STRATEGIEN ZUR TOKEN-MINIMIERUNG IN LLM-ANWENDUNGEN

Experimentelle Analyse von Prompt-Kompressionsverfahren

W1 Integrationsseminar – Onlinemedien, 5. Semester

Paul Klemm

DHBW Mosbach | Wintersemester 2025/26

AGENDA

1. Motivation & Problemstellung
2. Forschungsfrage & Ziele
3. Theoretischer Hintergrund
4. Methodik & Messgrößen
5. Kompressionsstrategien
6. Ergebnisse
7. Diskussion & Fazit
8. Quellenverzeichnis

1. MOTIVATION & PROBLEMSTELLUNG

391 €
Kosten pro 1 Mio. Anfragen
(GPT-3.5 Turbo, nur Input)

20×
teurer bei GPT-4
→ bis zu 8.000 €/Monat

75 %
mögliche Kosteneinsparung
durch Token-Minimierung

Kernfrage: Wie können Tokens reduziert werden – ohne die Antwortqualität zu verlieren?

- Mehr Tokens = höhere Kosten + höhere Latenz
- Prompts enthalten oft **Redundanzen** (Floskeln, irrelevanter Kontext)
- Systematischer Vergleich von Minimierungsstrategien fehlt

Vgl. Brown et al. (2020); Jiang et al. (2023); OpenAI Pricing (2024)

2. FORSCHUNGSFRAGE & ZIELE

Forschungsfrage: Inwiefern können verschiedene Token-Minimierungsstrategien die Betriebskosten und Latenz von LLM-Anwendungen reduzieren, ohne die semantische Qualität signifikant zu beeinträchtigen?

ZIELE

- Vergleich von 4 Strategien + Baseline
- Quantifizierung: Kosten, Latenz, Qualität
- Handlungsempfehlungen für Praxis

ABGRENZUNG

- Fokus: **Input-Token-Reduktion**
- Kein Fine-Tuning / Modellwechsel
- Deutschsprachige Testprompts

3. THEORETISCHER HINTERGRUND

TOKENISIERUNG (BPE)

- Text → **Subwort-Einheiten** (Byte-Pair Encoding)
- 1 Token ≈ 0,75 Wörter (Deutsch)
- Input- **und** Output-Tokens werden berechnet

$$\textbf{Kosten} = \text{Tokens} / 1000 \times \text{Preis/1K}$$

KOSTENMODELL (OPENAI, 2024)

Modell	Input/1K	Output/1K
GPT-3.5 Turbo	0,0015 €	0,002 €
GPT-4	0,03 €	0,06 €

Für diese Studie: **0,00138 € / 1K Input-Tokens**

Sennrich et al. (2016); OpenAI (2024); tiktoken-Dokumentation

3.1 STAND DER FORSCHUNG

PROMPT COMPRESSION

- **LLMLingua** (Jiang et al., 2023) – Token-Wichtigkeit per kleinem Sprachmodell
- **Selective Context** (Li et al., 2023) – Self-Information-basiert
- **LongLLMLingua** (Jiang et al., 2024) – Für lange Kontexte

KONTEXTNUTZUNG

- **Lost in the Middle** (Liu et al., 2023) – Anfang & Ende werden besser genutzt

SEMANTISCHE ÄHNLICHKEIT

- **Sentence-BERT** (Reimers & Gurevych, 2019) – Satz-Embeddings
- **BERTScore** (Zhang et al., 2020) – Evaluation von Textgenerierung

> **Lücke:** Vergleichende Evaluation unter einheitlichem Messrahmen fehlt.

