Test Documentation

For our automated testing setup, we make use of `PyTest`.

We write **unit tests** for the following modules:

- Paper data retrieval from ArXiv
- Research paper embedding process
- Retrieval-Augmented Generation (RAG) pipeline
- Front-end functionality
- Model fine-tuning

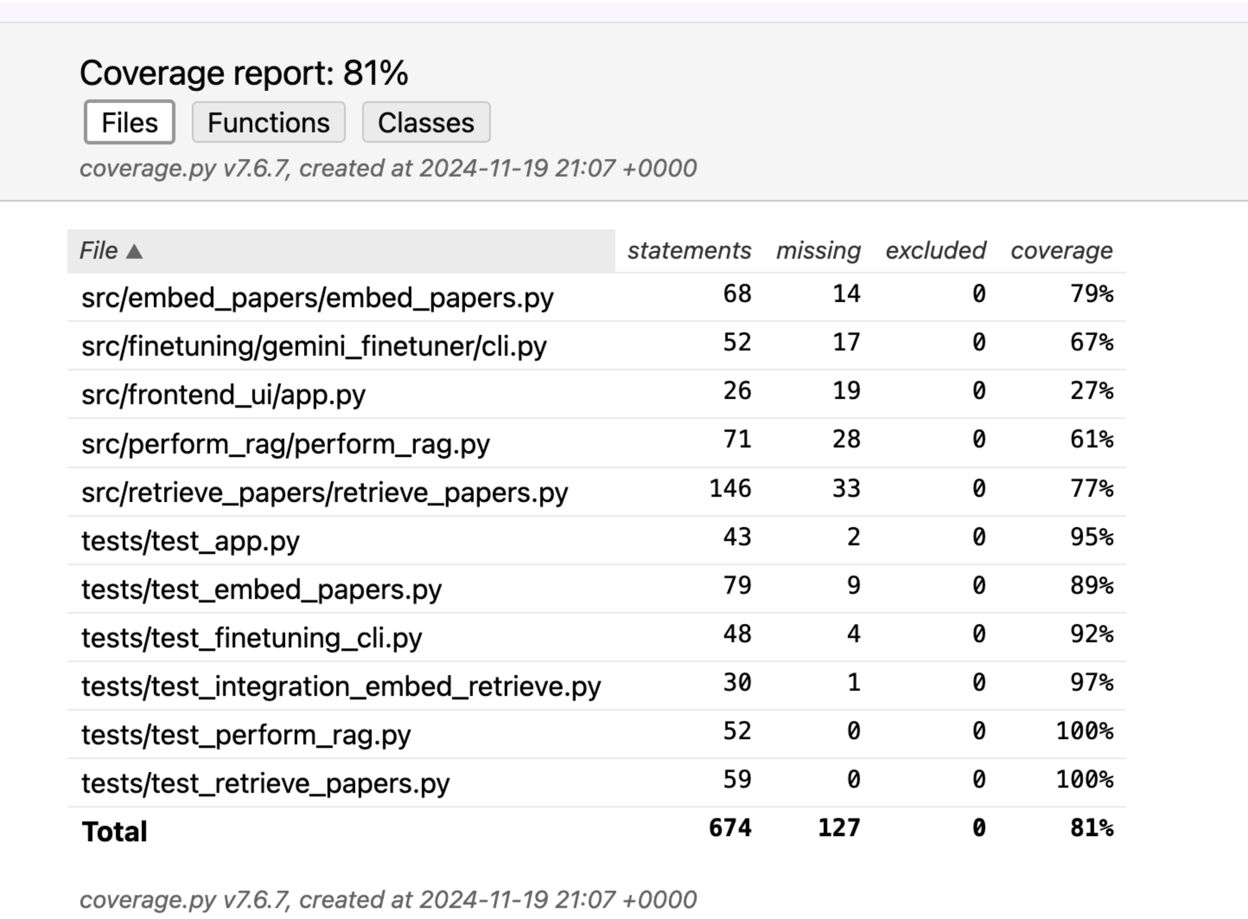
We design each module to operate independently in a modular structure. Any shared components are cloud-stored to minimize dependency conflicts. Due to this modular structure, there are not many possibilities for integration testing. Nonetheless, we write an **integration test** that verifies the seamless connection between data retrieval (ArXiv paper loading) and the embedding process for the same papers. Additionally, we write a **system test** which makes sure that the backend API is triggered correctly whenever a user submits a query via the front end.

For all testing, we use mocks to simulate endpoints such as the vector database and LLM API by using the `Mock` and `patch` modules.

All tests are automatically executed on each push to our GitHub repository via **GitHub Actions**.

Instructions for running manually:

1. Make sure you are in the directory `AC215_AIResearchForGood`/
2. Run `pytest --cov=src` (see below for sample output)



Coverage report: 81%

coverage.py v7.6.7, created at 2024-11-19 21:07 +0000

File ▲	*statements*	*missing*	*excluded*	*coverage*
src/embed_papers/embed_papers.py	68	14	0	79%
src/finetuning/gemini_finetuner/cli.py	52	17	0	67%
src/frontend_ui/app.py	26	19	0	27%
src/perform_rag/perform_rag.py	71	28	0	61%
src/retrieve_papers/retrieve_papers.py	146	33	0	77%
tests/test_app.py	43	2	0	95%
tests/test_embed_papers.py	79	9	0	89%
tests/test_finetuning_cli.py	48	4	0	92%
tests/test_integration_embed_retrieve.py	30	1	0	97%
tests/test_perform_rag.py	52	0	0	100%
tests/test_retrieve_papers.py	59	0	0	100%
Total	**674**	**127**	**0**	**81%**

coverage.py v7.6.7, created at 2024-11-19 21:07 +0000