

ARTICLE

Clinical information extraction for lower-resource languages and domains with few-shot learning using pretrained language models and prompting

Phillip Richter-Pechanski^{1,2,3,4,5} , Philipp Wiesenbach^{1,3,4,5}, Dominic Mathias Schwab³, Christina Kiriakou³, Nicolas Geis^{3,5}, Christoph Dieterich^{1,3,4,5} and Anette Frank²

¹Klaus Tschira Institute for Integrative Computational Cardiology, Heidelberg, Germany, ²Department of Computational Linguistics, Heidelberg University, Heidelberg, Germany, ³Department of Internal Medicine III, University Hospital Heidelberg, Heidelberg, Germany, ⁴German Center for Cardiovascular Research – Partner Site Heidelberg/Mannheim, Heidelberg, Germany, and ⁵Informatics for Life, Heidelberg, Germany

Corresponding author: Phillip Richter-Pechanski; Email: phillip.richter-pechanski@med.uni-heidelberg.de

(Received 19 January 2024; revised 5 July 2024; accepted 12 July 2024)

Abstract

A vast amount of clinical data are still stored in unstructured text. Automatic extraction of medical information from these data poses several challenges: high costs of clinical expertise, restricted computational resources, strict privacy regulations, and limited interpretability of model predictions. Recent domain adaptation and prompting methods using lightweight masked language models showed promising results with minimal training data and allow for application of well-established interpretability methods. We are first to present a systematic evaluation of advanced domain-adaptation and prompting methods in a lower-resource medical domain task, performing multi-class section classification on German doctor's letters. We evaluate a variety of models, model sizes (further-pre)training and task settings, and conduct extensive class-wise evaluations supported by Shapley values to validate the quality of small-scale training data and to ensure interpretability of model predictions. We show that in few-shot learning scenarios, a lightweight, domain-adapted pretrained language model, prompted with just 20 shots per section class, outperforms a traditional classification model, by increasing accuracy from 48.6% to 79.1%. By using Shapley values for model selection and training data optimization, we could further increase accuracy up to 84.3%. Our analyses reveal that pretraining of masked language models on general-language data is important to support successful domain-transfer to medical language, so that further-pretraining of general-language models on domain-specific documents can outperform models pretrained on domain-specific data only. Our evaluations show that applying prompting based on general-language pretrained masked language models combined with further-pretraining on medical-domain data achieves significant improvements in accuracy beyond traditional models with minimal training data. Further performance improvements and interpretability of results can be achieved, using interpretability methods such as Shapley values. Our findings highlight the feasibility of deploying powerful machine learning methods in clinical settings and can serve as a process-oriented guideline for lower-resource languages and domains such as clinical information extraction projects.

Keywords: Prompting; few-shot learning; pretraining; medical information extraction; language models

* AF and CD jointly supervised this work and are shared last authors.

1. Introduction

Vast amounts of clinical data are stored in unstructured text, such as doctor's letters. Natural language processing (NLP) and machine learning (ML) can make their information available for research and clinical routine. While supervised ML approaches rely on large amounts of manually annotated training data, recent developments in NLP showed promising results in text classification tasks using pretrained language models (PLM) and prompting (Brown *et al.* 2020). Prompting exploits the ability of PLMs to make correct predictions if guided through context; in combination with supervised methods, they achieve state-of-the-art results on various text classification tasks (Liu *et al.* 2023).

Doctor's letters are typically divided into sections, such as anamnesis (patient medical history), diagnosis or medication, containing semantically related sentences. Typically, it is not necessary to consider all sections to obtain specific medical information (Richter-Pechanski *et al.* 2021) or medication information (Uzuner, Solti, and Cadag 2010). Instead, medical information extraction (MIE) tasks, such as medication extraction or patient cohort retrieval, can be improved by contextualizing the information in a doctor's letter (Edinger *et al.* 2017). However, automatic section classification is non-trivial due to a high variability of the structuring of information across physicians and time periods (Lohr *et al.* 2018).

In close collaboration with physicians from clinical routine, we identified four challenges of MIE projects in the clinical domain (Hahn and Oleynik 2020) (Fig. 1).

- Ch₁ **Domain-and-Expert-dependent:** Annotation projects often require an active involvement of domain experts for data annotation and model evaluation. This is particularly relevant for lower-resource languages and domains such as the clinics and German language.
- Ch₂ **Resource-constrained:** Domain experts are costly and have only limited time resources. By contrast, external expert involvement is difficult due to strict data protection measures (Richter-Pechanski *et al.* 2021).
- Ch₃ **On-premise:** Personal data are confidential, which means that many MIE projects are carried out entirely on premise, that is, in the clinical IT infrastructure. However, computational resources in clinical infrastructures are often a limiting factor (Taylor *et al.* 2023).
- Ch₄ **Transparency:** Due to the sensitivity of clinical information, safety standards for using MIE results in clinical routine are high: model predictions must be of high quality, transparent, explainable, and as comprehensible as possible (Tjoa and Guan 2020).

We evaluate best-practice strategies to identify an ideal setup to address the multifaceted challenges of conducting a MIE task such as clinical section classification. Specifically, we identify and propose the following solutions:

- S₁ We reduce the demand for clinical knowledge in MIE by exploiting existing domain knowledge available in hospitals, such as clinical routine documents. We evaluate domain- and task-adapted (Gururangan *et al.* 2020) general-use PLMs, as well as PLMs pretrained on clinical data from scratch (Bressem *et al.* 2024) in combination with prompt-based learning methods (Schick and Schütze 2021a), which require only limited training data.
- S₂ To reduce time investment and costs of manual data annotation through clinical experts, we apply few-shot learning (Lake, Salakhutdinov, and Tenenbaumt 2015) and context-enriched training data using prompt-based fine-tuning with pattern-exploiting training (PET + PETAL) (Schick and Schütze 2021a; Schick, Schmid, and SchÄijtze 2020) and compare the results with supervised sequence classification methods. We further evaluate the

