, and contribute to the advancement of the field.
  - Language Learning Tools: Support interactive language learning experiences, aiding in grammar correction or providing writing practice.
  - Knowledge Exploration: Assist researchers in exploring large bodies of text by generating summaries or answering questions about specific topics.

Limitations

- Training Data
  - The quality and diversity of the training data significantly influence the model's capabilities. Biases or gaps in the training data can lead to limitations in the model's responses.
  - The scope of the training dataset determines the subject areas the model can handle effectively.
- Context and Task Complexity
  - LLMs are better at tasks that can be framed with clear prompts and instructions. Open-ended or highly complex tasks might be challenging.
  - A model's performance can be influenced by the amount of context provided (longer context generally leads to better outputs, up to a certain point).
- Language Ambiguity and Nuance
  - Natural language is inherently complex. LLMs might struggle to grasp subtle nuances, sarcasm, or figurative language.
- Factual Accuracy
  - LLMs generate responses based on information they learned from their training datasets, but they are not knowledge bases. They may generate incorrect or outdated factual statements.
- Common Sense
  - LLMs rely on statistical patterns in language. They might lack the ability to apply common sense reasoning in certain situations.

Ethical Considerations and Risks

The development of large language models (LLMs) raises several ethical concerns. In creating an open model, we have carefully considered the following:

- Bias and Fairness
  - LLMs trained on large-scale, real-world text data can reflect socio-cultural biases embedded in the training material. These models underwent careful scrutiny, input data pre-processing described and posterior evaluations reported in this card.
- Misinformation and Misuse
  - LLMs can be misused to generate text that is false, misleading, or harmful.
  - Guidelines are provided for responsible use with the model, see the Responsible Generative AI Toolkit.
- Transparency and Accountability:
  - This model card summarizes details on the models' architecture, capabilities, limitations, and evaluation processes.
  - A responsibly developed open model offers the opportunity to share innovation by making LLM technology accessible to developers and researchers across the AI ecosystem.

Risks identified and mitigations:

- Perpetuation of biases: It's encouraged to perform continuous monitoring (using evaluation metrics, human review) and the exploration of de-biasing techniques during model training, fine-tuning, and other use cases.
- Generation of harmful content: Mechanisms and guidelines for content safety are essential. Developers are encouraged to exercise caution and implement appropriate content safety safeguards based on their specific product policies and application use cases.
- Misuse for malicious purposes: Technical limitations and developer and end-user education can help mitigate against malicious applications of LLMs. Educational resources and reporting mechanisms for users to flag misuse are provided. Prohibited uses of Gemma models are outlined in the Gemma Prohibited Use Policy.
- Privacy violations: Models were trained on data filtered for removal of PII (Personally Identifiable Information). Developers are encouraged to adhere to privacy regulations with privacy-preserving techniques.

Benefits

At the time of release, this family of models provides high-performance open large language model implementations designed from the ground up for Responsible AI development compared to similarly sized models.