DOI 10.5162/iCCC2026/P29

Effekte der Quantisierung auf Retrieval-Augmented-Generation Language Modelle für die Parkinson-Versorgung

Patrick Gaudl, Christoph-Alexander Holst und Volker Lohweg, Institut für industrielle Informationstechnik (inIT), Technische Hochschule Ostwestfalen-Lippe, Lemgo, Deutschland, vorname.nachname@th-owl.de

Kurzfassung

Large Language Models bieten großes Potenzial für assistive Anwendungen in der Parkinsonversorgung, sind jedoch aufgrund ihres hohen Speicherbedarfs bislang schwer in der Praxis einsetzbar. Dieser Beitrag untersucht den Einfluss verschiedener Quantisierungsmethoden auf ein Retrieval-Augmented-Generation-System auf Basis eines Qwen-Modells. Die Ergebnisse zeigen, dass sich der Speicherbedarf auf bis zu 36 % reduzieren lässt, bei leichter bis moderater Beeinträchtigung der Antwortqualität. Die Ergebnisse demonstrieren, dass lokal ausführbare Parkinson-Assistenzsysteme technisch realisierbar sind.

Abstract

Large Language Models hold significant potential for assistive applications in Parkinson's care, yet their substantial memory requirements currently limit deployment in real-world clinical environments. This paper investigates the impact of different quantization methods on a Retrieval-Augmented Generation system based on a Qwen model. Our results show that model size can be reduced to as little as 36% of the original footprint, with only mild to moderate degradation in response quality. These findings demonstrate that locally executable, Parkinson's-specific assistant systems are technically feasible and enable privacy-preserving assistance systems.

1 Motivation

Parkinson ist die weltweit am schnellsten zunehmende neurodegenerative Erkrankung und geht mit einer Vielzahl motorischer und nichtmotorischer Symptome einher [1]. Durch die Heterogenität der Symptomprogression sowie die oft fragmentierte Versorgung entstehen erhebliche Herausforderungen bei der Dokumentation, Bewertung und Therapieanpassung [2].

Digitale Assistenzsysteme bieten hier einen Ansatz zur strukturierten Aufbereitung individueller Krankheitsverläufe und können damit Fachkräfte im Gesundheitswesen sowie Betroffene unterstützen [3].Gleichzeitig stehen gegenwärtige Large Language Models (LLMs) aufgrund ihres enormen Speicher- und Rechenbedarfs einer breiten Anwendung in der klinischen Praxis entgegen. Die Ausführung solcher Modelle erfolgt typischerweise auf leistungsfähiger Cloud-Infrastruktur. Für den Einsatz in hochsensiblen, personenbezogenen Anwendungsszenarien wie der Parkinson-Versorgung ist eine lokale Verarbeitung auf Endgeräten im Hinblick auf Datenschutz, Verfügbarkeit und regulatorischer Anforderungen besonders attraktiv.

Damit LLM-basierte Assistenzsysteme im Routinebetrieb der Parkinsonversorgung ankommen können, müssen (1) der Speicherbedarf großer Modelle drastisch reduziert werden und dabei (2) gleichzeitig die Antwortqualität gewahrt werden. Quantisierungsmethoden ermöglichen eine erhebliche Verringerung des Modell-Memory-Footprints, ohne das Modell neu trainieren zu müssen [4]. Für domänenspezifische Assistenzsysteme ist zusätzlich eine Wissensintegration erforderlich, um Halluzinationen zu minimieren und medizinische Aussagequalität sicherzustellen. *Retrieval-Augmented-Generation* (RAG) hat sich hierfür als vielversprechende Methode etabliert [5].

Vor diesem Hintergrund untersucht dieser Beitrag den Einfluss unterschiedlicher Quantisierungsverfahren auf ein RAG-basiertes LLM im Kontext der Parkinsonversorgung. Dazu wird ein offenes Modell der Qwen-Familie in mehreren Quantisierungsstufen evaluiert und auf eigens kuratierten Frage-Antwort-Datensätzen aus Leitlinieninhalten getestet. Ziel ist die Beantwortung der Kernfrage, wie stark ein LLM quantisiert werden kann, bevor relevante Einbußen in der Ausgabequalität auftreten.