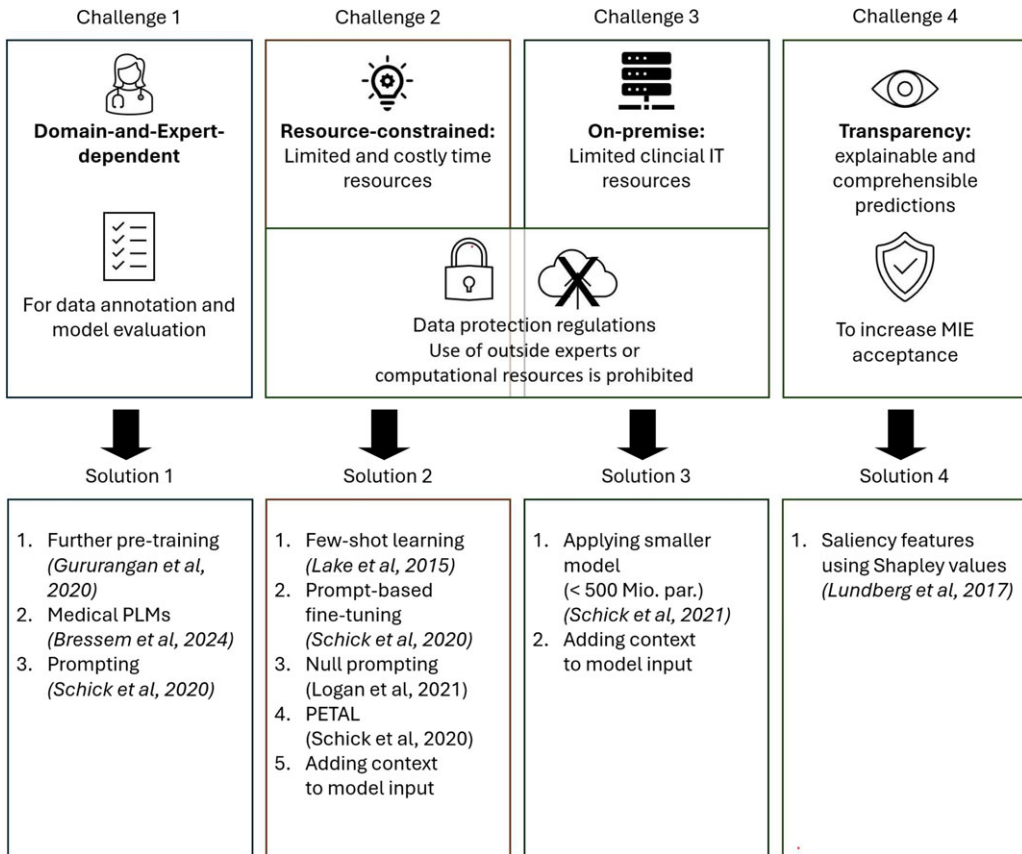


Figure 1. Challenges for MIE projects in clinics: Our proposed solutions to main challenges for MIE projects in a clinical setting. Abbreviation: “*par.*” refers to parameters.

feasibility of null prompts (Logan et al. 2022), which have been shown to alleviate the search for effective prompts while achieving improved results.

S₃ While large language models (LLMs) have recently shown impressive medical capabilities (Singhal et al. 2023), their demands of compute power, and currently unsolved issues regarding automatic evaluation, faithfulness control, and trustworthiness make their use in clinical contexts often impractical (Parnami and Lee 2022; Thirunavukarasu et al. 2023). We, therefore, focus on smaller PLMs (110–345 million learnable parameters) in a few-shot learning setting. Notably, prompt-based fine-tuning already achieves higher accuracy with smaller, encoder-based PLMs compared to PLMs fine-tuned for sequence labeling with a full-fledged training dataset in German (Schick and Schütze 2021a).

S₄ To address the need for transparent and trustworthy model predictions in clinical routine, we use well-established masked-language-models. They allow application of state-of-the-art interpretability methods that rely on saliency features computed with, for example, Shapley values (Lundberg and Lee 2017), to explain our model predictions.

In what follows we conduct in-depth evaluations of these proposed solutions in a real-world section classification task, applied to German doctor’s letters from the cardiovascular domain. To our knowledge, this is the first in-depth evaluation of a prompt-based fine-tuning method such as PET on real-world clinical routine data in German language.

1.1 State of research

From fine-tuning to few-shot learning with prompting. Since 2017, most NLP tasks apply a *pretrain-then-finetune* paradigm: neural models are pretrained with a language modeling objective on large amounts of unlabeled text and then fine-tuned for a down-stream task on a smaller amount of annotated data. However, even fine-tuned PLMs often perform poorly with sparse training data (Gao, Fisch, and Chen 2021) and require significant amounts of manually labeled training data to perform well (Liu *et al.* 2023). Especially with low(er)-resource languages and in special domains, we often face a scarcity of high-quality labeled data. With recent scaled-up language models, we observe another shift to a *pretrain-then-prompt* paradigm, where tasks are formulated using natural language prompts (Shin *et al.* 2020; Schick and Schütze 2021a; Reynolds and McDonnell 2021; Gao *et al.* 2021), revealing impressive zero-shot capabilities of these models (Kojima *et al.* 2022; Liu *et al.* 2023). While in many applications at least a few training samples are still required to guide model predictions, prompt-based learning soon matched and even surpassed the performance of fine-tuning in various few-shot learning settings (Liu *et al.* 2023; Taylor *et al.* 2023).

Although model size played a critical role in this development (Chowdhery *et al.* 2023), smaller, encoder-based PLMs have also been successfully applied in few-shot scenarios using prompt-based fine-tuning in combination with a semi-supervised approach (Schick and Schütze 2021b; Wang *et al.* 2022). Especially, framing text classification tasks as cloze-style problems using pattern-exploiting training (PET) showed promising results for various classification tasks (Schick and Schütze 2022) (cf. Section 2).

Domain adaptation through further-pretraining. Further-pretraining means training an already pretrained language model further on domain-specific texts using a language model objective. This allows domain-adaptation of general-purpose language models. General-purpose LMs achieve high performance across many tasks (Sun *et al.* 2019), yet performance typically drops in out-of domain settings. Several studies explored *further-pretraining* on domain-specific data (Zhu *et al.* 2021), in such cases, demonstrating that further-pretraining even on small-sized task-specific data can improve results in out-of-domain down-stream tasks (Gururangan *et al.* 2020).

