Beyond Q&A: Entwicklung eines agentenbasierten Retrieval-Augmented-Generation-Systems bei der imos AG

Raphael Bourdick [1] und Frederik S. Bäumer [2]

Abstract

Dieser Praxisbeitrag zeigt, wie ein *Retrieval-Augmented-Generation*-System zu einer agentenbasierten Architektur weiterentwickelt wurde, das über reine Frage-Antwort-Funktionalitäten hinausgeht und komplexe, mehrstufige Unternehmensprozesse automatisiert. Die Untersuchung adressiert die Anforderungen kleiner und mittlerer Unternehmen, die trotz begrenzter IT-Ressourcen von modernen KI-Verfahren profitieren möchten.

1 Einleitung

Kleine und mittlere Unternehmen (KMU) stehen vor einer doppelten Herausforderung. Einerseits entwickeln sich Large Language Models (LLMs) rasant weiter und ermöglichen vielfältige Anwendungen – von dialogbasierten Chatbots bis zu komplexen Textanalysen. Andererseits fehlen vielen KMU die notwendigen Ressourcen, um diese Technologien eigenständig zu betreiben oder zu integrieren. Insbesondere Rechenkapazitäten und Fachpersonal sind limitierende Faktoren. Damit bleiben Potenziale ungenutzt, obwohl gerade KMU erheblich von Automatisierung und intelligenten Assistenzsystemen profitieren könnten [Bä24; LG21; UF21]. Ein vielversprechender Lösungsansatz ist hier *Retrieval-Augmented Generation* (RAG). Er kombiniert die generische Ausdrucksfähigkeit großer Sprachmodelle mit domänenspezifischem Unternehmenswissen [Ba24]. Bei der imos AG wurde auf dieser Grundlage das System KIKUS (KI im Kundenservice) entwickelt, das als klassisches RAG-System in einem technischen Frage-Antwort-Chatbot zum Einsatz kam.

Im praktischen Einsatz zeigte sich, dass der statische Charakter klassischer RAG-Architekturen an Grenzen stößt. Sobald Kundenanfragen nicht nur eine Information, sondern weiterführende Aktionen erfordern – etwa das Anstoßen interner Prozesse – reicht eine rein reaktive Antwortlogik nicht mehr aus. Als Reaktion darauf wurde KIKUS schrittweise um agentenbasierte Mechanismen erweitert. Das System konnte fortan eigenständig Entscheidungen treffen, Zwischenschritte planen und gezielt auf externe Tools zugreifen. Diese Weiterentwicklung markiert den Übergang zu einer agentenbasierten RAG-Architektur innerhalb desselben Anwendungskontexts. Gleichzeitig entstand aus dem wachsenden

[1] imos AG, Planckstraße 24, 32052 Herford, RBourdick@imos3d.com, https://orcid.org/0009-0005-3210-9443
[2] Hochschule Bielefeld, AG Angewandte KI, Interaktion 1, 33619 Bielefeld, frederik.baeumer@hsbi.de, https://orcid.org/0000-0002-0826-0144

Bedarf nach automatisierter Prozessausführung ein eigenständiges Projekt: KITS (KI-Transformer-Systeme). Anders als KIKUS, das weiterhin auf textbasierte Kundenanfragen fokussiert bleibt, ist KITS darauf ausgelegt, komplexe, mehrstufige Arbeitsabläufe, wie die Umwandlung ganzer Produktkataloge in XML-Formate, vollständig zu automatisieren.

