ViPATalk meets AI:
Advancing Multimodal Virtual Anamnesis Simulation using LLM Fine-tuning

Martin Sboron [1] Oliver Thews [2] Thomas Schmid [1,3]

[1]Martin-Luther-University Halle-Wittenberg, Digital Research Methods in Medicine Group, Halle, Germany
[2]Martin-Luther-University Halle-Wittenberg, Julius Bernstein Institute of Physiology, Halle, Germany
[3]Lancaster University Leipzig, School of Computing and Communications, Leipzig, Germany

Abstract

Virtual simulated patients (VSP) have become an education tool for medical students and are implemented by frameworks of varying degrees of immersion. The path to free-form interaction between the user and the VSP has been of ever-growing interest. For a step in this direction, we extend the ViPATalk framework by Lippitsch et al. by replacing the current rule-based system for question categorization with a fine-tuned Large Language Model (LLM). We demonstrate that smaller LLMs perform similarly to larger models, when trained in the same domain, and that recent base models perform similarly to their language-specific fine-tuned counterparts. Further, our findings indicate that generated training data can be helpful when little data is available for certain categories.

ViPATalk-Framework

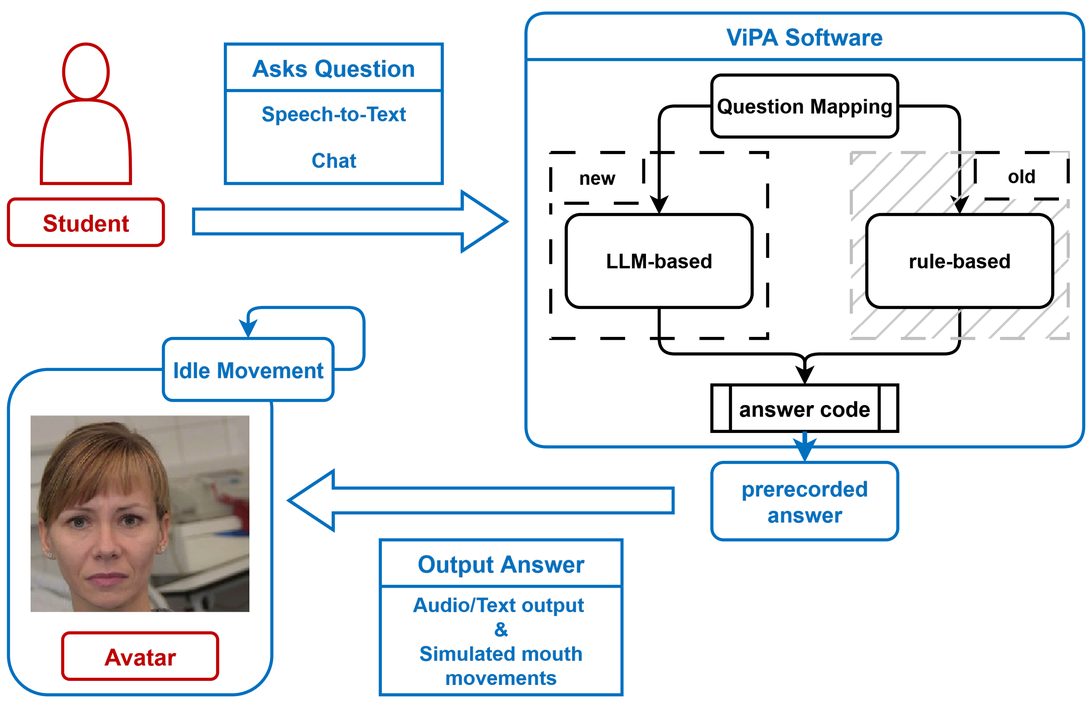

Figure: *ViPATalk-Framework pipeline, introduced by Lippitsch et al. [1], hatched: old question mapping: rule-base, white: new: LLM-based*

- ***Virtual Patient Anamnesis - Talk:*** multimodal interactive environment for anamnesis training for medical students
- questions are mapped to ***predefined answer codes*** with specific pattern for pre-recorded answers: e.g. "2_2_8" (Current Condition_Pain_Pain Duration)
- ***1600 to 2300 rules*:** own rules for word forms and tenses
- ***LLMs more general*:** can even map unseen question contents to at least similar answer codes
- responses of the avatar are carefully specified in advance to represent the underlying illness with realistic symptoms

