```
.
├─Combining documents with different chain types and LLM chains____
│    ├─MapReduce Chain Loading and Combining____
│    │    ├─Question answering chain with sources loading and combining____
│    │    │    ├─■──Loading question answering with sources chain with multiple loader mappings and chains.____ ── Topic: 53
│    │    │    └─■──Loading and Combining Documents with Language Models for Summarizing and QA____ ── Topic: 71
│    │    └─Map Reduce Chain Loading Function____
│    │         ├─Document Refinement using LLM Chains____
│    │         │    ├─■──Combining Documents with Stuffing and LLM Chain in Python____ ── Topic: 97
│    │         │    └─BaseQAWithSourcesChain document handling and processing.____
│    │         │         ├─■──Question Answering with Sources over Documents Chain____ ── Topic: 60
│    │         │         └─■──Python class for chatbot with vector database and question generation____ ── Topic: 16
│    │         └─MapReduce chain implementation____
│    │              ├─■──MapReduceDocumentsChain document combination with chaining and mapping____ ── Topic: 12
│    │              └─■──MapReduce Chain Loading Function____ ── Topic: 95
│    └─LLMBashChain document examples and related keywords____
│         ├─Bash operations and language modeling chain implementation____
│         │    ├─LLMSummarizationCheckerChain document samples____
│         │    │    ├─■──Working with SQL databases in Python using SQLDatabaseChain____ ── Topic: 46
│         │    │    └─Document processing with LLMSummarizationCheckerChain____
│         │    │         ├─■──Implementation of Program-Aided Language Models with PALChain class and related prompts and assertio ── Topic: 31
│         │    │         └─■──LLMSummarizationCheckerChain class and its functionality____ ── Topic: 93
│         │    └─LLMBashChain - interpreting prompts and executing bash code____
│         │         ├─■──LLMMathChain - Python code execution for math prompts____ ── Topic: 92
│         │         └─■──Bash execution with LLMBashChain____ ── Topic: 80
│         └─■──MRKLChain implementation with ChainConfig and API integration____ ── Topic: 59
└─Code organization and structure in Python including several classes related to self-hosted embedding
     ├─Code organization and improvement suggestions for a class definition.____
     │    ├─Code Loading and Organization Best Practices____
     │    │    ├─Web scraping Hacker News webpage titles____
     │    │    │    ├─Loading files using unstructured in Python____
     │    │    │    │    ├─Unstructured file loading with retry and partitioning capabilities.____
     │    │    │    │    │    ├─■──Retry Decorator for OpenAI API Calls____ ── Topic: 45
     │    │    │    │    │    └─Unstructured File Loading and Partitioning____
     │    │    │    │    │         ├─■──Unstructured File Loader for Partitioning Files in Various Formats____ ── Topic: 25
     │    │    │    │    │         └─■──Loading files with Unstructured package in different modes (Python code).____ ── Topic: 26
     │    │    │    │    └─PDF manipulation in Python with pypdf, pdfminer, fitz and pymupdf libraries____
     │    │    │    │         ├─■──PDF file loading and text extraction using PyMuPDF and PDFMiner____ ── Topic: 69
     │    │    │    │         └─■──Extracting Text from Paged PDF using PyPDF and PDFMiner____ ── Topic: 96
     │    │    │    └─Extracting Hacker News Webpage Information using WebBaseLoader and BeautifulSoup.____
     │    │    │         ├─■──Web scraping Hacker News with BeautifulSoup and WebBaseLoader____ ── Topic: 21
     │    │    │         └─■──Web Scraping for College Confidential and Lyrics Websites____ ── Topic: 76
     │    │    └─Code organization and structure in various Python modules____
     │    │         ├─Compliments on clear and structured codebase with good use of type hints for memory handling and con
     │    │         │    ├─Implementation of ReAct paper using ReActChain with examples in Python____
     │    │         │    │    ├─■──Implementation of ReAct paper in ReActChain agent with OpenAI LLC model and tools____ ── Topic: 101
     │    │         │    │    └─In-memory Docstore for Efficient Lookup and Exploration____
     │    │         │    │         ├─■──Document Store Exploration with DocstoreExplorer____ ── Topic: 87
     │    │         │    │         └─■──InMemoryDocstore for Storing and Searching Documents with AddableMixin____ ── Topic: 61
     │    │         │    └─Compliments on Code Readability and Organization in Python Codebase.____
     │    │         │         ├─Memory Handling and Conversation Management____
     │    │         │         │    ├─Memory Conversation Summarizer Implementation____
