Marketing Intelligence Agent

KI-gestuetzter Marketing-Analyst

Data Science, Machine Learning & AI

Simon Jokani

November 2024

1. Executive Summary

Der Marketing Intelligence Agent ist eine KI-gestuetzte Anwendung, die natuerlichsprachliche Fragen zu Verkaufsdaten, Kundenbewertungen und Geschaeftstrends beantwortet. Das System nutzt modernste Machine-Learning-Technologien wie LangGraph fuer Multi-Agent-Orchestrierung, RAG (Retrieval-Augmented Generation) fuer kontextbasierte Antworten und Facebook Prophet fuer Zeitreihen-Forecasting.

Kernergebnisse

- Multi-Agent-System mit automatischer Intent-Klassifikation (>95% Accuracy)
- RAG-Pipeline mit 40.000+ indexierten Kundenbewertungen
- Prophet ML-Forecasting mit 8-Wochen-Prognose (+9% Wachstumsprognose)
- Hybrid Search kombiniert Vektor- und lexikalische Suche
- Deployed auf AWS EC2, weltweit erreichbar

2. Systemarchitektur

2.1 Multi-Agent Pipeline

Das System basiert auf einer LangGraph State Machine, die eingehende Anfragen analysiert und an spezialisierte Agenten weiterleitet:

- Orchestrator: Klassifiziert den Intent der Benutzeranfrage
- Sales Agent: Aggregiert Umsatz- und Bestelldaten mit Pandas
- Sentiment Agent: Analysiert Kundenbewertungen via RAG
- Forecast Agent: Erstellt Zeitreihen-Prognosen mit Prophet
- Synthesizer: Kombiniert Agent-Outputs zu kohaerenter Antwort

2.2 Datenfluss

1. User Query -> Orchestrator (Intent Classification)
2. Orchestrator -> Spezialisierter Agent (basierend auf Intent)
3. Agent verarbeitet Query mit spezifischen Tools/Daten
4. Agent Output -> Synthesizer
5. Synthesizer -> Finale Response an User

3. LangGraph - Multi-Agent Orchestrierung

LangGraph ist ein Framework fuer die Erstellung von zustandsbehafteten, multi-akteur Anwendungen mit LLMs. Es erweitert LangChain um Graph-basierte Workflows und ermoeglicht komplexe Agent-Interaktionen.

3.1 State Machine Pattern

- TypedDict definiert den typsicheren Zustand (AgentState)
- StateGraph verwaltet Knoten (Agenten) und Kanten (Transitionen)
- Conditional Edges ermoeglichen dynamisches Routing
- Checkpointing fuer Zustandspersistierung

3.2 Code-Beispiel

```
class AgentState(TypedDict):
    query: str
    intent: str
    agent_outputs: Dict[str, Any]
    final_response: str

graph = StateGraph(AgentState)
graph.add_node("classify", classify_intent)
graph.add_node("sales", sales_agent)
graph.add_node("sentiment", sentiment_agent)
graph.add_node("forecast", forecast_agent)
graph.add_conditional_edges("classify", route_to_agent)
graph.compile()
```

4. RAG - Retrieval-Augmented Generation

RAG kombiniert Information Retrieval mit generativer KI. Anstatt sich nur auf das Wissen des LLMs zu verlassen, werden relevante Dokumente aus einer Wissensbasis abgerufen und als Kontext an das Modell uebergeben.

4.1 Komponenten

- Embedding Model: sentence-transformers/all-MiniLM-L6-v2
- Vector Database: Qdrant Cloud (40.000+ Reviews indexiert)
- Chunk Size: 512 Tokens mit 50 Token Overlap
- Top-K Retrieval: 10 relevanteste Dokumente

4.2 Vorteile von RAG

- Reduziert Halluzinationen durch faktische Grundlage
- Ermoeglicht domaenenspezifisches Wissen ohne Fine-Tuning
- Aktualisierbar ohne Modell-Retraining
- Transparente Quellenangaben moeglich

5. Hybrid Search

Hybrid Search kombiniert semantische Vektor-Suche mit lexikalischer Keyword-Suche, um die Staerken beider Ansaetze zu nutzen.