2 Stand der Forschung

2.1 LLMs in Medizin- und Gesundheitswesen

Generative künstliche Intelligenz und LLMs eröffnen zunehmend neue Anwendungsmöglichkeiten im Gesundheitswesen, insbesondere im klinischen Informationsmanagement [6]. LLMs können an verschiedenen Stellen der Versorgung eingesetzt werden, etwa zur Unterstützung klinischer Entscheidungsprozesse [7], zur Automatisierung medizinischer Dokumentation [8], zur Analyse von Fachliteratur oder als Bestandteil patientenorientierter Assistenzsysteme [3].

Gleichzeitig bestehen wesentliche Herausforderungen für den Einsatz im klinischen Kontext. Allgemeine LLMs verfügen über keine explizite domänenspezifische Wissensbasis und interpretieren medizinische Fachterminologie mitunter unvollständig, was ihre Leistungsfähigkeit in spezialisierten Aufgaben einschränkt [6]. Zudem sind LLMs anfällig für Halluzinationen, Jailbreak-Prompts und Datenvergiftungsangriffe. Forschungsarbeiten zu Unsicherheitsabschätzung [3], Modell-Governance und hybride Wissenszugriffstechniken adressieren diese Risiken, befinden sich jedoch noch in einem frühen Entwicklungsstadium

DOI 10.5162/iCCC2026/P29

[9]. Damit LLMs im medizinischen Alltag zuverlässig eingesetzt werden können, sind Mechanismen erforderlich, die domänenspezifisches Wissen integrieren, Fehlverhalten reduzieren und nachprüfbare Aussagen ermöglichen.

2.2 Retrieval-Augmented Generation und Fine-Tuning

RAG und Fine-Tuning gehören zu den zentralen Methoden, um allgemeine Sprachmodelle für medizinische Spezialaufgaben anzupassen. Die Ansätze ermöglichen es, LLMs mit einer Wissensbasis zu verknüpfen und so natürlichsprachliche Interaktion über medizinische Dokumente hinweg zu unterstützen [5], [10]. Der RAG-Prozess gliedert sich in: (1) Indexing, bei dem Textquellen in kleinere Einheiten zerlegt und mittels Embedding-Modellen als Vektoren in einem Wissensspeicher abgelegt werden, (2) Retrieval, bei dem Nutzereingaben als Embeddings kodiert und die semantisch ähnlichsten Wissenseinheiten identifiziert werden, sowie (3) Generation, in welcher das LLM die abgerufenen Inhalte in seine Antwort integriert [5].
RAG reduziert Halluzinationen, indem es die Ausgabe auf verifizierbare Quellen stützt. Fine-Tuning nutzt zusätzliche domänenspezifische Trainingsdaten, wodurch Modelle medizinische Terminologie präziser interpretieren können. Erste prototypische Systeme wie Alpacare [10] oder erste Parkinson-spezifische LLM-Implementationen [11] decken bislang nur eng definierte Fragestellungen ab. Offene Forschungsfragen bestehen hinsichtlich der Skalierbarkeit, der Robustheit gegen Angriffe und den Grenzen komprimierter Modellausführungen im klinischen Umfeld.

2.3 Quantisierung von LLMs

LLMs sind während ihres Trainings und der Inferenz oft mit hohen Anforderungen an Hardware bezüglich Speicher und Rechenleistung verbunden, da sie eine hohe Anzahl trainierbarer Parameter beinhalten [4]. Um dieses Problem zu lindern und die Hardwareanforderungen von LLMs zu reduzieren, dient die Modellquantisierung. Dabei handelt es sich um die Skalierung der jeweiligen Modellparameter von ihrem ursprünglichen Datentyp auf einen Datentyp mit reduziertem Speicherbedarf [4].
Die Quantisierung eines Tensors lässt sich am Beispiel der *absmax*-Quantisierung [12] darstellen: Seien $\boldsymbol{X}_{FP16}$ der als *16-Bit Floating Point (FP16)* vorliegende unquantisierte Tensor und $s_{x,FP16}$ die für ein Skalierungsfaktor, so lässt sich die Repräsentation ebendieses Tensors $\boldsymbol{X}_{Int8}$ mit *8-Bit-Integer (Int8),* wie folgt, berechnen [12]:

$$\boldsymbol{X}_{\text{Int8}} = round\left(\frac{127}{max_{ij}(|\boldsymbol{X}_{\text{FP16}}|)} \cdot \boldsymbol{X}_{\text{FP16}}\right) = round\left(s_{x,\text{FP16}} \cdot \boldsymbol{X}_{\text{FP16}}\right).$$