2 RAG als Brücke zwischen generischen LLMs und KMU-Wissen

RAG ist ein hybrider Ansatz [Ba24], bei dem ein generatives Sprachmodell durch kontextuelle Dokumentenfragmente aus einer externen Wissensbasis ergänzt wird. Das Modell verarbeitet dabei sowohl die Anfrage als auch die abgerufenen Inhalte, was in der Regel zu faktentreueren und kontextsensitiveren Antworten führt [Bä24]. Besonders für KMU bietet dieses Verfahren Vorteile. Einerseits ermöglicht es die Integration des „Weltwissens“ großer LLMs, andererseits können unternehmensspezifische Dokumente kontinuierlich aktualisiert und in den Antwortprozess einbezogen werden [Bä24]. In der Praxis entwickelte sich RAG schnell weiter. Moderne Varianten nutzen zusätzlich intelligentes *Chunking*, *Re-Ranking* relevanter Textpassagen oder iteratives *Retrieval*, um auch komplexe Anfragen mehrstufig bearbeiten zu können. Zwar steigt dadurch die Komplexität der Architektur, doch eröffnet dies auch neue Einsatzfelder, bei denen Informationsquellen konsolidiert werden müssen.

Bei der imos AG kam RAG zunächst in klassischer Form zum Einsatz. Das System KIKUS nutzte eine semantische Suche zur Auswahl relevanter Textstellen aus Dokumentationen im DITA-Format (technische Handbüchern und FAQs). Diese Passagen wurden zusammen mit der ursprünglichen Anfrage an ein generatives Modell übergeben, das daraus eine fundierte Antwort ableitete. Die Resultate erwiesen sich als qualitativ hochwertig und effizient – insbesondere bei standardisierten Anfragen im technischen Support.

Zur Performanzsteigerung wurden einzelne Komponenten des Systems gezielt optimiert. Neben dem Einsatz stärkerer Sprachmodelle (z. B. GPT-4o) erfolgten Verbesserungen im *Prompting*, eine feinjustierte Kombination aus semantischer Suche und klassischen Rankingverfahren (BM25, *Bi-* und *Cross-Encoder*), sowie die schrittweise Automatisierung der Wissensbasispflege. Letztere erfolgte über eine *Pipeline*, die DITA-Dokumente in *Markdown* überführte und tokenoptimiert *chunkte*. Die Wissenbasis kann so bei Updates voll automatisch neu geladen werden. Ergänzend wurden Q&A-Paare direkt aus Nutzungsdaten extrahiert, geprüft und eingepflegt, um punktuell und bedarfsgerecht Wissenslücken der Dokumentation durch Frage- Antwortpaare zu füllen. Doch trotz dieser Fortschritte blieb eine zentrale Limitation bestehen: Der statische Charakter. Eine Frage löst stets denselben Prozess aus – *Retrieval*, Übergabe, Generierung – unabhängig von der Präzision oder Zielrichtung der Anfrage. Unklare Fragen führen so oft zu ungeeigneten Kontexten und minderwertigen Antworten. Ansätze, diese Limitation durch Arbeitsteilung auf mehrere LLMs zu umgehen, scheiterten an der fehlenden Koordination zwischen den Modellen. Die Konfiguration komplexer *if-then*-Strukturen im *Prompt* erwies sich als fehleranfällig und schwer skalierbar, insbesondere in *Edge Cases*. Diese treten z. B. bei unerwartetem Verhalten der Nutzenden auf, das den Standard-Ablauf durchbricht. Stellen Nutzende etwa eine domänenfremde Frage,

löst die klassische *Pipeline* dennoch ein *Retrieval* aus. Das System erhält damit irrelevanten Kontext, weil das *Retrieval*-Modul zwingend textuellen *Input* erwartet. Wenn darüber hinaus lediglich ein Bild, z. B. ein *Screenshot* einer Fehlermeldung, übermittelt wird, kann es an jeglichem Kontext mangeln. Die Folge sind Halluzinationen oder Fehlantworten. Diese Beobachtungen legten den Grundstein für die Hinwendung zu flexibleren, agentenbasierten Architekturen, die Entscheidungslogik und Prozesssteuerung dynamisch interpretieren.

