**Инструкция по конвертации модели в gguf формат!**

**Шаг 1. Установите Git и Python**

1. Скачайте и установите **Git for Windows**:
   * [Ссылка для скачивания](https://git-scm.com/).
   * Убедитесь, что Git добавлен в PATH во время установки (опция "Use Git from the command line").
2. Установите **Python** (рекомендуется версия 3.8+):
   * [Ссылка для скачивания](https://www.python.org/downloads/).
   * Во время установки выберите "Add Python to PATH".

**Шаг 2. Установите CMake и компилятор**

1. Скачайте **CMake** (для сборки llama.cpp):
   * [Ссылка для скачивания](https://cmake.org/download/).
2. Установите компилятор GCC для Windows:
   * Рекомендуется использовать [**MinGW-w64**](https://github.com/niXman/mingw-builds-binaries/releases):
     + Скачайте MinGW-w64 отсюда.
     + Убедитесь, что путь к bin добавлен в переменную окружения PATH.

**Шаг 3. Скачайте и соберите llama.cpp**

1. Клонируйте репозиторий llama.cpp:

bash

Копировать код

git clone https://github.com/ggerganov/llama.cpp.git

cd llama.cpp

1. Соберите проект:

bash

Копировать код

cmake .

make

* + Если make не работает, попробуйте:

bash

Копировать код

mingw32-make

После сборки исполняемый файл main.exe появится в папке llama.cpp.

**Шаг 4. Установите Python-зависимости**

Установите библиотеку **transformers**, которая потребуется для конвертации модели:

bash

Копировать код

pip install transformers huggingface-hub

**Шаг 5. Скачайте модель из Hugging Face**

1. Авторизуйтесь в Hugging Face через CLI:

bash

Копировать код

huggingface-cli login

1. Скачайте модель:

bash

Копировать код

huggingface-cli download meta-llama/Llama-3.2-3B-Instruct --revision hf --cache-dir ./models

**Шаг 6. Конвертируйте модель в GGUF**

Используйте Python-скрипт для конвертации:

1. Перейдите в папку llama.cpp:

bash

Копировать код

cd llama.cpp

1. Запустите скрипт:

bash

Копировать код

python convert.py \

--input-dir ./models/meta-llama/Llama-3.2-3B-Instruct \

--model-size 3B \

--output ./models/llama-3.2-3b-instruct.Q8\_0.gguf \

--quantize Q8\_0

* **--input-dir**: Укажите путь к модели.
* **--output**: Укажите имя выходного файла.
* **--quantize Q8\_0**: Используйте 8-битное квантование для уменьшения размера модели.

**Шаг 7. Проверьте модель**

После успешной конвертации проверьте модель:

1. Запустите main.exe с моделью:

bash

Копировать код

main.exe -m ./models/llama-3.2-3b-instruct.Q8\_0.gguf --ctx-size 4096

**Шаг 8. Используйте модель в вашем проекте**

Скопируйте готовый файл модели (llama-3.2-3b-instruct.Q8\_0.gguf) в папку проекта и укажите путь в коде:

python

Копировать код

MODEL\_PATH = "models/llama-3.2-3b-instruct.Q8\_0.gguf"

cd

python convert\_hf\_to\_gguf.py "F:/voice\_assistant/VoiceAssistant/saved\_mistral\_model " --outfile "F:/voice\_assistant/VoiceAssistant/saved\_mistral\_model/ Ministral-8B-Instruct-2410.Q8\_0.gguf" --outtype q8\_0 --model-name Ministral-8B-Instruct-2410

Ministral-8B-Instruct-2410