PLMs for the medical domain. Medical PLMs, pretrained on medical data from scratch and further-pretrained medical PLMs, have outperformed general PLMs in several tasks (Sivarajkumar and Wang 2022; Taylor *et al.* 2023). However, clinical routine texts, as used in this study, have unique textual properties compared to biomedical texts on which such models are trained. This increases the complexity of medical NLP tasks in clinical routine (Leaman, Khare, and Lu 2015; Hahn and Oleynik 2020). Also, only a limited number of further-pretrained and clinical PLMs have been published to date, mostly for English, primarily due to strict data protection regulations (Lee *et al.* 2020; Li *et al.* 2023; Bressem *et al.* 2024).

Prompting methods in clinical NLP. Despite extensive research on PLMs for medical domain, previous research has mainly focused on supervised fine-tuning with full-fledged training data approaches that use large amounts of training data (Wu *et al.* 2020; Taylor *et al.* 2023) with the exception of Taylor *et al.* (2023) who investigated prompting on English clinical data. Thus, there is a need to investigate how further-pretraining influences prompting methods in few-shot scenarios.

PET performed well in various downstream tasks in English, i.a. in biomedical text classification, where for adverse drug effect classification it outperformed GPT3 with an *F1*-score of 82.2% versus 68.6% (Schick and Schütze 2022). This highlights the need for thorough evaluation of PET in clinical routine tasks with medical domain PLMs, particularly for lower-resource languages such as German (Leaman *et al.* 2015; Hahn and Oleynik 2020).

Clinical section classification. Identifying sections in clinical texts has been shown to enhance performance on several MIE tasks (Pomares-Quimbaya, Kreuzthaler, and Schulz 2019). However, this research field remains underdeveloped, partly due to the lack of benchmark datasets (cf. comprehensive survey Landolsi, Hlaoua, and Ben Romdhane 2023). Therefore, most studies focus on

English clinical texts (Denny 2008; Edinger 2017). In-depth studies focusing on few-shot learning scenarios and prompting are still lacking (Ge 2023). Our work is the first to thoroughly investigate these methods on a freely available clinical German benchmark corpus. Furthermore, we extensively explore German PLMs (Bressem 2024) for clinical domains to detect suitable (further-)pretraining methods for prompting and their effect on section classification.

Interpretability. Given the black-box nature of deep learning architectures, the interpretability of model outputs is challenging and attracts much interest (Fan *et al.* 2021), especially in safety-critical domains such as clinical routine. Various feature attribution methods have been developed to address these issues (Ribeiro, Singh, and Guestrin 2016; Sundararajan, Taly, and Yan 2017; Lundberg and Lee 2017), but we still face challenges in assessing their quality (Jacovi and Goldberg 2020; Attanasio 2023). Shapley values provide a theoretically well-founded approach to determine the contribution of individual input features to a model prediction. A computationally optimized implementation called SHAP (Shapley *et al.* 1953) can be applied out-of-the-box on transformer-based models. To our knowledge, we are the first to study the use of Shapley values for data and model optimization in clinical tasks.

Progress in the area of LLMs. Recently, generative LLMs with billions of parameters deliver impressive results in various general (Brown *et al.* 2020; Scao *et al.* 2022; Chowdhery *et al.* 2023; Touvron *et al.* 2023) and biomedical and clinical NLP tasks (Singhal *et al.* 2023; Thirunavukarasu *et al.* 2023; Clusmann *et al.* 2023; Peng *et al.* 2023). However, many challenges need to be addressed before LLMs can be applied in clinical tasks (Wang, Zhao, and Petzold 2023): Running them via external APIs is typically prohibited due to data protection regulations. Despite efforts to make LLMs available for use in protected infrastructures (cf. <https://github.com/bentoml/OpenLLM>), model deployment in clinical infrastructures is often not feasible (Taylor *et al.* 2023). Moreover, out-of-the-box local GPT and Llama models have shown poor performance in biomedical tasks (Moradi *et al.* 2021; Wu *et al.* 2023). Finally, due to the generated outputs of autoregressive PLMs, their use in clinical NLP implies unsolved issues concerning automatic evaluation (Guo *et al.* 2023; Chang *et al.* 2024) and judging the faithfulness of model predictions (Parcalabescu and Frank 2024), which are both critical in the clinical domain.

While evaluation of autoregressive LLMs will mature in the future, our study on encoder-based models serves as a process-oriented guideline for MIE projects in clinical routine tasks for lower-resource languages. All constraints discussed in this study: (1) expert-dependency, (2) data protection regulations, (3) demand for on-premise solutions, and (4) transparency requirements, invariably apply to popular local LLMs such as Llama (Touvron *et al.* 2023) or Mistral (Jiang *et al.* 2023), and can serve as guidelines for evaluating these models, too.

2. Methods

2.1 Pattern-exploiting training (S_1 and S_2)

In our experiments, we systematically evaluate methods for few-shot learning, that is, using minimal training data, in a lower-resource domain and language, in our case German clinical routine (Hahn and Oleynik 2020; Jantscher *et al.* 2023; Idrissi-Yaghir *et al.* 2024). Specifically, we evaluate PET, a semi-supervised prompting method optimized for few-shot learning scenarios (Schick and Schütze 2021a) which is designed to recast classical text classification or information extraction tasks as a language modeling problem. In our study, we classify paragraphs of German doctor's letters into a set of nine section categories (Table 1). The objective is, for instance, to accurately categorize a paragraph such as *The patient reports pressure pain in the left chest* under the section class *Anamnesis*.

To conduct PET experiments, we need a pretrained masked language model M with a vocabulary V , a few-shot dataset with training instances $x_i \in X$ and target labels $y_i \in Y$. We further need a pattern function P that maps instances to a set of cloze sentences (templates) $P: X \mapsto V^*$, and a verbalizer function $v: Y \mapsto V$ that maps each label to a single token from the vocabulary of M .

Table 1. Distribution of section classes: Number of samples per section class per corpus split. English translation in round brackets.