Models, Methods and Data

Model	Company	Architecture	Params	Size	Ger-Var.
GBert-large	Deepset	Encoder	0.3B	1.3	+
Gemma 2	Google	Decoder	2B	5.3	+
Llama 3.2	Meta	Decoder	3B	6.5	-
Llama 3.1	Meta	Decoder	8B	16.1	+
Gemma 2	Google	Decoder	9B	18.5	+

Table: *Used LLMs, Ger-Var. stands for "German-Variant", for which "+" means used and "-" not used. Size is given in Gigabyte and parameters are given in billions.*

Chosen LLMs

- ***GBert***[2]: variant of Bert[3] pretrained on fully German corpus
- ***LLaMA***[4] and ***Gemma***[5] as generation-based competitors in NLP tasks
- all models were pretrained by the corresponding company and fine-tuned on our data set

Data

- 3 patients: AS, LD, MS
  - 108, 110 and 136 answer codes, resp.
  - 2983, 2965 and 2650 unique questions, resp.
- 70/30 training/testing split and 70/30 training/validation split (49% of data for training)
- splitting with answer code stratification

Methodology

- **Classification:**
  - GBert base model kept frozen and training of parameters of add-on classification adapter
  - direct mapping from question to labels of answer codes
- **Generation:**
  - base model kept frozen and training with LORA-method[6]
  - instruction prompt with the available answer codes and a description of the task
  - auto-regressive generation of the answer code

Results

Classification vs. Generation

Model	Type	Params	GV	Accuracy↑	Diff. to Vanilla
GBert-large	Classf.	0.3B	+	**90.3**	+90.3
Gemma 2	Gen.	2B	+	89.5	+55.6
Llama 3.2	Gen.	3B	-	88.8	+50.8
Llama 3.1	Gen.	8B	+	89.3	+32.3
Gemma 2	Gen.	9B	+	89.7	+29.7

Table: *Classification vs. Generation-based models: "Accuracy" and "Difference to Vanilla model" are given in % and GV stands for "German-Variant", for which "+" means used and "-" not used. All models are trained and tested on the data for patient AS.*

Comparing Model Size via Cross Validation

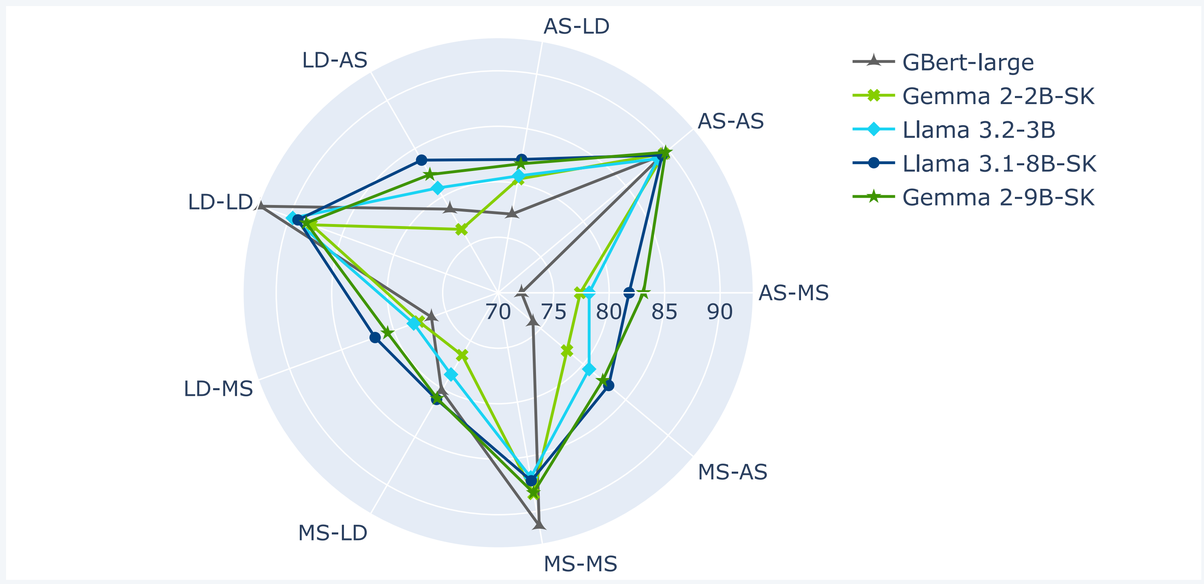

Figure: *Cross-validation over patients, shown in %. "AS-LD" means the model was trained on patient AS and tested on patient LD.*

