

Использование модели llama2 для разработки бота

1. Информация о модели.

В ходе проверки модели ollama с подключённой моделью llama2 использовалась версия на 7 миллионов параметров (см. Рисунок 1). При тестировании запросов с таким количеством параметров не возникало проблем с пониманием контекста и распознаванием запросов на русском языке.

```
>>> /show info
Model details:
Family          llama
Parameter Size  7B
Quantization Level Q4_0
```

Рисунок 1 – Информация о модели.

Уровень квантования Q4_0 означает, что каждый промпт в модели был сжат до размера 4 бит (при 16 битах без сжатия).

Во время процесса дообучения новые промпты изначально будут весить также 16 бит. Но при необходимости их можно будет также сжать до уровня Q4_0.

2. Шаблон входных данных, используемый в модели.

Шаблон, по которому llama2 обрабатывает контекстное окно выглядит следующим образом:

```
“[INST] <<SYS>>{{ .System }}</SYS>>
{{ .Prompt }} [/INST]”
```

Вместо {{ .System }} записывается инструкция для модели, чтобы она понимала, как следует реагировать на {{ .Prompt }}, т. е. на сообщение пользователя. Пример, как может выглядеть такой шаблон:

```
“<s>[INST] <<SYS>> You are a helpful, respectful and honest assistant.
Always answer as helpfully as possible, while being safe. Your answers should not
include any harmful, unethical, racist, sexist, toxic, dangerous, or illegal content. Please
ensure that your responses are socially unbiased and positive in nature. If a question
does not make any sense, or is not factually coherent, explain why instead of answering
```

something not correct. If you don't know the answer to a question, please don't share false information. <</SYS>> There's a llama in my garden 🤪 What should I do? [/INST]”

“[/INST]”, “<<SYS>>” и т. д. – параметры, которые показывают модели откуда и докуда считывать данные.

Таким образом, чтобы модель отвечала на русском, достаточно добавить в {{.System}} указание: “Говори на русском языке” (см. Рисунок 2 – Рисунок 4):

```
>>> Объясни, что значит .System и .Prompt в шаблоне llama2
In the template file for Ollama, there are two placeholders that you see repeated throughout the file: '.System'
and '.Prompt'. Here's what they represent:

1. **.System**: This placeholder represents the entire input data for the Ollama model. It is used as a wildcard
to capture any input data, regardless of its format or structure. For example, if you have a sentence with
multiple words, '.System' will capture all of them as separate entities.
2. **.Prompt**: This placeholder represents a specific entity in the input data that the model is trying to
classify. It is used to indicate the type of entity that the model should be looking for, such as a person,
organization, location, etc. The '.Prompt' placeholder is often followed by a list of categories or labels that
the model can recognize for that particular entity.
```

Рисунок 2 – Ответ модели без указания в System.

```
>>> /set system "Говори на русском языке."
Set system message.
>>> /show system
"Говори на русском языке."
```

Рисунок 3 – Изменение System.

```
>>> Объясни, что значит .System и .Prompt в шаблоне llama2

.System и .Prompt – это параметры шаблона LLMA, которые определяют поведение шаблона при выводе значений.

.System – это параметр, который определяет тему шаблона. Он может быть изменен для того, чтобы задать другой язык или другую тему для вывода значений. Например, если .System = "rus", то шаблон б
работать на русском языке.

.Prompt – это параметр, который определяет значение по умолчанию для входного сообщения. Он может быть изменен для того, чтобы задать другое значение по умолчанию. Например, если .Prompt = "Hello"
шаблон будет выводить сообщение "Hello" при выполнении команды.
```

Рисунок 4 – Ответ модели после изменения.