	Training set	Test set
Anrede (Salutation)	402	99
Diagnosen (Diagnosis)	8,023	1,738
AllergienUnverträglichkeitenRisiken (AllergiesIntolerancesRisks)	1,031	236
Anamnese (Patient Medical History)	1,188	281
Medikation (Medication)	6,148	1,627
Befunde (Findings)	15,396	3,914
Zusammenfassung (Summary)	3,645	843
Mix (Mix)	945	242
Abschluss (Closing Remarks)	2,805	695
Total	39,583	9,675

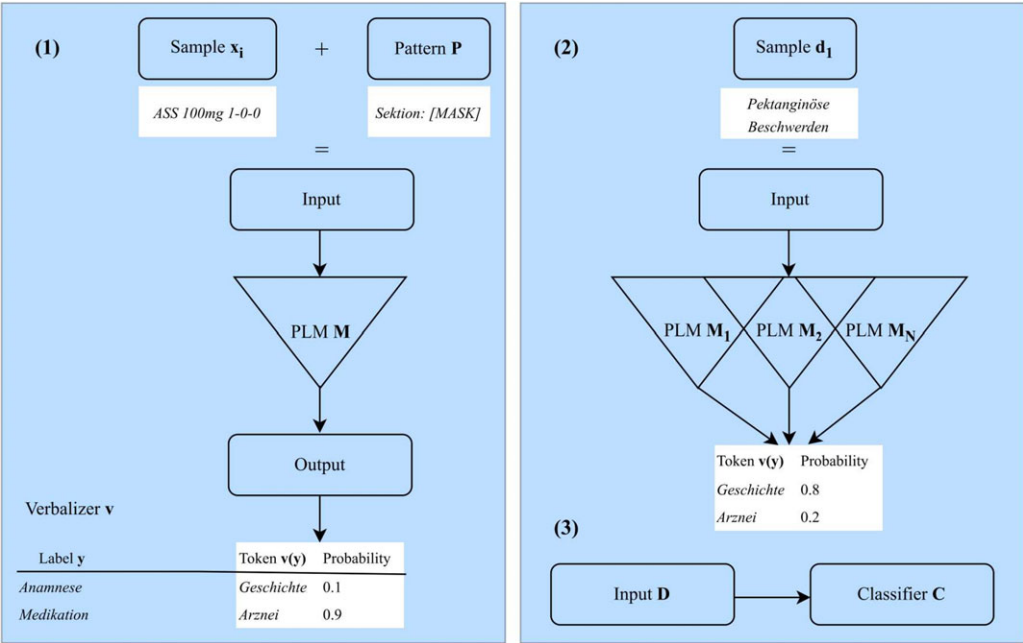


Figure 2. PET workflow: Three main steps: (a) Apply pattern function $P(x)$ to all few-shot training instances X . Fine-tune a PLM M using a language model objective on each pattern. The output of the PLM is mapped using a verbalizer function $v(y)$. (b) An ensemble of M trained on each pattern is used to annotate an unlabeled dataset D with soft labels. (c) A classifier C with a classification head is trained on D .

The PET workflow contains three basic steps (see Fig. 2): (1) applying P to each input instance x_i and fine-tune a model M for each template to obtain the most likely token for the *MASK* token $v(y)$, (2) use the ensemble of fine-tuned models M from the previous step and annotate a large unlabeled dataset D with soft labels, and (3) train a final classifier C with a traditional sequence classification head on the labeled dataset D .

2.1.1 Creating templates

Template engineering is a crucial hyperparameter in a PET experiment. For the *core experiments*, we used four different template types (including examples and English translations (in brackets)):

- Null prompt: SAMPLE [MASK]
Keine peripheren Ödeme [MASK]
(*No peripheral edema* [MASK])
- Punctuation: SAMPLE : [MASK] and SAMPLE - [MASK]
Keine peripheren Ödeme : [MASK]
(*No peripheral edema :* [MASK])
- Prompt: SAMPLE Sektion [MASK]
Keine peripheren Ödeme Sektion [MASK]
(*No peripheral edema Section* [MASK])
- Q&A: SAMPLE Frage: Zu welcher Sektion gehört dieser Text?
Antwort: [MASK]
Keine peripheren Ödeme Frage: Zu welcher Sektion gehört dieser Text? Antwort: [MASK]
(*No peripheral edema Question: To which section does this text belong? Answer:* [MASK])

To minimize engineering costs we also evaluated the feasibility of using exclusively null prompts, by removing all tokens from prompt templates, as proposed by Logan *et al.* (2022). We defined three null prompt templates: (1) SAMPLE [MASK]; (2) [MASK] SAMPLE; and (3) [MASK] SAMPLE [MASK].

2.1.2 Verbalizer

Defining the verbalizer token can be tedious, because domain knowledge and technical expertise about the used PLM is required. This can be a significant issue, as such a comprehensive knowledge is uncommon in the clinical setting. Moreover, PET restricts the verbalizer token to a single token. Hence, an appropriate and intuitive token may not be applicable for a label mapping, if it is not included in the PLM's vocabulary. For instance, the word *Anamnese* is not part of the *gbert* vocabulary. This makes a verbalizer search for clinicians quite challenging. Therefore, we use PET with automatic labels (PETAL) for all our experiments, except for the zero-shot baselines (Schick *et al.* 2020). This can reduce engineering costs and makes our experimental setup more comparable and reproducible. As visualized in Suppl. Fig. S2 PETAL calculates the most likely verbalizer token per label, given the few-shot training data for each pattern and given a PLM. We created a separate verbalizer for each few-shot size for each training set.

