
council_analytics

Christoph Blattgerste

Dec 20, 2024

CONTENTS:

1	llm_rag package	3
1.1	Submodules	3
1.2	llm_rag.query module	3
1.3	llm_rag.web_app module	3
1.4	Module contents	3
2	preprocessing package	5
2.1	Submodules	5
2.2	preprocessing.download module	5
2.3	preprocessing.embedding module	5
2.4	preprocessing.extractor module	5
2.5	preprocessing.preprocessing module	5
2.6	Module contents	6
	Python Module Index	7
	Index	9

Add your content using reStructuredText syntax. See the [reStructuredText](#) documentation for details.

LLM_RAG PACKAGE

1.1 Submodules

1.2 llm_rag.query module

```
class llm_rag.query.RAG_LLM
    Bases: object
    configure_query_engine(index) → RetrieverQueryEngine
    display_prompt_dict(prompts_dict)
    huggingface_login()
    init_embedding_model(embed_name)
    init_llm_model(llm_name, token)
    load_index_storage(index_dir)
    query_rag_llm(user_query)
```

1.3 llm_rag.web_app module

1.4 Module contents

PREPROCESSING PACKAGE

2.1 Submodules

2.2 preprocessing.download module

`preprocessing.download.download_pdf(idx, verbose=False)`

`preprocessing.download.request_pdf(idx, verbose=False)`

Request the PDF file from the municipal council website. Returns the content of the file if it's a PDF, otherwise None.

2.3 preprocessing.embedding module

`preprocessing.embedding.initVectorStore(embedding_model)`

`preprocessing.embedding.initialize_embedding_model()`

`preprocessing.embedding.load_txt_files(directory)`

`preprocessing.embedding.load_txt_files_nextcloud(start_idx, end_idx)`

2.4 preprocessing.extractor module

`preprocessing.extractor.extract_text(doc)`

`preprocessing.extractor.save_text(text: str, fout: str)`

2.5 preprocessing.preprocessing module

`preprocessing.preprocessing.download_from_nextcloud(folder, filename)`

`preprocessing.preprocessing.parallel_process_pdf(args)`

A helper function to process the PDFs in parallel.

`preprocessing.preprocessing.process_pdf(idx, verbose=False)`

Process the PDF by downloading, uploading to Nextcloud, extracting text, and uploading the text file to Nextcloud.

```
preprocessing.preprocessing.upload_to_nextcloud(folder, filename, content, content_type='binary',  
                                                verbose=False)
```

Upload the given content (PDF or text data) to Nextcloud. If content_type is 'text', it will upload the content as text (encoded in utf-8), otherwise it will treat it as binary content.

2.6 Module contents

PYTHON MODULE INDEX

I

`llm_rag`, 3
`llm_rag.query`, 3

P

`preprocessing`, 6
`preprocessing.download`, 5
`preprocessing.embedding`, 5
`preprocessing.extractor`, 5
`preprocessing.preprocessing`, 5

C

`configure_query_engine()`
(*llm_rag.query.RAG_LLM method*), 3

D

`display_prompt_dict()` (*llm_rag.query.RAG_LLM method*), 3

`download_from_nextcloud()` (*in module preprocessing*), 5

`download_pdf()` (*in module preprocessing.download*), 5

E

`extract_text()` (*in module preprocessing.extractor*), 5

H

`huggingface_login()` (*llm_rag.query.RAG_LLM method*), 3

I

`init_embedding_model()` (*llm_rag.query.RAG_LLM method*), 3

`init_llm_model()` (*llm_rag.query.RAG_LLM method*), 3

`initialize_embedding_model()` (*in module preprocessing.embedding*), 5

`initVectorStore()` (*in module preprocessing.embedding*), 5

L

`llm_rag`
module, 3

`llm_rag.query`
module, 3

`load_index_storage()` (*llm_rag.query.RAG_LLM method*), 3

`load_txt_files()` (*in module preprocessing.embedding*), 5

`load_txt_files_nextcloud()` (*in module preprocessing.embedding*), 5

M

module

`llm_rag`, 3

`llm_rag.query`, 3

`preprocessing`, 6

`preprocessing.download`, 5

`preprocessing.embedding`, 5

`preprocessing.extractor`, 5

`preprocessing.preprocessing`, 5

P

`parallel_process_pdf()` (*in module preprocessing*), 5

`preprocessing`

module, 6

`preprocessing.download`

module, 5

`preprocessing.embedding`

module, 5

`preprocessing.extractor`

module, 5

`preprocessing.preprocessing`

module, 5

`process_pdf()` (*in module preprocessing*), 5

Q

`query_rag_llm()` (*llm_rag.query.RAG_LLM method*), 3

R

`RAG_LLM` (*class in llm_rag.query*), 3

`request_pdf()` (*in module preprocessing.download*), 5

S

`save_text()` (*in module preprocessing.extractor*), 5

U

`upload_to_nextcloud()` (*in module preprocessing*), 5