Verlässlichkeit domänenspezifischer Sprache-nach-Text-Umwandlung am Beispiel des digitalisierten Zugleitbetriebs

Individual Research Project
Betreut von Lukas Pirl

Luca Kleinschmidt

2. Juni 2026

Idee

- Verlässlichkeit in Machine-Learning-Komponenten des ZliC
- Ausgabe der Sprache-zu-Text entlang einer Grammatik
- Vergleichbarkeit für Abstimmungen zwischen verschiedenen Modellen und Grammatikstrategien

Ziele der Verlässlichkeit

- Verwendete Modelle sind inherent nicht erklärbar
- Zuverlässigkeit des Sprache-zu-Text so *weit wie möglich* erhöhen, dass System meistens in funktionalem Zustand ist.
- Schnellere Reaktionszeit des Systems durch neuere Technologien
- Zugelassenes Sprachprotokoll als letzter Mechanismus der funktionalen Sicherheit [1]

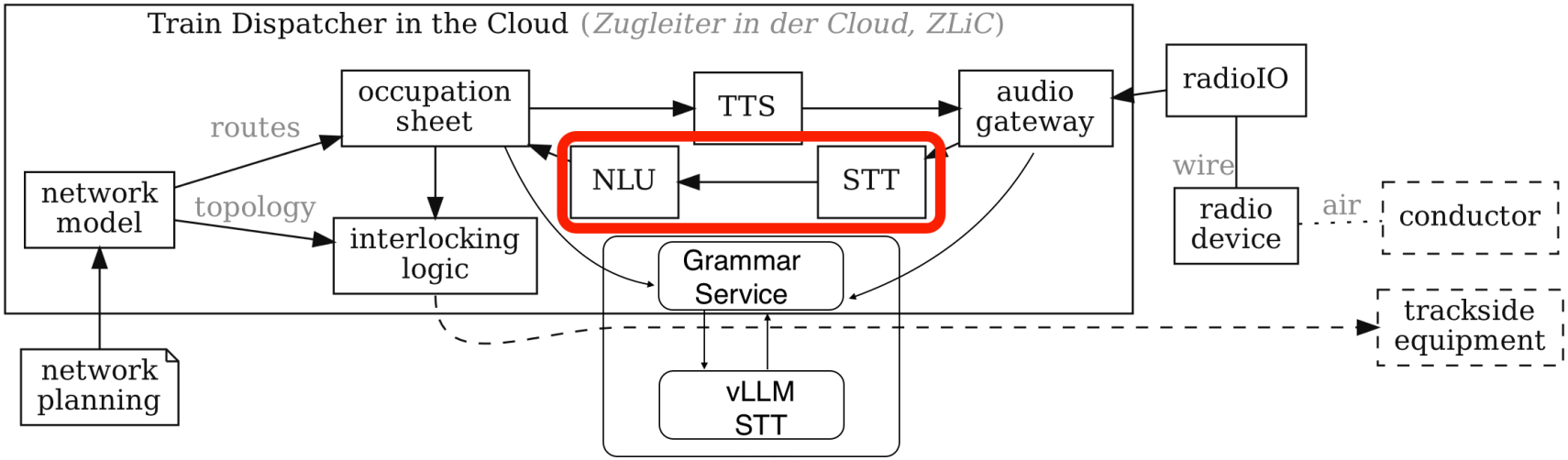


Figure 1: Architekturdiagramm des “Zugleiter in der Cloud”[1] Projekts

Modelle: “Mistral Voxtral Mini” [2] und “Google Gemma 4 E2B” [3] mit vLLM [4] und Xgrammar [5]

Grammatiken in ML

- Backus-Naur-Form

1. Modell sagt nächsten Token voraus
2. Menge mit allen von der Grammatik akzeptierten Tokens
3. Auswahl eines Tokens unter den akzeptierten Tokens

- Syntaktisch korrekt mit Punkt und Komma [6]

Aufbau der Grammatiken

- Grammatik striktem Zwang
- Anpassung der Grammatiken basierend auf dem Systemzustand

```
root ::= " " zugnummer " in " station
zugnummer ::= "1901" | "42" | "23"
station ::= "S-Berg" | "C-Dorf"
```

Probleme

- Garantiert syntaktisch korrekte Ausgabe auch bei *jeder* Eingabe [6]
- Semantik der Eingabe unterscheidet sich ggf. von der Grammatik akzeptierten Semantik [7]
- Wenn sich die Eingabesemantik von der akzeptierten Semantik nicht unterscheidet: Grammatik “führt” zu syntaktisch korrekten Ausgaben
- Token-Triggered-Grammar: Grammatik erst bei Beginn der Syntax [8]

Anwendungsziel des Grammar Constrained Decodings

- Eingabe semantisch korrekt → Ausgabe syntaktisch korrekt
- Eingabe semantisch inkorrekt → Ausgabe syntaktisch inkorrekt
- “Provozieren” einer falschen Ausgabe bei falscher Eingabe durch verschieden starke “Führung”
- Ermöglichen von Abstimmungen zwischen Ausgaben mittels gleicher Syntax

