oakai:alpha10

Benchmark & Modellprofil

Domänenspezifisches Sprachmodell
für Retail- und Kundenanalytik

„Klarheit – statt Hype.“

OAKAI · Ralph Ray Schwehr
Juli 2026

Proof of Concept

oakai:alpha10

Benchmark & Modellprofil

Domänenspezifisches Sprachmodell für Retail- und Kundenanalytik · OAKAI / Ralph Ray Schwehr · Juli 2026

oakai:alpha10 ist ein auf Basis von Qwen2.5-7B (Apache 2.0) feinabgestimmtes, vollständig lokal lauffähiges Sprachmodell. Es ist auf vier wiederkehrende Analyseaufgaben spezialisiert: B2B/B2C-Segmentierung, Sentiment- und Defekterkennung, semantisches Anliegen-Matching und Themen-Clustering. Ziel war nicht breiteres Weltwissen, sondern **präzise, maschinenlesbare Antworten in festem Format bei minimaler Latenz** – gebaut für automatisierte Pipelines. Klarheit statt Hype, im Modell selbst.

Warum ein eigenes, spezialisiertes Modell?

Große generische Modelle können nahezu jede Aufgabe – aber langsam, gesprächig und meist in der Cloud. Für wiederkehrende, klar umrissene Klassifikationsaufgaben ist das teuer und unpräzise. Ein kompaktes, auf die konkrete Domäne trainiertes Modell dreht dieses Verhältnis um: **gleiche Trefferquote, ein Bruchteil der Zeit, direkt maschinenlesbare Antworten, vollständig im eigenen Haus.** Das senkt Latenz und Kosten, schützt sensible Daten und macht die Auswertung großer Textmengen überhaupt erst wirtschaftlich.

Genauigkeit & Geschwindigkeit

40 ungesehene Testfälle (10 je Aufgabe), deterministisch (Temperatur 0). Genauigkeit = Anteil korrekt klassifizierter Fälle. Geschwindigkeit gemessen auf Mac Studio M4 Max.

Modell	B2B	Sentim.	Cluster	Match	Gesamt	Zeit	Tok.
oakai:alpha10	100%	90%	80%	90%	**90%**	**0,34s**	**4**
qwen2.5:7b *(Basis)*	100%	100%	70%	70%	85%	2,87s	220
gemma4:latest *(12B)*	100%	90%	100%	80%	92%	8,41s	634

Kernergebnis: oakai:alpha10 erreicht mit 90% praktisch die Genauigkeit des rund 3× größeren gemma4 (92%) – bei **25× kürzerer Antwortzeit** (0,34s vs. 8,41s) und einem um Faktor 150 knapperen, direkt weiterverarbeitbaren Output (4 vs. 634 Tokens).

Optimales Einsatzszenario

Das Modell spielt seine Stärken aus, wenn **große Mengen kurzer Texte automatisiert und strukturiert klassifiziert** werden – Bestell-Segmentierung, Review-Auswertung, Support-Ticket-Routing. Der feste, knappe Output (ein Label statt Fließtext) fließt ohne Nachbearbeitung direkt in Datenbanken oder Dashboards. Weil es vollständig lokal läuft, eignet es sich besonders für **datenschutzsensible Kundendaten**, die das Haus nicht verlassen dürfen.

Stärken und Grenzen

Vorteile	Grenzen / wo nicht ideal
• **Tempo:** ~0,3s je Klassifikation, 8–25× schneller als Vergleichsmodelle	• **Keine Begründungen:** liefert Labels, keine erklärenden Fließtexte
• **Maschinenlesbar:** knappes Label-Format, keine Ausgabe-Nachbearbeitung	• **Auf 4 Aufgaben spezialisiert:** außerhalb kein Vorteil
• **Datenschutz:** läuft 100% lokal, keine Daten an externe APIs	• **Seltene Sprach-Artefakte:** vereinzeltes Durchschlagen der Basis möglich
• **Kostenfrei im Betrieb:** keine Token- oder API-Gebühren	• **PoC-Datenbasis:** synthetische Trainingsdaten (Muster real, Inhalte erfunden)
• **Kompakt:** 4,7 GB (4-bit), läuft auf Apple-Silicon-Hardware	• **Kein produktives Tool-Format:** nicht auf exakte projektspezifische JSON-Formate abgestimmt
• **Lizenzsauber:** Apache-2.0-Basis, frei benenn- und weitergebbar	• **Feste Kategorien:** neue Cluster/Segmente erfordern erneutes Training

Bezug & Nutzung

Das Modell ist quelloffen unter Apache 2.0 veröffentlicht und über mehrere Wege sofort einsatzbereit – alles lokal, ohne Cloud-Anbindung:

Plattform	Bezug	Nutzung
Hugging Face	`huggingface.co/RaSchwehr/oakai-alpha10`	Modellkarte, Benchmark und GGUF-Download (4,7 GB)
Ollama	`ollama pull RaSchwehr/oakai-alpha10`	Direkt aus der Ollama-Registry – ein Befehl, sofort nutzbar
Ollama (via HF)	`ollama run hf.co/RaSchwehr/oakai-alpha10:Q4_K_M`	Alternativ direkt aus Hugging Face laden und starten
llama.cpp	`llama-cli -hf RaSchwehr/oakai-alpha10:Q4_K_M`	Direkt im Terminal oder als OpenAI-kompatibler Server

Technischer Ablauf (Kurzüberblick)

Von der Datenerzeugung bis zum lauffähigen Modell in einer reproduzierbaren Pipeline auf einem einzelnen Mac Studio M4 Max:

• **1. Datensatz:** 5.000 synthetische Beispiele, Muster realen Aufgaben nachempfunden, Inhalte frei erfunden (kein Kundendatenbezug).

• **2. Training:** QLoRA (4-bit) mit MLX, 1200 Iterationen – nur 0,15% der Modellgewichte trainiert, Validierungs-Loss 4,12 → 0,30.

• **3. Fusion & Export:** LoRA-Adapter fest ins Modell integriert, Konvertierung nach GGUF, Quantisierung auf Q4_K_M (~4,7 GB).

• **4. Bereitstellung:** lokal via Ollama nutzbar und öffentlich auf Hugging Face veröffentlicht – mit vollständiger Lizenz- und Herkunftskette.

Methodik & Transparenz

Basismodell: Qwen2.5-7B-Instruct (Apache 2.0). **Training:** QLoRA (4-bit), 1200 Iterationen, MLX auf Mac Studio M4 Max; Validierungs-Loss von 4,12 auf 0,30 gesunken. **Trainingsdaten:** 5.000 synthetische Beispiele einer fiktiven Marke (Retail-Kontext); Muster realen Analyse-Aufgaben nachempfunden, alle Namen, Firmen und Produkte frei erfunden. **Auswertung:** 40 ungesehene Fälle, tolerante Prüfung auf die korrekte Klasse. **Hinweis:** Ein weiteres Modell (gemma3:12b) wurde wegen eines Konfigurationsfehlers aus der Wertung genommen, um die Vergleichbarkeit zu wahren.
