

langchain API Reference

langchain. api

Helper functions for managing the LangChain API.

This module is only relevant for LangChain developers, not for users.

Warning: This module and its submodules are for internal use only. Do not use them in your own code. We may change the API at any time with no warning.

Classes

api.deprecation.LangChainDeprecationWarning	A class for issuing deprecation warnings for LangChain users.
---	---

Functions

api.deprecation.deprecated (since, *, ...)	Decorator to mark a function, a class, or a property as deprecated.
api.deprecation.suppress_langchain_deprecation_warning ()	Context manager to suppress LangChainDeprecationWarning.

langchain.adapters

Classes

adapters.openai.ChatCompletion ()

Functions

adapters.openai.aenumerate (iterable[, start])	Async version of enumerate.
adapters.openai.convert dict to message (_dict)	
adapters.openai.convert message to dict (message)	
adapters.openai.convert openai messages (messages)	Convert dictionaries representing OpenAI messages to LangChain format.

langchain.agents

Agent is a class that uses an LLM to choose a sequence of actions to take.

In Chains, a sequence of actions is hardcoded. In Agents, a language model is used as a reasoning engine to determine which actions to take and in which order.

Agents select and use **Tools** and **Toolkits** for actions.

Class hierarchy:

BaseSingleActionAgent	-->	LLMSingleActionAgent
		OpenAIFunctionsAgent
		XMLAgent
		Agent --> <name>Agent # Examples: ZeroShotAgent, ChatAgent
BaseMultiActionAgent	-->	OpenAIMultiFunctionsAgent

Main helpers:

Toggle Menu

AgentType, AgentExecutor, AgentOutputParser, AgentExecutorIterator, AgentAction, AgentFinish

Classes

agents.agent_iterator.AgentExecutorIterator(...)	Iterator for AgentExecutor.
agents.agent_iterator.BaseAgentExecutorIterator()	Base class for AgentExecutorIterator.
agents.agent.Agent	Agent that calls the language model and deciding the action.
agents.agent.AgentExecutor	Agent that is using tools.
agents.agent.AgentOutputParser	Base class for parsing agent output into agent action/finish.
agents.agent.BaseMultiActionAgent	Base Multi Action Agent class.
agents.agent.BaseSingleActionAgent	Base Single Action Agent class.
agents.agent.ExceptionTool	Tool that just returns the query.
agents.agent.LLMSingleActionAgent	Base class for single action agents.
agents.tools.InvalidTool	Tool that is run when invalid tool name is encountered by agent.
agents.schema.AgentScratchPadChatPromptTemplate	Chat prompt template for the agent scratchpad.
agents.agent_types.AgentType(value[, names, ...])	Enumerator with the Agent types.
agents.xml.base.XMLAgent	Agent that uses XML tags.
agents.xml.base.XMLAgentOutputParser	Output parser for XMLAgent.
agents.conversational_chat.output_parser.ConvoOutputParser	Output parser for the conversational agent.
agents.conversational_chat.base.ConversationalChatAgent	An agent designed to hold a conversation in addition to using tools.
agents.structured_chat.output_parser.StructuredChatOutputParser	Output parser for the structured chat agent.
agents.structured_chat.output_parser.StructuredChatOutputParserWithRetries	Output parser with retries for the structured chat agent.
agents.structured_chat.base.StructuredChatAgent	Structured Chat Agent.
agents.self_ask_with_search.output_parser.SelfAskOutputParser	Output parser for the self-ask agent.
agents.self_ask_with_search.base.SelfAskWithSearchAgent	Agent for the self-ask-with-search paper.
agents.self_ask_with_search.base.SelfAskWithSearchChain	Chain that does self-ask with search.
agents.conversational.output_parser.ConvoOutputParser	Output parser for the conversational agent.
agents.conversational.base.ConversationalAgent	An agent that holds a conversation in addition to using tools.
agents.react.output_parser.ReActOutputParser	Output parser for the ReAct agent.
agents.react.base.DocstoreExplorer(docstore)	Class to assist with exploration of a document store.
agents.react.base.ReActChain	Chain that implements the ReAct paper.
agents.react.base.ReActDocstoreAgent	Agent for the ReAct chain.
agents.react.base.ReActTextWorldAgent	Agent for the ReAct TextWorld chain.
agents.chat.output_parser.ChatOutputParser	Output parser for the chat agent.
agents.chat.base.ChatAgent	Chat Agent.
agents.openai_functions_agent.base.OpenAIFunctionsAgent	An Agent driven by OpenAIs function powered API.
agents.openai_functions_agent.agent token buffer memory.AgentTokenBufferMemory	Memory used to save agent output AND intermediate steps.
agents.mrkl.output_parser.MRKLOutputParser	MRKL Output parser for the chat agent.
agents.mrkl.base.ChainConfig(action_name, ...)	Configuration for chain to use in MRKL system.
agents.mrkl.base.MRKLChain	Chain that implements the MRKL system.
agents.mrkl.base.ZeroShotAgent	Agent for the MRKL chain.
agents.agent_toolkits.base.BaseToolkit	Base Toolkit representing a collection of related tools.
agents.agent_toolkits.azure_cognitive_services.AzureCognitiveServicesToolkit	Toolkit for Azure Cognitive Services.
agents.agent_toolkits.spark_sql.toolkit.SparkSQLToolkit	Toolkit for interacting with Spark SQL.
agents.agent_toolkits.amadeus.toolkit.AmadeusToolkit	Toolkit for interacting with Office365.
agents.agent_toolkits.playwright.toolkit.PlayWrightBrowserToolkit	Toolkit for PlayWright browser tools.
agents.agent_toolkits.vectorstore.toolkit.VectorStoreInfo	Information about a VectorStore.
agents.agent_toolkits.vectorstore.toolkit.VectorStoreRouterToolkit	Toolkit for routing between Vector Stores.
agents.agent_toolkits.vectorstore.toolkit.VectorStoreToolkit	Toolkit for interacting with a Vector Store.
agents.agent_toolkits.zapier.toolkit.ZapierToolkit	Zapier Toolkit.
agents.agent_toolkits.office365.toolkit.O365Toolkit	Toolkit for interacting with Office 365.
agents.agent_toolkits.nla.toolkit.NLAToolkit	Natural Language API Toolkit.
agents.agent_toolkits.nla.tool.NLATool	Natural Language API Tool.
agents.agent_toolkits.multion.toolkit.MultionToolkit	Toolkit for interacting with the Browser Agent

Toggle Menu

agents.agent_toolkits.gmail.toolkit.GmailToolkit	Toolkit for interacting with Gmail.
agents.agent_toolkits.sql.toolkit.SQLDatabaseToolkit	Toolkit for interacting with SQL databases.
agents.agent_toolkits.openapi.spec.ReducedOpenAPISpec(...)	A reduced OpenAPI spec.
agents.agent_toolkits.openapi.planner.RequestsDeleteToolWithParsing	A tool that sends a DELETE request and parses the response.
agents.agent_toolkits.openapi.planner.RequestsGetToolWithParsing	Requests GET tool with LLM-instructed extraction of truncated responses.
agents.agent_toolkits.openapi.planner.RequestsPatchToolWithParsing	Requests PATCH tool with LLM-instructed extraction of truncated responses.
agents.agent_toolkits.openapi.planner.RequestsPostToolWithParsing	Requests POST tool with LLM-instructed extraction of truncated responses.
agents.agent_toolkits.openapi.toolkit.OpenAPIToolkit	Toolkit for interacting with an OpenAPI API.
agents.agent_toolkits.openapi.toolkit.RequestsToolkit	Toolkit for making REST requests.
agents.agent_toolkits.file_management.toolkit.FileManagementToolkit	Toolkit for interacting with a Local Files.
agents.agent_toolkits.jira.toolkit.JiraToolkit	Jira Toolkit.
agents.agent_toolkits.github.toolkit.GitHubToolkit	GitHub Toolkit.
agents.agent_toolkits.powerbi.toolkit.PowerBIToolkit	Toolkit for interacting with Power BI dataset.
agents.agent_toolkits.json.toolkit.JsonToolkit	Toolkit for interacting with a JSON spec.
agents.openai_functions_multi_agent.base.OpenAIMultiFunctionsAgent	An Agent driven by OpenAIs function powered API.

Functions

agents.agent_iterator.rebuild_callback_manager_on_set(...)	Decorator to forc setters to rebuild callback mgr
agents.agent_toolkits.conversational_retrieval.openai_functions.create_conversational_retrieval_agent(...)	A convenience method for creati a conversational retrieval agent.
agents.agent_toolkits.conversational_retrieval.tool.create_retriever_tool(...)	Create a tool to d retrieval of documents.
agents.agent_toolkits.csv.base.create_csv_agent(...)	Create csv agent loading to a dataframe and using pandas age
agents.agent_toolkits.json.base.create_json_agent(...)	Construct a json agent from an LL and tools.
agents.agent_toolkits.openapi.base.create_openapi_agent(...)	Construct an OpenAPI agent fr an LLM and tools
agents.agent_toolkits.openapi.planner.create_openapi_agent(...)	Instantiate Open/ API planner and controller for a given spec.
agents.agent_toolkits.openapi.spec.dereference_refs(...)	Try to substitute \$refs.
agents.agent_toolkits.openapi.spec.reduce_openapi_spec(spec)	Simplify/distill/mi a spec somehow.
agents.agent_toolkits.pandas.base.create_pandas_dataframe_agent(llm, df)	Construct a pand agent from an LL and dataframe.
agents.agent_toolkits.powerbi.base.create_pbi_agent(llm)	Construct a Powe BI agent from an LLM and tools.
agents.agent_toolkits.powerbi.chat_base.create_pbi_chat_agent(llm)	Construct a Powe BI agent from a Chat LLM and toc
agents.agent_toolkits.python.base.create_python_agent(...)	Construct a pytho agent from an LL and tool.
agents.agent_toolkits.spark.base.create_spark_dataframe_agent(llm, df)	Construct a Spar agent from an LL and dataframe.
agents.agent_toolkits.spark_sql.base.create_spark_sql_agent(...)	Construct a Spar SQL agent from a LLM and tools.
agents.agent_toolkits.sql.base.create_sql_agent(...)	Construct an SQL agent from an LL and tools.

agents.agent_toolkits.vectorstore.base.create_vectorstore_agent(...)	Construct a VectorStore agent from an LLM and tools.
agents.agent_toolkits.vectorstore.base.create_vectorstore_router_agent(...)	Construct a VectorStore router agent from an LLM and tools.
agents.agent_toolkits.xorbits.base.create_xorbits_agent(...)	Construct a xorbits agent from an LLM and dataframe.
agents.initialize.initialize_agent(tools, llm)	Load an agent executor given tools and LLM.
agents.load_tools.get_all_tool_names()	Get a list of all possible tool names.
agents.load_tools.load_huggingface_tool(...)	Loads a tool from the HuggingFace Hub.
agents.load_tools.load_tools(tool_names[, ...])	Load tools based on their name.
agents.loading.load_agent(path, **kwargs)	Unified method for loading an agent from LangChain or local fs.
agents.loading.load_agent_from_config(config)	Load agent from Config Dict.
agents.utils.validate_tools_single_input(...)	Validate tools for single input.

langchain.cache

Warning: Beta Feature!

Cache provides an optional caching layer for LLMs.

Cache is useful for two reasons:

- It can save you money by reducing the number of API calls you make to the LLM provider if you're often requesting the same completion multiple times.
- It can speed up your application by reducing the number of API calls you make to the LLM provider.

Cache directly competes with Memory. See documentation for Pros and Cons.

Class hierarchy:

```
BaseCache --> <name>Cache # Examples: InMemoryCache, RedisCache, GPTCache
```

Classes

cache.BaseCache()	Base interface for cache.
cache.FullLLMCache(**kwargs)	SQLite table for full LLM Cache (all generations).
cache.GPTCache([init_func])	Cache that uses GPTCache as a backend.
cache.InMemoryCache()	Cache that stores things in memory.
cache.MomentoCache(cache_client, cache_name, *)	Cache that uses Momento as a backend.
cache.RedisCache(redis_, *, ttl)	Cache that uses Redis as a backend.
cache.RedisSemanticCache(redis_url, embedding)	Cache that uses Redis as a vector-store backend.
cache.SQLAlchemyCache(engine, cache_schema)	Cache that uses SQLAlchemy as a backend.
cache.SQLiteCache([database_path])	Cache that uses SQLite as a backend.

Functions

langchain.callbacks

Toggle Menu

Callbacks allow listening to events in LangChain.

Class hierarchy:

BaseCallbackHandler --> <name>CallbackHandler # Example: AimCallbackHandler

Classes

callbacks.sagemaker_callback.SageMakerCallbackHandler (run)	Callback Handler that logs prompt artifacts and metrics to SageMaker Experiments.
callbacks.argilla_callback.ArgillaCallbackHandler (...)	Callback Handler that logs into Argilla.
callbacks.mlflow_callback.MlflowCallbackHandler ([...])	Callback Handler that logs metrics and artifacts to mlflow server.
callbacks.mlflow_callback.MlflowLogger (**kwargs)	Callback Handler that logs metrics and artifacts to mlflow server.
callbacks.human.HumanApprovalCallbackHandler (...)	Callback for manually validating values.
callbacks.human.HumanRejectedException	Exception to raise when a person manually review and rejects a value.
callbacks.streaming_aiter_final_only.AsyncFinalIteratorCallbackHandler (*)	Callback handler that returns an async iterator.
callbacks.streaming_aiter.AsyncIteratorCallbackHandler ()	Callback handler that returns an async iterator.
callbacks.streaming_stdout_final_only.FinalStreamingStdOutCallbackHandler (*)	Callback handler for streaming in agents.
callbacks.arize_callback.ArizeCallbackHandler ([...])	Callback Handler that logs to Arize.
callbacks.promptlayer_callback.PromptLayerCallbackHandler ([...])	Callback handler for promptlayer.
callbacks.openai_info.OpenAICallbackHandler ()	Callback Handler that tracks OpenAI info.
callbacks.base.AsyncCallbackHandler ()	Async callback handler that can be used to handle callbacks from langchain.
callbacks.base.BaseCallbackHandler ()	Base callback handler that can be used to handle callbacks from langchain.
callbacks.base.BaseCallbackManager (handlers)	Base callback manager that handles callbacks from LangChain.
callbacks.base.CallbackManagerMixin ()	Mixin for callback manager.
callbacks.base.ChainManagerMixin ()	Mixin for chain callbacks.
callbacks.base.LLMManagerMixin ()	Mixin for LLM callbacks.
callbacks.base.RetrieverManagerMixin ()	Mixin for Retriever callbacks.
callbacks.base.RunManagerMixin ()	Mixin for run manager.
callbacks.base.ToolManagerMixin ()	Mixin for tool callbacks.
callbacks.comet_ml_callback.CometCallbackHandler ([...])	Callback Handler that logs to Comet.
callbacks.manager.AsyncCallbackManager (handlers)	Async callback manager that handles callbacks from LangChain.
callbacks.manager.AsyncCallbackManagerForChainRun (*, ...)	Async callback manager for chain run.
callbacks.manager.AsyncCallbackManagerForLLMRun (*, ...)	Async callback manager for LLM run.
callbacks.manager.AsyncCallbackManagerForRetrieverRun (*, ...)	Async callback manager for retriever run.
callbacks.manager.AsyncCallbackManagerForToolRun (*, ...)	Async callback manager for tool run.
callbacks.manager.AsyncParentRunManager (*, ...)	Async Parent Run Manager.
callbacks.manager.AsyncRunManager (*, run_id, ...)	Async Run Manager.
callbacks.manager.BaseRunManager (*, run_id, ...)	Base class for run manager (a bound callback manager).
callbacks.manager.CallbackManager (handlers)	Callback manager that handles callbacks from langchain.
callbacks.manager.CallbackManagerForChainRun (*, ...)	Callback manager for chain run.
callbacks.manager.CallbackManagerForLLMRun (*, ...)	Callback manager for LLM run.
callbacks.manager.CallbackManagerForRetrieverRun (*, ...)	Callback manager for retriever run.
callbacks.manager.CallbackManagerForToolRun (*, ...)	Callback manager for tool run.
callbacks.manager.ParentRunManager (*, ...[, ...])	Sync Parent Run Manager.
callbacks.manager.RunManager (*, run_id, ...)	Sync Run Manager.
callbacks.infino_callback.InfinoCallbackHandler ([...])	Callback Handler that logs to Infino.
callbacks.clearml_callback.ClearMLCallbackHandler ([...])	Callback Handler that logs to ClearML.
callbacks.file.FileCallbackHandler (filename)	Callback Handler that writes to a file.
callbacks.labelstudio_callback.LabelStudioCallbackHandler ([...])	Label Studio callback handler.
callbacks.labelstudio_callback.LabelStudioMode (value)	
callbacks.wandb_callback.WandbCallbackHandler ([...])	Callback Handler that logs to Weights and Biases.
callbacks.flyte_callback.FlyteCallbackHandler ()	This callback handler that is used within a Flyte task.
callbacks.utils.BaseMetadataCallbackHandler ()	This class handles the metadata and associated function states for callbacks.
callbacks.streaming_stdout.StreamingStdOutCallbackHandler ()	Callback handler for streaming.
callbacks.whylabs_callback.WhyLabsCallbackHandler (...)	Callback Handler for logging to WhyLabs.
callbacks.aim_callback.AimCallbackHandler ([...])	Callback Handler that logs to Aim.