3 Von statischer RAG-Architektur zu einem agentenbasierten System

Ein Agent agiert nicht bloß als Reaktionsmechanismus, sondern als kognitive Instanz mit Entscheidungsautonomie. Er analysiert eine Anfrage, plant eigenständig notwendige Handlungsschritte, greift bei Bedarf auf externe *Tools* oder Datenbanken zu und kann Zwischenergebnisse iterativ verarbeiten. Im Gegensatz zu statischen *Workflows*, die durch feste Regeln definiert sind, handelt ein Agent situativ – ein Schritt in Richtung robuster Automatisierung. Im Rahmen der Weiterentwicklung von KIKUS wurde ein hierarchisches Agentendesign implementiert, bei dem ein zentraler Agent die gesamte Konversation steuert. Dieser besitzt Zugriff auf die *Chat*-Historie und nutzt bei Bedarf Werkzeuge wie ein *Retrieval*-Modul oder *Re-Ranking*-Komponenten. Der Agent auf Basis von *GPT-4o* (gehostet über Azure OpenAI in einer EU-Region) steuert zwei Werkzeuge. Das essenzielle *Retrieval*-Tool erzeugt seine Vektorrepräsentationen (`text-embedding-3-large`, OpenAI) und kombiniert eine semantische Suche auf Basis der Kosinusähnlichkeit mit einer klassischen BM25-Keywordsuche. Ein optionales *Re-Ranking*-Tool nutzt ein leichtgewichtiges *GPT-4o mini*, um die Top-*k* *Retrieval*-Treffer qualitativ neu zu ordnen. Das vollständige System wurde jedoch anbieteragnostisch entwickelt, um Abhängigkeiten von einzelnen Anbietern zu vermeiden. Die Modelle lassen sich durch Alternativen (z. B. Voyage *Embeddings* und Anthropic Sprachmodelle) austauschen. Alle Komponenten werden verbrauchsabhängig („*pay as you go*“) abgerechnet; die imos AG stuft diese Kosten im Vergleich zu traditionellen Support-Aufwänden als marginal ein. Die Architektur erlaubt es, auf überflüssige *Retrievals* zu verzichten, etwa bei banalen oder bereits beantworteten Fragen, und dynamisch zwischen kontextbezogenen Operationen zu wählen.

Die Leistungsbewertung agentenbasierter RAG-Systeme ist bislang nur rudimentär erforscht. Wir führten erste Tests mit den Open-Source-Frameworks *Ragas* und *UpTrain* durch und verglichen deren Ergebnisse mit einer manuellen Begutachtung durch Fachexperten. Beide Frameworks nutzen ein *LLM-as-a-Judge*, das Antworten entlang verschiedener Metriken (u. a. Kontexttreue, Kohärenz, Relevanz) bewertet. In den Versuchen korrelierten die automatischen *Scores* jedoch nicht ausreichend stabil mit der menschlichen Einschätzung, sodass der *Human-in-the-Loop* weiterhin als kritischer *Evaluator* eingeplant ist.

Da das zugrundeliegende LLM kein domänenspezifisches Wissen besitzt, muss sämtliches Fachwissen über das *Retrieval* bereitgestellt werden. Die Qualität des Gesamtsystems steht somit in direktem Verhältnis zur Güte der Dokumentation. Optimierungen des *Retrieval*-Tools – etwa verbessertes *Chunking* und die Kombination aus semantischer

Suche und BM25 – steigern zwar die *Retrieval*-Qualität, können aber fehlende oder veraltete Inhalte nicht kompensieren. Daher stellt die Qualität und Vollständigkeit der Datengrundlage eine entscheidende Limitation dar. Insgesamt führte die Kombination aus LLMs und agentenbasiertem Entscheidungsverhalten zu einem robusteren Q&A-System. Hervorzuheben ist das positive Feedback des Service-Personals, auch wenn die Adaption noch Herausforderungen im Bereich des *Change Managements* mit sich bringt.