Dadurch ergibt sich eine Integer-Repräsentation des Tensors, dessen Elemente sich im von Int8 abgebildeten Wertebereich von $[-127{,}127]$ befinden. Der dequantisierte Tensor $\boldsymbol{X}_Q$ wird anschließend wie folgt berechnet [12]: $\boldsymbol{X}_Q = S_{\text{FP16}} \cdot \boldsymbol{X}_{\text{Int8}} = \frac{1}{s_{x,FP16}} \cdot \boldsymbol{X}_{\text{Int8}}$. Bei der Dequantisierung können durch Rundung Quantisierungsfehler E_Q auftreten [13]: $E_Q = \boldsymbol{X}_{\text{FP16}} - \boldsymbol{X}_Q$. Quantisierungsfehler können sich auf folgende Berechnungen auswirken und somit die Leistung des Modells beeinflussen [13], weshalb diese minimiert werden sollten.

3 Methodik

3.1 Systemarchitektur

3.1.1 RAG-Pipeline

Die Implementierung (Bild 1) umfasst eine RAG-Architektur, welche mithilfe des Frameworks *Langchain* realisiert wird. Für die Implementierung der Wissensbasis als Vektordatenbank wird *Facebook AI Similarity Search (FAISS)* genutzt. Für das Erstellen der Embeddings wurde der Sentence-Transformer *all-MiniLM-L6-v2* verwendet.

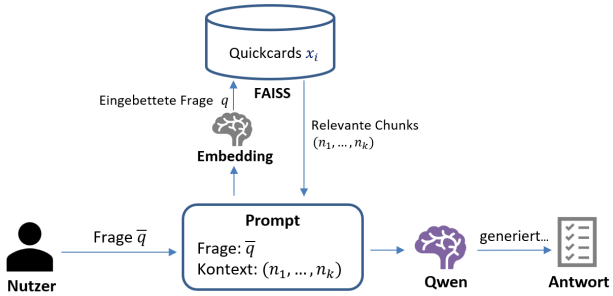

Bild 1 Aufbau des RAG-Systems: Die Frage $\overline{q}$ wird dem Prompt zugeführt und per Embedding in ihre Vektorrepräsentation umgewandelt, welche als Anfrage an die Vektordatenbank dient. Die k abgerufenen Dokumente werden dem Prompt als Kontext zugeführt.

Als Sprachmodell dient das von der Alibaba-Group entwickelte *Qwen2.5-7B-Instruct*. Das Modell umfasst 7,61 Milliarden Parameter und ist auf dem Open-LLM-Leaderboard (huggingface.co; Stand Februar 2025) als eines der beliebtesten Modelle vertreten.
Bei der Implementierung wird eine Temperatur von $t = 10^{-2}$ verwendet, um die Zufälligkeit der generierten Antworten gering zu halten. Zudem wurde dem Modell durch einen System-Prompt die Anweisung gegeben, möglichst kurz anhand des bereitgestellten Kontexts zu antworten, um den Spielraum für Halluzinationen zu minimieren.

3.1.2 Leitlinien als Wissensbasis

Bei den für die Vektordatenbank verwendet Inhalten handelt es sich um sogenannte *Quickcards*, die vom *Parkinsonnetz Münsterland+* herausgegeben wurden, auf aktuellen Parkinson-Leitlinien basieren und Bereiche, wie die Logopädie, Physiotherapie, Ergotherapie und die Neurologie abbilden. Konkret handelt es sich dabei um Symptome und die dazugehörigen Empfehlungen. Die initial als PDF vorliegenden Quickcards wurden per Hand in Textform überschrieben und im *MarkDown*-Format eingespeichert.
Bei der Einspeisung in die Wissensbasis wird auf eine fixe Chunk-Größe verzichtet, da sich die Länge der Einträge zu verschiedenen Symptomen deutlich voneinander unterscheiden und der Abbruch von Sätzen sowie das Einbringen kontextfremder Informationen verhindert werden soll.