Toggle Menu

callbacks.aim_callback.BaseMetadataCallbackHandler()	This class handles the metadata and associated function states for callbacks.
callbacks.stdout.StdoutCallbackHandler([color])	Callback Handler that prints to std out.
callbacks.arthur_callback.ArthurCallbackHandler(...)	Callback Handler that logs to Arthur platform.
callbacks.context_callback.ContextCallbackHandler([...])	Callback Handler that records transcripts to the Context service.
callbacks.streamlit.mutable_expander.ChildRecord(...)	The child record as a NamedTuple.
callbacks.streamlit.mutable_expander.ChildType(value)	The enumerator of the child type.
callbacks.streamlit.mutable_expander.MutableExpander(...)	A Streamlit expander that can be renamed and dynamically expanded/collapsed.
callbacks.streamlit.streamlit_callback_handler.LLMThought(...)	A thought in the LLM's thought stream.
callbacks.streamlit.streamlit_callback_handler.LLMThoughtLabeler()	Generates markdown labels for LLMThought containers.
callbacks.streamlit.streamlit_callback_handler.LLMThoughtState(value)	Enumerator of the LLMThought state.
callbacks.streamlit.streamlit_callback_handler.StreamlitCallbackHandler(...)	A callback handler that writes to a Streamlit app.
callbacks.streamlit.streamlit_callback_handler.ToolRecord(...)	The tool record as a NamedTuple.
callbacks.tracers.schemas.BaseRun	Base class for Run.
callbacks.tracers.schemas.ChainRun	Class for ChainRun.
callbacks.tracers.schemas.LLMRun	Class for LLMRun.
callbacks.tracers.schemas.Run	Run schema for the V2 API in the Tracer.
callbacks.tracers.schemas.ToolRun	Class for ToolRun.
callbacks.tracers.schemas.TracerSession	TracerSessionV1 schema for the V2 API.
callbacks.tracers.schemas.TracerSessionBase	Base class for TracerSession.
callbacks.tracers.schemas.TracerSessionV1	TracerSessionV1 schema.
callbacks.tracers.schemas.TracerSessionV1Base	Base class for TracerSessionV1.
callbacks.tracers.schemas.TracerSessionV1Create	Create class for TracerSessionV1.
callbacks.tracers.run_collector.RunCollectorCallbackHandler([...])	A tracer that collects all nested runs in a list.
callbacks.tracers.base.BaseTracer(**kwargs)	Base interface for tracers.
callbacks.tracers.base.TracerException	Base class for exceptions in tracers module.
callbacks.tracers.langchain.LangChainTracer([...])	An implementation of the SharedTracer that POSTS to the langchain endpoint.
callbacks.tracers.langchain_v1.LangChainTracerV1(...)	An implementation of the SharedTracer that POSTS to the langchain endpoint.
callbacks.tracers.wandb.RunProcessor(...)	Handles the conversion of a LangChain Runs into a WBTraceTree.
callbacks.tracers.wandb.WandbRunArgs	Arguments for the WandbTracer.
callbacks.tracers.wandb.WandbTracer([run_args])	Callback Handler that logs to Weights and Biases.
callbacks.tracers.stdout.ConsoleCallbackHandler(...)	Tracer that prints to the console.
callbacks.tracers.stdout.FunctionCallbackHandler(...)	Tracer that calls a function with a single str parameter.
callbacks.tracers.evaluation.EvaluatorCallbackHandler(...)	A tracer that runs a run evaluator whenever a run is persisted.

Functions

callbacks.aim_callback.import_aim()	Import the aim python package and raise an error if it is not installed.
callbacks.clearml_callback.import_clearml()	Import the clearml python package and raise an error if it is not installed.
callbacks.comet_ml_callback.import_comet_ml()	Import comet_ml and raise an error if it is not installed.
callbacks.context_callback.import_context()	Import the <i>getcontext</i> package.
callbacks.flyte_callback.analyze_text(text)	Analyze text using textstat and spacy.
callbacks.flyte_callback.import_flytekit()	Import flytekit and flytekitplugins-deck-standard.
callbacks.infino_callback.import_infino()	Import the infino client.
callbacks.labelstudio_callback.get_default_label_configs(mode)	
callbacks.manager.atrace_as_chain_group(...)	Get an async callback manager for a chain group in a context manager.
callbacks.manager.env_var_is_set(env_var)	Check if an environment variable is set.
callbacks.manager.get_openai_callback()	Get the OpenAI callback handler in a context manager.
callbacks.manager.trace_as_chain_group(...)	Get a callback manager for a chain group in a context manager.
callbacks.manager.tracing_enabled([session_name])	Get the Deprecated LangChainTracer in a context manager.
callbacks.manager.tracing_v2_enabled([...])	Instruct LangChain to log all runs in context to LangSmith.
callbacks.manager.wandb_tracing_enabled([...])	Get the WandbTracer in a context manager.
callbacks.mlflow_callback.analyze_text(text)	Analyze text using textstat and spacy.

Toggle Menu

callbacks.mlflow_callback.construct_html_from_prompt_and_generation(...)	Construct an html element from a prompt and a generation.
callbacks.mlflow_callback.import_mlflow()	Import the mlflow python package and raise an error if it is not installed.
callbacks.openai_info.get_openai_token_cost_for_model(...)	Get the cost in USD for a given model and number of tokens.
callbacks.openai_info.standardize_model_name(...)	Standardize the model name to a format that can be used in the OpenAI API.
callbacks.sagemaker_callback.save_json(data, ...)	Save dict to local file path.
callbacks.tracers.evaluation.wait_for_all_evaluators()	Wait for all tracers to finish.
callbacks.tracers.langchain.log_error_once(...)	Log an error once.
callbacks.tracers.langchain.wait_for_all_tracers()	Wait for all tracers to finish.
callbacks.tracers.langchain_v1.get_headers()	Get the headers for the LangChain API.
callbacks.tracers.schemas.RunTypeEnum()	RunTypeEnum.
callbacks.tracers.stdout.elapsed(run)	Get the elapsed time of a run.
callbacks.tracers.stdout.try_json_stringify(...)	Try to stringify an object to JSON.
callbacks.utils.flatten_dict(nested_dict[, ...])	Flattens a nested dictionary into a flat dictionary.
callbacks.utils.hash_string(s)	Hash a string using sha1.
callbacks.utils.import_pandas()	Import the pandas python package and raise an error if it is not installed.
callbacks.utils.import_spacy()	Import the spacy python package and raise an error if it is not installed.
callbacks.utils.import_textstat()	Import the textstat python package and raise an error if it is not installed.
callbacks.utils.load_json(json_path)	Load json file to a string.
callbacks.wandb_callback.analyze_text(text)	Analyze text using textstat and spacy.
callbacks.wandb_callback.construct_html_from_prompt_and_generation(...)	Construct an html element from a prompt and a generation.
callbacks.wandb_callback.import_wandb()	Import the wandb python package and raise an error if it is not installed.
callbacks.wandb_callback.load_json_to_dict(...)	Load json file to a dictionary.
callbacks.whylabs_callback.import_langkit([...])	Import the langkit python package and raise an error if it is not installed.

langchain.chains

Chains are easily reusable components linked together.

Chains encode a sequence of calls to components like models, document retrievers, other Chains, etc., and provide a simple interface to this sequence.

The Chain interface makes it easy to create apps that are:

- **Stateful:** add Memory to any Chain to give it state,
- **Observable:** pass Callbacks to a Chain to execute additional functionality, like logging, outside the main sequence of component calls,
- **Composable:** combine Chains with other components, including other Chains.

Class hierarchy:

Chain --> <name>Chain # Examples: LLMChain, MapReduceChain, RouterChain

Classes

chains.transform.TransformChain	Chain that transforms the chain output.
chains.base.Chain	Abstract base class for creating structured sequences of calls to components.
chains.mapreduce.MapReduceChain	Map-reduce chain.
chains.moderation.OpenAIModerationChain	Pass input through a moderation endpoint.
chains.llm_requests.LLMRequestsChain	Chain that requests a URL and then uses an LLM to parse results.
chains.llm.LLMChain	Chain to run queries against LLMs.
chains.prompt_selector.BasePromptSelector	Base class for prompt selectors.
chains.prompt_selector.ConditionalPromptSelector	Prompt collection that goes through conditionals.
chains.sequential.SequentialChain	Chain where the outputs of one chain feed directly into next.
chains.sequential.SimpleSequentialChain	Simple chain where the outputs of one step feed directly into next.

Toggle Menu

chains.llm_summarization_checker.base.LLMSummarizationCheckerChain	Chain for question-answering with self-verification.
chains.openai_functions.openapi.SimpleRequestChain	Chain for making a simple request to an API endpoint.
chains.openai_functions.citation_fuzzy_match.FactWithEvidence	Class representing a single statement.
chains.openai_functions.citation_fuzzy_match.QuestionAnswer	A question and its answer as a list of facts each one should have a source.
chains.openai_functions.qa_with_structure.AnswerWithSources	An answer to the question, with sources.
chains.query_constructor.base.StructuredQueryOutputParser	Output parser that parses a structured query.
chains.query_constructor.ir.Comparator (value)	Enumerator of the comparison operators.
chains.query_constructor.ir.Comparison	A comparison to a value.
chains.query_constructor.ir.Expr	Base class for all expressions.
chains.query_constructor.ir.FilterDirective	A filtering expression.
chains.query_constructor.ir.Operation	A logical operation over other directives.
chains.query_constructor.ir.Operator (value)	Enumerator of the operations.
chains.query_constructor.ir.StructuredQuery	A structured query.
chains.query_constructor.ir.Visitor ()	Defines interface for IR translation using visitor pattern.
chains.query_constructor.schema.AttributeInfo	Information about a data source attribute.
chains.hyde.base.HypotheticalDocumentEmbedder	Generate hypothetical document for query, and then embed that.
chains.router.multi_prompt.MultiPromptChain	A multi-route chain that uses an LLM router chain to choose amongst prompts.
chains.router.base.MultiRouteChain	Use a single chain to route an input to one of multiple candidate chains.
chains.router.base.Route (destination, ...)	Create new instance of Route(destination, next_inputs)
chains.router.base.RouterChain	Chain that outputs the name of a destination chain and the inputs to it.
chains.router.llm_router.LLMRouterChain	A router chain that uses an LLM chain to perform routing.
chains.router.llm_router.RouterOutputParser	Parser for output of router chain int he multi-prompt chain.
chains.router.multi_retrieval_qa.MultiRetrievalQAChain	A multi-route chain that uses an LLM router chain to choose amongst retrieval qa chains.
chains.router.embedding_router.EmbeddingRouterChain	Chain that uses embeddings to route between options.
chains.constitutional_ai.base.ConstitutionalChain	Chain for applying constitutional principles.
chains.constitutional_ai.models.ConstitutionalPrinciple	Class for a constitutional principle.
chains.qa_with_sources.base.BaseQAWithSourcesChain	Question answering chain with sources over documents.
chains.qa_with_sources.base.QAWithSourcesChain	Question answering with sources over documents.
chains.qa_with_sources.vector_db.VectorDBQAWithSourcesChain	Question-answering with sources over a vector database.
chains.qa_with_sources.loading.LoadingCallable (...)	Interface for loading the combine documents chain.
chains.qa_with_sources.retrieval.RetrievalQAWithSourcesChain	Question-answering with sources over an index.
chains.llm_bash.base.LLMBashChain	Chain that interprets a prompt and executes bash operations.
chains.llm_bash.prompt.BashOutputParser	Parser for bash output.
chains.retrieval_qa.base.BaseRetrievalQA	Base class for question-answering chains.
chains.retrieval_qa.base.RetrievalQA	Chain for question-answering against an index.
chains.retrieval_qa.base.VectorDBQA	Chain for question-answering against a vector database.
chains.api.base.APIChain	Chain that makes API calls and summarizes the responses to answer a question.
chains.api.openapi.chain.OpenAPIEndpointChain	Chain interacts with an OpenAPI endpoint using natural language.
chains.api.openapi.response_chain.APIResponderChain	Get the response parser.
chains.api.openapi.response_chain.APIResponderOutputParser	Parse the response and error tags.
chains.api.openapi.requests_chain.APIRequesterChain	Get the request parser.
chains.api.openapi.requests_chain.APIRequesterOutputParser	Parse the request and error tags.
chains.elasticsearch_database.base.ElasticsearchDatabaseChain	Chain for interacting with Elasticsearch Database.
chains.llm_math.base.LLMMathChain	Chain that interprets a prompt and executes python code to do math.
chains.combine_documents.base.AnalyzeDocumentChain	Chain that splits documents, then analyzes it in pieces.
chains.combine_documents.base.BaseCombineDocumentsChain	Base interface for chains combining documents.
chains.combine_documents.reduce.AsyncCombineDocsProtocol (...)	Interface for the combine_docs method.
chains.combine_documents.reduce.CombineDocsProtocol (...)	Interface for the combine_docs method.
chains.combine_documents.reduce.ReduceDocumentsChain	Combine documents by recursively reducing them.
chains.combine_documents.refine.RefineDocumentsChain	Combine documents by doing a first pass and then refining on more documents.
chains.combine_documents.stuff.StuffDocumentsChain	Chain that combines documents by stuffing into context.
chains.combine_documents.map_rerank.MapRerankDocumentsChain	Combining documents by mapping a chain over them, then reranking results.

chains.combine_documents.map_reduce.MapReduceDocumentsChain	Combining documents by mapping a chain over them, then combining results.
chains.llm_symbolic_math.base.LLMSymbolicMathChain	Chain that interprets a prompt and executes python code to do symbolic math.
chains.llm_checker.base.LLMCheckerChain	Chain for question-answering with self-verification.
chains.sql_database.query.SQLInput	Input for a SQL Chain.
chains.sql_database.query.SQLInputWithTables	Input for a SQL Chain.
chains.conversational_retrieval.base.BaseConversationalRetrievalChain	Chain for chatting with an index.
chains.conversational_retrieval.base.ChatVectorDBChain	Chain for chatting with a vector database.
chains.conversational_retrieval.base.ConversationalRetrievalChain	Chain for having a conversation based on retrieved documents.
chains.natbot.base.NatBotChain	Implement an LLM driven browser.
chains.natbot.crawler.Crawler()	A crawler for web pages.
chains.natbot.crawler.ElementInViewPort	A typed dictionary containing information about elements in the viewport.
chains.qa_generation.base.QAGenerationChain	Base class for question-answer generation chains.
chains.graph_qa.cypher.GraphCypherQAChain	Chain for question-answering against a graph by generating Cypher statements.
chains.graph_qa.neptune_cypher.NeptuneOpenCypherQAChain	Chain for question-answering against a Neptune graph by generating openCypher statements.
chains.graph_qa.base.GraphQAChain	Chain for question-answering against a graph.
chains.graph_qa.sparql.GraphSparqlQAChain	Chain for question-answering against an RDF or OWL graph by generating SPARQL statements.
chains.graph_qa.hugegraph.HugeGraphQAChain	Chain for question-answering against a graph by generating gremlin statements.
chains.graph_qa.arangodb.ArangoGraphQAChain	Chain for question-answering against a graph by generating AQL statements.
chains.graph_qa.kuzu.KuzuQAChain	Chain for question-answering against a graph by generating Cypher statements for Kùzu.
chains.graph_qa.nebulagraph.NebulaGraphQAChain	Chain for question-answering against a graph by generating nGQL statements.
chains.flare.prompts.FinishedOutputParser	Output parser that checks if the output is finished.
chains.flare.base.FlareChain	Chain that combines a retriever, a question generator, and a response generator.
chains.flare.base.QuestionGeneratorChain	Chain that generates questions from uncertain spans.
chains.conversation.base.ConversationChain	Chain to have a conversation and load context from memory.

Functions

chains.example_generator.generate_example(...)	Return another example given a list of examples for a prompt.
chains.graph_qa.cypher.extract_cypher(text)	Extract Cypher code from a text.
chains.graph_qa.neptune_cypher.extract_cypher(text)	Extract Cypher code from text using Regex.
chains.loading.load_chain(path, **kwargs)	Unified method for loading a chain from LangChainHub or local fs.
chains.loading.load_chain_from_config(...)	Load chain from Config Dict.
chains.openai_functions.base.convert_python_function_to_openai_function(...)	Convert a Python function to an OpenAI function-calling API compatible dict.
chains.openai_functions.base.convert_to_openai_function(...)	Convert a raw function/class to an OpenAI function.
chains.openai_functions.base.create_openai_fn_chain(...)	Create an LLM chain that uses OpenAI functions.
chains.openai_functions.base.create_structured_output_chain(...)	Create an LLMChain that uses an OpenAI function to get a structured output.
chains.openai_functions.citation_fuzzy_match.create_citation_fuzzy_match_chain(llm)	Create a citation fuzzy match chain.
chains.openai_functions.extraction.create_extraction_chain(...)	Creates a chain that extracts information from a passage.
chains.openai_functions.extraction.create_extraction_chain_pydantic(...)	Creates a chain that extracts information from a passage using pydantic schema.
chains.openai_functions.openapi.get_openapi_chain(spec)	Create a chain for querying an API from a OpenAPI spec.
chains.openai_functions.openapi.openapi_spec_to_openai_fn(spec)	Convert a valid OpenAPI spec to the JSON Schema format expected for OpenAI
chains.openai_functions.qa_with_structure.create_qa_with_sources_chain(...)	Create a question answering chain that returns an answer with sources.
chains.openai_functions.qa_with_structure.create_qa_with_structure_chain(...)	Create a question answering chain that returns an answer with sources
chains.openai_functions.tagging.create_tagging_chain(...)	Creates a chain that extracts information from a passage

Toggle Menu

chains.openai_functions.tagging.create_tagging_chain_pydantic(...)	Creates a chain that extracts information from a passage
chains.openai_functions.utils.get_llm_kwargs(...)	Returns the kwargs for the LLMChain constructor.
chains.prompt_selector.is_chat_model(llm)	Check if the language model is a chat model.
chains.prompt_selector.is_llm(llm)	Check if the language model is a LLM.
chains.qa_with_sources.loading.load_qa_with_sources_chain(llm)	Load a question answering with sources chain.
chains.query_constructor.base.load_query_constructor_chain(...)	Load a query constructor chain.
chains.query_constructor.parser.get_parser([...])	Returns a parser for the query language.
chains.query_constructor.parser.v_args(...)	Dummy decorator for when lark is not installed.
chains.sql_database.query.create_sql_query_chain(llm, db)	Create a chain that generates SQL queries.

langchain.chat models

Chat Models are a variation on language models.

While Chat Models use language models under the hood, the interface they expose is a bit different. Rather than expose a “text in, text out” API, they expose an interface where “chat messages” are the inputs and outputs.

Class hierarchy:

BaseLanguageModel --> BaseChatModel --> <name> # Examples: ChatOpenAI, ChatGooglePalm

Main helpers:

AIMessage, BaseMessage, HumanMessage

Classes

chat_models.openai.ChatOpenAI	OpenAI Chat large language models API.
chat_models.human.HumanInputChatModel	ChatModel which returns user input as the response.
chat_models.azureml_endpoint.AzureMLChatOnlineEndpoint	AzureML Chat models API.
chat_models.azureml_endpoint.LlamaContentFormatter()	Content formatter for LLaMA.
chat_models.base.BaseChatModel	Base class for Chat models.
chat_models.base.SimpleChatModel	Simple Chat Model.
chat_models.vertexai.ChatVertexAI	Vertex AI Chat large language models API.
chat_models.azure_openai.AzureChatOpenAI	Azure OpenAI Chat Completion API.
chat_models.jinachat.JinaChat	Jina AI Chat models API.
chat_models.google_palm.ChatGooglePalm	Google PaLM Chat models API.
chat_models.google_palm.ChatGooglePalmError	Error with the Google PaLM API.
chat_models.anthropic.ChatAnthropic	Anthropic chat large language models.
chat_models.ernie.ErnieBotChat	ERNIE-Bot large language model.
chat_models.mlflow_ai_gateway.ChatMLflowAIGateway	MLflow AI Gateway chat models API.
chat_models.mlflow_ai_gateway.ChatParams	Parameters for the MLflow AI Gateway LLM.
chat_models.litellm.ChatLiteLLM	LiteLLM Chat models API.
chat_models.litellm.ChatLiteLLMException	Error with the LiteLLM I/O library
chat_models.litellm.FunctionMessage	Message for passing the result of executing a function back to a model.
chat_models.litellm.FunctionMessageChunk	Message Chunk for passing the result of executing a function back to a model.
chat_models.anyscale.ChatAnyscale	Anyscale Chat large language models.
chat_models.fake.FakeListChatModel	Fake ChatModel for testing purposes.
chat_models.promptlayer_openai.PromptLayerChatOpenAI	PromptLayer and OpenAI Chat large language models API.

