Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external
retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing
accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,
chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external
retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing
accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,
chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external
retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing
accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,
chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models

with external retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings, chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings, chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings, chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external retrieval from a knowledge base. RAG allows models to access a large corpus of documents,

increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external

retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external

retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external

retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector

stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external

retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external

retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external

retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external
retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing
accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,
chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external
retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing
accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,
chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external
retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing
accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,
chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models

with external
retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing
accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,
chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external
retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing
accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,
chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external
retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing
accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,
chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external
retrieval from a knowledge base. RAG allows models to access a large corpus of documents,

increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external

retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external

retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external

retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector

stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external

retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external

retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external

retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external
retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing
accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,
chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external
retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing
accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,
chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external
retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing
accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,
chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models

with external

retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external

retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external

retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external

retrieval from a knowledge base. RAG allows models to access a large corpus of documents,

increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external

retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external

retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external

retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector

stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external

retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external

retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external

retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external
retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing
accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,
chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external
retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing
accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,
chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external
retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing
accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,
chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models

with external
retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing
accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,
chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external
retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing
accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,
chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external
retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing
accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,
chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external
retrieval from a knowledge base. RAG allows models to access a large corpus of documents,

increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external

retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external

retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external

retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector

stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.

Retrieval-Augmented Generation (RAG) is a technique that combines traditional language models with external

retrieval from a knowledge base. RAG allows models to access a large corpus of documents, increasing

accuracy and relevance of answers. This document explains RAG concepts, workflows, vector stores, embeddings,

chunking, and fallback mechanisms. It is intended for testing the RAG backend service.