     │    │         │         │    │    ├─Memory and Conversation Summarization in AI-assisted dialogues.____
     │    │         │         │    │    │    ├─■──Purpose of ChatPromptValue class in chat.py____ ── Topic: 30
     │    │         │         │    │    │    └─■──Memory management and conversation summarization in AI chatbot system.____ ── Topic: 6
     │    │         │         │    │    └─■──Implementation of Chain class with CallbackManager and Memory attributes.____ ── Topic: 52
     │    │         │         │    └─Potential bugs and suggestions for loading LLM, few-shot prompts, and examples from JSON and YAML fi
     │    │         │         │         ├─Code structure and organization tips for loading examples and templates from files in Python.____
     │    │         │         │         │    ├─Compliments on code structure and organization____
     │    │         │         │         │    │    ├─■──Loading few-shot prompts from config with prefix and suffix templates____ ── Topic: 34
     │    │         │         │         │    │    └─Code organization and structure for creating chat prompt templates____
     │    │         │         │         │    │         ├─■──Chat prompt template and message prompt templates for generating chatbot prompts.____ ── Topic: 8
     │    │         │         │         │    │         └─■──Purpose of `_load_prompt_from_file` function in loading.py module.____ ── Topic: 13
     │    │         │         │         │    └─■──Function for Loading a Chain of LLM Checkers from a Configuration Dictionary.____ ── Topic: 3
     │    │         │         │         └─Documented class definitions for tools used in handling API requests, including OpenSearchVectorSear
     │    │         │         │              ├─Handling API requests using tools such as RequestsPostTool and OpenSearchVectorSearch____
     │    │         │         │              │    ├─Python requests wrapper for making HTTP requests with various tools and methods____
     │    │         │         │              │    │    ├■─DeepInfra API token and text generation model wrapper____ ── Topic: 41
     │    │         │         │              │    │    └─RequestsWrapper and BaseRequestsTool for making HTTP requests (POST, GET, PATCH, DELETE) to API endp
     │    │         │         │              │    │         ├■─Checking Validity of Template Strings with Input Variables and Formatter Mapping____ ── Topic: 14
     │    │         │         │              │    │         └■─Requests tools for making HTTP requests with Python____ ── Topic: 10
     │    │         │         │              │    └─Code organization and positive feedback____
     │    │         │         │              │         ├─Bing Search API Wrapper and Handler Classes____
     │    │         │         │              │         │    ├■─Langchain callback manager and codebase organization____ ── Topic: 2
     │    │         │         │              │         │    └■─Bing Search API Wrapper and SERP API Usage in Python____ ── Topic: 1
     │    │         │         │              │         └─Handling iFixit devices with models and remote hardware____
     │    │         │         │              │              ├■─Loading iFixit repair guides and device wikis with transformer model inference.____ ── Topic: 0
     │    │         │         │              │              └■─Potential Issues with Modifying Input Dictionary in a Prompt Loading Function____ ── Topic: 9
     │    │         │         │              └─Implementation and Usage of SearxSearchWrapper with Environment Variables and SSL Support____
     │    │         │         │                   ├─Python Libraries for API Wrappers and Search Engines____
     │    │         │         │                   │    ├─Python packages for integrating with search engines: SearxSearchWrapper and QdrantClient.____
     │    │         │         │                   │    │    ├■─Implementation of Searx API Wrapper (SearxSearchWrapper) using Python's BaseModel with QdrantClient  ── Topic: 33
     │    │         │         │                   │    │    └■─Handling environment variables and dictionaries with get_from_dict_or_env function____ ── Topic: 72
     │    │         │         │                   │    └─Purpose and Issues with `print_text` Function in `langchain` Repository's `input.py` File____
     │    │         │         │                   │         ├■─Printing Highlighted Text with Options in Python____ ── Topic: 51
     │    │         │         │                   │         └■─Converting Python Objects to String Representation with Nested Structures and Joining on Newline Cha ── Topic: 66
     │    │         │         │                   └─GitbookLoader class and its methods____
     │    │         │         │                        ├■─Handling newlines recursively in data structures using pandas____ ── Topic: 29