3.1.3 Quantisierungsstufen

Das Modell wurde in drei verschiedenen Quantisierungsstufen untersucht. Die Modellparameter liegen initial im Datenformat *BrainFloat16 (BF16)* [14] vor, wobei jeder Modellparameter 16 Bits Speicherplatz benötigt. Als Quantisierungen werden die von Dettmers et al. vorgestellten Algorithmen *LLM.Int8* [12] und *4-bit block-wise quantization* [15] verwendet..

Quantisierungsstufe	Datentyp	Speicherbedarf*
Keine	BF16	15.23 GB
LLM.Int8	Int8	8.7 GB
4-bit block-wise	NF4	5.44 GB

Tabelle 1 Übersicht über die verwendeten Quantisierungsmethoden und ihre Auswirkungen auf den Speicherbedarf (VRAM) des Modells.

3.2 Evaluation

Die Auswirkungen der jeweiligen Quantisierungsstufen wurden anhand von drei Evaluationsdatensätze getestet und anhand verschiedener Metriken evaluiert. Die bei der Evaluation verwendeten *Quickcard-QA (Question-Answering)*-Datensätze beziehen sich inhaltlich auf die als Wissensbasis verwendeten Quickcards und orientieren sich in ihrem Aufbau an gängigen Benchmarks wie z.B. *PubMedQA* [16]. So handelt es sich bei *MC (Multiple-Choice)-QQA-1* und *-2)* um Multiple-Choice-Fragen mit direkten symptombezogenen Fragen, bei denen das Modell zwischen Antwortmöglichkeiten auswählen muss, und bei *Gen (Generative)-QQA* um offene Fragen, bei denen das Modell Antwortsequenzen generieren muss.

Datensatz	Fragentyp	Metriken
MC-QQA-1	Single/Multiple-Choice QA, direkte symptombezogene Fragen	Accuracy, Precision, Recall, F1
MC-QQA-2	Single/Multiple-Choice QA mit fiktiven Fallbeispielen	Accuracy, Precision, Recall, F1
Gen-QQA	Generative QA mit fiktiven Fallbeispielen	Accuracy durch menschliche Evaluation, BERTScore

Tabelle 2 Übersicht über die drei verwendeten Evaluationsdatensätze, deren verwendeten Fragentypen und deren angewendete Evaluationsmetriken.

Die Antworten der Modelle auf die Fragen aus MC-QQA-1 und -2 wurden anhand der Klassifikationsmetriken *Accuracy*, *Precision*, *Recall* und *F1-Score* evaluiert. Für die Evaluation der Frei-Text-Modellantworten für Gen-QQA wurde sowohl eine menschlichen Beurteilung als auch der *BERTScore [17]* genutzt. BERTScore vergleicht die Token des zu prüfenden Satzes mit denen des Referenzsatzes und berechnet darauf basierend die Metriken Precision (P_{BERT}), Recall (R_{BERT}) und einen F1-Score (F_{BERT}) [17].

Für die Evaluation wurde eine *Nvidia Titan RTX* mit 24 GB Grafikspeicher und einer Leistung von 130 Tensor TFLOPS und ein *AMD Ryzen 7 3800X 8-Core*-Prozessor verwendet. Für jede neue Frage wurde ein neuer Kontext eröffnet, um keine Informationen aus den vorherigen Fragen einfließen zu lassen.

4 Ergebnisse

Tabelle 3 zeigt auf, dass Int8 über alle Datensätze hinweg die höchsten Accuracy-Scores und den höchsten F_{BERT} liefert. Die Quantisierung mit NF4 zeigt eine leichte Performance-Degradation, während die Leistung von BF16 nahe an der von Int8 liegt.

