SocietyByte
BFH-Magazin für die Humane Digitale Transformation

Chatbot für Stillstands-Daten eröffnet neue Wege in der Produktion

Von Kilian Schürch , Jürgen Vogel | 0 Kommentare

Von der Betriebsstörung zum datengetriebenen Erkenntnisgewinn – Wie ein KI-Chatbot auf Basis von Retrieval Augmented Generation die Auswertung von Maschinenausfällen revolutioniert.

Einleitung: Wenn Maschinenstillstand zum Informationsproblem wird

In modernen Fertigungsumgebungen ist es entscheidend, Stillstände zu minimieren, da jede Minute Maschinenstillstand erhebliche Kosten verursacht und Produktionszeit vernichtet. Trotz Software-Unterstützung durch Manufacturing Execution Systems (MES) bleibt viel Potenzial zur schnellen Datenauswertung ungenutzt. Berichte zu Ausfällen sind oft unstrukturiert und über verschiedene Systeme verstreut, was die Analyse erschwert.
Hier setzt ein Prototyp an, der auf Large Language Models (LLMs) in Kombination mit Retrieval Augmented Generation (RAG) basiert. Ziel ist es, Stillstands-Daten in natürlicher Sprache sofort und präzise abzurufen, zu interpretieren und für Optimierungen im Produktionsalltag nutzbar zu machen.

Stillstands-Daten und ihre ungenutzten Schätze

Im konkreten Projektbeispiel bei Hoffmann Neopac AG, einem führenden Hersteller hochwertiger Kunststoffverpackungen, werden alle Anlagenstörungen als „Downtime-Notes“ direkt am Linien-Panel erfasst. Zusätzlich werden Leistungsdaten wie Geschwindigkeit und Ausschussraten pro Maschine mittels Sensoren aufgezeichnet.
Folgende Herausforderungen treten dabei auf:

- Stillstands-Daten sind oft unstrukturiert, da sie in frei formulierten Notizen vorliegen.
- Differenzierte Analysen, etwa welche Materialien häufig Stillstände verursachen oder welche Fehler bei bestimmten Linien gehäuft auftreten, erfordern fundierte Datenanalysefähigkeiten.
- Viele Mitarbeiterinnen und Mitarbeiter haben nicht das technische Know-how oder die Zeit, komplexe Auswertungen zu erstellen.

Das Ergebnis: Wertvolle Erkenntnisse bleiben oft ungenutzt, weil die Datenaufbereitung als zu aufwendig erscheint.

RAG als Schlüssel: Wie KI-Modelle externe Wissensquellen einbinden

RAG adressiert genau dieses Problem. Während klassische Sprachmodelle (LLMs) ihre „Welt“ nur aus den riesigen Textmengen im Trainingsdatensatz beziehen, erweitert RAG das Modell um eine zusätzliche Wissensquelle. Dieses zusätzliche Wissen kann eine Datenbank, ein Dateisystem oder das Internet sein.

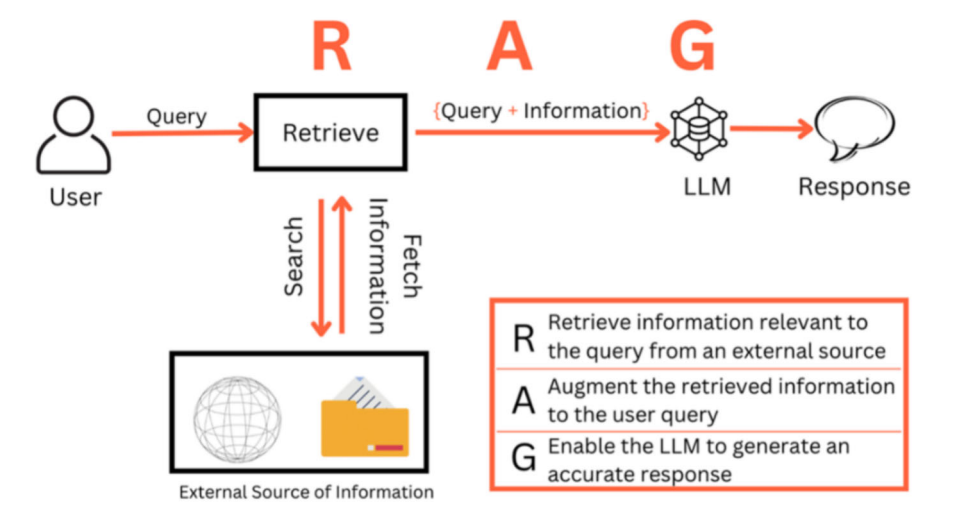

Abbildung 1: Ablauf Retrieval-Augmented Generation (RAG) (Quelle: [1] [#_edn1])