     │    │         │         │                        └─GitBookLoader class for loading web pages with options to load all or single pages____
     │    │         │         │                             ├■─GitbookLoader class for loading single or multiple pages from GitBook with relative paths in the nav ── Topic: 28
     │    │         │         │                             └■─Length-Based Example Selection and Text Length Calculation____ ── Topic: 57
     │    │         │         └─Ngram overlap score using sentence_bleu and method1 smoothing function____
     │    │         │              ├─Ngram overlap score using sentence_bleu method1 smoothing function and auto reweighting____
     │    │         │              │    ├■─Code structure and organization in langchain document loaders with support for parsing comma-separat ── Topic: 70
     │    │         │              │    └─Ngram overlap score using sentence_bleu and method1 smoothing function with auto reweighting in nltk
     │    │         │              │         ├■─Compliments on well-structured and organized code in different classes and methods____ ── Topic: 65
     │    │         │              │         └■─Sentence BLEU score and ngram overlap computation with method1 smoothing function and auto reweighti ── Topic: 49
     │    │         │              └─Model Definition and Experimentation with Datetime and UTCNow Attributes____
     │    │         │                   ├■─Data Modeling with Time Zones in Python____ ── Topic: 91
     │    │         │                   └■─Constitutional Principles and Tracing in Python____ ── Topic: 68
     │    │         └─Text splitting for knowledge triple extraction____
     │    │              ├─Text Splitting Toolkit____
     │    │              │    ├─Text splitting interface and implementation____
     │    │              │    │    ├─Python REPL Tool and AST Implementation____
     │    │              │    │    │    ├─Python REPL Tool Implementation____
     │    │              │    │    │    │    ├─SQL database metadata retrieval tool____
     │    │              │    │    │    │    │    ├■─Python function to concatenate cell information for AI and human usage____ ── Topic: 44
     │    │              │    │    │    │    │    └─SQL database metadata tool for listing table schema and metadata____
     │    │              │    │    │    │    │         ├■─SQL database metadata extraction tool for specified tables____ ── Topic: 75
     │    │              │    │    │    │    │         └■─JSON and SQL database tools for listing and getting values____ ── Topic: 15
     │    │              │    │    │    │    └─Python REPL Tool using AST and Coroutine____
     │    │              │    │    │    │         ├■─Tool implementation with direct function or coroutine input and error handling.____ ── Topic: 99
     │    │              │    │    │    │         └■─Python REPL Tool with AST and version validation____ ── Topic: 74
     │    │              │    │    │    └─Implementing API wrappers for news, movie information, and weather using APIChain____
     │    │              │    │    │         ├─Implementing APIs for News, Weather, and Movie Information in LangChain's Load Tools Module____
     │    │              │    │    │         │    ├■─Language model for reasoning about position and color attributes of objects in weather forecasting w ── Topic: 73
     │    │              │    │    │         │    └─Implementing APIs for fetching news and movies using Python____
     │    │              │    │    │         │         ├■─well-structured and readable implementation of API initialization functions in load_tools.py for Too ── Topic: 85
     │    │              │    │    │         │         └■─Working with API authentication and chaining for news and movie information retrieval (using news_ap ── Topic: 100
     │    │              │    │    │         └■─Wolfram Alpha SDK querying using WolframAlphaQueryRun class and api_wrapper attribute____ ── Topic: 89
     │    │              │    │    └─TextSplitting for Vector Storage with Overlapping Chunks____
     │    │              │    │         ├─Python's StrictFormatter class and its check_unused_args method for formatting and validation of inp
     │    │              │    │         │    ├─L2 distance search using ndarray in Python____
     │    │              │    │         │    │    ├■─L2 search for nearest neighbors with np.linalg.norm____ ── Topic: 32
     │    │              │    │         │    │    └■─Parsing and Organizing Notes with Hashing and Embeddings____ ── Topic: 67
     │    │              │    │         │    └─Python Class for Strict Formatter with Check on Unused Args____