Datensatz	Metrik	BF16	Int8	NF4
	Speicherbedarf	15.23 GB	8.7 GB	5.4 GB
MC-QQA-1	Accuracy	0.757	**0.779**	0.764
	Precision	0.657	**0.691**	0.662
	Recall	0.891	**0.905**	0.843
	F1	0.733	**0.76**	0.724
MC-QQA-2	Accuracy	0.71	**0.75**	0.68
	Precision	0.55	**0.6**	0.547
	Recall	**0.78**	**0.78**	0.76
	F1	0.621	**0.659**	0.608
Gen-QQA	Accuracy*	**0.6**	0.52	0.4
	P_{BERT}	0.673	**0.678**	0.632
	R_{BERT}	0.759	**0.762**	0.717
	F_{BERT}	0.712	**0.717**	0.671

Tabelle 3 Übersicht der über den jeweiligen Datensatz gemittelten Evaluationsergebnisse des Qwen2.5-7B-Instruct in den verwendeten Quantisierungsstufen. *Bewertung durch menschlichen Experten

Die Recall-Scores im Falle von MC-QQA-1 und -2 liegen zudem deutlich über den Precision-Scores. Dies ist ein Indiz dafür, dass die Modelle viele tatsächlich falsche Antworten als richtig interpretieren, was im Rahmen einer medizinischen Assistenzanwendung problematisch wäre.
Ferner fällt auf, dass sich bei Gen-QQA die BERTScore-Metriken nur leicht voneinander unterscheiden, während die Evaluation durch Menschen eine deutliche Leistungsdegradation mit steigender Quantisierungsstufe feststellt.
Allgemein lassen sich zwischen den einzelnen Quantisierungsstufen nur geringe Abweichungen beobachten, wobei diese bei NF4 etwas stärker ausgeprägt sind. Das beste Verhältnis zwischen Speicherplatz und Leistung liefert jedoch das Modell mit Int8-Quantisierung.

5 Diskussion

5.1 Limitationen

Die Bewertung der generierten Antworten erfolgte notwendigerweise über qualitative Beurteilungen, bei denen individuelle Einschätzungen eine Rolle spielen. Solche Einschätzungen sind in domänenspezifischen Benchmarks üblich, beinhalten jedoch unvermeidbare subjektive Komponenten, die in zukünftigen Arbeiten durch Mehrgutachter oder ergänzende automatische Metriken weiter objektiviert werden können.
Als Wissensbasis dienten Quickcards, die eine kontrollierte, aber vereinfachte Repräsentation klinischer Informationen darstellen. Komplexe Versorgungssituationen,

wie Mehrfacherkrankungen, Kontraindikationen oder vollständige Patientenhistorien, wurden nicht abgebildet.
Schließlich bestehen systembedingte Limitierungen in der technischen Pipeline. Das verwendete Embedding-Modell besitzt eine begrenzte Kontextlänge und ist nicht auf deutsche medizinische Terminologie spezialisiert, was das Retrieval in Einzelfällen beeinträchtigen kann. Auch das generative Modell zeigt gelegentliche Unsicherheiten im Umgang mit fachspezifischen Begriffen, was weiteres Fine-Tuning nahelegt.

5.2 Fazit

Die Arbeit zeigt, dass sich die Quantisierung als bewährtes Mittel zur ressourcenschonenden Implementierung von RAG-Systemen bewährt, da die Leistungseinbußen mit zunehmender Stärke der Quantisierung moderat ausfallen. Selbst die stärkste Quantisierungsstufe, die den Speicherbedarf des Modells um fast Dreiviertel senkt, geht mit nur einem moderaten Leistungsabfall einher. Ebenfalls unterstreichen die Ergebnisse, dass sich RAG als geeignete Methode bewährt, um LLMs bei der Generation kuratiertes Fachwissen zur Verfügung zu stellen. Durch die Kombination von geeigneten Quantisierungsmethoden und einem RAG-System mit einer kuratierten Wissensbasis, lassen sich fachlich hochspezialisierte LLMs auch auf handelsüblichen Grafikkarten ausführen.

5.3 Ausblick

Zur Verbesserung des Retrievals bieten sich weiterentwickelte RAG-Verfahren wie *Query-Rewriting*, modulare RAG-Architekturen und LLM-basierte Bewertungsinstanzen an. Ein domänenspezifischer Ausbau des Systems stellt weiteres Potenzial dar. Fine-Tuning mit medizinischen Datensätzen – beispielsweise mittels QLoRA – sowie der Einsatz spezialisierter deutscher Embedding-Modelle können die Robustheit der Generierung und Terminologieverarbeitung erhöhen. Darüber hinaus eröffnen Multi-Agenten-Ansätze und Reasoning-Strategien wie Chain-of-Thought neue Möglichkeiten, komplexere klinische Fragestellungen abzubilden und die Qualität datenbasierter Entscheidungsunterstützung weiter zu steigern.