2.2 Pretrained language models (S_1 & S_3)

To evaluate the feasibility of exploiting existing clinical domain knowledge by further-pretraining, we used a set of three language models, all based on the BERT architecture (Devlin *et al.* 2019) and available at Hugging Face Hub: (1) *deepset/gbert-base* (Chan, Schweter, and Möller 2020), (2) *deepset/gbert-large* (*gbert*), (3) *Smanjil/German-MedBERT* (*medbertde*) (Bressem *et al.* 2024). The largest model *gbert-large* contains 340 million parameters. In our clinical infrastructure, which contains a maximum of two NVIDIA RTX6000 GPUs, we were able to perform all further-pretraining experiments within a reasonable timeframe (cf. Suppl. Section S3). Compared to current foundation models with billions of parameters, we consider these models as lightweight. For both *gbert* and *medbertde*, we create medical-adapted variants by further-pretraining, as proposed by Gururangan *et al.* (2020) to assess the impact of different pretraining datasets on section classification results (Fig. 3). We defined datasets for three different pretraining approaches:

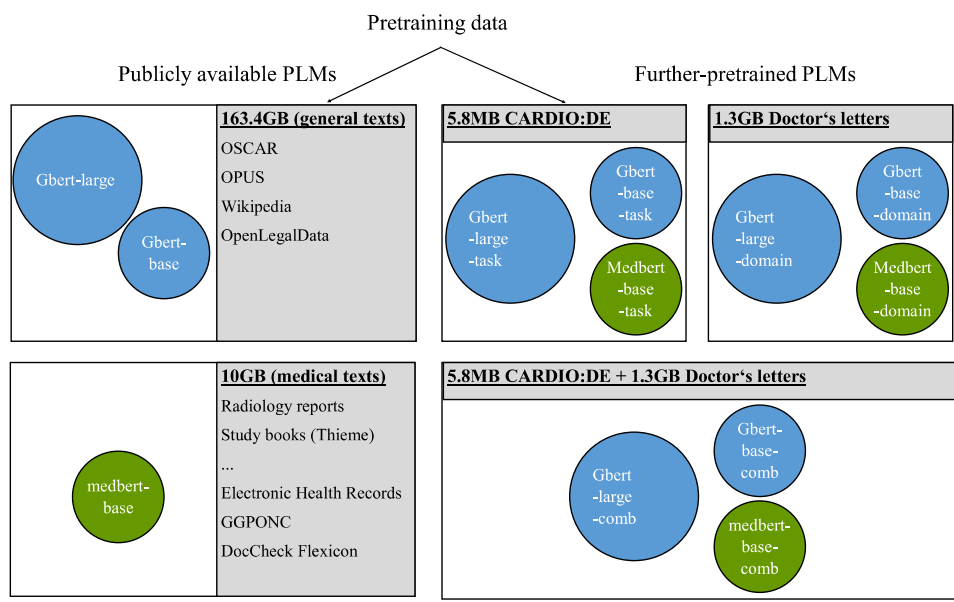


Figure 3. Pretrained language models: We use two publicly available PLMs: *gbert* and *medbertde*. We evaluate base and large *gbert* models. Four pretraining methods are used: (a) publicly available, (b) task-adapted, (c) domain-adapted, and (d) task- and domain-adapted combined.

1. *task-adaptation*. Using CARDIO:DE, cf. Section 2.4.1. This dataset contains unlabeled data extracted from the same source as the training and test data of the section classification task. It is relatively small, only 5.8 MB (megabytes). (PLMs appended with suffix *-task*)
2. *domain-adaptation*. Using 179,000 doctor's letters from the Cardiology department at the University Hospital, cf. Section 2.4.2. This dataset contains a broad range of texts from clinical routine in cardiovascular domain. With 1.3 GB (gigabytes), it is significantly larger than the task-adaptation dataset. (PLMs appended with suffix *-domain*)
3. *combination of both approaches* Further-pretrain a domain-adapted PLM on our task specific data (PLMs appended with suffix *-comb*)

We performed pretraining using a masked language modelling objective (cf. <https://tinyurl.com/5n8bjnbh>). For hyperparameters and further training details see Suppl. Section S3.

2.3 Shapley values (S_4)

In many safety-critical domains, in particular in the clinical domain, it is crucial to (1) understand the inner workings of a model (faithfulness) and to (2) evaluate how convincing a model interpretation is for a human observer (plausibility) (Jacovi and Goldberg 2020). This can increase trust in model predictions (explainable AI) by identifying which token contributed to a specific prediction. Furthermore, if a model makes incorrect predictions, allocating such tokens can help to understand and address these issues.

In recent years, Shapley values became a valuable tool in NLP for local model interpretations using saliency features (Attanasio *et al.* 2023). Shapley values offer a systematic approach to attribute the impact of individual textual components (token, token sequences) on a model prediction. In our setup, we apply Shapley values in two ways: (1) From a clinical routine perspective: to make deep learning model predictions more transparent and explainable and (2) from an engineering perspective: to detect biases or errors in the training data and to support choosing

the most optimal model architecture. Shapley values, originating from cooperative game theory, allocate the importance of each feature by averaging its marginal contribution across all possible feature combinations in predicting an outcome (Lundberg *et al.* 2017). The Shapley value for a feature i is given by

$$\phi_i(f) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} [f(S \cup \{i\}) - f(S)] \quad (1)$$

Here f is the prediction function, S is a subset of all features without feature i , and N is the set of all features.

In our experiments, we use SHAP (SHapley Additive exPlanations) because it offers an optimized algorithm that approximates Shapley values with reduced computational costs, making its application feasible for practical use (Mosca *et al.* 2022). Furthermore, we conducted experimental explorations and compared several interpretability methods in advance with ferret, a framework for benchmarking popular explainers on transformers (Attanasio *et al.* 2023), finding that SHAP was the best-performing method for our setup.

2.4 Data

2.4.1 Annotated corpus

For our experiments, we used a German clinical corpus from the cardiovascular domain, CARDIO:DE, encompassing 500 doctor's letters from the Cardiology Department at the Heidelberg University Hospital. For more details about the dataset, preprocessing steps, data annotation, and data distribution, cf. Richter-Pechanski *et al.* (2023). The corpus can be accessed via heiData, a public research repository; see Richter-Pechanski and Dieterich (2023). The complete corpus contains 993,143 tokens, with approximately 31,952 unique tokens. The corpus was randomly split into CARDIO:DE400 containing 400 letters (805,617 tokens) for training and CARDIO:DE100, containing 100 letters (187,526 tokens) for testing. The corpus was automatically de-identified, by replacing protected health information (PHI) containing patient sensitive identifiers with placeholders using an in-house deep learning model (Richter-Pechanski *et al.* 2019). This was followed by a manual review involving domain experts to fix de-identification errors. To increase readability and semantic consistency and to decrease the chance for re-identification, all PHI placeholders were replaced with semantic surrogates, as proposed in Lohr, Eder, and Hahn (2021).