Jiang et al. (2023/2024); Li et al. (2023); Liu et al. (2023); Reimers & Gurevych (2019); Zhang et al. (2020)

4. METHODIK & MESSGRÖSSEN

EXPERIMENTDESIGN

Parameter	Wert
Tokenizer	tiktoken (GPT-3.5 BPE)
Testszenarien	3 (deutschsprachig)
Strategien	4 + Baseline
Einzelmessungen	15 (3 × 5)

MESSGRÖSSEN

Metrik	Methode
Tokens	tiktoken (exakt)
Kosten	0,00138 €/1K Tokens
Latenz	Kompressionszeit + 50ms + 2ms/Token
Qualität	Cosine-Similarity (Embeddings)

TESTSZENARIEN

KUNDENSUPPORT – 264 TOKENS

E-Mail mit Bestelldaten, Grußformeln

DOKUMENTENZUSAMMENFASSUNG – 353 TOKENS

Marktbericht mit Zahlen & Prognosen

CODE-REVIEW – 234 TOKENS

Python-Code mit Analyseanfrage

Ø 284 Tokens/Prompt

4.1 QUALITÄTSMESSUNG

METHODE: SEMANTISCHE ÄHNLICHKEIT

- Modell: **paraphrase-multilingual-MiniLM-L12-v2**
- Original-Text → Embedding (384-dim)
- Komprimierter Text → Embedding
- Qualität = Cosine-Similarity der Vektoren

$$\text{Qualität} = \cos(v_{\text{orig}}, v_{\text{komp}}) \times 100\%$$

≥ 90%: Hoch | 70-89%: Mittel | < 70%: Niedrig

Reimers & Gurevych (2019); Zhang et al. (2020)

VORTEILE

- Etabliert in NLP (vgl. BERTScore)
- Multilingual (Deutsch + Englisch)
- Automatisiert & reproduzierbar
- Keine manuelle Bewertung nötig

Limitation: Misst semantische Nähe, nicht tatsächliche LLM-Antwortqualität (Proxy-Metrik)

5. KOMPRESSIONSSTRATEGIEN

1. REGELBASIERTE KOMPRESSION

Regex-basiertes Entfernen von Anrede/Grußformeln, Reduktion von Leerzeichen, optionale Stoppwort-Entfernung.
→ *Deterministisch, kein ML nötig, gut erklärbar*

2. STRUKTURIERTE KOMPRESSION

Satz-Embeddings → Centrality-Bewertung (Ähnlichkeit zum Dokument-Mittelwert) → nur zentrale Sätze behalten (~45%).
→ *LLMLingua-inspiriert, semantisch gesteuert*

3. TOKEN-BUDGET (100 TOKENS)

Festes Token-Limit. Sätze nach Position gewichtet (U-Kurve: Anfang & Ende höher). Greedy-Auswahl bis Budget voll.
→ *Aggressivste Kompression, Lost in the Middle*

4. CHUNKING MIT ÜBERLAPPUNG

Text in Chunks mit Überlappung teilen. Pro Chunk: erste + letzte Sätze als Repräsentation. Alle Chunk-Repräsentationen konkatenieren.
→ *LongLLMLingua-inspiriert, für lange Dokumente*

Jiang et al. (2023/2024); Li et al. (2023); Liu et al. (2023)