6 Literatur

[1] D. Su *et al.,* "Projections for prevalence of Parkinson" disease and its driving factors in 195 countries and territories to 2050: modelling study of Global Burden of Disease Study 2021," *BMJ*, Jg. 388, 2025.

[2] C. Eggers *et al.,* "Versorgung von Parkinson-Patienten in Deutschland: Status quo und Perspektiven im Spiegel des digitalen Wandels," *Der Nervenarzt*, Jg. 92, Nr. 6, S. 602–610, 2021.

[3] C.-A. Holst, F. Wiegräbe, C. Redecker und V. Lohweg, "Die Parkinson-Erkrankung im Zeitalter der KI: Eine Smartphone-App als Schlüssel zur nutzerzentrierten Patientenversorgung," in *Künstliche Intelligenz im Einsatz für die erfolgreiche Patientenreise: Innovation, Integration und Stärkung*, M. A. Pfannstiel, Hg., Wiesbaden: Springer Gabler, 2025, S. 191–218.

[4] R. Gong *et al.,* "A survey of low-bit large language models: Basics, systems, and algorithms," *Neural Networks*, Jg. 192, S. 107856, 2025.

[5] Y. Gao *et al.,* "Retrieval-Augmented Generation for Large Language Models: A Survey," *arXiv:*2312.10997, 2024.

[6] A. J. Thirunavukarasu *et al.,* "Large language models in medicine," *Nature Medicine*, Jg. 29, Nr. 8, S. 1930–1940, 2023.

[7] Y. Kim *et al.,* "MDAgents: An Adaptive Collaboration of LLMs for Medical Decision-Making," in *Advances in Neural Information Processing Systems*, A. Globerson et al., Hg., Bd. 37, 2024, S. 79410–79452.

[8] M. Rehman *et al.,* "Advancement in medical report generation: current practices, challenges, and future directions," *Medical & Biological Engineering & Computing*, Jg. 63, Nr. 5, S. 1249–1270, 2025.

[9] Y. Huang *et al.,* "TrustLLM: Trustworthiness in Large Language Models," *arXiv:*2401.05561, 2024.

[10] X. Zhang, C. Tian, X. Yang, L. Chen, Z. Li und L. R. Petzold, "AlpaCare: Instruction-tuned Large Language Models for Medical Application," *arXiv:*2310.14558, 2025.

[11] L. Cardenas, K. Parajes, M. Zhu und S. Zhai, "AutoHealth: Advanced LLM-Empowered Wearable Personalized Medical Butler for Parkinson's Disease Management," in *2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC)*, 2024, S. 375–379.

[12] T. Dettmers, M. Lewis, Y. Belkada und L. Zettlemoyer, "GPT3.int8: 8-bit Matrix Multiplication for Transformers at Scale," in *Advances in Neural Information Processing Systems*, S. Koyejo et al., Hg., Bd. 35, 2022, S. 30318–30332.

[13] C. Zhang, J. Cheng, G. A. Constantinides und Y. Zhao, "LQER: low-rank quantization error reconstruction for LLMs," in *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[14] G. Henry, P. T. P. Tang und A. Heinecke, "Leveraging the bfloat16 Artificial Intelligence Datatype For Higher-Precision Computations," in *2019 IEEE 26th Symposium on Computer Arithmetic (ARITH)*, 2019, S. 69–76.

[15] T. Dettmers, A. Pagnoni, A. Holtzman und L. Zettlemoyer, "QLoRA: Efficient Finetuning of Quantized LLMs," in *Advances in Neural Information Processing Systems*, A. Oh et al., Hg., Bd. 36, 2023, S. 10088–10115.

[16] Q. Jin *et al.,* "PubMedQA: A Dataset for Biomedical Research Question Answering," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui et al., Hg., 2019, S. 2567–2577.

[17] T. Zhang *et al.,* "BERTScore: Evaluating Text Generation with BERT," in *International Conference on Learning Representations*, 2020.