Functions

chat_models.google_palm.achat_with_retry(...)	Use tenacity to retry the async completion call.
chat_models.google_palm.chat_with_retry(llm, ...)	Use tenacity to retry the completion call.
chat_models.jinachat.acompletion_with_retry(...)	Use tenacity to retry the async completion call.
chat_models.litellm.acompletion_with_retry(llm)	Use tenacity to retry the async completion call.
chat_models.openai.acompletion_with_retry(llm)	Use tenacity to retry the async completion call.

Toggle Menu

langchain.docstore

Docstores are classes to store and load Documents.

The **Docstore** is a simplified version of the Document Loader.

Class hierarchy:

Docstore --> <name> # Examples: InMemoryDocstore, Wikipedia

Main helpers:

Document, AddableMixin

Classes

docstore.base.AddableMixin()	Mixin class that supports adding texts.
docstore.base.Docstore()	Interface to access to place that stores documents.
docstore.wikipedia.Wikipedia()	Wrapper around wikipedia API.
docstore.arbitrary_fn.DocstoreFn(lookup_fn)	Langchain Docstore via arbitrary lookup function.
docstore.in_memory.InMemoryDocstore([_dict])	Simple in memory docstore in the form of a dict.

langchain.document loaders

Document Loaders are classes to load Documents.

Document Loaders are usually used to load a lot of Documents in a single run.

Class hierarchy:

BaseLoader --> <name>Loader # Examples: TextLoader, UnstructuredFileLoader
--

Main helpers:

Document, <name>TextSplitter

Classes

document_loaders.docugami.DocugamiLoader	Load from <i>Docugami</i> .
document_loaders.git.GitLoader(repo_path[, ...])	Load <i>Git</i> repository files.
document_loaders.url_selenium.SeleniumURLLoader(urls)	Load <i>HTML</i> pages with <i>Selenium</i> and parse with <i>Unstructured</i> .
document_loaders.cube_semantic.CubeSemanticLoader(...)	Load <i>Cube semantic layer</i> metadata.
document_loaders.azure_blob_storage_file.AzureBlobStorageFileLoader(...)	Load from <i>Azure Blob Storage</i> files.
document_loaders.powerpoint.UnstructuredPowerPointLoader(...)	Load <i>Microsoft PowerPoint</i> files using <i>Unstructured</i> .
document_loaders.psychic.PsychicLoader(...)	Load from <i>Psychic.dev</i> .
document_loaders.html.UnstructuredHTMLLoader(...)	Load <i>HTML</i> files using <i>Unstructured</i> .
document_loaders.spreadly.SpreadlyLoader(...)	Load from <i>Spreadly</i> API.
document_loaders.whatsapp_chat.WhatsAppChatLoader(path)	Load <i>WhatsApp</i> messages text file.
document_loaders.diffbot.DiffbotLoader(...)	Load <i>Diffbot</i> json file.
document_loaders.mastodon.MastodonTootsLoader(...)	Load the <i>Mastodon</i> 'toots'.
document_loaders.image.UnstructuredImageLoader(...)	Load <i>PNG</i> and <i>JPG</i> files using <i>Unstructured</i> .
document_loaders.roam.RoamLoader(path)	Load <i>Roam</i> files from a directory.
document_loaders.s3_directory.S3DirectoryLoader(bucket)	Load from <i>Amazon AWS S3</i> directory.
document_loaders.iugu.IuguLoader(resource[, ...])	Load from <i>IUGU</i> .
document_loaders.imsdb.IMSDBLoader(web_path)	Load <i>IMSDb</i> webpages.
document_loaders.gutenberg.GutenbergLoader(...)	Load from <i>Gutenberg.org</i> .
document_loaders.larksuite.LarkSuiteDocLoader(...)	Load from <i>LarkSuite (FeiShu)</i> .
document_loaders.directory.DirectoryLoader(...)	Load from a directory.
document_loaders.duckdb_loader.DuckDBLoader(query)	Load from <i>DuckDB</i> .
document_loaders.conllu.CoNLLULoader(file_path)	Load <i>CoNLL-U</i> files.
document_loaders.snowflake_loader.SnowflakeLoader(...)	Load from <i>Snowflake</i> API.

Toggle Menu

document_loaders.bigquery.BigQueryLoader (query)	Load from the Google Cloud Platform <i>BigQuery</i> .
document_loaders.datadog_logs.DatadogLogsLoader (...)	Load <i>Datadog</i> logs.
document_loaders.weather.WeatherDataLoader (...)	Load weather data with <i>Open Weather Map</i> API.
document_loaders.notebook.NotebookLoader (path)	Load <i>Jupyter notebook</i> (.ipynb) files.
document_loaders.pyspark_dataframe.PySparkDataFrameLoader ([...])	Load <i>PySpark</i> DataFrames.
document_loaders.chromium.AsyncChromiumLoader (urls)	Scrape HTML pages from URLs using a headless instance of the Chromium.
document_loaders.gcs_directory.GCSDirectoryLoader (...)	Load from GCS directory.
document_loaders.notiondb.NotionDBLoader (...)	Load from <i>Notion DB</i> .
document_loaders.rss.RSSFeedLoader ([urls, ...])	Load news articles from RSS feeds using <i>Unstructured</i> .
document_loaders.embaas.BaseEmbaasLoader	Base loader for <i>Embaas</i> document extraction API.
document_loaders.embaas.EmbaasBlobLoader	Load <i>Embaas</i> blob.
document_loaders.embaas.EmbaasDocumentExtractionParameters	Parameters for the embaas document extraction API.
document_loaders.embaas.EmbaasDocumentExtractionPayload	Payload for the Embaas document extraction API.
document_loaders.embaas.EmbaasLoader	Load from <i>Embaas</i> .
document_loaders.onedrive.OneDriveLoader	Load from <i>Microsoft OneDrive</i> .
document_loaders.ifixit.IFixitLoader (web_path)	Load <i>iFixit</i> repair guides, device wikis and answers.
document_loaders.discord.DiscordChatLoader (...)	Load <i>Discord</i> chat logs.
document_loaders.trello.TrelloLoader (client, ...)	Load cards from a <i>Trello</i> board.
document_loaders.etherscan.EtherscanLoader (...)	Load transactions from <i>Ethereum</i> mainnet.
document_loaders.url.UnstructuredURLLoader (urls)	Load files from remote URLs using <i>Unstructured</i> .
document_loaders.base.BaseBlobParser ()	Abstract interface for blob parsers.
document_loaders.base.BaseLoader ()	Interface for Document Loader.
document_loaders.evernote.EverNoteLoader (...)	Load from <i>EverNote</i> .
document_loaders.async_html.AsyncHtmlLoader (...)	Load <i>HTML</i> asynchronously.
document_loaders.wikipedia.WikipediaLoader (query)	Load from <i>Wikipedia</i> .
document_loaders.youtube.GoogleApiClient ([...])	Generic Google API Client.
document_loaders.youtube.GoogleApiYoutubeLoader (...)	Load all Videos from a <i>YouTube</i> Channel.
document_loaders.youtube.YoutubeLoader (video_id)	Load <i>YouTube</i> transcripts.
document_loaders.python.PythonLoader (file_path)	Load <i>Python</i> files, respecting any non-default encoding if specified.
document_loaders.blackboard.BlackboardLoader (...)	Load a <i>Blackboard</i> course.
document_loaders.pubmed.PubMedLoader (query)	Load from the <i>PubMed</i> biomedical library.
document_loaders.dataframe.DataFrameLoader (...)	Load <i>Pandas</i> DataFrame.
document_loaders.tensorflow_datasets.TensorflowDatasetLoader (...)	Load from <i>TensorFlow Dataset</i> .
document_loaders.web_base.WebBaseLoader (web_path)	Load HTML pages using <i>urllib</i> and parse them with ' <i>BeautifulSoup</i> '.
document_loaders.tencent_cos_file.TencentCOSFileLoader (...)	Load from <i>Tencent Cloud COS</i> file.
document_loaders.browserless.BrowserlessLoader (...)	Load webpages with <i>Browserless</i> /content endpoint.
document_loaders.helpers.FileEncoding (...)	File encoding as the NamedTuple.
document_loaders.srt.SRTLoader (file_path)	Load <i>.srt</i> (subtitle) files.
document_loaders.azure_blob_storage_container.AzureBlobStorageContainerLoader (...)	Load from <i>Azure Blob Storage</i> container.
document_loaders.open_city_data.OpenCityDataLoader (...)	Load from <i>Open City</i> .
document_loaders.mhtml.MHTMLLoader (file_path)	Parse <i>MHTML</i> files with <i>BeautifulSoup</i> .
document_loaders.html_bs.BSHTMLLoader (file_path)	Load <i>HTML</i> files and parse them with <i>beautiful soup</i> .
document_loaders.generic.GenericLoader (...)	Generic Document Loader.
document_loaders.recursive_url_loader.RecursiveUrlLoader (url)	Load all child links from a URL page.
document_loaders.twitter.TwitterTweetLoader (...)	Load <i>Twitter</i> tweets.
document_loaders.markdown.UnstructuredMarkdownLoader (...)	Load <i>Markdown</i> files using <i>Unstructured</i> .
document_loaders.merge.MergedDataLoader (loaders)	Merge documents from a list of loaders
document_loaders.github.BaseGithubLoader	Load <i>GitHub</i> repository Issues.
document_loaders.github.GitHubIssuesLoader	Load issues of a GitHub repository.
document_loaders.bibtex.BibtexLoader (...[, ...])	Load a <i>bibtex</i> file.
document_loaders.image_captions.ImageCaptionLoader (...)	Load image captions.
document_loaders.joplin.JoplinLoader ([...])	Load notes from <i>Joplin</i> .
document_loaders.concurrent.ConcurrentLoader (...)	Load and pars Documents concurrently.
document_loaders.modern_treasury.ModernTreasuryLoader (...)	Load from <i>Modern Treasury</i> .

Toggle Menu

document_loaders.sitemap.SitemapLoader (web_path)	Load a sitemap and its URLs.
document_loaders.pdf.AmazonTextractPDFLoader (...)	Load <i>PDF</i> files from a local file system, HTTP or S3.
document_loaders.pdf.BasePDFLoader (file_path)	Base Loader class for <i>PDF</i> files.
document_loaders.pdf.MathpixPDFLoader (file_path)	Load <i>PDF</i> files using <i>Mathpix</i> service.
document_loaders.pdf.OnLinePDFLoader (file_path)	Load online <i>PDF</i> .
document_loaders.pdf.PDFMinerLoader (file_path)	Load <i>PDF</i> files using <i>PDFMiner</i> .
document_loaders.pdf.PDFMinerPDFasHTMLLoader (...)	Load <i>PDF</i> files as HTML content using <i>PDFMiner</i> .
document_loaders.pdf.PDFPlumberLoader (file_path)	Load <i>PDF</i> files using <i>pdfplumber</i> .
document_loaders.pdf.PyMuPDFLoader (file_path)	Load <i>PDF</i> files using <i>PyMuPDF</i> .
document_loaders.pdf.PyPDFDirectoryLoader (path)	Load a directory with <i>PDF</i> files using <i>pypdf</i> and chunks at character level.
document_loaders.pdf.PyPDFLoader (file_path)	Load <i>PDF</i> using <i>`pypdf</i> and chunks at character level.
document_loaders.pdf.PyPDFium2Loader (file_path)	Load <i>PDF</i> using <i>pypdfium2</i> and chunks at character level.
document_loaders.pdf.UnstructuredPDFLoader (...)	Load <i>PDF</i> files using <i>Unstructured</i> .
document_loaders.brave_search.BraveSearchLoader (...)	Load with <i>Brave Search</i> engine.
document_loaders.tsv.UnstructuredTSVLoader (...)	Load <i>TSV</i> files using <i>Unstructured</i> .
document_loaders.dropbox.DropboxLoader	Load files from <i>Dropbox</i> .
document_loaders.s3_file.S3FileLoader (...)	Load from <i>Amazon AWS</i> S3 file.
document_loaders.max_compute.MaxComputeLoader (...)	Load from <i>Alibaba Cloud MaxCompute</i> table.
document_loaders.airbyte.AirbyteCDKLoader (...)	Load with an <i>Airbyte</i> source connector implemented using the <i>CDK</i> .
document_loaders.airbyte.AirbyteGongLoader (...)	Load from <i>Gong</i> using an <i>Airbyte</i> source connector.
document_loaders.airbyte.AirbyteHubspotLoader (...)	Load from <i>Hubspot</i> using an <i>Airbyte</i> source connector.
document_loaders.airbyte.AirbyteSalesforceLoader (...)	Load from <i>Salesforce</i> using an <i>Airbyte</i> source connector.
document_loaders.airbyte.AirbyteShopifyLoader (...)	Load from <i>Shopify</i> using an <i>Airbyte</i> source connector.
document_loaders.airbyte.AirbyteStripeLoader (...)	Load from <i>Stripe</i> using an <i>Airbyte</i> source connector.
document_loaders.airbyte.AirbyteTypeformLoader (...)	Load from <i>Typeform</i> using an <i>Airbyte</i> source connector.
document_loaders.airbyte.AirbyteZendeskSupportLoader (...)	Load from <i>Zendesk Support</i> using an <i>Airbyte</i> source connector.
document_loaders.xml.UnstructuredXMLLoader (...)	Load <i>XML</i> file using <i>Unstructured</i> .
document_loaders.nuclia.NucliaLoader (path, ...)	Load from any file type using <i>Nuclia Understanding API</i> .
document_loaders.toml.TomlLoader (source)	Load <i>TOML</i> files.
document_loaders.url_playwright.PlaywrightURLLoader (urls)	Load <i>HTML</i> pages with <i>Playwright</i> and parse with <i>Unstructured</i> .
document_loaders.unstructured.UnstructuredAPIFileIOLoader (file)	Load files using <i>Unstructured</i> API.
document_loaders.unstructured.UnstructuredAPIFileLoader ([...])	Load files using <i>Unstructured</i> API.
document_loaders.unstructured.UnstructuredBaseLoader ([...])	Base Loader that uses <i>Unstructured</i> .
document_loaders.unstructured.UnstructuredFileIOLoader (file)	Load files using <i>Unstructured</i> .
document_loaders.unstructured.UnstructuredFileLoader (...)	Load files using <i>Unstructured</i> .
document_loaders.gitbook.GitbookLoader (web_page)	Load <i>GitBook</i> data.
document_loaders.reddit.RedditPostsLoader (...)	Load <i>Reddit</i> posts.
document_loaders.slack_directory.SlackDirectoryLoader (...)	Load from a <i>Slack</i> directory dump.
document_loaders.excel.UnstructuredExcelLoader (...)	Load Microsoft Excel files using <i>Unstructured</i> .
document_loaders.rst.UnstructuredRSTLoader (...)	Load <i>RST</i> files using <i>Unstructured</i> .
document_loaders.acreom.AcreomLoader (path[, ...])	Load <i>acreom</i> vault from a directory.
document_loaders.obs_directory.OBSDirectoryLoader (...)	Load from <i>Huawei OBS directory</i> .
document_loaders.text.TextLoader (file_path)	Load text file.
document_loaders.figma.FigmaFileLoader (...)	Load <i>Figma</i> file.
document_loaders.obs_file.OBSFileLoader (...)	Load from the <i>Huawei OBS file</i> .
document_loaders.csv_loader.CSVLoader (file_path)	Load a CSV file into a list of Documents.
document_loaders.csv_loader.UnstructuredCSVLoader (...)	Load CSV files using <i>Unstructured</i> .
document_loaders.airbyte_json.AirbyteJSONLoader (...)	Load local <i>Airbyte</i> json files.
document_loaders.hugging_face_dataset.HuggingFaceDatasetLoader (path)	Load from <i>Hugging Face Hub</i> datasets.
document_loaders.arxiv.ArxivLoader (query[, ...])	Load a query result from <i>Arxiv</i> .
document_loaders.news.NewsURLLoader (urls[, ...])	Load news articles from URLs using <i>Unstructured</i> .