     │    │              │    │         │         ├─Vector Store Toolkit and Deployment____
     │    │              │    │         │         │    ├■─Vector Store Toolkit and Deployment with OpenAI LLM____ ── Topic: 35
     │    │              │    │         │         │    └■─Working with AirbyteJSONLoader to load local Airbyte JSON files____ ── Topic: 47
     │    │              │    │         │         └─Python Formatter class with check_unused_args method and strict validation____
     │    │              │    │         │              ├─Python's StrictFormatter class and its check_unused_args method for validating unused and extra argu
     │    │              │    │         │              │    ├─■─Finding TODO Tasks in Code Snippets____ ── Topic: 4
     │    │              │    │         │              │    └─Python Formatter and StrictFormatter with check_unused_args method____
     │    │              │    │         │              │         ├─■─Color Mapping Function for Prompt Inputs with Exclusions____ ── Topic: 88
     │    │              │    │         │              │         └─■─Implementing strict checking of unused and extra keys in a subclass of formatter____ ── Topic: 48
     │    │              │    │         │              └─Python module for loading and manipulating language chain data with verbosity control.____
     │    │              │    │         │                   ├─■─Python function for getting verbosity from language chaining with Azure OpenAI and difference from O ── Topic: 64
     │    │              │    │         │                   └─■─Purpose of functions in loading.py and csv toolkit of langchain repository____ ── Topic: 42
     │    │              │    │         └─Text splitting using chunk size and overlap with various libraries and interfaces.____
     │    │              │    │              ├─Text splitting and chunking with overlap and length functions____
     │    │              │    │              │    ├─■─Developing and Maintaining Docker Compose Modules in Python____ ── Topic: 79
     │    │              │    │              │    └─Text splitting and chunking using TextSplitter interface____
     │    │              │    │              │         ├─Text Splitting Interface and Implementation____
     │    │              │    │              │         │    ├─■─Text splitting using TokenTextSplitter class.____ ── Topic: 7
     │    │              │    │              │         │    └─■─Document Loading and Splitting with Text Splitting and Callback Management.____ ── Topic: 84
     │    │              │    │              │         └─■─Python code for initializing an agent with various optional arguments____ ── Topic: 18
     │    │              │    │              └─Loading Google Docs from Google Drive using Credentials and Tokens with Python____
     │    │              │    │                   ├─Document Loading from Cloud Storage (GCS and S3) using BaseLoader Class____
     │    │              │    │                   │    ├─■─Online PDF loading and caching using SQLite and temporary directories____ ── Topic: 98
     │    │              │    │                   │    └─■─Loading documents from cloud storage using GCSFileLoader and S3FileLoader classes.____ ── Topic: 36
     │    │              │    │                   └─■─Google Drive Loader and Credentials for Loading Google Docs____ ── Topic: 86
     │    │              │    └─StreamlitCallbackHandler for logging to streamlit in Python code____
     │    │              │         ├─Streaming with LLMs and Callback Handlers____
     │    │              │         │    ├─Networkx wrapper for entity graph operations with Redis caching.____
     │    │              │         │    │    ├─NetworkX Entity Graph with Missing Tables and Callback Manager____
     │    │              │         │    │    │    ├─■─Graph Index Creation and Operations using NetworkX Library in Python____ ── Topic: 58
     │    │              │         │    │    │    └─■─NetworkxEntityGraph and entity graph operations.____ ── Topic: 20
     │    │              │         │    │    └─Redis cache implementation in Python____
     │    │              │         │    │         ├─■─Implementing a SQAlchemy-based cache system with missing and existing prompts for better performance ── Topic: 17
     │    │              │         │    │         └─■─Implementation of a Redis cache as a backend in Python____ ── Topic: 39
     │    │              │         │    └─Python Callback Handler for Streamlit Logging____
     │    │              │         │         ├─■─Callback handlers for printing to standard output.____ ── Topic: 43
     │    │              │         │         └─■─StreamlitCallbackHandler for logging prompts and actions to Streamlit____ ── Topic: 90