Doch trotz dieser Fortschritte bleibt eine Grenze bestehen: KIKUS bleibt ein reaktives System, das informiert, aber nicht handelt. Handlung ist jedoch in vielen Anwendungsfällen von Bedeutung. So beispielhaft bei der XML-basierten Artikelkonfiguration. Die imos AG entwickelt Softwarelösungen für die Möbelbranche, die den gesamten Prozess von der Planung über die Konstruktion bis zur Fertigung abdecken. Der Produktbereich Verkauf umfasst dabei *Point-of-Sale*-Systeme. In diesen Planungs- und Konfigurationstools wählen Endkunden oder Händler interaktiv Möbelartikel aus, konfigurieren Varianten und platzieren Produkte im Raum. Damit dieser Prozess funktioniert, müssen die Artikel inklusive Variantenlogik und Katalogstruktur in einem proprietären XML-Format vorliegen. Dabei sind zwei XML-Datentypen erforderlich: Die Artikel-XML beschreibt technische Eigenschaften, Konfigurationsregeln und Geometriedaten einzelner Produkte, während die Katalog-XML die übergeordnete Gliederung und Gruppierung im Sortiment abbildet. Nur im Zusammenspiel beider Dateien können Artikel im POS-System konfiguriert, angezeigt und in der späteren Konstruktions- und Fertigungslogik weiterverarbeitet werden. Die manuelle Erstellung dieser Dateien ist zeitaufwendig und fehleranfällig. Hier setzt das System KITS an, das die automatische Transformation gängiger Eingabedokumente (z. B. Produktkataloge als PDF) in die erforderlichen XML-Strukturen übernimmt.

Der Workflow gestaltet sich wie folgt: Nachdem Nutzende einen Katalog im PDF-Format in der *Chat*-Umgebung hochgeladen haben, initiiert der Agent automatisch die Analyse durch spezialisierte *Parser*. Die extrahierten Informationen werden zunächst in ein *Markdown*-Format überführt und anschließend durch ein Sprachmodell in strukturierte JSON-Objekte transformiert. Aus diesen JSON-Daten generiert das System die benötigten XML-Strukturen, die in das POS-System überführt werden. In einem optionalen Nachbearbeitungsschritt können Nutzende zusätzlich eine Farbauswahltabelle hochladen, wodurch die erzeugten XML-Dateien gezielt modifiziert und erweitert werden. Der gesamte Prozess ist dabei auf Effizienz ausgelegt: Die digitale Befüllung des Shops erfolgt in unter zwei Minuten.

Im Gegensatz zu KIKUS verfügt KITS bislang über kein vollwertiges *Retrieval*-Modul, kann jedoch bei Bedarf auf RAG-Komponenten zurückgreifen – etwa zur Ergänzung fehlender Kontextinformationen. Diese Fähigkeit zur Kontextanreicherung sowie die flexible Ansteuerung externer Tools erfordern klare Prinzipien: Sicherheitsmechanismen müssen unkontrollierte Aktionen verhindern, strukturierte *Prompt*-Designs definieren zulässige Funktionsaufrufe, und ein *Monitoring*-Framework protokolliert Prozesse nachvollziehbar.

Langfristig sollen weitere Workflows automatisiert werden – insbesondere solche mit klar definierten Ein- und Ausgabestrukturen. Entscheidend ist die Erkenntnis, dass agentenbasier-

te RAG-Systeme nicht nur klassische Q&A-Anwendungen weiterentwickeln, sondern eine neue Grundlage für proaktive Prozessautomatisierung in KMU bilden. KITS demonstriert exemplarisch die Potenziale, aber auch die Herausforderungen eines solchen Ansatzes.

