1. RAG- .
2. (, Llama, Mistral Falcon).
   : , , .
3. .
4. : , , /

5. colab ,

```
!pip install langchain langchain_community langchain-openai langchain_huggingface faiss-cpu s
```

```
                          0.0/2.5 MB ? eta -:--:--                         2.5/2.5 MB 88.1 MB
                          0.0/1.0 MB ? eta -:--:--                         1.0/1.0 MB 58.1 MB
                          60.9/60.9 kB 5.8 MB/s eta 0:00:00
                          30.7/30.7 MB 66.8 MB/s eta 0:00:00
                          76.1/76.1 MB 12.1 MB/s eta 0:00:00
                          580.2/580.2 kB 39.5 MB/s eta 0:00:00
                          1.2/1.2 MB 49.9 MB/s eta 0:00:00
                          363.4/363.4 MB 3.7 MB/s eta 0:00:00
                          13.8/13.8 MB 52.6 MB/s eta 0:00:00
                          24.6/24.6 MB 31.8 MB/s eta 0:00:00
                          883.7/883.7 kB 54.3 MB/s eta 0:00:00
                          664.8/664.8 MB 2.0 MB/s eta 0:00:00
                          211.5/211.5 MB 5.6 MB/s eta 0:00:00
                          56.3/56.3 MB 13.4 MB/s eta 0:00:00
                          127.9/127.9 MB 7.4 MB/s eta 0:00:00
                          207.5/207.5 MB 7.2 MB/s eta 0:00:00
                          21.1/21.1 MB 67.3 MB/s eta 0:00:00
                          50.9/50.9 kB 3.5 MB/s eta 0:00:00
```

```
import requests
from bs4 import BeautifulSoup
import json
import numpy as np
import pickle
from sentence_transformers import SentenceTransformer
from transformers import pipeline
from langchain.embeddings import HuggingFaceEmbeddings
from langchain_core.documents import Document
from langchain.vectorstores.faiss import FAISS
from langchain_community.document_loaders import WebBaseLoader
from langchain.text_splitter import RecursiveCharacterTextSplitter
```

```
WARNING:langchain_community.utils.user_agent:USER_AGENT environment variable not set, conside
```

```
from urllib.parse import urlparse

#
def fetch_links(url):
    links = []
    response = requests.get(url)
    soup = BeautifulSoup(response.text, 'html.parser')
    domain = urlparse(url).netloc

    for ul in soup.find_all('ul'):
        for li in ul.find_all('li'):
            link = li.find('a')
            if link and "href" in link.attrs:
                href = link.attrs["href"]
                if "/wiki" in href[:5]:
                    links.append(f"https://{domain}{href}")

    return links

# URL
url = 'https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%82%D0%B5%D0%B3%D0%BE%D1%80%D0%B8%D1%8F:
links = fetch_links(url)
```

```python
import os

#
os.environ["USER_AGENT"] = "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KH

loader = WebBaseLoader(links, show_progress=True)
docs = loader.load()
```

```python
#           ,
with open("docs.pickle", "wb") as f:
    pickle.dump(docs, f)
```

```python
#            ,
with open("docs.pickle", "rb") as f:
    docs = pickle.load(f)
```

```python
text_splitter = RecursiveCharacterTextSplitter(chunk_size=1000, chunk_overlap=100)
split_docs = text_splitter.split_documents(docs)
```

```python
from langchain_huggingface import HuggingFaceEmbeddings

model_name = "sentence-transformers/paraphrase-multilingual-mpnet-base-v2"
model_kwargs = {'device': 'cuda'}
encode_kwargs = {'normalize_embeddings': False}
embedding = HuggingFaceEmbeddings(model_name=model_name,
                                  model_kwargs=model_kwargs,
                                  encode_kwargs=encode_kwargs)
```

```python
vector_store = FAISS.from_documents(split_docs, embedding=embedding)
```

```python
#
with open("vector_store.pickle", "wb") as f:
    pickle.dump(vector_store, f)
```

```python
with open("vector_store.pickle", "rb") as f:
    vector_store = pickle.load(f)
```