Language specialization

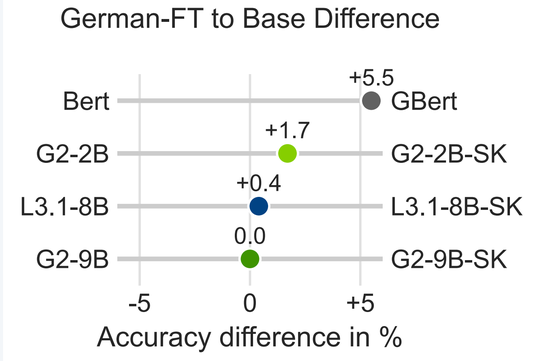

Figure: *German-language fine-tuned models vs. their base models: "G" = Gemma, "L" = Llama, "SK" = Sauerkraut-variant from VAGOsolutions[7]*

- all German-variants (except GBert) taken from VAGOsolutions
- for Llama 3.2 no Sauerkraut-variant was available at the time of experiments

Reaction to Data Augmentation

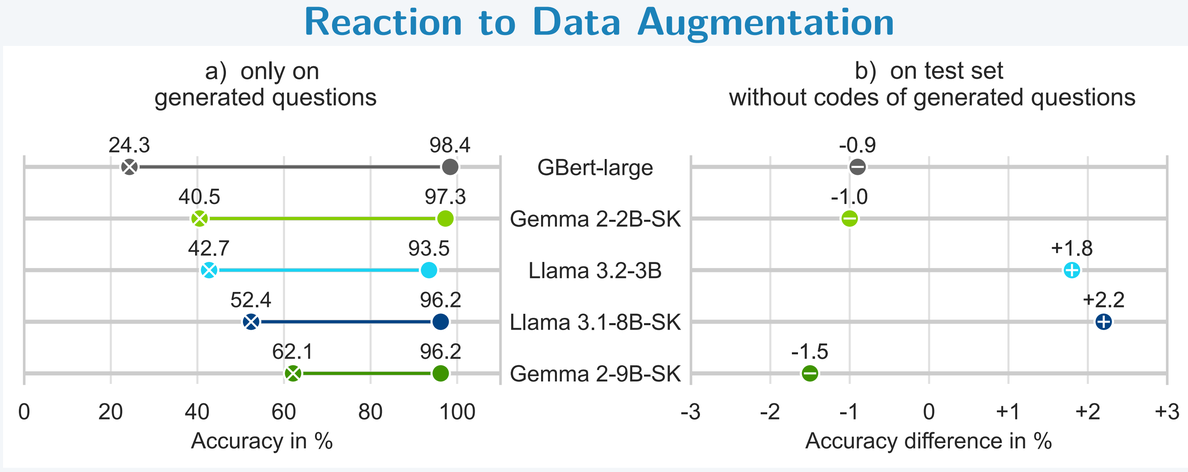

Figure: *a) Models tested on the set of generated questions without being trained on them (x-circle), and with being trained on them (full-circle). b) The accuracy difference when tested on the test set without the answer codes of the generated questions: between the model when trained without the generated questions and when trained with the generated questions (plus/minus-circle).*

- Evaluation of data augmentation for fine-tuning LLMs for patient MS using 185 questions for 12 answer codes (at least 10 per code) generated with ChatGPT4

Conclusion and Outlook

- **Our Decision:** GBert as extension of the ViPATalk framework
  ↪ best cost-effectiveness on single-patient performance
  - one model for each patient: significant question mapping improvement compared to rule-based system
  - can be operated without GPU → average latency per sample: tens of ms
    → using quantization, ONNX runtime [8] and Optimum framework [9]
- **Future Work:** Exploring merits of generation-based models for more generalization

References

[1] A. Lippitsch et al., "Development and evaluation of a software system for medical students to teach and practice anamnestic interviews with virtual patient avatars," Computer Methods and Programs in Biomedicine, vol. 244, 2024, p. 107964.

[2] B. Chan, S. Schweter, and T. Möller, "German's next language model," CoRR, vol. abs/2010.10906, 2020.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 4171–4186.

[4] A. Grattafiori et al., "The llama 3 herd of models," 2024.

[5] Gemma Team, "Gemma 2: Improving open language models at a practical size," 2024.

[6] E. J. Hu et al., "Lora: Low-rank adaptation of large language models," 2021.

[7] VAGO solutions, https://huggingface.co/VAGOsolutions, accessed: 2025-06-09.

[8] ONNX Runtime developers, "ONNX Runtime," https://onnxruntime.ai/, 2021, accessed: 2025-06-09.

[9] Hugging Face, "Optimum framework," https://github.com/huggingface/optimum, 2021, accessed: 2025-06-09.