Toggle Menu

document_loaders.onedrive_file.OneDriveFileLoader	Load a file from <i>Microsoft OneDrive</i> .
document_loaders.azlyrics.AZLyricsLoader (...)	Load <i>AZLyrics</i> webpages.
document_loaders.apify_dataset.ApifyDatasetLoader	Load datasets from <i>Apify</i> web scraping, crawling, and data extraction platform.
document_loaders.facebook_chat.FacebookChatLoader (path)	Load <i>Facebook Chat</i> messages directory dump.
document_loaders.tomarkdown.ToMarkdownLoader (...)	Load <i>HTML</i> using <i>2markdown API</i> .
document_loaders.tencent_cos_directory.TencentCOSDirectoryLoader (...)	Load from <i>Tencent Cloud COS</i> directory.
document_loaders.json_loader.JSONLoader (...)	Load a <i>JSON</i> file using a <i>jq</i> schema.
document_loaders.stripe.StripeLoader (resource)	Load from <i>Stripe</i> API.
document_loaders.org_mode.UnstructuredOrgModeLoader (...)	Load <i>Org-Mode</i> files using <i>Unstructured</i> .
document_loaders.googledrive.GoogleDriveLoader	Load <i>Google Docs</i> from <i>Google Drive</i> .
document_loaders.odt.UnstructuredODTLoader (...)	Load <i>OpenOffice ODT</i> files using <i>Unstructured</i> .
document_loaders.email.OutlookMessageLoader (...)	Loads <i>Outlook Message</i> files using <code>extract_msg</code> .
document_loaders.email.UnstructuredEmailLoader (...)	Load email files using <i>Unstructured</i> .
document_loaders.gcs_file.GCSFileLoader (...)	Load from <i>GCS</i> file.
document_loaders.epub.UnstructuredEPubLoader (...)	Load <i>EPub</i> files using <i>Unstructured</i> .
document_loaders.notion.NotionDirectoryLoader (path)	Load <i>Notion</i> directory dump.
document_loaders.arcgis_loader.ArcGISLoader (layer)	Load records from an <i>ArcGIS FeatureLayer</i> .
document_loaders.geodataframe.GeoDataFrameLoader (...)	Load <i>geopandas</i> <i>Dataframe</i> .
document_loaders.mediawikidump.MWDumpLoader (...)	Load <i>MediaWiki</i> dump from an <i>XML</i> file.
document_loaders.blockchain.BlockchainDocumentLoader (...)	Load elements from a blockchain smart contract.
document_loaders.blockchain.BlockchainType (value)	Enumerator of the supported blockchains.
document_loaders.readthedocs.ReadTheDocsLoader (path)	Load <i>ReadTheDocs</i> documentation directory.
document_loaders.xorbits.XorbitsLoader (...)	Load <i>Xorbits</i> <i>DataFrame</i> .
document_loaders.college_confidential.CollegeConfidentialLoader (...)	Load <i>College Confidential</i> webpages.
document_loaders.chatgpt.ChatGPTLoader (log_file)	Load conversations from exported <i>ChatGPT</i> data.
document_loaders.fauna.FaunaLoader (query, ...)	Load from <i>FaunaDB</i> .
document_loaders.airtable.AirtableLoader (...)	Load the <i>Airtable</i> tables.
document_loaders.bilibili.BiliBiliLoader (...)	Load <i>BiliBili</i> video transcripts.
document_loaders.rocksetdb.ColumnNotFoundError (...)	Column not found error.
document_loaders.rocksetdb.RocksetLoader (...)	Load from a <i>Rockset</i> database.
document_loaders.confluence.ConfluenceLoader (url)	Load <i>Confluence</i> pages.
document_loaders.confluence.ContentFormat (value)	Enumerator of the content formats of <i>Confluence</i> page.
document_loaders.hn.HNLoader (web_path[, ...])	Load <i>Hacker News</i> data.
document_loaders.word_document.Docx2txtLoader (...)	Load <i>DOCX</i> file using <i>docx2txt</i> and chunks at character level.
document_loaders.word_document.UnstructuredWordDocumentLoader (...)	Load <i>Microsof Word</i> file using <i>Unstructured</i> .
document_loaders.obsidian.ObsidianLoader (path)	Load <i>Obsidian</i> files from directory.
document_loaders.telegram.TelegramChatApiLoader ([...])	Load <i>Telegram</i> chat json directory dump.
document_loaders.telegram.TelegramChatFileLoader (path)	Load from <i>Telegram chat</i> dump.
document_loaders.rtf.UnstructuredRTFLoader (...)	Load <i>RTF</i> files using <i>Unstructured</i> .
document_loaders.parsers.generic.MimeTypeBasedParser (...)	Parser that uses <i>mime</i> -types to parse a blob.
document_loaders.parsers.txt.TextParser ()	Parser for text blobs.
document_loaders.parsers.pdf.AmazonTextractPDFParser ([...])	Send <i>PDF</i> files to <i>Amazon Textract</i> and parse them.
document_loaders.parsers.pdf.PDFMinerParser ()	Parse <i>PDF</i> using <i>PDFMiner</i> .
document_loaders.parsers.pdf.PDFPlumberParser ([...])	Parse <i>PDF</i> with <i>PDFPlumber</i> .
document_loaders.parsers.pdf.PyMuPDFParser ([...])	Parse <i>PDF</i> using <i>PyMuPDF</i> .
document_loaders.parsers.pdf.PyPDFParser ([...])	Load <i>PDF</i> using <i>pypdf</i> and chunk at character level.
document_loaders.parsers.pdf.PyPDFium2Parser ()	Parse <i>PDF</i> with <i>PyPDFium2</i> .
document_loaders.parsers.audio.OpenAIWhisperParser ([...])	Transcribe and parse audio files.
document_loaders.parsers.audio.OpenAIWhisperParserLocal ([...])	Transcribe and parse audio files with <i>OpenAI Whisper</i> model.
document_loaders.parsers.grobid.GrobidParser (...)	Load article <i>PDF</i> files using <i>Grobid</i> .
document_loaders.parsers.grobid.ServerUnavailableException	Exception raised when the <i>Grobid</i> server is unavailable.
document_loaders.parsers.html.bs4.BS4HTMLParser (*)	Pparse <i>HTML</i> files using <i>Beautiful Soup</i> .

document_loaders.parsers.language.language_parser.LanguageParser ([...])	Parse using the respective programming language syntax.
document_loaders.parsers.language.python.PythonSegmenter (code)	Code segmenter for <i>Python</i> .
document_loaders.parsers.language.code_segmenter.CodeSegmenter (code)	Abstract class for the code segmenter.
document_loaders.parsers.language.javascript.JavaScriptSegmenter (code)	Code segmenter for JavaScript.
document_loaders.blob_loaders.file_system.FileSystemBlobLoader (path, *)	Load blobs in the local file system.
document_loaders.blob_loaders.schema.Blob	Blob represents raw data by either reference or value.
document_loaders.blob_loaders.schema.BlobLoader ()	Abstract interface for blob loaders implementation.
document_loaders.blob_loaders.youtube_audio.YoutubeAudioLoader (...)	Load YouTube urls as audio file(s).

Functions

document_loaders.chatgpt.concatenate_rows (...)	Combine message information in a readable format ready to be used.
document_loaders.facebook_chat.concatenate_rows (row)	Combine message information in a readable format ready to be used.
document_loaders.helpers.detect_file_encodings (...)	Try to detect the file encoding.
document_loaders.notebook.concatenate_cells (...)	Combine cells information in a readable format ready to be used.
document_loaders.notebook.remove_newlines (x)	Recursively remove newlines, no matter the data structure they are stored in.
document_loaders.parsers.registry.get_parser (...)	Get a parser by parser name.
document_loaders.rocksetdb.default_joiner (docs)	Default joiner for content columns.
document_loaders.telegram.concatenate_rows (row)	Combine message information in a readable format ready to be used.
document_loaders.telegram.text_to_docs (text)	Convert a string or list of strings to a list of Documents with metadata.
document_loaders.unstructured.get_elements_from_api ([...])	Retrieve a list of elements from the <i>Unstructured API</i> .
document_loaders.unstructured.satisfies_min_unstructured_version (...)	Check if the installed <i>Unstructured</i> version exceeds the minimum version for the feature in question.
document_loaders.unstructured.validate_unstructured_version (...)	Raise an error if the <i>Unstructured</i> version does not exceed the specified minimum.
document_loaders.whatsapp_chat.concatenate_rows (...)	Combine message information in a readable format ready to be used.

langchain.document transformers

Document Transformers are classes to transform Documents.

Document Transformers usually used to transform a lot of Documents in a single run.

Class hierarchy:

BaseDocumentTransformer --> <name> # Examples: DoctranQATransformer, DoctranTextTranslator
--

Main helpers:

Document

Classes

document_transformers.long_context_reorder.LongContextReorder	Lost in the middle: Performance degrades when models must access relevant information in the middle of long contexts.
document_transformers.doctran text translate.DoctranTextTranslator ([...])	Translate text documents using doctran.
document_transformers.embeddings_redundant_filter.EmbeddingsClusteringFilter	Perform K-means clustering on document vectors.
document_transformers.embeddings_redundant_filter.EmbeddingsRedundantFilter	Filter that drops redundant documents by comparing their embeddings.
document_transformers.doctran_text_qa.DoctranQATransformer ([...])	Extract QA from text documents using doctran.
document_transformers.doctran_text_extract.DoctranPropertyExtractor (...)	Extract properties from text documents using doctran.
document_transformers.openai_functions.OpenAIMetadataTagger	Extract metadata tags from document contents using OpenAI functions.
document_transformers.html2text.Html2TextTransformer ()	Replace occurrences of a particular search pattern with a replacement string .

Toggle Menu

document_transformers.nuclia_text_transform.NucliaTextTransformer (nua)	The Nuclia Understanding API splits into paragraphs and sentences, identifies entities, provides a summary of the text and generates embeddings for all sentences.
document_transformers.beautiful_soup_transformer.BeautifulSoupTransformer ()	Transform HTML content by extracting specific tags and removing unwanted ones.
Functions	
document_transformers.embeddings_redundant_filter.get_stateful_documents (...)	Convert a list of documents to a list of documents with state.
document_transformers.openai_functions.create_metadata_tagger (...)	Create a DocumentTransformer that uses an OpenAI function chain to automatically

langchain.embeddings

Embedding models are wrappers around embedding models from different APIs and services.

Embedding models can be LLMs or not.

Class hierarchy:

Embeddings --> <name>Embeddings # Examples: OpenAIEmbeddings, HuggingFaceEmbeddings
--

Classes

embeddings.jina.JinaEmbeddings	Jina embedding models.
embeddings.openai.OpenAIEmbeddings	OpenAI embedding models.
embeddings.cohere.CohereEmbeddings	Cohere embedding models.
embeddings.cache.CacheBackedEmbeddings (...)	Interface for caching results from embedding models.
embeddings.gpt4all.GPT4AllEmbeddings	GPT4All embedding models.
embeddings.mosaicml.MosaicMLInstructorEmbeddings	MosaicML embedding service.
embeddings.dashscope.DashScopeEmbeddings	DashScope embedding models.
embeddings.embaas.EmbaasEmbeddings	Embaas's embedding service.
embeddings.embaas.EmbaasEmbeddingsPayload	Payload for the Embaaas embeddings API.
embeddings.aleph_alpha.AlephAlphaAsymmetricSemanticEmbedding	Aleph Alpha's asymmetric semantic embedding.
embeddings.aleph_alpha.AlephAlphaSymmetricSemanticEmbedding	The symmetric version of the Aleph Alpha's semantic embeddings.
embeddings.clarifai.ClarifaiEmbeddings	Clarifai embedding models.
embeddings.base.Embeddings ()	Interface for embedding models.
embeddings.vertexai.VertexAIEmbeddings	Google Cloud VertexAI embedding models.
embeddings.bedrock.BedrockEmbeddings	Bedrock embedding models.
embeddings.self_hosted_hugging_face.SelfHostedHuggingFaceEmbeddings	HuggingFace embedding models on self-hosted remote hardware.
embeddings.self_hosted_hugging_face.SelfHostedHuggingFaceInstructEmbeddings	HuggingFace InstructEmbedding models on self-hosted remote hardware.
embeddings.spacy_embeddings.SpacyEmbeddings	Embeddings by SpaCy models.
embeddings.mlflow_gateway.MlflowAIGatewayEmbeddings	Wrapper around embeddings LLMs in the MLflow AI Gateway.
embeddings.modelscope_hub.ModelScopeEmbeddings	ModelScopeHub embedding models.
embeddings.minimax.MiniMaxEmbeddings	MiniMax's embedding service.
embeddings.tensorflow_hub.TensorflowHubEmbeddings	TensorflowHub embedding models.
embeddings.elasticsearch.ElasticsearchEmbeddings (...)	Elasticsearch embedding models.
embeddings.awa.AwaEmbeddings	Embedding documents and queries with Awa DB.
embeddings.octoai_embeddings.OctoAIEmbeddings	OctoAI Compute Service embedding models.
embeddings.huggingface.HuggingFaceBgeEmbeddings	HuggingFace BGE sentence_transformers embedding models.
embeddings.huggingface.HuggingFaceEmbeddings	HuggingFace sentence_transformers embedding models.
embeddings.huggingface.HuggingFaceInstructEmbeddings	Wrapper around sentence_transformers embedding models.
embeddings.xinference.XinferenceEmbeddings ([...])	Wrapper around xinference embedding models.
embeddings.google_palm.GooglePalmEmbeddings	Google's PaLM Embeddings APIs.
embeddings.sagemaker_endpoint.EmbeddingsContentHandler ()	Content handler for LLM class.
embeddings.sagemaker_endpoint.SagemakerEndpointEmbeddings	Custom Sagemaker Inference Endpoints.
embeddings.eepinfra.DeepInfraEmbeddings	Deep Infra's embedding inference service.

Toggle Menu

embeddings.huggingface_hub.HuggingFaceHubEmbeddings	HuggingFaceHub embedding models.
embeddings.edenai.EdenAiEmbeddings	EdenAI embedding.
embeddings.localai.LocalAIEmbeddings	LocalAI embedding models.
embeddings.nlpcloud.NLPCloudEmbeddings	NLP Cloud embedding models.
embeddings.fake.DeterministicFakeEmbedding	Fake embedding model that always returns the same embedding vector for the same text.
embeddings.fake.FakeEmbeddings	Fake embedding model.
embeddings.self_hosted.SelfHostedEmbeddings	Custom embedding models on self-hosted remote hardware.
embeddings.llamacpp.LlamaCppEmbeddings	llama.cpp embedding models.

Functions

embeddings.dashscope.embed_with_retry(...)	Use tenacity to retry the embedding call.
embeddings.google_palm.embed_with_retry(...)	Use tenacity to retry the completion call.
embeddings.localai.async_embed_with_retry(...)	Use tenacity to retry the embedding call.
embeddings.localai.embed_with_retry(...)	Use tenacity to retry the embedding call.
embeddings.minimax.embed_with_retry(...)	Use tenacity to retry the completion call.
embeddings.openai.async_embed_with_retry(...)	Use tenacity to retry the embedding call.
embeddings.openai.embed_with_retry(...)	Use tenacity to retry the embedding call.
embeddings.self_hosted_hugging_face.load_embedding_model(...)	Load the embedding model.

langchain.evaluation

Evaluation chains for grading LLM and Chain outputs.

This module contains off-the-shelf evaluation chains for grading the output of LangChain primitives such as language models and chains.

Loading an evaluator

To load an evaluator, you can use the [load_evaluators](#) or [load_evaluator](#) functions with the names of the evaluators to load.

```
from langchain.evaluation import load_evaluator

evaluator = load_evaluator("qa")
evaluator.evaluate_strings(
    prediction="We sold more than 40,000 units last week",
    input="How many units did we sell last week?",
    reference="We sold 32,378 units",
)
```

The evaluator must be one of [EvaluatorType](#).

Datasets

To load one of the LangChain HuggingFace datasets, you can use the [load_dataset](#) function with the name of the dataset to load.

```
from langchain.evaluation import load_dataset
ds = load_dataset("llm-math")
```

Some common use cases for evaluation include:

- Grading the accuracy of a response against ground truth answers: [QAEvalChain](#)
- Comparing the output of two models: [PairwiseStringEvalChain](#) or [LabeledPairwiseStringEvalChain](#) when there is additionally a reference label.
- Judging the efficacy of an agent's tool usage: [TrajectoryEvalChain](#)
- Checking whether an output complies with a set of criteria: [CriteriaEvalChain](#) or [LabeledCriteriaEvalChain](#) when there is additionally a reference label.
- Computing semantic difference between a prediction and reference: [EmbeddingDistanceEvalChain](#) or between two predictions: [PairwiseEmbeddingDistanceEvalChain](#)
- Measuring the string distance between a prediction and reference [StringDistanceEvalChain](#) or between two predictions [PairwiseStringDistanceEvalChain](#)

Low-level API

These evaluators implement one of the following interfaces:

Evaluator: Evaluate a prediction string against a reference label and/or input context.

Toggle Menu

- [PairwiseStringEvaluator](#): Evaluate two prediction strings against each other. Useful for scoring preferences, measuring similarity between two chain or llm agents, or comparing outputs on similar inputs.
- [AgentTrajectoryEvaluator](#) Evaluate the full sequence of actions taken by an agent.

These interfaces enable easier composability and usage within a higher level evaluation framework.

Classes

evaluation.schema.AgentTrajectoryEvaluator()	Interface for evaluating agent trajectories.
evaluation.schema.EvaluatorType(value[, ...])	The types of the evaluators.
evaluation.schema.LLMEvalChain	A base class for evaluators that use an LLM.
evaluation.schema.PairwiseStringEvaluator()	Compare the output of two models (or two outputs of the same model).
evaluation.schema.StringEvaluator()	Grade, tag, or otherwise evaluate predictions relative to their inputs and/or reference labels.
evaluation.criteria.eval_chain.Criteria(value)	A Criteria to evaluate.
evaluation.criteria.eval_chain.CriteriaEvalChain	LLM Chain for evaluating runs against criteria.
evaluation.criteria.eval_chain.CriteriaResultOutputParser	A parser for the output of the CriteriaEvalChain.
evaluation.criteria.eval_chain.LabeledCriteriaEvalChain	Criteria evaluation chain that requires references.
evaluation.parsing.base.JsonEqualityEvaluator([...])	Evaluates whether the prediction is equal to the reference after
evaluation.parsing.base.JsonValidityEvaluator(...)	Evaluates whether the prediction is valid JSON.
evaluation.qa.generate_chain.QAGenerateChain	LLM Chain for generating examples for question answering.
evaluation.qa.eval_chain.ContextQAEvalChain	LLM Chain for evaluating QA w/o GT based on context
evaluation.qa.eval_chain.CotQAEvalChain	LLM Chain for evaluating QA using chain of thought reasoning.
evaluation.qa.eval_chain.QAEvalChain	LLM Chain for evaluating question answering.
evaluation.agents.trajectory_eval_chain.TrajectoryEval	A named tuple containing the score and reasoning for a trajectory.
evaluation.agents.trajectory_eval_chain.TrajectoryEvalChain	A chain for evaluating ReAct style agents.
evaluation.agents.trajectory_eval_chain.TrajectoryOutputParser	Trajectory output parser.
evaluation.comparison.eval_chain.LabeledPairwiseStringEvalChain	A chain for comparing two outputs, such as the outputs
evaluation.comparison.eval_chain.PairwiseStringEvalChain	A chain for comparing two outputs, such as the outputs
evaluation.comparison.eval_chain.PairwiseStringResultOutputParser	A parser for the output of the PairwiseStringEvalChain.
evaluation.embedding_distance.base.EmbeddingDistance(value)	Embedding Distance Metric.
evaluation.embedding_distance.base.EmbeddingDistanceEvalChain	Use embedding distances to score semantic difference between a prediction and reference.
evaluation.embedding_distance.base.PairwiseEmbeddingDistanceEvalChain	Use embedding distances to score semantic difference between two predictions.
evaluation.string_distance.base.PairwiseStringDistanceEvalChain	Compute string edit distances between two predictions.
evaluation.string_distance.base.StringDistance(value)	Distance metric to use.
evaluation.string_distance.base.StringDistanceEvalChain	Compute string distances between the prediction and the reference.

Functions

evaluation.comparison.eval_chain.resolve_pairwise_criteria(...)	Resolve the criteria for the pairwise evaluator.
evaluation.criteria.eval_chain.resolve_criteria(...)	Resolve the criteria to evaluate.
evaluation.loading.load_dataset(uri)	Load a dataset from the LangChainDatasets on HuggingFace .
evaluation.loading.load_evaluator(evaluator, *)	Load the requested evaluation chain specified by a string.
evaluation.loading.load_evaluators(evaluators, *)	Load evaluators specified by a list of evaluator types.

[langchain.graphs](#)

Graphs provide a natural language interface to graph databases.