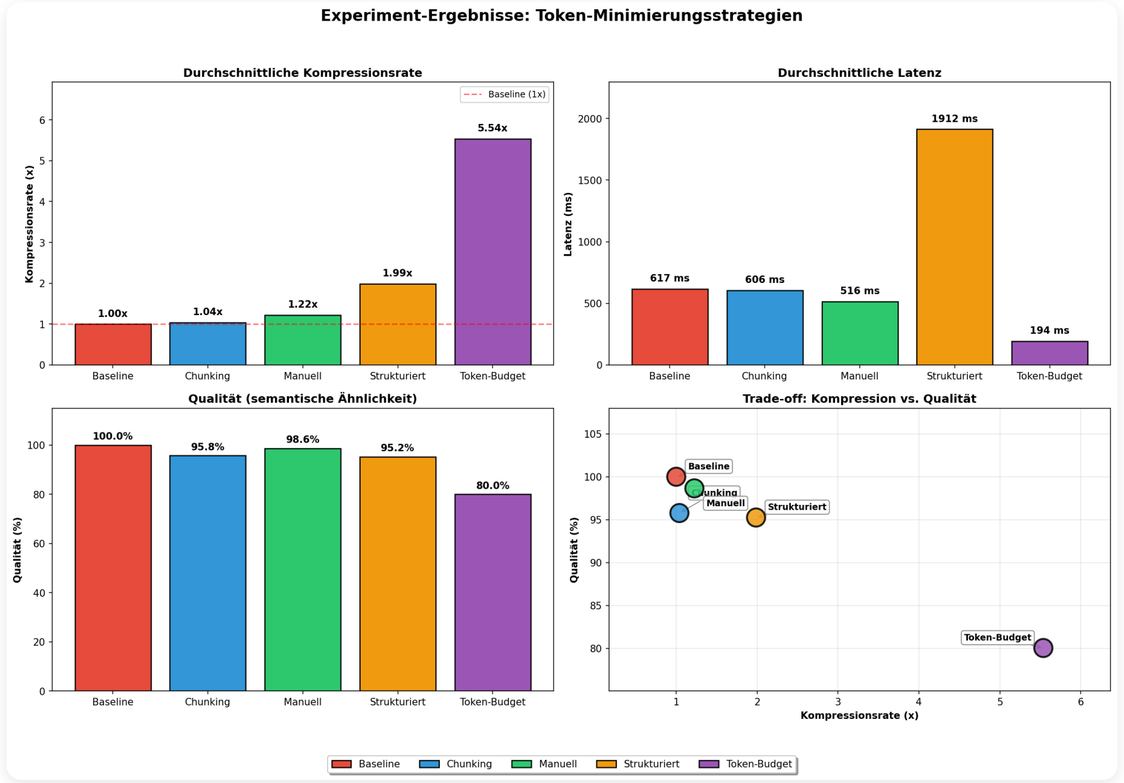
6. ERGEBNISSE: GESAMTÜBERSICHT

Strategie	Kompression	Tokens (Ø)	Latenz	Kosten/1K	Einsparung	Qualität
Baseline	1,0×	284	617 ms	0,391 €	–	**100 %**
Chunking	1,0×	278	607 ms	0,384 €	2 %	**92 %**
Regelbasiert	1,2×	233	516 ms	0,321 €	**18 %**	**97 %**
Strukturiert	2,0×	161	**2.247 ms**	0,222 €	**43 %**	**91 %**
Token-Budget	5,5×	72	**194 ms**	0,099 €	**75 %**	**80 %**

Durchschnitt über 3 Szenarien (n = 15 Messungen). Kosten: nur Input-Tokens, GPT-3.5 Turbo.

Kritische Beobachtung: Strukturierte Kompression ist durch Embedding-Berechnung **3,6× langsamer** als Baseline!

6.1 ERGEBNISSE: DETAILVERGLEICH



Eigene Darstellung. 15 Einzelmessungen (3 Szenarien × 5 Strategien). Qualität = semantische Ähnlichkeit.

6.2 KOSTENANALYSE

PRO 1.000 ANFRAGEN

Strategie	Kosten	Einsparung
Baseline	0,391 €	–
Regelbasiert	0,321 €	**18 %**
Strukturiert	0,222 €	**43 %**
Token-Budget	0,099 €	**75 %**

HOCHRECHNUNG: 1 MIO. ANFRAGEN/MONAT

169 €

Ersparnis
Strukturiert

293 €

Ersparnis
Token-Budget

Bei GPT-4: Einsparungen × 20 (bis zu 5.860 €/Monat)

Praxisbeispiel: Ein Kundensupport-Chatbot mit 10.000 Anfragen/Tag spart mit Token-Budget **~3.000 €/Jahr** (nur Input-Tokens).