     │    │              │         └─ZeroShotAgent class and observation prefix property in Python____
     │    │              │              ├─Creating a JSON agent using a toolkit for zeroshot agent execution with format instructions and inpu
     │    │              │              │    ├─■─Creating Pandas DataFrames using Agent Scratchpad and Python AST REPL Tool.____ ── Topic: 82
     │    │              │              │    └─Creating a JSON agent with toolkit, format instructions, and prefix/suffix____
     │    │              │              │         ├─■─SQL agent creation with SQLDatabaseToolkit, BaseLLM and BaseCallbackManager____ ── Topic: 11
     │    │              │              │         └─■─Creating a JSON agent with OpenAPI toolkit and interacting with it using JSON tools____ ── Topic: 56
     │    │              │              └─Classes for language model-driven decision making and use of "agent_scratchpad" in LLMChain prompts_
     │    │              │                   ├─■─Agent class and entity extraction using "agent_scratchpad" variable____ ── Topic: 38
     │    │              │                   └─■─Code for a text-based game-playing agent using self-ask-with-search approach in TextWorld environmen ── Topic: 102
     │    │              └─Text Mapping for Approximate k-NN Search using nmslib in Python____
     │    │                   ├─Script Scoring with KNN Search____
     │    │                   │    ├─■─Document bulk-ingest function for embeddings in Elasticsearch index____ ── Topic: 23
     │    │                   │    └─■─Script Scoring Search with Cosine Similarity and k-Nearest Neighbors (k-NN) Algorithm____ ── Topic: 19
     │    │                   └─Default text mapping for Approximate k-NN Search in dense vector fields using NMSLIB engine____
     │    │                        ├─■─Default Mapping for Approximate k-NN Search using NMSLIB Engine____ ── Topic: 81
     │    │                        └─■─Elasticsearch indexing and scripting with default mappings and painless scripting____ ── Topic: 94
     │    └─Tracing and Recording Runs with SharedTracer and TracerStack____
     │         ├─Python classes ToolRun and ChainRun in schemas.py file with additional attributes and their purpose.
     │         │    ├─■─Extracting information about ElementInViewPort instances in chainrun toolrun runs.____ ── Topic: 77
     │         │    └─■─Purpose and attributes of the ChainRun class in schemas.py file____ ── Topic: 78
     │         └─Tracing and thread-safe execution with SharedTracer Singleton class____
     │              ├─■─Tracing Execution Order with BaseTracer in a Thread-Safe Manner____ ── Topic: 55
     │              └─■─TracerStack and SharedTracer Implementation in Python____ ── Topic: 63
     └─Python wrapper for OpenAI and Hugging Face language models____
          ├─Self-Hosted Hugging Face Instructor Embedding Models on Remote Hardware____
          │    ├─HuggingFace and Sentence-Transformers Embeddings for Cohere____
          │    │    ├─■─Output parsing using regular expressions and the BaseOutputParser class____ ── Topic: 54
          │    │    └─NLP Embeddings using Hugging Face and Sentence Transformers____
          │    │         ├─■─Neural Embeddings with Hugging Face and Cohere API____ ── Topic: 24
          │    │         └─■─Loading sentence embedding model with sentence_transformers library.____ ── Topic: 27
          │    └─Self-hosted HuggingFace pipeline API for running models on remote hardware____
          │         ├─Self-hosted HuggingFace pipeline for remote GPU hardware inference with autolaunched instances on va
          │         │    ├─■──Self-hosted HuggingFace pipeline for remote hardware with HuggingFace Transformers and AutoTokenizer ── Topic: 40
          │         │    └─■──Self-hosted embeddings for sentence_transformers with remote hardware support.____ ── Topic: 22
          │         └─■──Self-hosted embeddings for running custom embedding models on remote hardware____ ── Topic: 62
          └─Python wrapper for OpenAI language model with API key authentication and model parameters configurat
               ├─OpenAI Language Model Wrapper Class with API Key Authentication and Model Parameters Configuration__
               │    ├─■──StochasticAI Wrapper for Large Language Models with Environment Key Validation and PDF Partitioning_ ── Topic: 50
               │    └─Integration of OpenAI Language Model with GooseAI class for Text Generation____
               │         ├─■──OpenAI Chat Model Implementation____ ── Topic: 37
               │         └─■──Python Wrapper for OpenAI Language Models____ ── Topic: 5
               └─■──Anthropic Large Language Models and API Usage in AtlasDB Project Management____ ── Topic: 83
```