We split the corpus by newline characters, which are part of the MS-DOC source documents. Sentence splitting the corpus with publicly available sentence splitting methods or by pattern heuristics showed unsatisfactory results. Furthermore, sequence length of newline split paragraphs rarely exceed 512 token (min: 3, max: 599, mean: 30.9, median: 16, 99th percentile: 205), thus, comply with most PLM sequence length restrictions. If a paragraph exceeds the maximum sequence length of the PLM, we trim the sample accordingly.

The corpus contains 116,898 paragraphs manually annotated with 14 section classes: *Anrede* (Salutation/Greeting), *AktuellDiagnosen* (Current Diagnosis), *Diagnosen* (Diagnosis), *AllergienUnverträglichkeitenRisiken* (AllergiesIntolerancesRisks), *Anamnese* (Patient Medical History), *AufnahmeMedikation* (Admission Medication), *KUBefunde* (Body Findings), *Befunde* (Findings), *EchoBefunde* (Echocardiogram Findings), *Labor* (Laboratory), *Zusammenfassung* (Summary), *Mix* (Mix), *EntlassMedikation* (Discharge Medication), *Abschluss* (Closing Remarks) (see CARDIO:DE section classes, Suppl. Tab. S1). Manual annotation was conducted on the paragraph level, no nested annotations were allowed. For our experiments, we reduced the section classes to the most significant sections. We removed the *Labor* section, as it contains flattened tables resulting in a large amount of relatively well structured and short numeric samples. Internal experiments showed that they can be sufficiently identified using regular expressions

Table 2. Contextualized paragraphs: A sample annotated as *AllergiesIntolerancesRisks* with three different context types, each separated by the [SEP] token. English translation in *italics*.

Context type	Example
nocontext	Cvrf: Hypertonie, Nikotinkonsum, Hypercholesterinämie <i>Cardiovascular risk factors: high blood pressure, smoker, high cholesterol</i>
context	– OP am 02.01.2011 [SEP] Cvrf: Hypertonie, Nikotinkonsum, Hypercholesterinämie [SEP] Anamnese: <i>– Surgery on January 2, 2011 [SEP] Cardiovascular risk factors: high blood pressure, smoker, high cholesterol [SEP] Patient medical history:</i>
prevcontext	– OP am 02.01.2011 [SEP] Cvrf: Hypertonie, Nikotinkonsum, Hypercholesterinämie <i>– Surgery on January 2, 2011 [SEP] Cardiovascular risk factors: high blood pressure, smoker, high cholesterol</i>

and patterns. Furthermore, we merged seven semantically similar classes in CARDIO:DE annotations to three meta classes: (1) *Diagnosen*: (*AktuellDiagnosen* + *Diagnosen*), (2) *Medikation*: (*AufnahmeMedikation* + *EntlassMedikation*), and (3) *Befunde*: (*KUBefunde* + *EchoBefunde* + *Befunde*). Our final dataset contains 49, 258 paragraphs annotated with 9 section classes (Table 1).

During annotation human annotators of CARDIO:DE were presented the whole document (for further annotation details, see Richter-Pechanski *et al.* 2023). To mimic this setup for our automatic section classifiers in this study, we introduced basic information about document structure to the model without introducing additional preprocessing steps or external knowledge. In addition to our training data containing single paragraph samples, we assessed two types of context-enriched datasets for our experiments (Examples Table 2):

- no-context (a single paragraph to be classified)
- context (previous paragraph + main paragraph + subsequent paragraph)
- prevcontext (previous paragraph + main paragraph)

The context-enriched samples still mostly comply with sequence length restrictions of PLMs (minimum 7, maximum 967, mean length 90.2, median length 63 and 99th percentile 371 sub tokens). If the sequence length of the context enriched sample is exceeded, we trim the sequence of the context to fit the maximum sequence length of the PLM.

2.4.2 Pretraining data

For all pretraining experiments, we used an internal clinical routine corpus containing approximately 179, 000 German doctor’s letters in a binary MS-DOC format covering the time period 2004–2020. We collected the letters from the Cardiology Department of the University Hospital Heidelberg. The pretraining corpus is disjoint from the annotated corpus. We conducted the following preprocessing steps: each letter was converted into a UTF-8 encoded raw text file using the LibreOffice command line tool *soffice* (version 6.2.8). We chose LibreOffice, as it best preserved the structure of newlines and blanklines. We automatically de-identified all letters using a method based on a deep learning model trained on internal data, see Richter-Pechanski *et al.* (2019). We replaced PHI tokens with semantic surrogates, see Lohr *et al.* (2021). All doctor’s letters were concatenated into a single raw text file. We separated each new letter by the sequence *###BEGINN*. All empty lines and all tables containing laboratory values were removed. The corpus is sentence splitted using NLTK’s (version 3.7) *PunktSentenceTokenizer*.

The doctor's letters were further supplemented by the GGPONC corpus, which contains German oncology guidelines, with a total of 2 million tokens (Borchert 2022). The final corpus covers 1.3 GB of raw text, approximately 218, 084, 190 tokens and 667, 903 unique tokens.

2.5 Experimental setup

2.5.1 Metrics

We measure section classification performance with accuracy for per-model results. In a multi-class text classification task, the accuracy is defined as the ratio of text documents correctly classified to their respective classes over the total number of text documents:

$$\text{Accuracy} = \frac{\sum_{i=1}^n \text{TP}_i}{\text{Total Number of Texts}} \quad (2)$$

where TP_i represents the true predictions for each class i and n is the total number of classes.

To measure section classification performance per-section class, we use the F_1 -score. It is defined as the harmonic mean of precision and recall given by

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (3)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (4)$$

Hence, the F_1 -score is defined by

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

where TP, FP, and FN represent true positives, false positives, and false negatives, respectively.

We used approximate randomization tests (Yeh 2000) to measure statistical significance for accuracy and F_1 -score results. Results are considered significant if $p < 0.05$, cf. (<https://github.com/smartschat/art>).