Die Abbildung 1 veranschaulicht den Ablauf des RAG-Workflow. Der Prozess erfolgt in drei Schritten:

1. **Retrieve**: Eine Abfrage (Query), wie beispielsweise „Zeige mir die häufigsten Fehlerursachen im August auf Linie 1“, wird vom Nutzer (User) an einen Retriever gesendet. Dieser durchsucht eine Wissensquelle und ruft relevante Informationen ab.
2. **Augment**: Die gefundenen Einträge werden mit der ursprünglichen Abfrage kombiniert und als erweiterter Kontext ({Query + Information}) an das Sprachmodell (LLM) übergeben.
3. **Generate**: Das LLM erstellt eine kontextbezogene und faktenbasierte Antwort.

Dieser Prozess verhindert erfundene Inhalte („Halluzinationen“) und ermöglicht präzise Antworten aus firmeneigenen Daten, ohne das Modell selbst neu trainieren zu müssen.

Neo4j und Chatbot: Architektur für strukturierte Daten

Damit das System Zusammenhänge zwischen Produkt, Linie und Ausfallkategorie versteht, wird ein Graphdatenbank-Ansatz verwendet. Im Prototyp kommt Neo4j AuraDB [2] [#_edn2] zum Einsatz, wo Entitäten wie „Downtime“ und „TubeLine“ als Knoten und Beziehungen wie „OCCURS_ON“ (Stillstand auf einer Linie) als Kanten gespeichert werden.
Ein solches Graphen-Modell ist besonders geeignet für LLMs, da es sich leicht in natürlicher Sprache abbilden lässt. LLMs erkennen sprachliche Muster und Beziehungen in der Graphenstruktur. Die speziell für Graphdatenbanken entwickelte Abfragesprache Cypher [3] [#_edn3] ist zudem intuitiver und übersichtlicher als SQL (Structured Query Language), besonders bei komplexen Beziehungen zwischen mehreren Entitäten.

Die Abbildung 2 veranschaulicht die Architektur des Chatbots und zeigt die Interaktion zwischen den einzelnen Komponenten:

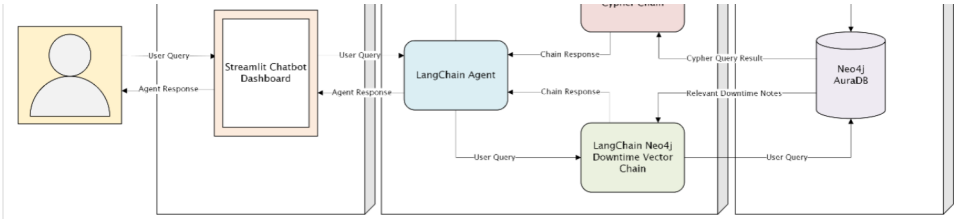

Abbildung 2: Architektur des Chatbots

1. **Chatbot Frontend:** Über ein Streamlit-Dashboard [4] [#_edn4] können Nutzerinnen und Nutzer Anfragen in natürlicher Sprache stellen. Das Frontend leitet die Anfrage an den LangChain Agent weiter, der eine passende Abfrage generiert und das Ergebnis als Klartext zurückgibt.
2. **Chatbot API:** Hier läuft der eigentliche RAG-Workflow, gesteuert durch die LangChain-Bibliothek [5] [#_edn5] . Der LangChain Agent entscheidet basierend auf der Frage, ob die Neo4j Cypher Chain (für strukturierte Datenabfragen) oder die Downtime Vector Chain (für semantische Suchen) genutzt wird. Die entsprechenden Abfragen werden an die Neo4j AuraDB weitergeleitet, und die Ergebnisse fliessen in die Antwortgenerierung ein.
3. **Neo4j Datenbank:** Enthält alle aufbereiteten Daten aus den verschiedenen Datenquellen.

Cypher Chain und Vector Chain: Zwei Ansätze für unterschiedliche Daten

Für strukturierte Informationen wie Produktionsdaten eignet sich die Cypher-Chain, bei dem das LLM mithilfe von Cypher-Queries in der Graphdatenbank recherchiert. Anders ist es bei Downtime-Notes, die kurze, ähnliche Störungsmeldungen enthalten. Hier wird die semantische Vektorsuche (Vector Chain) genutzt. Downtime-Notes werden mit Embedding-Modellen in Vektoren umgewandelt und gespeichert. Diese Modelle übersetzen Wörter oder Sätze in numerische Vektoren, um ähnliche Begriffe zu erfassen. So werden auch sinngleiche Formulierungen erkannt.