7. DISKUSSION

KERNERKENNTNISSE

- **Trade-off bestätigt:** Mehr Kompression → weniger Qualität
- **Texttyp entscheidend:** Code ≠ E-Mail ≠ Bericht
- **Embedding-Overhead:** Strukturierte Kompression 3,6× langsamer
- **Token-Budget:** Höchste Einsparung (75%), aber Qualitätseinbußen (80%)

EMPFEHLUNGEN

QUALITÄTSKRITISCH (SUPPORT, MEDIZIN, RECHT)

→ **Regelbasiert**: 97% Qualität · 18% Ersparnis · 516ms Latenz

BALANCIERT (ANALYTICS, MASSENVERARBEITUNG)

→ **Strukturiert**: 91% Qualität · 43% Ersparnis · aber 2.247ms Latenz

KOSTENOPTIMIERT (PRE-FILTERING, INTERNE TOOLS)

→ **Token-Budget**: 80% Qualität · 75% Ersparnis · 194ms Latenz

7.1 LIMITATIONEN

- **n = 3 Szenarien** – begrenzte Generalisierbarkeit
- **Proxy-Metrik** – semantische Ähnlichkeit ≠ LLM-Antwortqualität
- **Keine echten API-Calls** – Latenz modelliert (50ms + 2ms/Token)
- **Nur Input-Tokens** – Output-Kosten nicht gemessen
- **Nur Deutsch** – Sprachübertragbarkeit offen
- **Ein Embedding-Modell** – andere Modelle könnten abweichen

AUSBLICK

- Dynamische Kompressionsraten je nach Query-Komplexität
- Kombination mit Output-Token-Optimierung
- Evaluation mit verschiedenen LLM-Modellen (GPT-4, Claude, Llama)

7.2 FAZIT

BEANTWORTUNG DER FORSCHUNGSFRAGE

- **Kompressionsraten:** 1,2× (regelbasiert) bis 5,5× (Token-Budget) erreichbar
- **Kosteneinsparungen:** 18% bis 75% realistisch (Input-Tokens)
- **Qualität:** Ab ~2× Kompression sinkt Qualität merklich; bei 5,5× nur noch 80%
- **Latenz-Trade-off:** Embedding-basierte Methoden haben signifikanten Overhead

Kernbotschaft: Regelbasierte Kompression sollte **immer als Grundlage** eingesetzt werden – sie liefert 18 % Einsparung bei 97 % Qualität ohne Latenz-Overhead. Weiterführende Strategien (strukturiert, Token-Budget) können **bei Bedarf ergänzt** werden, um noch mehr Kontext zu komprimieren.

8. QUELLENVERZEICHNIS (APA)

- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems, 33*, 1877–1901. https://arxiv.org/abs/2005.14165
- Jiang, H., Wu, Q., Luo, X., Li, D., Lin, C.-Y., Yang, Y., & Qiu, L. (2023). LLMLingua: Compressing prompts for accelerated inference of large language models. *arXiv preprint arXiv:2310.05736*. https://arxiv.org/abs/2310.05736
- Jiang, H., Wu, Q., Luo, X., Li, D., Lin, C.-Y., Yang, Y., & Qiu, L. (2024). LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. *arXiv preprint arXiv:2310.06839*. https://arxiv.org/abs/2310.06839
- Li, X., Zhu, J., & Zhao, Y. (2023). Selective context: Efficient processing of long documents with language models. *arXiv preprint arXiv:2305.07726*.
- Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., & Liang, P. (2023). Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*. https://arxiv.org/abs/2307.03172
- Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. https://arxiv.org/abs/1908.10084
- Sennrich, R., Haddow, B., & Birch, A. (2016). Neural machine translation of rare words with subword units. *Proceedings of the 54th Annual Meeting of the ACL*, 1715–1725.
- Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2020). BERTScore: Evaluating text generation with BERT. *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1904.09675
- OpenAI. (2024). *Pricing for OpenAI API*. https://openai.com/pricing