2.5.2 Creating few-shot data

To conduct PET experiments, we created six few-shot datasets. Each dataset contains N paragraphs per section class with size $N = 10, 20, 50, 100, 200$, and 400 randomly selected from the CARDIO:DE400 data (random seed 42). Each paragraph includes the previous and subsequent context paragraph. All other context types (*nocontext*, *prevcontext*) are derived from this dataset. Each few-shot set includes three labeled training files and three unlabeled files with the remaining samples from the CARDIO:DE400 dataset (Suppl. Fig. S3). All experiments were evaluated on the complete CARDIO:DE100 held-out dataset.

2.5.3 Core experiments

We conducted *core experiments* to assess the performance of different section classification models along three dimensions to compare: (1) fine-tuned sequence classification model variants (SC) to few-shot prompt-based learning with PET (S_2 , Fig. 2), (2) four different pretraining methods for clinical adaptation (S_1), and (3) six different few-shot sizes: 10 – 400 (S_2).

The SC model is trained using a BERT-architecture with an additional output layer for a sequence-classification task as described in Devlin *et al.* (2019). We use the SC implementation of the PET framework, defined by the parameter — `method sequence_classifier`.

For all *core experiments*, we used base-sized BERT models (S_3) (*gbert-base-** and *medbertde-base-**) using all five templates combined and *nocontext* samples (Suppl. Tab. S2). To measure

standard deviation in *core experiments* and *additional experiments*, we used three disjoint training sets including their unlabeled sets for each few-shot set. Furthermore, we conducted all experiments with two random initial seeds (123 and 234).

2.5.4 Additional experiments

In *additional experiments*, we investigate the effectiveness of further parameters, using the model that performed best in *core experiments*, with reduced few-shot sets: 20, 50, 100, and 400. We investigate the impact of (1) *model size* comparing BERT-large and BERT-base models, (2) *null prompt patterns*, and (3) *contextualization*. In *core* and *additional experiments*, we further perform *class-based evaluations* on two primary classes, which were selected with clinical experts: (1) *Anamnesis* (mostly unstructured) and (2) *Medikation* (semi-structured).

Model size (S_3): We evaluated the impact of adding model parameters, by comparing *gbert-base* (110 million) vs *gbert-large* (340 million) PLMs. We limited this setup to *gbert* PLMs, since a large *medbertde* was not published.

Null prompts (S_2): Logan *et al.* (2022) discovered that the usage of null prompts without manually crafted templates achieve competitive accuracy to manually tuned prompt templates on a wide range of tasks. This is of particular interest in the clinical domain, to further reduce costly engineering efforts.

Adding context (S_2): To introduce further information to the document structure, we added further context to each input sample to evaluate the effect of adding context paragraphs to each sample. We evaluated three types of context (Table 2).

3. Results

3.1 Baselines

We define two baselines to assess model performance in our *core* and *additional experiments*: as *lower bound* we use a *zero-shot prompting* approach; as *upper bound* we use a *fine-tuned sequence classifier* trained on the *full* size of the training corpus. Fig. 4 shows the accuracy results for both baselines. The upper bound results exceed 96% accuracy for both models. The further-pretrained *gbert* models yield a minimal (statistically significant) advance of 0.4–0.6 accuracy points above the original *gbert-base*. For *medbertde*, no such difference is observed.

The zero-shot results are all below 16% accuracy, except for the public *medbert-base* that with 28.3% achieves a great advance over *gbert-base* with 7.2% accuracy. However, the *gbert* models further-pretrained on both task- and domain-specific data more than double the performance of the original model to 15% accuracy, beyond *gbert* pretrained on domain-specific data only (*-domain). All performance differences for *gbert* are statistically significant, except *gbert-base* and *gbert-domain*.

3.2 Core experiments

Fig. 5 presents our *core experiment* results compared to the baselines introduced above.

PET versus SC. The PET model variants significantly outperform SC models at shot sizes ≤ 100 in 31 out of 32 setups when comparing the same pretraining methods. Only SC *medbertde-base-comb* outperforms all PET models with shot size 100.

Few-shot size. Both PET and SC models benefit from an increase in few-shot size. We observed statistical significance at shot sizes ≤ 200 . The smaller the shot size, the greater the relative performance gain of PET over SC models.

Further-pretraining. We observe notably different results for further-pretrained *gbert* and *medbertde* PLMs.

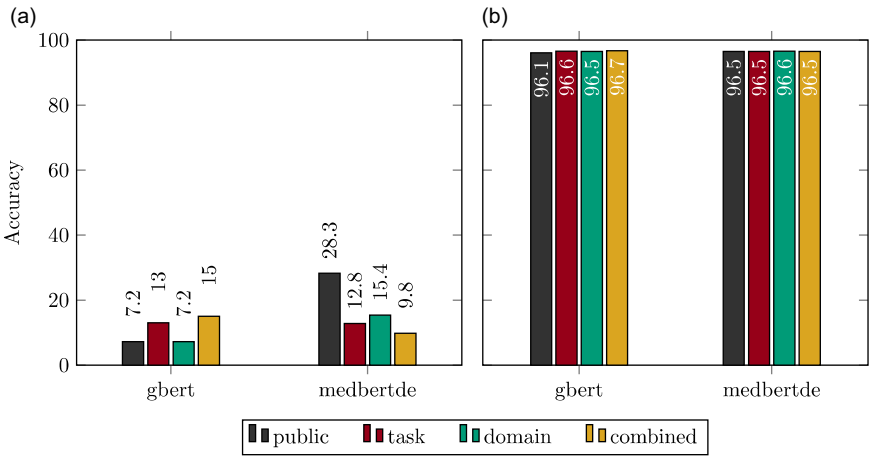


Figure 4. Section classification baseline results (lower/upper bound): We show accuracy scores in percentage per pretraining method (public, task-adapted, domain-adapted, and combination of both) per model: *gbert*-base and *medbertde*-base. (a) Lower-bound: used in zero-shot prompting (b) Upper bound: *full* training set.