Classes

graphs.networkx_graph.KnowledgeTriple(...)	A triple in the graph.
graphs.networkx_graph.NetworkxEntityGraph([graph])	Networkx wrapper for entity graph operations.
graphs.kuzu_graph.KuzuGraph(db[, database])	Kùzu wrapper for graph operations.
graphs.rdf_graph.RdfGraph([source_file, ...])	RDFlib wrapper for graph operations.
<div><div>Toggle Menu</div><div>a_graph.NebulaGraph(space[, ...])</div></div>	NebulaGraph wrapper for graph operations NebulaGraph inherits methods from Neo4jGraph to bring ease to the user space.

graphs.hugegraph.HugeGraph ([username, ...])	HugeGraph wrapper for graph operations
graphs.memgraph_graph.MemgraphGraph (url, ...)	Memgraph wrapper for graph operations.
graphs.neo4j_graph.Neo4jGraph (url, username, ...)	Neo4j wrapper for graph operations.
graphs.arangodb_graph.ArangoGraph (db)	ArangoDB wrapper for graph operations.
graphs.neptune_graph.NeptuneGraph (host[, ...])	Neptune wrapper for graph operations.
graphs.neptune_graph.NeptuneQueryException (...)	A class to handle queries that fail to execute

Functions

graphs.arangodb_graph.get_arangodb_client ([...])	Get the Arango DB client from credentials.
graphs.networkx_graph.get_entities (entity_str)	Extract entities from entity string.
graphs.networkx_graph.parse_triples (...)	Parse knowledge triples from the knowledge string.

langchain.hub

Push and pull to the LangChain Hub.

Functions

hub.pull (owner_repo_commit, *, [, api_url, ...])	Pulls an object from the hub and returns it.
hub.push (repo_full_name, object, *, [, ...])	Pushes an object to the hub and returns the URL.

langchain.indexes

Index utilities.

Classes

indexes.vectorstore.VectorStoreIndexWrapper	Wrapper around a vectorstore for easy access.
indexes.vectorstore.VectorstoreIndexCreator	Logic for creating indexes.
indexes.graph.GraphIndexCreator	Functionality to create graph index.

Functions

langchain.llms

LLM classes provide access to the large language model (LLM) APIs and services.

Class hierarchy:

BaseLanguageModel --> BaseLLM --> LLM --> <name> # Examples: AI21, HuggingFaceHub, OpenAI

Main helpers:

LLMResult, PromptValue, CallbackManagerForLLMRun, AsyncCallbackManagerForLLMRun, CallbackManager, AsyncCallbackManager, AIMessage, BaseMessage
--

Classes

llms.openai.AzureOpenAI	Azure-specific OpenAI large language models.
llms.openai.BaseOpenAI	Base OpenAI large language model class.
llms.openai.OpenAI	OpenAI large language models.
llms.openai.OpenAIChat	OpenAI Chat large language models.
llms.cohere.Cohere	Cohere large language models.
api_gateway.AmazonAPIGateway	Amazon API Gateway to access LLM models hosted on AWS.

Toggle Menu

llms.amazon_api_gateway.ContentHandlerAmazonAPIGateway()	Adapter to prepare the inputs from Langchain to a format that LLM model expects.
llms.manifest.ManifestWrapper	HazyResearch's Manifest library.
llms.huggingface_pipeline.HuggingFacePipeline	HuggingFace Pipeline API.
llms.baseten.Baseten	Baseten models.
llms.huggingface_text_gen_inference.HuggingFaceTextGenInference	HuggingFace text generation API.
llms.human.HumanInputLLM	It returns user input as the response.
llms.databricks.Databricks	Databricks serving endpoint or a cluster driver proxy app for LLM.
llms.gpt4all.GPT4All	GPT4All language models.
llms.bittensor.NIBittensorLLM	NIBittensorLLM is created by Neural Internet (https://neuralinternet.ai/), powered by Bittensor, a decentralized network full of different AI models.
llms.titan_takeoff.TitanTakeoff	Create a new model by parsing and validating input data from keyword arguments.
llms.openlm.OpenLM	OpenLM models.
llms.mosaicml.MosaicML	MosaicML LLM service.
llms.azureml_endpoint.AzureMLEndpointClient(...)	AzureML Managed Endpoint client.
llms.azureml_endpoint.AzureMLOnlineEndpoint	Azure ML Online Endpoint models.
llms.azureml_endpoint.ContentFormatterBase()	Transform request and response of AzureML endpoint to match with required schema.
llms.azureml_endpoint.DollyContentFormatter()	Content handler for the Dolly-v2-12b model
llms.azureml_endpoint.GPT2ContentFormatter()	Content handler for GPT2
llms.azureml_endpoint.HFContentFormatter()	Content handler for LLMs from the HuggingFace catalog.
llms.azureml_endpoint.LlamaContentFormatter()	Content formatter for LLaMa
llms.azureml_endpoint.OSSContentFormatter()	Deprecated: Kept for backwards compatibility
llms.aleph_alpha.AlephAlpha	Aleph Alpha large language models.
llms.predibase.Predibase	Use your Predibase models with Langchain.
llms.clarifai.Clarifai	Clarifai large language models.
llms.base.BaseLLM	Base LLM abstract interface.
llms.base.LLM	Base LLM abstract class.
llms.vertexai.VertexAI	Google Vertex AI large language models.
llms.bedrock.Bedrock	Bedrock models.
llms.bedrock.LLMInputOutputAdapter()	Adapter class to prepare the inputs from Langchain to a format that LLM model expects.
llms.self_hosted_hugging_face.SelfHostedHuggingFaceLLM	HuggingFace Pipeline API to run on self-hosted remote hardware.
llms.bananadev.Banana	Banana large language models.
llms.minimax.Minimax	Wrapper around Minimax large language models.
llms.beam.Beam	Beam API for gpt2 large language model.
llms.ctransformers.CTransformers	C Transformers LLM models.
llms.xinference.Xinference	Wrapper for accessing Xinference's large-scale model inference service.
llms.google_palm.GooglePaLm	Google PaLM models.
llms.predictionguard.PredictionGuard	Prediction Guard large language models.
llms.sagemaker_endpoint.ContentHandlerBase()	A handler class to transform input from LLM to a format that SageMaker endpoint expects.
llms.sagemaker_endpoint.LLMContentHandler()	Content handler for LLM class.
llms.sagemaker_endpoint.SagemakerEndpoint	Sagemaker Inference Endpoint models.
llms.ai21.AI21	AI21 large language models.
llms.ai21.AI21PenaltyData	Parameters for AI21 penalty data.
llms.forefrontai.ForefrontAI	ForefrontAI large language models.
llms.koboldai.KoboldApiLLM	Kobold API language model.
llms.deepinfra.DeepInfra	DeepInfra models.
llms.anthropic.Anthropic	Anthropic large language models.
llms.modal.Modal	Modal large language models.
llms.symblai_nebula.Nebula	Nebula Service models.
llms.aviary.Aviary	Aviary hosted models.
llms.aviary.AviaryBackend(backend_url, bearer)	Aviary backend.
llms.huggingface_hub.HuggingFaceHub	HuggingFaceHub models.
llms.huggingface_endpoint.HuggingFaceEndpoint	HuggingFace Endpoint models.
llms.vllm.VLLM	VLLM language model.
llms.vllm.VLLMOpenAI	vLLM OpenAI-compatible API client
llms.tongyi.Tongyi	Tongyi Qwen large language models.
llms.ollama.Ollama	Ollama locally run large language models.
llms.petals.Petals	Petals Bloom models.
llms.petals.PetalsEndpoint.OctoAIEndpoint	OctoAI LLM Endpoints.

Toggle Menu

llms.edenai.EdenAI	Wrapper around edenai models.
llms.mlflow_ai_gateway.MlflowAIGateway	Wrapper around completions LLMs in the MLflow AI Gateway.
llms.mlflow_ai_gateway.Params	Parameters for the MLflow AI Gateway LLM.
llms.cerebriumai.CerebriumAI	CerebriumAI large language models.
llms.rwkv.RWKV	RWKV language models.
llms.openllm.IdentifyingParams	Parameters for identifying a model as a typed dict.
llms.openllm.OpenLLM	OpenLLM, supporting both in-process model instance and remote OpenLLM servers.
llms.anyscale.Anyscale	Anyscale Service models.
llms.writer.Writer	Writer large language models.
llms.textgen.TextGen	text-generation-webui models.
llms.deepsparse.DeepSparse	Neural Magic DeepSparse LLM interface.
llms.gooseai.GooseAI	GooseAI large language models.
llms.nlpcloud.NLPCloud	NLPCloud large language models.
llms.chatglm.ChatGLM	ChatGLM LLM service.
llms.stochasticai.StochasticAI	StochasticAI large language models.
llms.pipelineai.PipelineAI	PipelineAI large language models.
llms.fireworks.BaseFireworks	Wrapper around Fireworks large language models.
llms.fireworks.Fireworks	Wrapper around Fireworks large language models.
llms.fireworks.FireworksChat	Wrapper around Fireworks Chat large language models.
llms.fake.FakeListLLM	Fake LLM for testing purposes.
llms.fake.FakeStreamingListLLM	Fake streaming list LLM for testing purposes.
llms.promptlayer_openai.PromptLayerOpenAI	PromptLayer OpenAI large language models.
llms.promptlayer_openai.PromptLayerOpenAIChat	Wrapper around OpenAI large language models.
llms.self_hosted.SelfHostedPipeline	Model inference on self-hosted remote hardware.
llms.replicate.Replicate	Replicate models.
llms.llamacpp.LlamaCpp	llama.cpp model.

Functions

llms.aviary.get_completions (model, prompt[, ...])	Get completions from Aviary models.
llms.aviary.get_models ()	List available models
llms.base.create_base_retry_decorator (...[, ...])	Create a retry decorator for a given LLM and provided list of error types.
llms.base.get_prompts (params, prompts)	Get prompts that are already cached.
llms.base.update_cache (existing_prompts, ...)	Update the cache and get the LLM output.
llms.cohere.acompletion_with_retry (llm, **kwargs)	Use tenacity to retry the completion call.
llms.cohere.completion_with_retry (llm, **kwargs)	Use tenacity to retry the completion call.
llms.databricks.get_default_api_token ()	Gets the default Databricks personal access token.
llms.databricks.get_default_host ()	Gets the default Databricks workspace hostname.
llms.databricks.get_repl_context ()	Gets the notebook REPL context if running inside a Databricks notebook.
llms.fireworks.acompletion_with_retry (llm, ...)	Use tenacity to retry the async completion call.
llms.fireworks.completion_with_retry (llm, ...)	Use tenacity to retry the completion call.
llms.fireworks.execute (prompt, model, api_key)	Execute LLM query
llms.fireworks.update_token_usage (keys, ...)	Update token usage.
llms.google_palm.generate_with_retry (llm, ...)	Use tenacity to retry the completion call.
llms.koboldai.clean_url (url)	Remove trailing slash and /api from url if present.
llms.loading.load_llm (file)	Load LLM from file.
llms.loading.load_llm_from_config (config)	Load LLM from Config Dict.
llms.openai.acompletion_with_retry (llm[, ...])	Use tenacity to retry the async completion call.
llms.openai.completion_with_retry (llm[, ...])	Use tenacity to retry the completion call.
llms.openai.update_token_usage (keys, ...)	Update token usage.
llms.symblai_nebula.completion_with_retry (...)	Use tenacity to retry the completion call.
llms.symblai_nebula.make_request (self, ...)	Generate text from the model.
llms.tongyi.generate_with_retry (llm, **kwargs)	Use tenacity to retry the completion call.
llms.tongyi.stream_generate_with_retry (llm, ...)	Use tenacity to retry the completion call.
llms.utils.enforce_stop_tokens (text, stop)	Cut off the text as soon as any stop words occur.
llms.vertexai.completion_with_retry (llm, ...)	Use tenacity to retry the completion call.
llms.vertexai.is_codey_model (model_name)	Returns True if the model name is a Codey model.

langchain.load

Serialization and deserialization.

Toggle Menu

Classes

load.serializable.BaseSerialized	Base class for serialized objects.
load.serializable.Serializable	Serializable base class.
load.serializable.SerializedConstructor	Serialized constructor.
load.serializable.SerializedNotImplemented	Serialized not implemented.
load.serializable.SerializedSecret	Serialized secret.
load.load.Reviver ([secrets_map, ...])	Reviver for JSON objects.

Functions

load.dump.default (obj)	Return a default value for a Serializable object or a SerializedNotImplemented object.
load.dump.dumpd (obj)	Return a json dict representation of an object.
load.dump.dumps (obj, *[, pretty])	Return a json string representation of an object.
load.load.load (obj, *[, secrets_map, ...])	Revive a LangChain class from a JSON object.
load.load.loads (text, *[, secrets_map, ...])	Revive a LangChain class from a JSON string.
load.serializable.to_json_not_implemented (obj)	Serialize a "not implemented" object.

[langchain.memory](#)

Memory maintains Chain state, incorporating context from past runs.

Class hierarchy for Memory:

BaseMemory --> BaseChatMemory --> <name>Memory <i># Examples: ZepMemory, MotorheadMemory</i>
--

Main helpers:

BaseChatMessageHistory

Chat Message History stores the chat message history in different stores.

Class hierarchy for ChatMessageHistory:

BaseChatMessageHistory --> <name>ChatMessageHistory <i># Example: ZepChatMessageHistory</i>

Main helpers:

AIMessage, BaseMessage, HumanMessage

Classes

memory.summary.ConversationSummaryMemory	Conversation summarizer to chat memory.
memory.summary.SummarizerMixin	Mixin for summarizer.
memory.vectorstore.VectorStoreRetrieverMemory	VectorStoreRetriever-backed memory.
memory.buffer_window.ConversationBufferWindowMemory	Buffer for storing conversation memory inside a limited size window.
memory.combined.CombinedMemory	Combining multiple memories' data together.
memory.summary_buffer.ConversationSummaryBufferMemory	Buffer with summarizer for storing conversation memory.
memory.kg.ConversationKGMemory	Knowledge graph conversation memory.
memory.motorhead_memory.MotorheadMemory	Chat message memory backed by Motorhead service.
memory.zep_memory.ZepMemory	Persist your chain history to the Zep Memory Server.
memory.entity.BaseEntityStore	Abstract base class for Entity store.
memory.entity.ConversationEntityMemory	Entity extractor & summarizer memory.
memory.entity.InMemoryEntityStore	In-memory Entity store.
memory.entity.RedisEntityStore	Redis-backed Entity store.
memory.entity.SQLiteEntityStore	SQLite-backed Entity store
memory.token_buffer.ConversationTokenBufferMemory	Conversation chat memory with token limit.
memory.buffer.ConversationBufferMemory	Buffer for storing conversation memory.
memory.buffer.ConversationStringBufferMemory	Buffer for storing conversation memory.
memory.chat_memory.BaseChatMemory	Abstract base class for chat memory.

Toggle Menu

memory.simple.SimpleMemory	Simple memory for storing context or other information that shouldn't ever change between prompts.
memory.readonly.ReadOnlySharedMemory	A memory wrapper that is read-only and cannot be changed.
memory.chat_message_histories.cosmos_db.CosmosDBChatMessageHistory(...)	Chat message history backed by Azure CosmosDB.
memory.chat_message_histories.mongodb.MongoDBChatMessageHistory(...)	Chat message history that stores history in MongoDB.
memory.chat_message_histories.cassandra.CassandraChatMessageHistory(...)	Chat message history that stores history in Cassandra.
memory.chat_message_histories.postgres.PostgresChatMessageHistory(...)	Chat message history stored in a Postgres database.
memory.chat_message_histories.zep.ZepChatMessageHistory(...)	Chat message history that uses Zep as a backend.
memory.chat_message_histories.dynamodb.DynamoDBChatMessageHistory(...)	Chat message history that stores history in AWS DynamoDB.
memory.chat_message_histories.momento.MomentoChatMessageHistory(...)	Chat message history cache that uses Momento as a backend.
memory.chat_message_histories.file.FileChatMessageHistory(...)	Chat message history that stores history in a local file.
memory.chat_message_histories.streamlit.StreamlitChatMessageHistory([key])	Chat message history that stores messages in Streamlit session state.
memory.chat_message_histories.in_memory.ChatMessageHistory	In memory implementation of chat message history.
memory.chat_message_histories.firestore.FirestoreChatMessageHistory(...)	Chat message history backed by Google Firestore.
memory.chat_message_histories.redis.RedisChatMessageHistory(...)	Chat message history stored in a Redis database.
memory.chat_message_histories.sql.SQLChatMessageHistory(...)	Chat message history stored in an SQL database.
memory.chat_message_histories.rocksetdb.RocksetChatMessageHistory(...)	Uses Rockset to store chat messages.

Functions

memory.chat_message_histories.sql.create_message_model(...)	Create a message model for a given table name.
memory.utils.get_prompt_input_key(inputs, ...)	Get the prompt input key.

[langchain.model_laboratory](#)

Experiment with different models.

Classes

model_laboratory.ModelLaboratory(chains[, names])	Experiment with different models.
---	-----------------------------------

[langchain.output_parsers](#)

OutputParser classes parse the output of an LLM call.

Class hierarchy:

BaseLLMOutputParser --> BaseOutputParser --> <name>OutputParser # <i>ListOutputParser, PydanticOutputParser</i>

Main helpers:

Serializable, Generation, PromptValue

Classes

output_parsers.list.CommaSeparatedListOutputParser	Parse the output of an LLM call to a comma-separated list.
output_parsers.list.ListOutputParser	Parse the output of an LLM call to a list.
output_parsers.boolean.BooleanOutputParser	Parse the output of an LLM call to a boolean.
output_parsers.datetime.DatetimeOutputParser	Parse the output of an LLM call to a datetime.
output_parsers.combining.CombiningOutputParser	Combine multiple output parsers into one.
output_parsers.regex.RegexParser	Parse the output of an LLM call using a regex.
output_parsers.fix.OutputFixingParser	Wraps a parser and tries to fix parsing errors.
output_parsers.regex_dict.RegexDictParser	Parse the output of an LLM call into a Dictionary using a regex.
output_parsers.structured.ResponseSchema	A schema for a response from a structured output parser.

Toggle Menu

output_parsers.structured.StructuredOutputParser	Parse the output of an LLM call to a structured output.
output_parsers.openai_functions.JsonKeyOutputFunctionsParser	Parse an output as the element of the Json object.
output_parsers.openai_functions.JsonOutputFunctionsParser	Parse an output as the Json object.
output_parsers.openai_functions.OutputFunctionsParser	Parse an output that is one of sets of values.
output_parsers.openai_functions.PydanticAttrOutputFunctionsParser	Parse an output as an attribute of a pydantic object.
output_parsers.openai_functions.PydanticOutputFunctionsParser	Parse an output as a pydantic object.
output_parsers.rail_parser.GuardrailsOutputParser	Parse the output of an LLM call using Guardrails.
output_parsers.json.SimpleJsonOutputParser	Parse the output of an LLM call to a JSON object.
output_parsers.pydantic.PydanticOutputParser	Parse an output using a pydantic model.
output_parsers.enum.EnumOutputParser	Parse an output that is one of a set of values.
output_parsers.retry.RetryOutputParser	Wraps a parser and tries to fix parsing errors.
output_parsers.retry.RetryWithErrorOutputParser	Wraps a parser and tries to fix parsing errors.