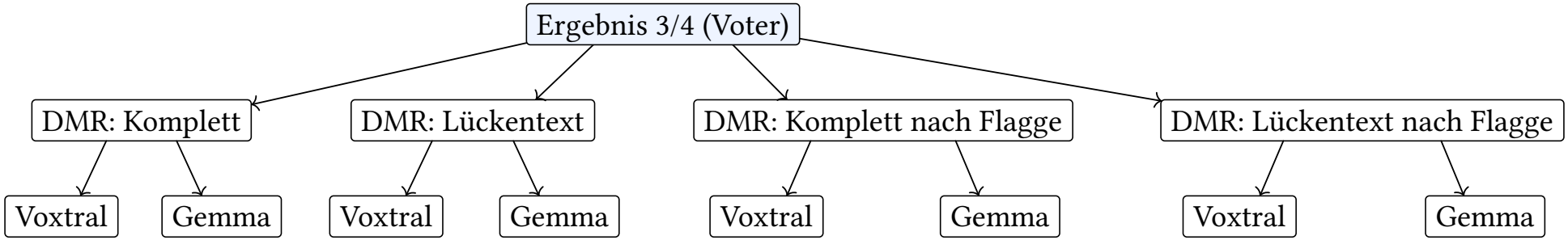


Figure 2: Schema eines Voters zwischen verschiedenen Grammatikstrategien und Modellen.

Vereinfachtes Beispiel

- Eingesprochen: “Zug 1901 in Berlin Hauptbahnhof.”
- Starke Führung: “Zug 1901 in Berlin Hbf.”
- Führung nach Flagge: “Zug 1901 in Berlin Hbf.”
- Abstimmung: “Zug 1901 in Berlin Hbf.”

Vereinfachtes Beispiel

- Eingesprochen: “Das Pferd frisst keinen Gurkensalat.”
- Starke Führung: “Zug 1901 in Berlin Hbf.”
- Führung nach Flagge: “Das Pferd frisst keinen Gurkensalat”
- Abstimmung: Fehlgeschlagen

Videodemo

- Typst kann leider keine Videos https://github.com/typst/typst/issues/955

Meilensteine des Projekts

- ZliC auf Grammatiken umrüsten (Ziel erreicht)
- Aufbau eines Testdatensatzes (Begonnen)
- Vermessung/Fehlerinjektion
  - Antwortzeit
  - Benchmark mit Testdaten
  - Testen der Strategien
- Verbesserungen an den Grammatikstrategien
- Abstimmungsstrategien

Testdaten

- Diverse Testdaten nicht existent ;)
- zlic.floe.cc dcl/dcl



Demo des Testdatenwerkzeugs

- Bitte um eure Mithilfe (5 min) zlic.floe.cc PW: dcl/dcl



Zusammenfassung

- ZliC mit Grammatik und Geschwindigkeitsvorteil
- Grammatik kann bei natürlicher Sprache Vergleichbarkeit schaffen
- Vergleichbarkeit für Fehlertoleranz
- Grammatikstrategie mit hohen Ausgabeabweichungen für gezielte Fehlzustände
- Bitte um eure Mithilfe (5 min) zlic.floe.cc PW: dcl/dcl



References

[1] L. Pirl, H. Herholz, D. Friedenberger, A. Boockmeyer, A. Polze, and B. Milius, "Train Dispatcher in the Cloud – Digitalising Track Warrant Control for Safe Train Operations in Structurally Transforming Areas," in *Transport Transitions: Advancing Sustainable and Inclusive Mobility*, C. McNally, P. Carroll, B. Martinez-Pastor, B. Ghosh, M. Efthymiou, and N. Valantasis-Kanellos, Eds., Cham: Springer Nature Switzerland, 2024, pp. 13–19.

[2] A. H. Liu *et al.*, "Voxtral." [Online]. Available: https://arxiv.org/abs/2507.13264

[3] "Gemma 4 model card." [Online]. Available: https://ai.google.dev/gemma/docs/core/model_card_4

[4] W. Kwon *et al.*, "Efficient Memory Management for Large Language Model Serving with PagedAttention," in *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[5] Y. Dong *et al.*, "Xgrammar: Flexible and efficient structured generation engine for large language models," *Proceedings of Machine Learning and Systems 7*, 2024.

[6] S. Geng, M. Josifoski, M. Peyrard, and R. West, "Grammar-Constrained Decoding for Structured NLP Tasks without Finetuning," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds., Singapore: Association for Computational Linguistics, Dec. 2023, pp. 10932–10952. doi: 10.18653/v1/2023.emnlp-main.674.

[7] D. Banerjee, T. Suresh, S. Ugare, S. Misailovic, and G. Singh, "CRANE: Reasoning with constrained LLM generation," in *Proceedings of the 42nd International Conference on Machine Learning*, in Proceedings of Machine Learning Research, vol. 267. Vancouver, Canada: PMLR, 2025.

[8] L. Li, Y. Dong, G. Wang, Z. Xu, A. Jiang, and T. Chen, “XGrammar-2: Efficient Dynamic Structured Generation Engine for Agentic LLMs.” [Online]. Available: https://arxiv.org/abs/2601.04426