LLM Math

Contents

Evaluating chains that know how to do math.

```
# Comment this out if you are NOT using tracing
import os
os.environ["LANGCHAIN_HANDLER"] = "langchain"
```

```
from langchain.evaluation.loading import load_dataset
dataset = load_dataset("llm-math")
```

```
Downloading and preparing dataset json/LangChainDatasets--llm-math to
/Users/harrisonchase/.cache/huggingface/datasets/LangChainDatasets___json/LangChain
Datasets--llm-math-
509b11d101165afa/0.0.0/0f7e3662623656454fcd2b650f34e886a7db4b9104504885bd462096cc7a
9f51...
```

```
Dataset json downloaded and prepared to
/Users/harrisonchase/.cache/huggingface/datasets/LangChainDatasets___json/LangChain
Datasets--llm-math-
509b11d101165afa/0.0.0/0f7e3662623656454fcd2b650f34e886a7db4b9104504885bd462096cc7a
9f51. Subsequent calls will reuse this data.
```

Setting up a chain

Now we need to create some pipelines for doing math.

```
from langchain.llms import OpenAI
from langchain.chains import LLMMathChain
```

```
llm = OpenAI()
```

```
chain = LLMMathChain(llm=llm)
```

```
predictions = chain.apply(dataset)
```

```
numeric_output = [float(p['answer'].strip().strip("Answer: ")) for p in
predictions]
```

```
correct = [example['answer'] == numeric_output[i] for i, example in
enumerate(dataset)]
```

```
sum(correct) / len(correct)
```

```
1.0
```

```
for i, example in enumerate(dataset):
    print("input: ", example["question"])
    print("expected output :", example["answer"])
    print("prediction: ", numeric_output[i])
```

```
input:  5
expected output : 5.0
prediction:  5.0
input:  5 + 3
expected output : 8.0
prediction:  8.0
input:  2^3.171
expected output : 9.006708689094099
prediction:  9.006708689094099
input:    2 ^3.171
expected output : 9.006708689094099
prediction:  9.006708689094099
input:  two to the power of three point one hundred seventy one
expected output : 9.006708689094099
prediction:  9.006708689094099
input:  five + three squared minus 1
expected output : 13.0
```

Skip to main content

```
expected output : 57269.07
prediction:  57269.07
input:  two thousand ninety seven times twenty seven point thirty one
expected output : 57269.07
prediction:  57269.07
input:  209758 / 2714
expected output : 77.28739867354459
prediction:  77.28739867354459
input:  209758.857 divided by 2714.31
expected output : 77.27888745205964
prediction:  77.27888745205964
```