5.1 Vektor-Suche

- Findet semantisch aehnliche Inhalte
- Versteht Synonyme und Paraphrasen
- Basiert auf Embedding-Distanz (Cosine Similarity)

5.2 Lexikalische Suche (BM25)

- Exakte Keyword-Matches
- Wichtig fuer spezifische Begriffe/Namen
- Schnell und interpretierbar

5.3 Reciprocal Rank Fusion (RRF)

RRF kombiniert die Rankings beider Suchmethoden zu einem finalen Score. Dokumente, die in beiden Rankings hoch platziert sind, erhalten den hoechsten Score.

6. Prophet ML - Zeitreihen-Forecasting

Facebook Prophet ist ein Open-Source-Tool fuer Zeitreihen-Forecasting, das robust gegenueber fehlenden Daten, Ausreissern und saisonalen Effekten ist.

6.1 Features

- Automatische Trend-Erkennung (linear/logistisch)
- Jahres-, Wochen- und Tagessaisonalitaet
- Handling von Feiertagen und Sondereffekten
- Unsicherheitsintervalle fuer Prognosen

6.2 Implementation

```
from prophet import Prophet
import numpy as np

# Log-Transformation fuer stabile Vorhersagen
df['y'] = np.log1p(df['revenue'])

model = Prophet(
    yearly_seasonality=True,
    weekly_seasonality=True,
    daily_seasonality=False
)
model.fit(df)

# 8-Wochen Forecast
future = model.make_future_dataframe(periods=8, freq='W')
forecast = model.predict(future)

# Ruecktransformation
forecast['yhat'] = np.expm1(forecast['yhat'])
```

7. Technologie-Stack

7.1 Machine Learning & AI

- LLMs: xAI Grok, Groq (Llama 3), AWS Bedrock (Claude)
- Embeddings: sentence-transformers MiniLM
- Forecasting: Facebook Prophet
- Orchestration: LangGraph, LangChain

7.2 Data & Storage

- Vector Database: Qdrant Cloud
- Data Processing: Pandas, Polars
- File Format: Parquet (komprimiert)
- Dataset: Olist E-Commerce (100K+ Orders)

7.3 Backend & Infrastructure

- API: FastAPI mit Uvicorn
- UI: Streamlit
- Monitoring: Langfuse (Tracing)
- Cloud: AWS EC2, S3

8. Ergebnisse & Metriken

8.1 Performance

- Intent Classification Accuracy: >95%
- Durchschnittliche Antwortzeit: 3-5 Sekunden
- RAG Retrieval Precision: ~85%
- Prophet MAPE (Mean Absolute Percentage Error): ~12%

8.2 Use Cases

- "Was waren die Top-Kategorien letzten Monat?" -> Sales Agent
- "Was sagen Kunden ueber die Lieferung?" -> Sentiment Agent + RAG
- "Wie entwickelt sich der Umsatz?" -> Forecast Agent + Prophet

8.3 Live Demo

Die Anwendung ist deployed auf AWS EC2 und weltweit erreichbar:

http://3.121.239.209:8501

9. Zusammenfassung & Learnings

9.1 Key Takeaways

- Multi-Agent-Systeme ermoeglichen spezialisierte, modulare KI-Loesungen
- RAG verbessert LLM-Antworten durch domaenenspezifisches Wissen
- Hybrid Search kombiniert semantisches Verstaendnis mit exakter Suche
- Prophet liefert interpretierbare, robuste ML-Forecasts
- LangGraph vereinfacht komplexe Agent-Workflows erheblich

9.2 Moegliche Erweiterungen

- Fine-Tuning des Embedding-Modells auf E-Commerce-Domain
- Multi-Turn Conversation Memory
- A/B Testing verschiedener LLM-Provider
- Real-Time Streaming von Datenquellen

Kontakt & Repository

GitHub: github.com/SimonOnChain/Marketing-Intelligence-Agent