Functions

output_parsers.json.parse_and_check_json_markdown(...)	Parse a JSON string from a Markdown string and check that it contains the expected keys.
output_parsers.json.parse_json_markdown(...)	Parse a JSON string from a Markdown string.
output_parsers.loading.load_output_parser(config)	Load an output parser.

[langchain.prompts](#)

Prompt is the input to the model.

Prompt is often constructed from multiple components. Prompt classes and functions make constructing

and working with prompts easy.

Class hierarchy:

BasePromptTemplate --> PipelinePromptTemplate	
StringPromptTemplate --> PromptTemplate	
	FewShotPromptTemplate
	FewShotPromptWithTemplates
BaseChatPromptTemplate --> AutoGPTPrompt	
	ChatPromptTemplate --> AgentScratchPadChatPromptTemplate
BaseMessagePromptTemplate --> MessagesPlaceholder	
BaseStringMessagePromptTemplate --> ChatMessagePromptTemplate	
	HumanMessagePromptTemplate
	AIMessagePromptTemplate
	SystemMessagePromptTemplate
PromptValue --> StringPromptValue	
	ChatPromptValue

Classes

prompts.base.StringPromptTemplate	String prompt that exposes the format method, returning a prompt.
prompts.base.StringPromptValue	String prompt value.
prompts.few_shot_with_templates.FewShotPromptWithTemplates	Prompt template that contains few shot examples.
prompts.few_shot.FewShotChatMessagePromptTemplate	Chat prompt template that supports few-shot examples.
prompts.few_shot.FewShotPromptTemplate	Prompt template that contains few shot examples.
prompts.prompt.Prompt	alias of PromptTemplate
prompts.prompt.PromptTemplate	A prompt template for a language model.
prompts.chat.AIMessagePromptTemplate	AI message prompt template.
prompts.chat.BaseChatPromptTemplate	Base class for chat prompt templates.
prompts.chat.BaseMessagePromptTemplate	Base class for message prompt templates.
prompts.chat.BaseStringMessagePromptTemplate	Base class for message prompt templates that use a string prompt template.
prompts.chat.ChatMessagePromptTemplate	Chat message prompt template.

Toggle Menu

prompts.chat.ChatPromptTemplate	A prompt template for chat models.
prompts.chat.ChatPromptValue	Chat prompt value.
prompts.chat.HumanMessagePromptTemplate	Human message prompt template.
prompts.chat.MessagesPlaceholder	Prompt template that assumes variable is already list of messages.
prompts.chat.SystemMessagePromptTemplate	System message prompt template.
prompts.pipeline.PipelinePromptTemplate	A prompt template for composing multiple prompt templates together.
prompts.example_selector.base.BaseExampleSelector()	Interface for selecting examples to include in prompts.
prompts.example_selector.ngram_overlap.NGramOverlapExampleSelector	Select and order examples based on ngram overlap score (sentence_bleu score).
prompts.example_selector.semantic_similarity.MaxMarginalRelevanceExampleSelector	ExampleSelector that selects examples based on Max Marginal Relevance.
prompts.example_selector.semantic_similarity.SemanticSimilarityExampleSelector	Example selector that selects examples based on SemanticSimilarity.
prompts.example_selector.length_based.LengthBasedExampleSelector	Select examples based on length.

Functions

prompts.base.check_valid_template (template, ...)	Check that template string is valid.
prompts.base.jinja2_formatter (template, **kwargs)	Format a template using jinja2.
prompts.base.validate_jinja2 (template, ...)	Validate that the input variables are valid for the template.
prompts.example_selector.ngram_overlap.ngram_overlap_score (...)	Compute ngram overlap score of source and example as sentence_bleu score.
prompts.example_selector.semantic_similarity.sorted_values (values)	Return a list of values in dict sorted by key.
prompts.loading.load_prompt (path)	Unified method for loading a prompt from LangChainHub or local fs.
prompts.loading.load_prompt_from_config (config)	Load prompt from Config Dict.

langchain.retrievers

Retriever class returns Documents given a text **query**.

It is more general than a vector store. A retriever does not need to be able to store documents, only to return (or retrieve) it. Vector stores can be used as the backbone of a retriever, but there are other types of retrievers as well.

Class hierarchy:

```
BaseRetriever --> <name>Retriever # Examples: ArxivRetriever, MergerRetriever
```

Main helpers:

```
Document, Serializable, Callbacks, CallbackManagerForRetrieverRun, AsyncCallbackManagerForRetrieverRun
```

Classes

retrievers.docarray.DocArrayRetriever	Retriever for DocArray Document Indices.
retrievers.docarray.SearchType (value[, ...])	Enumerator of the types of search to perform.
retrievers.remote_retriever.RemoteLangChainRetriever	Retriever for remote LangChain API.
retrievers.chatgpt_plugin_retriever.ChatGPTPluginRetriever	Retrieves documents from a ChatGPT plugin.
retrievers.web_research.LineList	List of questions.
retrievers.web_research.QuestionListOutputParser	Output parser for a list of numbered questions.
retrievers.web_research.SearchQueries	Search queries to run to research for the user's goal.
retrievers.web_research.WebResearchRetriever	Retriever for web research based on the Google Search API.
retrievers.databerry.DataberryRetriever	Retriever for the Databerry API.
retrievers.bm25.BM25Retriever	BM25 Retriever without elastic search.
retrievers.wikipedia.WikipediaRetriever	Retriever for Wikipedia API.
retrievers.zilliz.ZillizRetriever	Retriever for the Zilliz API.
retrievers.pinecone_hybrid_search.PineconeHybridSearchRetriever	Pinecone Hybrid Search Retriever.
retrievers.pubmed.PubMedRetriever	Retriever for PubMed API.
retrievers.zep.ZepRetriever	Retriever for the Zep long-term memory store.

Toggle Menu

retrievers.svm.SVMRetriever	SVM Retriever.
retrievers.llama_index.LlamaIndexGraphRetriever	Retriever for question-answering with sources over an LlamaIndex graph data structure.
retrievers.llama_index.LlamaIndexRetriever	Retriever for the question-answering with sources over an LlamaIndex data structure.
retrievers.time_weighted_retriever.TimeWeightedVectorStoreRetriever	Retriever that combines embedding similarity with recency in retrieving values.
retrievers.knn.KNNRetriever	KNN Retriever.
retrievers.multi_query.LineList	List of lines.
retrievers.multi_query.LineListOutputParser	Output parser for a list of lines.
retrievers.multi_query.MultiQueryRetriever	Given a user query, use an LLM to write a set of queries.
retrievers.vespa_retriever.VespaRetriever	Retriever that uses Vespa.
retrievers.milvus.MilvusRetriever	Retriever that uses the Milvus API.
retrievers.re_phraser.RePhraseQueryRetriever	Given a user query, use an LLM to re-phrase it.
retrievers.kendra.AdditionalResultAttribute	An additional result attribute.
retrievers.kendra.AdditionalResultAttributeValue	The value of an additional result attribute.
retrievers.kendra.AmazonKendraRetriever	Retriever for the Amazon Kendra Index.
retrievers.kendra.DocumentAttribute	A document attribute.
retrievers.kendra.DocumentAttributeValue	The value of a document attribute.
retrievers.kendra.Highlight	Represents the information that can be used to highlight key words in the excerpt.
retrievers.kendra.QueryResult	Represents an Amazon Kendra Query API search result, which is composed of:
retrievers.kendra.QueryResultItem	A Query API result item.
retrievers.kendra.ResultItem	Abstract class that represents a result item.
retrievers.kendra.RetrieveResult	Represents an Amazon Kendra Retrieve API search result, which is composed of:
retrievers.kendra.RetrieveResultItem	A Retrieve API result item.
retrievers.kendra.TextWithHighLights	Text with highlights.
retrievers.weaviate_hybrid_search.WeaviateHybridSearchRetriever	Retriever for the Weaviate's hybrid search.
retrievers.merger_retriever.MergerRetriever	Retriever that merges the results of multiple retrievers.
retrievers.ensemble.EnsembleRetriever	This class ensemble the results of multiple retrievers by using rank fusion.
retrievers.arxiv.ArxivRetriever	Retriever for Arxiv.
retrievers.elastic_search_bm25.ElasticSearchBM25Retriever	Retriever for the Elasticsearch using BM25 as a retrieval method.
retrievers.tfidf.TFIDFRetriever	TF-IDF Retriever.
retrievers.chaindesk.ChaindeskRetriever	Retriever for the Chaindesk API.
retrievers.google_cloud_enterprise_search.GoogleCloudEnterpriseSearchRetriever	Retriever for the Google Cloud Enterprise Search Service API.
retrievers.azure_cognitive_search.AzureCognitiveSearchRetriever	Retriever for the Azure Cognitive Search service.
retrievers.metal.MetalRetriever	Retriever that uses the Metal API.
retrievers.contextual_compression.ContextualCompressionRetriever	Retriever that wraps a base retriever and compresses the results.
retrievers.parent_document_retriever.ParentDocumentRetriever	Fetches small chunks, then fetches their parent documents.
retrievers.document_compressors.chain_extract.LLMChainExtractor	DocumentCompressor that uses an LLM chain to extract the relevant parts of documents.
retrievers.document_compressors.chain_extract.NoOutputParser	Parse outputs that could return a null string of some sort.
retrievers.document_compressors.embeddings_filter.EmbeddingsFilter	Document compressor that uses embeddings to drop documents unrelated to the query.
retrievers.document_compressors.cohere_rerank.CohereRerank	DocumentCompressor that uses Cohere's rerank API to compress documents.
retrievers.document_compressors.base.BaseDocumentCompressor	Base abstraction interface for document compression.
retrievers.document_compressors.base.DocumentCompressorPipeline	Document compressor that uses a pipeline of transformers.
retrievers.document_compressors.chain_filter.LLMChainFilter	Filter that drops documents that aren't relevant to the query.
retrievers.self_query.deeplake.DeepLakeTranslator()	Logic for converting internal query language elements to valid filters.
retrievers.self_query.base.SelfQueryRetriever	Retriever that uses a vector store and an LLM to generate the vector store queries.
retrievers.self_query.elasticsearch.ElasticsearchTranslator()	Translate the internal query language elements to valid filters.

Toggle Menu

retrievers.self_query.qdrant.QdrantTranslator(...)	Translate the internal query language elements to valid filters.
retrievers.self_query.chroma.ChromaTranslator()	Translate internal query language elements to valid filters.
retrievers.self_query.pinecone.PineconeTranslator()	Translate the internal query language elements to valid filters.
retrievers.self_query.myscale.MyScaleTranslator([...])	Translate internal query language elements to valid filters.
retrievers.self_query.weaviate.WeaviateTranslator()	Translate the internal query language elements to valid filters.

Functions

retrievers.bm25.default_preprocessing_func(text)	
retrievers.document_compressors.chain_extract.default_get_input(...)	Return the compression chain input.
retrievers.document_compressors.chain_filter.default_get_input(...)	Return the compression chain input.
retrievers.kendra.clean_excerpt(excerpt)	Cleans an excerpt from Kendra.
retrievers.kendra.combined_text(item)	Combines a ResultItem title and excerpt into a single string.
retrievers.knn.create_index(contexts, embeddings)	Create an index of embeddings for a list of contexts.
retrievers.milvus.MilvusRetreiver(*args, ...)	Deprecated MilvusRetreiver.
retrievers.pinecone_hybrid_search.create_index(...)	Create a Pinecone index from a list of contexts.
retrievers.pinecone_hybrid_search.hash_text(text)	Hash a text using SHA256.
retrievers.self_query.deeplake.can_cast_to_float(string)	Check if a string can be cast to a float.
retrievers.self_query.myscale.DEFAULT_COMPOSER(op_name)	Default composer for logical operators.
retrievers.self_query.myscale.FUNCTION_COMPOSER(op_name)	Composer for functions.
retrievers.svm.create_index(contexts, embeddings)	Create an index of embeddings for a list of contexts.
retrievers.zilliz.ZillizRetreiver(*args, ...)	Deprecated ZillizRetreiver.

langchain.runnables

Classes

runnables.openai_functions.OpenAIFunction	A function description for ChatOpenAI
runnables.openai_functions.OpenAIFunctionsRouter	A runnable that routes to the selected function.

langchain.schema

Schemas are the LangChain Base Classes and Interfaces.

Classes

schema.memory.BaseChatMessageHistory()	Abstract base class for storing chat message history.
schema.memory.BaseMemory	Abstract base class for memory in Chains.
schema.output_parser.BaseGenerationOutputParser	Base class to parse the output of an LLM call.
schema.output_parser.BaseLLMOutputParser	Abstract base class for parsing the outputs of a model.
schema.output_parser.BaseOutputParser	Base class to parse the output of an LLM call.
schema.output_parser.BaseTransformOutputParser	Base class for an output parser that can handle streaming input.
schema.output_parser.NoOpOutputParser	alias of StrOutputParser
schema.output_parser.OutputParserException(error)	Exception that output parsers should raise to signify a parsing error.
schema.output_parser.StrOutputParser	OutputParser that parses LLMResult into the top likely string.
schema.document.BaseDocumentTransformer()	Abstract base class for document transformation systems.
schema.document.Document	Class for storing a piece of text and associated metadata.
schema.messages.AIMessage	A Message from an AI.
schema.messages.AIMessageChunk	A Message chunk from an AI.
schema.messages.BaseMessage	The base abstract Message class.
schema.messages.BaseMessageChunk	A Message chunk, which can be concatenated with other Message chunks.
schema.messages.ChatMessage	A Message that can be assigned an arbitrary speaker (i.e.
schema.messages.ChatMessageChunk	A Chat Message chunk.
schema.messages.FunctionMessage	A Message for passing the result of executing a function back to a model.
schema.messages.FunctionMessageChunk	A Function Message chunk.
schema.messages.HumanMessage	A Message from a human.

Toggle Menu

schema.messages.HumanMessageChunk	A Human Message chunk.
schema.messages.SystemMessage	A Message for priming AI behavior, usually passed in as the first of a sequence of input messages.
schema.messages.SystemMessageChunk	A System Message chunk.
schema.output.ChatGeneration	A single chat generation output.
schema.output.ChatGenerationChunk	A ChatGeneration chunk, which can be concatenated with other
schema.output.ChatResult	Class that contains all results for a single chat model call.
schema.output.Generation	A single text generation output.
schema.output.GenerationChunk	A Generation chunk, which can be concatenated with other Generation chunks.
schema.output.LLMResult	Class that contains all results for a batched LLM call.
schema.output.RunInfo	Class that contains metadata for a single execution of a Chain or model.
schema.agent.AgentAction (tool, tool_input, log)	A full description of an action for an ActionAgent to execute.
schema.agent.AgentFinish (return_values, log)	The final return value of an ActionAgent.
schema.prompt_template.BasePromptTemplate	Base class for all prompt templates, returning a prompt.
schema.language_model.BaseLanguageModel	Abstract base class for interfacing with language models.
schema.exceptions.LangChainException	General LangChain exception.
schema.storage.BaseStore ()	Abstract interface for a key-value store.
schema.retriever.BaseRetriever	Abstract base class for a Document retrieval system.
schema.prompt.PromptValue	Base abstract class for inputs to any language model.
schema.runnable.passthrough.RunnablePassthrough	A runnable that passes through the input.
schema.runnable.config.RunnableConfig	Configuration for a Runnable.
schema.runnable.router.RouterInput	A Router input.
schema.runnable.router.RouterRunnable	A runnable that routes to a set of runnables based on Input['key'].
schema.runnable.base.Runnable ()	A Runnable is a unit of work that can be invoked, batched, streamed, or transformed.
schema.runnable.base.RunnableBinding	A runnable that delegates calls to another runnable with a set of kwargs.
schema.runnable.base.RunnableLambda (func)	A runnable that runs a callable.
schema.runnable.base.RunnableMap	A runnable that runs a mapping of runnables in parallel, and returns a mapping of their outputs.
schema.runnable.base.RunnableSequence	A sequence of runnables, where the output of each is the input of the next.
schema.runnable.base.RunnableWithFallbacks	A Runnable that can fallback to other Runnables if it fails.

Functions

schema.messages.get_buffer_string (messages)	Convert sequence of Messages to strings and concatenate them into one string.
schema.messages.messages_from_dict (messages)	Convert a sequence of messages from dicts to Message objects.
schema.messages.messages_to_dict (messages)	Convert a sequence of Messages to a list of dictionaries.
schema.prompt_template.format_document (doc, ...)	Format a document into a string based on a prompt template.
schema.runnable.base.coerce_to_runnable (thing)	
schema.runnable.base.patch_config (config, ...)	
schema.runnable.passthrough.aidentity (x)	
schema.runnable.passthrough.identity (x)	
schema.runnable.utils.gated_coro (semaphore, coro)	
schema.runnable.utils.gather_with_concurrency (n, ...)	

langchain.server

Script to run langchain-server locally using docker-compose.

Functions

server.main ()	Run the langchain server locally.
--------------------------------	-----------------------------------

langchain.smith

LangSmith utilities.

This module provides utilities for connecting to [LangSmith](#). For more information on LangSmith, see the [LangSmith documentation](#).

Evaluation

Toggle Menu

LangSmith helps you evaluate Chains and other language model application components using a number of LangChain evaluators. An example of this is shown below, assuming you’ve created a LangSmith dataset called `<my_dataset_name>`:

```
from langsmith import Client
from langchain.chat_models import ChatOpenAI
from langchain.chains import LLMChain
from langchain.smith import RunEvalConfig, run_on_dataset

# Chains may have memory. Passing in a constructor function lets the
# evaluation framework avoid cross-contamination between runs.
def construct_chain():
    llm = ChatOpenAI(temperature=0)
    chain = LLMChain.from_string(
        llm,
        "What's the answer to {your_input_key}"
    )
    return chain

# Load off-the-shelf evaluators via config or the EvaluatorType (string or enum)
evaluation_config = RunEvalConfig(
    evaluators=[
        "qa", # "Correctness" against a reference answer
        "embedding_distance",
        RunEvalConfig.Criteria("helpfulness"),
        RunEvalConfig.Criteria({
            "fifth-grader-score": "Do you have to be smarter than a fifth grader to answer this question?"
        }),
    ]
)

client = Client()
run_on_dataset(
    client,
    "<my_dataset_name>",
    construct_chain,
    evaluation=evaluation_config,
)
```

You can also create custom evaluators by subclassing the [StringEvaluator](#) or LangSmith’s *RunEvaluator* classes.

```
from typing import Optional
from langchain.evaluation import StringEvaluator

class MyStringEvaluator(StringEvaluator):

    @property
    def requires_input(self) -> bool:
        return False

    @property
    def requires_reference(self) -> bool:
        return True

    @property
    def evaluation_name(self) -> str:
        return "exact_match"

    def _evaluate_strings(self, prediction, reference=None, input=None, **kwargs) -> dict:
        return {"score": prediction == reference}

evaluation_config = RunEvalConfig(
    custom_evaluators = [MyStringEvaluator()],
)

run_on_dataset(
    client,
    "<my_dataset_name>",
    construct_chain,
    evaluation=evaluation_config,
)
```

Primary Functions

- [arun_on_dataset](#): Asynchronous function to evaluate a chain, agent, or other LangChain component over a dataset.
- [run_on_dataset](#): Function to evaluate a chain, agent, or other LangChain component over a dataset.

