

AI functions

The AI functions allow you to invoke a large language model (LLM) to perform various textual tasks. Multiple LLM providers are supported, specifically [OpenAI](https://platform.openai.com/) [https://platform.openai.com/] and [Anthropic](https://www.anthropic.com/api) [https://www.anthropic.com/api] directly, and many others such as Llama, DeepSeek, Phi, Mistral, or Gemma using [Ollama](https://ollama.com/) [https://ollama.com/].

The LLM must be provided outside Trino as an external service.

Configuration

Because the AI functions require an external LLM service, they are not available by default. To enable them, you must configure a [catalog properties file](#) [../installation/deployment.html#catalog-properties] to register the functions invoking the configured LLM under the specified catalog name.

Create a catalog properties file `etc/catalog/llm.properties` that references the `ai` connector:

```
connector.name=ai
```

The AI functions all use the `ai` schema name. For the preceding example, the functions use the `llm.ai` catalog and schema prefix.

To avoid needing to reference the functions using their fully qualified name, you can configure the `sql.path` [SQL environment property](#) [../admin/properties-sql-environment.html] in the `config.properties` file to include the catalog and schema prefix:

```
sql.path=llm.ai
```

Configure multiple catalogs to use the same functions with different LLM providers. In this case, the functions must be referenced using their fully qualified name, rather than relying on the SQL path.

Providers

The AI functions invoke an external LLM, which must be configured for the catalog. Performance, results, and cost of all AI function invocations are completely dependent on the LLM provider and the model used. You must specify a model that is suitable for textual analysis.

Anthropic

The Anthropic provider uses the [Anthropic API](https://www.anthropic.com/api) [https://www.anthropic.com/api] to perform the AI functions:

```
ai.provider=anthropic  
ai.model=claude-3-5-sonnet-latest  
ai.anthropic.api-key=xxx
```

OpenAI

The OpenAI provider uses the [OpenAI API](https://platform.openai.com/) [https://platform.openai.com/] to perform the AI functions:

```
ai.provider=openai  
ai.model=gpt-4o-mini  
ai.openai.api-key=xxx
```

Ollama

The OpenAI provider can be used with **Ollama** [https://ollama.com/] to perform the AI functions, as Ollama is compatible with the OpenAI API:

```
ai.provider=openai
ai.model=llama3.3
ai.openai.endpoint=http://localhost:11434
ai.openai.api-key=none
```

The API key must be specified, but will be ignored by Ollama.

Functions

The following functions are available in each catalog configured with the `ai` connector under the `ai` schema and use the configured LLM provider:

ai_analyze_sentiment(text) → varchar

Analyzes the sentiment of the input text.

The sentiment result is `positive`, `negative`, `neutral`, or `mixed`.

```
SELECT ai_analyze_sentiment('I love Trino');
-- positive
```

ai_classify(text, labels) → varchar

Classifies the input text according to the provided labels.

```
SELECT ai_classify('Buy now!', ARRAY['spam', 'not spam']);
-- spam
```

ai_extract(text, labels)

Extracts values for the provided labels from the input text.

```
SELECT ai_extract('John is 25 years old', ARRAY['name', 'age']);
-- {name=John, age=25}
```

ai_fix_grammar(*text*) → varchar

Corrects grammatical errors in the input text.

```
SELECT ai_fix_grammar('I are happy. What you doing?');  
-- I am happy. What are you doing?
```

ai_gen(*prompt*) → varchar

Generates text based on the input prompt.

```
SELECT ai_gen('Describe Trino in a few words');  
-- Distributed SQL query engine.
```

ai_mask(*text*, *labels*) → varchar

Masks values for the provided labels in the input text.

```
SELECT ai_mask(  
    'Contact me at 555-1234 or visit us at 123 Main St.',  
    ARRAY['phone', 'address']);  
-- Contact me at [MASKED] or visit us at [MASKED].
```

ai_translate(*text*, *language*) → varchar

Translates the input text to the specified language.

```
SELECT ai_translate('I like coffee', 'es');  
-- Me gusta el café  
  
SELECT ai_translate('I like coffee', 'zh-TW');  
--
```