4 Nutzen für KMU: Von Effizienz zu strategischem Mehrwert

Agentenbasierte RAG-Systeme eröffnen neue Wege, um bislang manuelle, fehleranfällige und personalintensive Abläufe zu automatisieren. Insbesondere KMU profitieren hiervon, da sie mit begrenzten Ressourcen bislang nur eingeschränkt Zugang zu skalierbarer KI-Technologie hatten. Damit einher geht jedoch die Notwendigkeit eines verantwortungsvollen Systemdesigns. Agenten, die tief in Geschäftsprozesse eingreifen, müssen klar strukturiert und kontrollierbar bleiben. Neben einem sorgfältigen *Prompt*-Design bedarf es transparenter Kontrollmechanismen, die nachvollziehbar machen, welche Informationen und *Tools* das System verwendet. Datenschutz spielt dabei eine zentrale Rolle, nicht zuletzt wegen Regulierungen und sensibler Kundendaten. Werden diese Anforderungen erfüllt, können agentenbasierte RAG-Lösungen einen Digitalisierungsschub für KMU bedeuten – ohne die Risiken klassischer Digitalisierungsprojekte, wie hohe Investitionskosten oder komplexe Integrationsprozesse. Möglich wird dies durch die Kombination vier wesentlicher Faktoren: die kontinuierliche Leistungssteigerung moderner LLMs, methodische Fortschritte in der Architektur und Interaktion von RAG-Komponenten, die Überführung dieser Komponenten in agentenbasierte, handlungsfähige Systeme und, zentral, aber häufig unterschätzt, die Qualität der Unternehmensdaten. Selbst Agentensysteme bleiben ineffektiv, wenn die zugrundeliegenden Daten veraltet, inkonsistent oder unstrukturiert sind. Die Effektivität ist unmittelbar an die Verlässlichkeit und semantische Tiefe der Wissensbasis gebunden. Für KMU bedeutet dies, dass Investitionen in Datenqualität nicht nur eine begleitende Maßnahme sind, sondern eine Voraussetzung für erfolgreiche Automatisierung.

KIKUS illustriert exemplarisch, wie sich diese Dynamik entfalten kann. Zunächst wurde eine komplexe Architektur notwendig, um zuverlässige Ergebnisse zu erzielen. Mit der Einführung leistungsstärkerer Modellgenerationen konnten viele dieser Komponenten vereinfacht oder ersetzt werden. Die gestiegene Robustheit der Modelle reduziert nicht nur die Komplexität, sondern erlaubt es KMU, von aktuellen Entwicklungen zu profitieren.

5 Fazit

Die Implementierung agentenbasierter RAG-Systeme zeigt das Potenzial dieser Technologie. Die Weiterentwicklung von Frage-Antwort-Systemen zu autonomen Agenten ermöglicht die Automatisierung komplexer Prozesse trotz begrenzter Ressourcen. Dabei stellen sich Herausforderungen wie *Parsing*-Fehler und Wissenslücken, die durch Validierung, bessere Modelle und systematische Datenpflege gelöst werden müssen. Die Projekte verdeutlichen den Wandel von reaktiven Assistenzsystemen zu eigenständigen Prozessakteuren – ein Paradigmenwechsel für KMU hin zu nachhaltiger, intelligenter Automatisierung.

Literaturverzeichnis

[Ba24] Barnett, S. et al.: Seven Failure Points When Engineering a Retrieval Augmented Generation System. In: Proc. of the IEEE/ACM 3rd Int. Conf. on AI Engineering - Software Engineering for AI. CAIN '24, Association for Computing Machinery, Lisbon, Portugal, S. 194–199, 2024, https://doi.org/10.1145/3644815.3644945.

[Bä24] Bäumer, F. S. et al.: Lektionen und Anwendungsfälle aus der Implementierung von Retrieval-Augmented-Generation-Systemen. In (Klein, M. et al., Hrsg.): INFORMATIK 2024 Lock-in or log out? Wie digitale Souveränität gelingt. Bd. 352, Köllen Druck+Verlag GmbH, 2024.

[LG21] Lundborg, M.; Gull, I.: Künstliche Intelligenz im Mittelstand–So wird KI für kleine und mittlere Unternehmen zum Game Changer. Begleitforschung Mittelstand-Digital/WIK-Consult, Abrufdatum: 04.07.2023, 2021.

[UF21] Ulrich, P.; Frank, V.: Relevance and Adoption of AI technologies in German SMEs – Results from Survey-Based Research. Procedia Computer Science 192 (1), Knowledge-Based and Intel. Inf. Engineering Systems: Proc. of the 25th Int. Conf. KES2021, S. 2152–2159, 2021.