Neo4j kann flexibel als Vektordatenbank erweitert werden. Anfragen werden ebenfalls in Embedding-Vektoren umgewandelt und mit Distanzmetriken verglichen, um passende Störungsmeldungen zu finden, auch wenn die Formulierungen voneinander abweichen.

Evaluation: Genauigkeit, Kosten, Geschwindigkeit

Ein zentrales Anliegen war die Evaluierung unterschiedlicher KI-Modelle. In einer ersten Prototypphase wurden LLMs von OpenAI getestet [6] [#_edn6] . Die Ergebnisse bei 30 Testfragen (Englisch und Deutsch) zeigen:

- **gpt-4o-mini** erwies sich als „Sweet Spot“: Hohe Genauigkeit (über 86 % im Englischen, bis zu 90 % im Deutschen) bei gleichzeitig niedrigen Kosten und kurzen Antwortzeiten.
- **gpt-4o** lieferte in einigen Fällen ähnlich präzise Ergebnisse, war jedoch um ein Vielfaches teurer.
- **gpt-3.5-turbo** war zwar günstig aber teils ungenau bei komplexen Anfragen.

Gerade für Unternehmen, die viele Fragen pro Woche an ihren Chatbot richten, sind Latenz und Kosten entscheidende Kriterien. Hier bietet das leichtere 4o-mini-Modell eine attraktive Balance.

Fazit: Ein neuer Standard für datenbasierte Entscheidungen

Der LLM-RAG-Chatbot zeigt, wie KI-Modelle echten Mehrwert in der Industrie schaffen können. Statt mühsamer Recherche erhalten Fachkräfte schnell fundierte Antworten zu Ausfallzeiten, Fehlerursachen und Performancezahlen.
Das Zusammenspiel aus Graphdatenbank und Retrieval Augmented Generation liefert präzise, kontextbasierte Einblicke.

Der Einsatz von KI im Produktionsumfeld geht längst über Prognosen oder vorausschauende Wartung hinaus. Mit RAG-Lösungen lassen sich unternehmenseigene Datenquellen effizient nutzen, um fundierte Entscheidungen zu treffen. Das Ergebnis ist höhere Effizienz, niedrigere Kosten und verbesserte Transparenz in der Produktion – ein deutlicher Schritt in Richtung *Smart Factory*.

Referenzen

1 [#_ednref1] A. Kimothi, „1 Large Language Models and the Need for Retrieval Augmented Generation,“ in A Simple Guide to Retrieval Augmented, Manning Publications, 2024, pp. 1-17.

2 [#_ednref2] Neo4j Inc., „Neo4j AuraDB: Fully Managed Graph Database," 2025. [Online]. Available: https://neo4j.com/product/auradb/. [Zugriff am 10 03 2025].

3 [#_ednref3] Neo4j Inc., „Cypher Manual," 2025. [Online]. Available: https://neo4j.com/docs/cypher-manual/current/introduction/. [Zugriff am 10 03 2025].

4 [#_ednref4] Snowflake Inc., „A faster way to build and share data apps," 2024. [Online]. Available: https://streamlit.io/. [Zugriff am 10 03 2025].

5 [#_ednref5] LangChain, „Applications that can reason. Powered by LangChain," 2025. [Online]. Available: https://www.langchain.com/. [Zugriff am 10 03 2025].

6 [#_ednref6] OpenAI, „OpenAI Platform," 2025. [Online]. Available: https://platform.openai.com/docs/models. [Zugriff am 10 03 2025].

AUTHOR: KILIAN SCHÜRCH

Kilian Schürch ist Teilzeitstudent im Master of Science in Engineering (MSE) mit Vertiefung Data Science an der Berner Fachhochschule und arbeitet als Projektleiter Process Development bei der Hoffmann Neopac AG. Sein Fokus liegt auf Datenanalyse, Prozessautomatisierung, LLM-RAG-Applikationen und Predictive Maintenance.

Posts from Kilian Schürch

AUTHOR: JÜRGEN VOGEL

Dr. Jürgen Vogel ist Professor am Institut IDAS der Berner Fachhochschule. Er lehrt und forscht in den Bereichen Data Engineering, KI und Maschinelles Lernen mit Schwerpunkt auf der Verarbeitung von natürlicher Sprache, beispielsweise anhand von LLMs.

Posts from Jürgen Vogel

Create PDF

Ähnliche Beiträge

Es wurden leider keine ähnlichen Beiträge gefunden.

0

COMMENTS