Toggle Menu

- RunEvalConfig**: Class representing the configuration for running evaluation. You can select evaluators by [EvaluatorType](#) or config, or you can pass in *custom_evaluators*

Classes

smith.evaluation.config.EvalConfig	Configuration for a given run evaluator.
smith.evaluation.config.RunEvalConfig	Configuration for a run evaluation.
smith.evaluation.runner_utils.InputFormatError	Raised when the input format is invalid.
smith.evaluation.string_run_evaluator.ChainStringRunMapper	Extract items to evaluate from the run object from a chain.
smith.evaluation.string_run_evaluator.LLMStringRunMapper	Extract items to evaluate from the run object.
smith.evaluation.string_run_evaluator.StringExampleMapper	Map an example, or row in the dataset, to the inputs of an evaluation.
smith.evaluation.string_run_evaluator.StringRunEvaluatorChain	Evaluate Run and optional examples.
smith.evaluation.string_run_evaluator.StringRunMapper	Extract items to evaluate from the run object.
smith.evaluation.string_run_evaluator.ToolStringRunMapper	Map an input to the tool.

Functions

smith.evaluation.runner_utils.arun_on_dataset(...)	Asynchronously run the Chain or language model on a dataset and store traces to the specified project name.
smith.evaluation.runner_utils.run_on_dataset(...)	Run the Chain or language model on a dataset and store traces to the specified project name.

[langchain.storage](#)

Implementations of key-value stores and storage helpers.

Module provides implementations of various key-value stores that conform to a simple key-value interface.

The primary goal of these storages is to support implementation of caching.

Classes

storage.encoder_backed.EncoderBackedStore(...)	Wraps a store with key and value encoders/decoders.
storage.exceptions.InvalidKeyException	Raised when a key is invalid; e.g., uses incorrect characters.
storage.in_memory.InMemoryStore()	In-memory implementation of the BaseStore using a dictionary.
storage.redis.RedisStore(*[, client, ...])	BaseStore implementation using Redis as the underlying store.
storage.file_system.LocalFileStore(root_path)	BaseStore interface that works on the local file system.

[langchain.text_splitter](#)

Text Splitters are classes for splitting text.

Class hierarchy:

BaseDocumentTransformer --> TextSplitter --> <name>TextSplitter # <i>Example: CharacterTextSplitter</i> RecursiveCharacterTextSplitter --> <name>TextSplitter
--

Note: **MarkdownHeaderTextSplitter** does not derive from TextSplitter.

Main helpers:

Document, Tokenizer, Language, LineType, HeaderType

Classes

text_splitter.CharacterTextSplitter([...])	Splitting text that looks at characters.
text_splitter.HeaderType	Header type as typed dict.
text_splitter.Language(value[, names, ...])	Enum of the programming languages.
text_splitter.LatexTextSplitter(**kwargs)	Attempts to split the text along Latex-formatted layout elements.
text_splitter.LineType	Line type as typed dict.
r.MarkdownHeaderTextSplitter(...)	Splitting markdown files based on specified headers.

Toggle Menu

text_splitter.MarkdownTextSplitter (**kwargs)	Attempts to split the text along Markdown-formatted headings.
text_splitter.NLTKTextSplitter ([separator])	Splitting text using NLTK package.
text_splitter.PythonCodeTextSplitter (**kwargs)	Attempts to split the text along Python syntax.
text_splitter.RecursiveCharacterTextSplitter ([...])	Splitting text by recursively look at characters.
text_splitter.SentenceTransformersTokenTextSplitter ([...])	Splitting text to tokens using sentence model tokenizer.
text_splitter.SpacyTextSplitter ([separator, ...])	Splitting text using Spacy package.
text_splitter.TextSplitter (chunk_size, ...)	Interface for splitting text into chunks.
text_splitter.TokenTextSplitter ([...])	Splitting text to tokens using model tokenizer.
text_splitter.Tokenizer (chunk_overlap, ...)	

Functions

text_splitter.split_text_on_tokens (* text, ...)	Split incoming text and return chunks using tokenizer.
--	--

[langchain.tools](#)

Tools are classes that an Agent uses to interact with the world.

Each tool has a **description**. Agent uses the description to choose the right tool for the job.

Class hierarchy:

ToolMetaclass --> BaseTool --> <name>Tool	# Examples: AIPluginTool, BaseGraphQLTool
<name>	# Examples: BraveSearch, HumanInputRun

Main helpers:

CallbackManagerForToolRun, AsyncCallbackManagerForToolRun

Classes

tools.ifttt.IFTTTWebhook	IFTTT Webhook.
tools.base.BaseTool	Interface LangChain tools must implement.
tools.base.SchemaAnnotationError	Raised when 'args_schema' is missing or has an incorrect type annotation.
tools.base.StructuredTool	Tool that can operate on any number of inputs.
tools.base.Tool	Tool that takes in function or coroutine directly.
tools.base.ToolException	An optional exception that tool throws when execution error occurs.
tools.base.ToolMetaclass (name, bases, dct)	Metaclass for BaseTool to ensure the provided args_schema
tools.plugin.AIPlugin	AI Plugin Definition.
tools.plugin.AIPluginTool	Tool for getting the OpenAPI spec for an AI Plugin.
tools.plugin.AIPluginToolSchema	Schema for AIPluginTool.
tools.plugin.ApiConfig	API Configuration.
tools.convert_to_openai.FunctionDescription	Representation of a callable function to the OpenAI API.
tools.wolfram_alpha.tool.WolframAlphaQueryRun	Tool that queries using the Wolfram Alpha SDK.
tools.spark_sql.tool.BaseSparkSQLTool	Base tool for interacting with Spark SQL.
tools.spark_sql.tool.InfoSparkSQLTool	Tool for getting metadata about a Spark SQL.
tools.spark_sql.tool.ListSparkSQLTool	Tool for getting tables names.
tools.spark_sql.tool.QueryCheckerTool	Use an LLM to check if a query is correct.
tools.spark_sql.tool.QuerySparkSQLTool	Tool for querying a Spark SQL.
tools.shell.tool.ShellInput	Commands for the Bash Shell tool.
tools.shell.tool.ShellTool	Tool to run shell commands.
tools.human.tool.HumanInputRun	Tool that asks user for input.
tools.golden_query.tool.GoldenQueryRun	Tool that adds the capability to query using the Golden API and get back JSON.
tools.wikipedia.tool.WikipediaQueryRun	Tool that searches the Wikipedia API.
tools.google_places.tool.GooglePlacesSchema	Input for GooglePlacesTool.
tools.google_places.tool.GooglePlacesTool	Tool that queries the Google places API.
tools.amadeus.base.AmadeusBaseTool	Base Tool for Amadeus.
tools.amadeus.flight_search.AmadeusFlightSearch	Tool for searching for a single flight between two airports.
tools.amadeus.flight_search.FlightSearchSchema	Schema for the AmadeusFlightSearch tool.

Toggle Menu

tools.amadeus.closest_airport.AmadeusClosestAirport	Tool for finding the closest airport to a particular location.
tools.amadeus.closest_airport.ClosestAirportSchema	Schema for the AmadeusClosestAirport tool.
tools.arxiv.tool.ArxivQueryRun	Tool that searches the Arxiv API.
tools.playwright.extract_text.ExtractTextTool	Tool for extracting all the text on the current webpage.
tools.playwright.base.BaseBrowserTool	Base class for browser tools.
tools.playwright.navigate.NavigateTool	Tool for navigating a browser to a URL.
tools.playwright.navigate.NavigateToolInput	Input for NavigateToolInput.
tools.playwright.click.ClickTool	Tool for clicking on an element with the given CSS selector.
tools.playwright.click.ClickToolInput	Input for ClickTool.
tools.playwright.navigate_back.NavigateBackTool	Navigate back to the previous page in the browser history.
tools.playwright.get_elements.GetElementsTool	Tool for getting elements in the current web page matching a CSS selector.
tools.playwright.get_elements.GetElementsToolInput	Input for GetElementsTool.
tools.playwright.extract_hyperlinks.ExtractHyperlinksTool	Extract all hyperlinks on the page.
tools.playwright.extract_hyperlinks.ExtractHyperlinksToolInput	Input for ExtractHyperlinksTool.
tools.playwright.current_page.CurrentWebPageTool	Tool for getting the URL of the current webpage.
tools.vectorstore.tool.BaseVectorStoreTool	Base class for tools that use a VectorStore.
tools.vectorstore.tool.VectorStoreQATool	Tool for the VectorDBQA chain.
tools.vectorstore.tool.VectorStoreQAWithSourcesTool	Tool for the VectorDBQAWithSources chain.
tools.zapier.tool.ZapierNLAListActions	Returns a list of all exposed (enabled) actions associated with
tools.zapier.tool.ZapierNLARunAction	Executes an action that is identified by action_id, must be exposed
tools.youtube.search.YouTubeSearchTool	Tool that queries YouTube.
tools.office365.events_search.0365SearchEvents	Class for searching calendar events in Office 365
tools.office365.events_search.SearchEventsInput	Input for SearchEmails Tool.
tools.office365.create_draft_message.CreateDraftMessageSchema	Input for SendMessageTool.
tools.office365.create_draft_message.0365CreateDraftMessage	Tool for creating a draft email in Office 365.
tools.office365.base.0365BaseTool	Base class for the Office 365 tools.
tools.office365.send_message.0365SendMessage	Tool for sending an email in Office 365.
tools.office365.send_message.SendMessageSchema	Input for SendMessageTool.
tools.office365.send_event.0365SendEvent	Tool for sending calendar events in Office 365.
tools.office365.send_event.SendEventSchema	Input for CreateEvent Tool.
tools.office365.messages_search.0365SearchEmails	Class for searching email messages in Office 365
tools.office365.messages_search.SearchEmailsInput	Input for SearchEmails Tool.
tools.multion.update_session.MultionUpdateSession	Tool that updates an existing Multion Browser Window with provided fields.
tools.multion.update_session.UpdateSessionSchema	Input for UpdateSessionTool.
tools.multion.create_session.CreateSessionSchema	Input for CreateSessionTool.
tools.multion.create_session.MultionCreateSession	Tool that creates a new Multion Browser Window with provided fields.
tools.gmail.get_message.GmailGetMessage	Tool that gets a message by ID from Gmail.
tools.gmail.get_message.SearchArgsSchema	Input for GetMessageTool.
tools.gmail.create_draft.CreateDraftSchema	Input for CreateDraftTool.
tools.gmail.create_draft.GmailCreateDraft	Tool that creates a draft email for Gmail.
tools.gmail.base.GmailBaseTool	Base class for Gmail tools.
tools.gmail.send_message.GmailSendMessage	Tool that sends a message to Gmail.
tools.gmail.send_message.SendMessageSchema	Input for SendMessageTool.
tools.gmail.search.GmailSearch	Tool that searches for messages or threads in Gmail.
tools.gmail.search.Resource(value[, names, ...])	Enumerator of Resources to search.
tools.gmail.search.SearchArgsSchema	Input for SearchGmailTool.
tools.gmail.get_thread.GetThreadSchema	Input for GetMessageTool.
tools.gmail.get_thread.GmailGetThread	Tool that gets a thread by ID from Gmail.
tools.brave_search.tool.BraveSearch	Tool that queries the BraveSearch.
tools.openapi.utils.api_models.APIOperation	A model for a single API operation.
tools.openapi.utils.api_models.APIProperty	A model for a property in the query, path, header, or cookie params.
tools.openapi.utils.api_models.APIPropertyBase	Base model for an API property.
tools.openapi.utils.api_models.APIPropertyLocation(value)	The location of the property.
tools.openapi.utils.api_models.APIRequestBody	A model for a request body.
tools.openapi.utils.api_models.APIRequestBodyProperty	A model for a request body property.

Toggle Menu

tools.metaphor_search.tool.MetaphorSearchResults	Tool that queries the Metaphor Search API and gets back json.
tools.dataforseo_api_search.tool.DataForSeoAPISearchResults	Tool that queries the DataForSeo Google Search API and get back json.
tools.dataforseo_api_search.tool.DataForSeoAPISearchRun	Tool that queries the DataForSeo Google search API.
tools.google_search.tool.GoogleSearchResults	Tool that queries the Google Search API and gets back json.
tools.google_search.tool.GoogleSearchRun	Tool that queries the Google search API.
tools.requests.tool.BaseRequestsTool	Base class for requests tools.
tools.requests.tool.RequestsDeleteTool	Tool for making a DELETE request to an API endpoint.
tools.requests.tool.RequestsGetTool	Tool for making a GET request to an API endpoint.
tools.requests.tool.RequestsPatchTool	Tool for making a PATCH request to an API endpoint.
tools.requests.tool.RequestsPostTool	Tool for making a POST request to an API endpoint.
tools.requests.tool.RequestsPutTool	Tool for making a PUT request to an API endpoint.
tools.pubmed.tool.PubmedQueryRun	Tool that searches the PubMed API.
tools.file_management.file_search.FileSearchInput	Input for FileSearchTool.
tools.file_management.file_search.FileSearchTool	Tool that searches for files in a subdirectory that match a regex pattern.
tools.file_management.read.ReadFileInput	Input for ReadFileTool.
tools.file_management.read.ReadFileTool	Tool that reads a file.
tools.file_management.move.FileMoveInput	Input for MoveFileTool.
tools.file_management.move.MoveFileTool	Tool that moves a file.
tools.file_management.copy.CopyFileTool	Tool that copies a file.
tools.file_management.copy.FileCopyInput	Input for CopyFileTool.
tools.file_management.write.WriteFileInput	Input for WriteFileTool.
tools.file_management.write.WriteFileTool	Tool that writes a file to disk.
tools.file_management.list_dir.DirectoryListingInput	Input for ListDirectoryTool.
tools.file_management.list_dir.ListDirectoryTool	Tool that lists files and directories in a specified folder.
tools.file_management.utils.BaseFileToolMixin	Mixin for file system tools.
tools.file_management.utils.FileValidationError	Error for paths outside the root directory.
tools.file_management.delete.DeleteFileTool	Tool that deletes a file.
tools.file_management.delete.FileDeleteInput	Input for DeleteFileTool.
tools.azure_cognitive_services.text2speech.AzureCogsText2SpeechTool	Tool that queries the Azure Cognitive Services Text2Speech API.
tools.azure_cognitive_services.speech2text.AzureCogsSpeech2TextTool	Tool that queries the Azure Cognitive Services Speech2Text API.
tools.azure_cognitive_services.image_analysis.AzureCogsImageAnalysisTool	Tool that queries the Azure Cognitive Services Image Analysis API.
tools.azure_cognitive_services.form_recognizer.AzureCogsFormRecognizerTool	Tool that queries the Azure Cognitive Services Form Recognizer API.
tools.openweathermap.tool.OpenWeatherMapQueryRun	Tool that queries the OpenWeatherMap API.
tools.jira.tool.JiraAction	Tool that queries the Atlassian Jira API.
tools.bing_search.tool.BingSearchResults	Tool that queries the Bing Search API and gets back json.
tools.bing_search.tool.BingSearchRun	Tool that queries the Bing search API.
tools.github.tool.GitHubAction	Tool for interacting with the GitHub API.
tools.sql_database.tool.BaseSQLDatabaseTool	Base tool for interacting with a SQL database.
tools.sql_database.tool.InfoSQLDatabaseTool	Tool for getting metadata about a SQL database.
tools.sql_database.tool.ListSQLDatabaseTool	Tool for getting tables names.
tools.sql_database.tool.QuerySQLCheckerTool	Use an LLM to check if a query is correct.
tools.sql_database.tool.QuerySQLDataBaseTool	Tool for querying a SQL database.
tools.scenexplain.tool.SceneXplainInput	Input for SceneXplain.
tools.scenexplain.tool.SceneXplainTool	Tool that explains images.
tools.sleep.tool.SleepInput	Input for CopyFileTool.
tools.sleep.tool.SleepTool	Tool that adds the capability to sleep.
tools.python.tool.PythonAstREPLTool	A tool for running python code in a REPL.
tools.python.tool.PythonREPLTool	A tool for running python code in a REPL.
tools.powerbi.tool.InfoPowerBITool	Tool for getting metadata about a PowerBI Dataset.
tools.powerbi.tool.ListPowerBITool	Tool for getting tables names.
tools.powerbi.tool.QueryPowerBITool	Tool for querying a Power BI Dataset.
tools.ddg_search.tool.DuckDuckGoSearchResults	Tool that queries the DuckDuckGo search API and gets back json.

Toggle Menu

tools.ddg_search.tool.DuckDuckGoSearchRun	Tool that queries the DuckDuckGo search API.
tools.graphql.tool.BaseGraphQLTool	Base tool for querying a GraphQL API.
tools.json.tool.JsonGetValueTool	Tool for getting a value in a JSON spec.
tools.json.tool.JsonListKeysTool	Tool for listing keys in a JSON spec.
tools.json.tool.JsonSpec	Base class for JSON spec.
tools.nuclia.tool.NUASchema	Input for Nuclia Understanding API.
tools.nuclia.tool.NucliaUnderstandingAPI	Tool to process files with the Nuclia Understanding API.
tools.steamship_image_generation.tool.ModelName (value)	Supported Image Models for generation.
tools.steamship_image_generation.tool.SteamshipImageGenerationTool	Tool used to generate images from a text-prompt.
tools.google_serper.tool.GoogleSerperResults	Tool that queries the Serper.dev Google Search API and get back json.
tools.google_serper.tool.GoogleSerperRun	Tool that queries the Serper.dev Google search API.
tools.searx_search.tool.SearxSearchResults	Tool that queries a Searx instance and gets back json.
tools.searx_search.tool.SearxSearchRun	Tool that queries a Searx instance.