Также это можно использовать, например, чтобы в конце ответа выводить сообщение об ограниченной ответственности бота за полученный результат (см. Рисунок 5):

```
>>> /set system "Выводи в конце сообщения "Данный бот не несёт ответственности за предоставленный результат. Проверьте данные на сайте ВУЗа!"
Set system message.
>>> /show system
"Выводи в конце сообщения "Данный бот не несёт ответственности за предоставленный результат. Проверьте данные на сайте ВУЗа!"
>>> Что такое море?
Море – это огромное водное пространство, которое занимает большую часть поверхности нашей планеты. Море представляет собой часть океана, который покрывает более 70% поверхности Земли. Оно играет важную роль в поддержании климата и жизни на Земле, а также обеспечивает множество ресурсов для человечества, таких как рыба, нефть и газ.
Море состоит из воды, которая постоянно движется в результате океанских течений и приливов. Оно также имеет разную температуру и солёность в зависимости от глубины и местоположения. Море может быть спокойным или бурным, и в его пределах можно найти множество различных экосистем, таких как коралловые рифы, мангровые болота и дельты рек.
Море также играет важную роль в транспорте и торговле, а также является источником пищи и энергии для человечества. Однако, море также может быть угрозой для человеческой жизни и экономики в результате таких явлений, как цунами, штормы и повышенный уровень океанического загрязнения.
В целом, море – это огромное и разнообразное водное пространство, которое играет важную роль в жизни нашей планеты и requires careful management and conservation to ensure its continued health and productivity.
Обратите внимание, что в конце сообщения есть заключение, которое указывает, что данный бот не несет ответственности за предоставленный результат. Пожалуйста, проверьте данные на сайте ВУЗа!
```

Рисунок 5 – Изменение System и пример вывода.

Кроме того, модель не умеет запоминать историю запросов в качестве отдельных промптов, поэтому весь предыдущий контекст запоминается в “текущий промт”. То есть фактически шаблон при диалоге будет выглядеть следующим образом:

```
<s>[INST] <<SYS>>
{{ system_prompt }}
<</SYS>>
{{ user_msg_1 }} [/INST] {{ model_answer_1 }} </s><s>[INST] {{
user_msg_2 }} [/INST]
```

Поэтому при расчёте пользовательских запросов надо также рассчитывать и уточняющие вопросы от ученика, а также сколько токенов использует для ответа Llama2. Стоит заметить, что в llama есть возможность ограничивать количество токенов, которые используются для генерации ответа модели, через параметры модели.

3. Исследование размера контекстного окна.

В Llama2 контекст по умолчанию позволяет распознавать до 4096 токенов. Один токен не равен одному символу и не имеет фиксированной величины. По информации в интернете (см. <https://habr.com/ru/articles/768844/>) в среднем размер

токена составляет 4 символа при использовании английского языка и 2 символов на русском.

Нам важен второй факт, проверим его на практике (см. Рисунок 6).

```
В этом промпте 26 символов.  
  
total duration:      681.530411ms  
load duration:      197.07µs  
prompt eval count:   48 token(s)  
prompt eval duration: 443.261ms  
prompt eval rate:    108.29 tokens/s  
eval count:          14 token(s)  
eval duration:       235.911ms  
eval rate:           59.34 tokens/s
```

Рисунок 6 – Вывод LLM статистики.

Подключив статистику для каждого ответа с помощью команды “/set verbose” было обнаружено, что сообщение “В этом промпте 26 символов.” занимает 14 токенов, то есть 1 токен на русском рассчитывается примерно как 2 символа, следовательно, информация подтверждается.

Таким образом, максимальный контекст для Llama2 составляет приблизительно 8192 символов. Это приблизительно 3 страницы текста на формата A4 с размером шрифта 14.

4. Альтернативная модель с большим контекстом.

Существует модель, основанная на Llama2, но имеющая больший контекст. Yarn-llama2 расширяет его до 128 тысяч токенов. Таким образом, yarn-llama2 имеет возможность использовать контекстное окно в размере примерно 256 тысяч символов. Однако, обработка такого большого контекста замедлит работу модели.