Dokumentation des Backends der Bachelorarbeit

Konstantin Benz

16. September 2024

Konfigurationsdatei

Parametername	Standardwert	Beschreibung
[FastAPI]		
enable_tls	"true"	Aktiviert TLS-Verschlüsselung für sichere Kommunikation.
ip_address	"0.0.0.0"	IP-Adresse, an die der FastAPI-Server gebunden wird.
port	"8000"	Portnummer, auf dem der FastAPI-Server hört.
[cors]		
allow_origins	["*"]	Liste der erlaubten Ursprünge für CORS-Anfragen („*“ erlaubt alle Ursprünge).
allow_credentials	"True"	Aktiviert die Unterstützung von Benutzerdaten in CORS-Anfragen.
allow_methods	["*"]	Liste der erlaubten HTTP-Methoden („*“ erlaubt alle Methoden).
allow_headers	["*"]	Liste der erlaubten Header in CORS-Anfragen.
[vectorDatabase]		
embedding_model	"BAAI/bge-m3"	Gibt das Embedding-Modell an, das zur Vektorisierung von Daten verwendet wird.
use_gpu	"false"	Bestimmt, ob eine GPU für Berechnungen verwendet werden soll.
[opcua]		
opcua_interval	"2000"	Intervall (in Millisekunden) für OPC UA-Subscriptions.
[llm]		
use_local_LLM	"false"	Legt fest, ob ein lokales LLM oder ein Cloud-basiertes Modell verwendet wird.
llm_local_fileName	"Beispiel.gguf"	Dateiname für das lokale LLM-Modell.
llm_local_ctxsize	"8192"	Kontextgröße des lokalen LLM (maximales Tokenfenster).
llm_local_layers	"40"	Anzahl der Schichten in der Architektur des lokalen LLM.
llm_local_batchsize	"256"	Batch-Größe, die für Inferenzen mit dem lokalen LLM verwendet wird.
llm_cloud_hoster	"cohere"	Cloud-Anbieter für das LLM, wenn kein lokales Modell verwendet wird.
llm_cloud_model	"command-r"	Cloud-LLM-Modell, das verwendet werden soll.
[Beispielmaschine]		(Parameter müssen ermittelt werden)
type	ß.B. nbh630"	Typkennung für die Maschine.
ip_address	"127.0.0.1"	IP-Adresse des OPC UA-Servers für diese Maschine.
port	"4840"	Portnummer des OPC UA-Servers für diese Maschine.
vdb_name	ß.B. nbh630"	Name der Vektordatenbank für diese Maschine.
from_node_id	"ns=2;i=2"	Start-Knoten-ID für die OPC UA-Datenextraktion.
to_node_id	"ns=2;i=34"	End-Knoten-ID für die OPC UA-Datenextraktion.
additional_prompt	"Beispiel"	Zusätzlicher Eingabetext für das LLM (falls vorhanden).
opcua_use_certificate	"false"	Gibt an, ob die Zertifikatsauthentifizierung für OPC UA verwendet wird.
opcua_username	üser"	Benutzername für die OPC UA-Authentifizierung.
opcua_password	"pass"	Passwort für die OPC UA-Authentifizierung.

Tabelle 1: Konfigurationsparameter des Backends

Cloud-Anbieter	Modellname	Beschreibung
[Groq]		
groq	llama3-8b-8192	LLaMA 3, 8 Milliarden Parameter, Kontextgröße 8192 Tokens.
groq	llama3-70b-8192	LLaMA 3, 70 Milliarden Parameter, Kontextgröße 8192 Tokens.
groq	mixtral-8x7b-32768	Mixtral-Modell, 8x7 Milliarden Parameter, Kontextgröße 32768 Tokens.
groq	gemma-7b-it	Gemma-Modell, 7 Milliarden Parameter, optimiert für Italienisch.
groq	gemma2-9b-it	Gemma 2, 9 Milliarden Parameter, optimiert für Italienisch.
[OpenAI]		
openai	gpt-4o	GPT-4o, optimierte Version von GPT-4.
openai	gpt-3.5	GPT-3.5, Modell für generelle Anwendungen.
openai	gpt-4	GPT-4, leistungsfähigeres Modell mit größerem Kontextfenster.
openai	gpt-3.5-turbo-instruct	GPT-3.5 Turbo, optimiert für schnelle Anweisungsverarbeitung.
[Cohere]		
cohere	command-r	Command-R, Modell von Cohere für Anweisungsverarbeitung.
cohere	command-r-plus	Command-R Plus, erweiterte Version mit besserer Leistung.

Tabelle 2: Verfügbare LLM-Modelle und deren Optionen

Erklärung zur Auswahl der LLMs

In der Konfigurationsdatei kann die Auswahl eines LLMs durch das Setzen der folgenden Parameter erfolgen:

- `llm_cloud_hoster`: Definiert den Cloud-Anbieter, der für das Hosting des LLMs verantwortlich ist. Mögliche Optionen sind `"groq"`, `öpenai"` oder `"cohere"`.

- `llm_cloud_model`: Legt das spezifische LLM-Modell fest, das verwendet werden soll. Zum Beispiel könnte `"gpt-4"` für OpenAI oder `"command-r"` für Cohere gewählt werden.

Beispielkonfiguration:

```
[llm]
llm_cloud_hoster = "openai"
llm_cloud_model  = "gpt-4"
```

In diesem Beispiel wird OpenAI als Cloud-Anbieter genutzt, und das GPT-4-Modell wird für die Verarbeitung ausgewählt.

Die Wahl des Modells hängt von der jeweiligen Anwendung ab und beeinflusst die Performance und die Verarbeitungsfähigkeit des Systems. LLMs mit mehr Parametern und größerem Kontextfenster, wie `gpt-4` oder `llama3-70b-8192`, bieten in der Regel genauere Ergebnisse, erfordern jedoch auch mehr Rechenleistung.