, .

```python
encoder = SentenceTransformer('all-MiniLM-L6-v2')

def retrieve(query, top_k=2):
    query_vector = np.array(encoder.encode([query]))
    documents = vector_store.search(query, "similarity")
    return documents[:top_k]
```

```
modules.json:   0%|          | 0.00/349 [00:00<?, ?B/s]

config_sentence_transformers.json:   0%|          | 0.00/116 [00:00<?, ?B/s]

README.md:   0%|          | 0.00/10.5k [00:00<?, ?B/s]

sentence_bert_config.json:   0%|          | 0.00/53.0 [00:00<?, ?B/s]

config.json:   0%|          | 0.00/612 [00:00<?, ?B/s]

model.safetensors:   0%|          | 0.00/90.9M [00:00<?, ?B/s]

tokenizer_config.json:   0%|          | 0.00/350 [00:00<?, ?B/s]

vocab.txt:   0%|          | 0.00/232k [00:00<?, ?B/s]

tokenizer.json:   0%|          | 0.00/466k [00:00<?, ?B/s]

special_tokens_map.json:   0%|          | 0.00/112 [00:00<?, ?B/s]

config.json:   0%|          | 0.00/190 [00:00<?, ?B/s]
```

HuggingFace ( )

```
# from huggingface_hub import login
# from google.colab import userdata

# login(token = userdata.get("HF_TOKEN"))
```

```
from transformers import AutoModelForCausalLM, AutoTokenizer, GenerationConfig
import torch

# MODEL_NAME = "facebook/opt-1.3b"
# MODEL_NAME = "deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B"
MODEL_NAME = "IlyaGusev/saiga_llama3_8b"

model = AutoModelForCausalLM.from_pretrained(
    MODEL_NAME,
    load_in_8bit=True,
    torch_dtype=torch.bfloat16,
    device_map="cuda",
)
model.eval()

tokenizer = AutoTokenizer.from_pretrained(MODEL_NAME)
generation_config = GenerationConfig.from_pretrained(MODEL_NAME)
```

```
config.json:   0%|          | 0.00/689 [00:00<?, ?B/s]

The `load_in_4bit` and `load_in_8bit` arguments are deprecated and will be removed in the fu

model.safetensors.index.json:   0%|          | 0.00/23.9k [00:00<?, ?B/s]

Downloading shards:   0%|          | 0/4 [00:00<?, ?it/s]

model-00001-of-00004.safetensors:   0%|          | 0.00/4.98G [00:00<?, ?B/s]

model-00002-of-00004.safetensors:   0%|          | 0.00/5.00G [00:00<?, ?B/s]

model-00003-of-00004.safetensors:   0%|          | 0.00/4.92G [00:00<?, ?B/s]
```

```
model-00004-of-00004.safetensors:   0%|          | 0.00/1.17G [00:00<?, ?B/s]

Loading checkpoint shards:   0%|          | 0/4 [00:00<?, ?it/s]

generation_config.json:   0%|          | 0.00/277 [00:00<?, ?B/s]

tokenizer_config.json:   0%|          | 0.00/51.0k [00:00<?, ?B/s]

tokenizer.json:   0%|          | 0.00/9.09M [00:00<?, ?B/s]

special_tokens_map.json:   0%|          | 0.00/446 [00:00<?, ?B/s]
```

```python
gen_pipeline = pipeline("text-generation",
                        model=model,
                        tokenizer=tokenizer,
                        return_full_text=False)
```

```
Device set to use cuda:0
```

```python
def generate_response(query):
    relevant_texts = retrieve(query)
    context = ' '.join([t.model_dump()['page_content'] for t in relevant_texts])
    prompt = f"""                .
                                 .
                ,                 .
         : '''{context}'''
        : {query}
       :
    """
    response = gen_pipeline(prompt)
    return response[0]["generated_text"]
```

```python
import warnings
warnings.filterwarnings("ignore")
```

```
query = "                              ?"

answer = generate_response(query)
print(answer)
```

30 . ,

```
query = "           ?"

answer = generate_response(query)
print(answer)
```

- , ,

. ,
. , .

```
query = "               ?"

answer = generate_response(query)
print(answer)
```

- , , (
, , !'''

```
---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---
```

–––

–––

–––

–––

–––

–––

–––

–––

–––

–––

–––

–––

–––

–––

–––

–––

–––

–––

–––

–––

–––

–––

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

```
---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---
```

```
---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---
```

```
---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---
```

```
---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---
```

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

−−−

–––

–––

–––

–––

–––

–––

–––

–––

–––

–––

–––

–––

–––

–––

–––

–––

–––

–––

–––

–––

–––

```
---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---
```

```
---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---
```

```
---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---
```