Functions

tools.amadeus.utils.authenticate()	Authenticate using the Amadeus API
tools.azure_cognitive_services.utils.detect_file_src_type (...)	Detect if the file is local or remote.
tools.azure_cognitive_services.utils.download_audio_from_url (...)	Download audio from url to local.
tools.base.create_schema_from_function (...)	Create a pydantic schema from a function's signature.
tools.base.tool (*args[, return_direct, ...])	Make tools out of functions, can be used with or without arguments.
tools.convert_to_openai.format_tool_to_openai_function (tool)	Format tool into the OpenAI function API.
tools.ddg_search.tool.DuckDuckGoSearchTool (...)	Deprecated.
tools.file_management.utils.get_validated_relative_path (...)	Resolve a relative path, raising an error if not within the root directory.
tools.file_management.utils.is_relative_to (...)	Check if path is relative to root.
tools.gmail.utils.build_resource_service ([...])	Build a Gmail service.
tools.gmail.utils.clean_email_body (body)	Clean email body.
tools.gmail.utils.get_gmail_credentials ([...])	Get credentials.
tools.gmail.utils.import_google ()	Import google libraries.
tools.gmail.utils.import_googleapiclient_resource_builder ()	Import googleapiclient.discovery.build function.
tools.gmail.utils.import_installed_app_flow ()	Import InstalledAppFlow class.
tools.interaction.tool.StdInInquireTool (...)	Tool for asking the user for input.
tools.office365.utils.authenticate()	Authenticate using the Microsoft Grah API
tools.office365.utils.clean_body (body)	Clean body of a message or event.
tools.playwright.base.lazy_import_playwright_browsers ()	Lazy import playwright browsers.
tools.playwright.utils.aget_current_page (browser)	Asynchronously get the current page of the browser.
tools.playwright.utils.create_async_playwright_browser ([...])	Create an async playwright browser.
tools.playwright.utils.create_sync_playwright_browser ([...])	Create a playwright browser.
tools.playwright.utils.get_current_page (browser)	Get the current page of the browser.
tools.playwright.utils.run_async (coro)	Run an async coroutine.
tools.plugin.marshal_spec (txt)	Convert the yaml or json serialized spec to a dict.
tools.python.tool.sanitize_input (query)	Sanitize input to the python REPL.
tools.steamship_image_generation.utils.make_image_public (...)	Upload a block to a signed URL and return the public URL.

langchain.utilities

Utilities are the integrations with third-part systems and packages.

Other LangChain classes use **Utilities** to interact with third-part systems and packages.

Classes

utilities.sql_database.SQLDatabase (engine[, ...])	SQLAlchemy wrapper around a database.
utilities.bing_search.BingSearchAPIWrapper	Wrapper for Bing Search API.
utilities.scenexplain.SceneXplainAPIWrapper	Wrapper for SceneXplain API.
utilities.golden_query.GoldenQueryAPIWrapper	Wrapper for Golden.
utilities.searx_search.SearxResults (data)	Dict like wrapper around search api results.
utilities.searx_search.SearxSearchWrapper	Wrapper for Searx API.
utilities.serpapi.HiddenPrints ()	Context manager to hide prints.

Toggle Menu

utilities.serpapi.SerpAPIWrapper	Wrapper around SerpAPI.
utilities.portkey.Portkey()	Portkey configuration.
utilities.requests.Requests	Wrapper around requests to handle auth and async.
utilities.requests.RequestsWrapper	alias of TextRequestsWrapper
utilities.requests.TextRequestsWrapper	Lightweight wrapper around requests library.
utilities.wikipedia.WikipediaAPIWrapper	Wrapper around WikipediaAPI.
utilities.openapi.HTTPVerb (value[, names, ...])	Enumerator of the HTTP verbs.
utilities.openapi.OpenAPISpec	OpenAPI Model that removes mis-formatted parts of the spec.
utilities.python.PythonREPL	Simulates a standalone Python REPL.
utilities.alpha_vantage.AlphaVantageAPIWrapper	Wrapper for AlphaVantage API for Currency Exchange Rate.
utilities.pubmed.PubMedAPIWrapper	Wrapper around PubMed API.
utilities.tensorflow_datasets.TensorflowDatasets	Access to the TensorFlow Datasets.
utilities.dataforseo_api_search.DataForSeoAPIWrapper	Wrapper around the DataForSeo API.
utilities.metaphor_search.MetaphorSearchAPIWrapper	Wrapper for Metaphor Search API.
utilities.google_search.GoogleSearchAPIWrapper	Wrapper for Google Search API.
utilities.powerbi.PowerBIDataset	Create PowerBI engine from dataset ID and credential or token.
utilities.github.GitHubAPIWrapper	Wrapper for GitHub API.
utilities.bibtex.BibtexparserWrapper	Wrapper around bibtexparser.
utilities.duckduckgo_search.DuckDuckGoSearchAPIWrapper	Wrapper for DuckDuckGo Search API.
utilities.brave_search.BraveSearchWrapper	Wrapper around the Brave search engine.
utilities.google_serper.GoogleSerperAPIWrapper	Wrapper around the Serper.dev Google Search API.
utilities.max_compute.MaxComputeAPIWrapper (client)	Interface for querying Alibaba Cloud MaxCompute tables.
utilities.dalle_image_generator.DallEAPIWrapper	Wrapper for OpenAI's DALL-E Image Generator.
utilities.twilio.TwilioAPIWrapper	Messaging Client using Twilio.
utilities.zapier.ZapierNLWrapper	Wrapper for Zapier NLA.
utilities.jira.JiraAPIWrapper	Wrapper for Jira API.
utilities.arxiv.ArxivAPIWrapper	Wrapper around ArxivAPI.
utilities.google_places_api.GooglePlacesAPIWrapper	Wrapper around Google Places API.
utilities.spark_sql.SparkSQL ([...])	SparkSQL is a utility class for interacting with Spark SQL.
utilities.awslambda.LambdaWrapper	Wrapper for AWS Lambda SDK.
utilities.wolfram_alpha.WolframAlphaAPIWrapper	Wrapper for Wolfram Alpha.
utilities.openweathermap.OpenWeatherMapAPIWrapper	Wrapper for OpenWeatherMap API using PyOWM.
utilities.bash.BashProcess ([strip_newlines, ...])	Wrapper class for starting subprocesses.
utilities.graphql.GraphQLAPIWrapper	Wrapper around GraphQL API.

Functions

utilities.loading.try_load_from_hub (path, ...)	Load configuration from hub.
utilities.powerbi.fix_table_name (table)	Add single quotes around table names that contain spaces.
utilities.powerbi.json_to_md (json_contents)	Converts a JSON object to a markdown table.
utilities.redis.get_client (redis_url, **kwargs)	Get a redis client from the connection url given.
utilities.sql_database.truncate_word (...[, ...])	Truncate a string to a certain number of words, based on the max string length.
utilities.vertexai.init_vertexai ([project, ...])	Init vertexai.
utilities.vertexai.raise_vertex_import_error ([...])	Raise ImportError related to Vertex SDK being not available.

langchain.utils

Utility functions for LangChain.

These functions do not depend on any other LangChain module.

Classes

utils.aiter.Tee (iterable[, n])	Create n separate asynchronous iterators over iterable
utils.aiter.ateee	alias of Tee
utils.formatting.StrictFormatter ()	A subclass of formatter that checks for extra keys.

Functions

utils.aiter.py_anext (iterator[, default])	Pure-Python implementation of anext() for testing purposes.
utils.aiter.tee_peer (iterator, buffer, peers)	An individual iterator of a tee()
utils.get_from_dict_or_env (data, key, ...)	Get a value from a dictionary or an environment variable.

Toggle Menu

utils.env.get_from_env (key, env_key[, default])	Get a value from a dictionary or an environment variable.
utils.input.get_boldded_text (text)	Get bolded text.
utils.input.get_color_mapping (items[, ...])	Get mapping for items to a support color.
utils.input.get_colored_text (text, color)	Get colored text.
utils.input.print_text (text[, color, end, file])	Print text with highlighting and no end characters.
utils.math.cosine_similarity (X, Y)	Row-wise cosine similarity between two equal-width matrices.
utils.math.cosine_similarity_top_k (X, Y[, ...])	Row-wise cosine similarity with optional top-k and score threshold filtering.
utils.pydantic.get_pydantic_major_version ()	Get the major version of Pydantic.
utils.strings.comma_list (items)	Convert a list to a comma-separated string.
utils.strings.stringify_dict (data)	Stringify a dictionary.
utils.strings.stringify_value (val)	Stringify a value.
utils.utils.build_extra_kwargs (extra_kwargs, ...)	Build extra kwargs from values and extra_kwargs.
utils.utils.check_package_version (package[, ...])	Check the version of a package.
utils.utils.get_pydantic_field_names (...)	Get field names, including aliases, for a pydantic class.
utils.utils.guard_import (module_name, *[, ...])	Dynamically imports a module and raises a helpful exception if the module is not installed.
utils.utils.mock_now (dt_value)	Context manager for mocking out datetime.now() in unit tests.
utils.utils.raise_for_status_with_text (response)	Raise an error with the response text.
utils.utils.xor_args (*arg_groups)	Validate specified keyword args are mutually exclusive.

langchain.vectorstores

Vector store stores embedded data and performs vector search.

One of the most common ways to store and search over unstructured data is to embed it and store the resulting embedding vectors, and then query the store and retrieve the data that are ‘most similar’ to the embedded query.

Class hierarchy:

VectorStore --> <name> <i># Examples: Annoy, FAISS, Milvus</i>
BaseRetriever --> VectorStoreRetriever --> <name>Retriever <i># Example: VespaRetriever</i>

Main helpers:

Embeddings, Document

Classes

vectorstores.xata.XataVectorStore (api_key, ...)	VectorStore for a Xata database.
vectorstores.bageldb.Bagel ([cluster_name, ...])	Wrapper around BagelDB.ai vector store.
vectorstores.awadb.AwaDB ([table_name, ...])	Interface implemented by AwaDB vector stores.
vectorstores.marqo.Marqo (client, index_name)	Wrapper around Marqo database.
vectorstores.opensearch_vector_search.OpenSearchVectorSearch (...)	Wrapper around OpenSearch as a vector database.
vectorstores.starrocks.StarRocks (embedding)	Wrapper around StarRocks vector database
vectorstores.starrocks.StarRocksSettings	StarRocks Client Configuration
vectorstores.singlestoredb.SingleStoreDB (...)	This class serves as a Pythonic interface to the SingleStore DB database.
vectorstores.singlestoredb.SingleStoreDBRetriever	Retriever for SingleStoreDB vector stores.
vectorstores.deeplake.DeepLake ([...])	Wrapper around Deep Lake, a data lake for deep learning applications.
vectorstores.cassandra.Cassandra (embedding, ...)	Wrapper around Cassandra embeddings platform.
vectorstores.tigris.Tigris (client, ...)	Initialize Tigris vector store
vectorstores.scann.ScaNN (embedding, index, ...)	Wrapper around ScaNN vector database.
vectorstores.clarifai.Clarifai ([user_id, ...])	Wrapper around Clarifai AI platform's vector store.
vectorstores.base.VectorStore ()	Interface for vector stores.
vectorstores.base.VectorStoreRetriever	Retriever class for VectorStore.
vectorstores.typesense.Typesense (...[, ...])	Wrapper around Typesense vector search.
vectorstores.vectara.Vectara ([...])	Implementation of Vector Store using Vectara.
vectorstores.vectara.VectaraRetriever	Retriever class for Vectara.
vectorstores.zilliz.Zilliz (embedding_function)	Initialize wrapper around the Zilliz vector database.
vectorstores.tair.Tair (embedding_function, ...)	Wrapper around Tair Vector store.
vectorstores.pgembedding.BaseModel (**kwargs)	A simple constructor that allows initialization from kwargs.

Toggle Menu

vectorstores.pgembedding.CollectionStore(...)	A simple constructor that allows initialization from kwargs.
vectorstores.pgembedding.EmbeddingStore(**kwargs)	A simple constructor that allows initialization from kwargs.
vectorstores.pgembedding.PGEmbedding(...[, ...])	VectorStore implementation using Postgres and the pg_embedding extension.
vectorstores.pgembedding.QueryResult()	QueryResult is a result from a query.
vectorstores.meilisearch.Meilisearch(embedding)	Initialize wrapper around Meilisearch vector database.
vectorstores.zep.CollectionConfig(name, ...)	A configuration class for a Zep Collection.
vectorstores.zep.ZepVectorStore(...[, ...])	ZepVectorStore is a VectorStore implementation that uses the Zep long-term memory store as a backend.
vectorstores.elasticsearch.ApproxRetrievalStrategy([...])	
vectorstores.elasticsearch.BaseRetrievalStrategy()	
vectorstores.elasticsearch.ElasticsearchStore(...)	Wrapper around Elasticsearch search database.
vectorstores.elasticsearch.ExactRetrievalStrategy()	
vectorstores.elasticsearch.SparseRetrievalStrategy([...])	
vectorstores.analyticdb.AnalyticDB(...[, ...])	VectorStore implementation using AnalyticDB.
vectorstores.sklearn.BaseSerializer(persist_path)	Abstract base class for saving and loading data.
vectorstores.sklearn.BsonSerializer(persist_path)	Serializes data in binary json using the bson python package.
vectorstores.sklearn.JsonSerializer(persist_path)	Serializes data in json using the json package from python standard library.
vectorstores.sklearn.ParquetSerializer(...)	Serializes data in Apache Parquet format using the pyarrow package.
vectorstores.sklearn.SKLearnVectorStore(...)	A simple in-memory vector store based on the scikit-learn library NearestNeighbors implementation.
vectorstores.sklearn.SKLearnVectorStoreException	Exception raised by SKLearnVectorStore.
vectorstores.clickhouse.Clickhouse(embedding)	Wrapper around ClickHouse vector database
vectorstores.clickhouse.ClickhouseSettings	ClickHouse Client Configuration
vectorstores.milvus.Milvus(embedding_function)	Initialize wrapper around the milvus vector database.
vectorstores.elastic_vector_search.ElasticKnnSearch(...)	ElasticKnnSearch is a class for performing k-nearest neighbor (k-NN) searches on text data using Elasticsearch.
vectorstores.elastic_vector_search.ElasticVectorSearch(...)	<i>[Deprecated]</i> Wrapper around Elasticsearch as a vector database.
vectorstores.qdrant.Qdrant(client, ...[, ...])	Wrapper around Qdrant vector database.
vectorstores.qdrant.QdrantException	Base class for all the Qdrant related exceptions
vectorstores.pgvector.BaseModel(**kwargs)	A simple constructor that allows initialization from kwargs.
vectorstores.pgvector.DistanceStrategy(value)	Enumerator of the Distance strategies.
vectorstores.pgvector.PGVector(...[, ...])	VectorStore implementation using Postgres and pgvector.
vectorstores.faiss.FAISS(embedding_function, ...)	Wrapper around FAISS vector database.
vectorstores.azuresearch.AzureSearch(...[, ...])	Azure Cognitive Search vector store.
vectorstores.azuresearch.AzureSearchVectorStoreRetriever	Retriever that uses Azure Search to find similar documents.
vectorstores.alibabacloud_opensearch.AlibabaCloudOpenSearch(...)	Alibaba Cloud OpenSearch Vector Store
vectorstores.alibabacloud_opensearch.AlibabaCloudOpenSearchSettings(...)	Alibaba Cloud Opensearch Client Configuration.
vectorstores.dingo.Dingo(embedding, text_key, *)	Wrapper around Dingo vector database.
vectorstores.redis.Redis(redis_url, ...[, ...])	Wrapper around Redis vector database.
vectorstores.redis.RedisVectorStoreRetriever	Retriever for Redis VectorStore.
vectorstores.chroma.Chroma([...])	Wrapper around ChromaDB embeddings platform.
vectorstores.utils.DistanceStrategy(value[, ...])	Enumerator of the Distance strategies for calculating distances between vectors.
vectorstores.dashvector.DashVector(...)	Wrapper around DashVector vector database.
vectorstores.lancedb.LanceDB(connection, ...)	Wrapper around LanceDB vector database.
vectorstores.atlas.AtlasDB(name[, ...])	Wrapper around Atlas: Nomic's neural database and rhizomatic instrument.
vectorstores.hologres.Hologres(...[, ndims, ...])	VectorStore implementation using Hologres.
vectorstores.hologres.HologresWrapper(...)	Wrapper around Hologres service.
vectorstores.mongodb_atlas.MongoDBAtlasVectorSearch(...)	Wrapper around MongoDB Atlas Vector Search.
vectorstores.pinecone.Pinecone(index, ...[, ...])	Wrapper around Pinecone vector database.
vectorstores.myscale.MyScale(embedding[, config])	Wrapper around MyScale vector database
vectorstores.myscale.MyScaleSettings	MyScale Client Configuration
vectorstores.matching_engine.MatchingEngine(...)	Vertex Matching Engine implementation of the vector store.
vectorstores.usearch.USearch(embedding, ...)	Wrapper around USearch vector database.
vectorstores.weaviate.Weaviate(client, ...)	Wrapper around Weaviate vector database.
<div>Toggle Menu</div> .rocksetdb.Rockset(client, ...)	Wrapper arpund Rockset vector database.

vectorstores.supabase.SupabaseVectorStore(...)	VectorStore for a Supabase postgres database.
vectorstores.annoy.Annoy(embedding_function, ...)	Wrapper around Annoy vector database.
vectorstores.docarray.base.DocArrayIndex(...)	Initialize a vector store from DocArray's DocIndex.
vectorstores.docarray.in_memory.DocArrayInMemorySearch(...)	Wrapper around in-memory storage for exact search.
vectorstores.docarray.hnsw.DocArrayHnswSearch(...)	Wrapper around HnswLib storage.

Functions

vectorstores.alibabacloud_opensearch.create_metadata(fields)	Create metadata from fields.
vectorstores.annoy.dependable_annoy_import()	Import annoy if available, otherwise raise error.
vectorstores.clickhouse.has_mul_sub_str(s, *args)	Check if a string contains multiple substrings.
vectorstores.faiss.dependable_faiss_import([...])	Import faiss if available, otherwise raise error.
vectorstores.myscale.has_mul_sub_str(s, *args)	Check if a string contains multiple substrings.
vectorstores.qdrant.sync_call_fallback(method)	Decorator to call the synchronous method of the class if the async method is not implemented.
vectorstores.scann.dependable_scann_import()	Import scann if available, otherwise raise error.
vectorstores.scann.normalize(x)	Normalize vectors to unit length.
vectorstores.starrocks.debug_output(s)	Print a debug message if DEBUG is True.
vectorstores.starrocks.get_named_result(...)	Get a named result from a query.
vectorstores.starrocks.has_mul_sub_str(s, *args)	Check if a string has multiple substrings.
vectorstores.usearch.dependable_usearch_import()	Import usearch if available, otherwise raise error.
vectorstores.utils.filter_complex_metadata(...)	Filter out metadata types that are not supported for a vector store.
vectorstores.utils.maximal_marginal_relevance(...)	Calculate maximal marginal relevance.

© 2023, Harrison Chase. Last updated on Aug 16, 2023. [Show this page source](#)