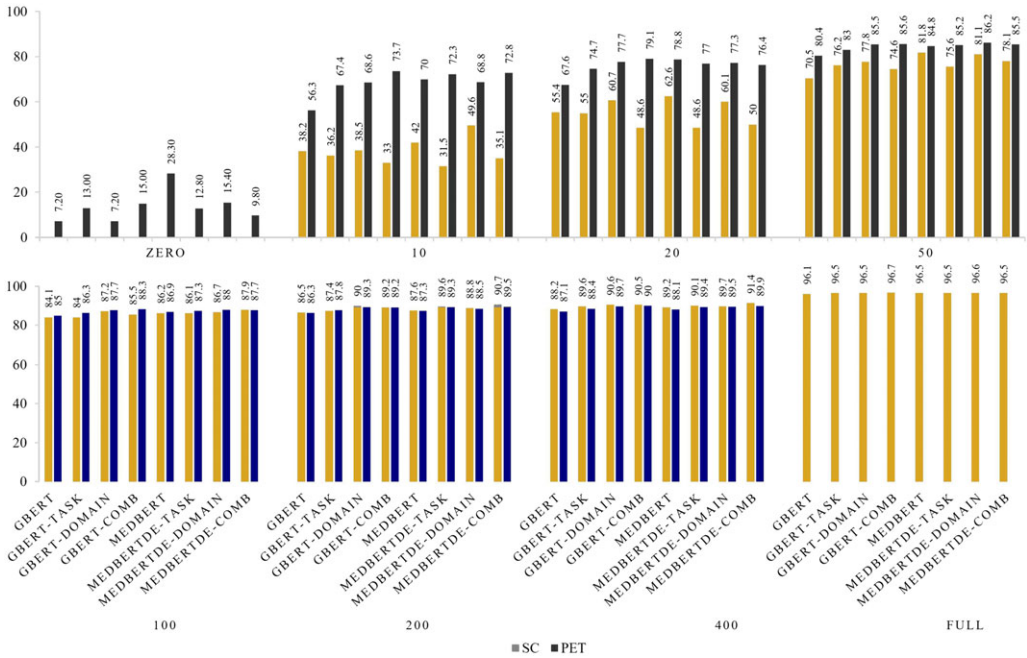


Figure 5. Accuracy scores in percentage for core experiments and lower/upper bound: comparing prompting using PET vs. SC, few-shot sizes 10 – 400 and pretraining methods using base BERT models. For reference, lower-bound PET baselines trained with zero-shots (ZERO) and upper-bound SC models trained on complete training set (FULL).

Gbert. PET models benefit significantly from further-pretraining with ≤ 100 shots. Accuracy gradually increases with task-specific, domain-specific and combined pretraining, in that order. *Gbert* SC models also benefit significantly from domain-adapted models over all shot sizes (except 10 and 400 shots), but not from task adaptation or their combination. Overall, we observe a more consistent effect of further-pretraining for PET models compared to SC models.

Medbertde. Further-pretraining shows no consistent performance improvement for *medbertde* model variants. In particular, with 20 shots, the *medbertde-base* PET model outperforms the further-pretrained models, achieving a statistically significant 79.1% accuracy. For few-shot sizes 10 and 50–400, the best performing model alternates between the *medbertde-base-domain* and *medbertde-base-comb* PET models. Similar to *gbert* models, the relative gain of pretraining decreases with increasing shot sizes. It appears that our pretraining method using cardiovascular doctor's letters has no impact or may even impair the *medbertde* model. A possible reason could be that the public *medbertde* model was only pretrained on 10G of clinical and medical texts, primarily from the oncology domain. However, future research is needed for further investigation (pretraining data information cf. Fig. 3).

Best-performing model variant. According to our core experiments, the overall best-performing model is the *gbert* domain- and task-adapted model (*gbert-base-comb*). This model achieved best accuracy scores with shot sizes ≤ 100 compared to other pretraining methods and to fine-tuned SC models with shot sizes ≤ 400 . When using only 20 shots, this model outperforms the SC model by 30.5 percentage points (pp.) and the public *gbert-base* PET model by 11.5 pp. Hence, we select this model for all *additional experiments*. If not further pretrained *medbertde-base* outperforms public *gbert-base*: this is similar to our baseline experiments. However, further-pretraining does not improve the performance of *medbertde-base*, possibly due to the relatively small pretraining data size of *medbertde-base* (10G).

Robustness. Experiments were performed using three training sets and two initial random seeds. For smaller shot sizes (≤ 50 shots), standard deviation was low ($\sim 2.5\%$) decreasing to less than 1% for larger sizes. We observed this for *gbert* and *medbertde* with no impact of different pretraining methods.

3.2.1 Inspecting primary classes

We investigate the impact of shot size on the accuracy of predicting the selected primary section classes (Fig. 6a). Across shot sizes 20–50, the *F1*-scores of both classes increase in average by 9.2% pp. *Anamnese*, with a lower *F1*-score, benefits more from larger few-shot sizes. However, the SC model trained on the full training set significantly outperforms the 50-shot models. This is especially true for the *Anamnese* class. Even if shot size is increased to our maximum of 400 shots, the results still differ significantly: (*Anamnese*: 82.4%, *Medikation*: 97.5%). Results for more semi-structured classes like *Medikation* are closest to the performance of the full model. For results of all shot sizes cf. Suppl. Fig. S4.

While our primary classes benefit from further-pretraining, *F1*-score of *Anrede* slightly decreased. A possible explanation could be that *Anrede* often contains non-clinical terminology that describes a patient's place of residence, date of birth and name (Suppl. Fig. S5).

3.2.2 Inspecting Shapley values

To better understand model predictions in a few-shot setting, we further analyzed Shapley values of the 20-shot model for the lower-performing class *Anamnese*. We chose a false positive sample as the running example for the remainder of this study because *Anamnese* belongs to our primary classes and often suffers from a low precision rate (for 20-shots, *gbert-base-comb* achieves 44.6% precision and 62.2% recall, cf. Suppl. Fig. S6). Table 6b illustrates selected Shapley values per token for the sample: 'Die Aufnahme der Patientin erfolgte bei akutem Myokardinfarkt -LRB- STEMI -RRB-. (English: The patient was admitted due to an acute myocardial infarction -LRB- STEMI -RRB-.)' toward the classes *Anamnese* and *Zusammenfassung*, respectively.

The model incorrectly classified this sample as *Anamnese*, with 76.8% probability, while the correct class is predicted with 18.2% probability score. Tokens such as *Die* (*the*), *Aufnahme* (*admission*), *Patient* (*patient*), *erfolgte* (*took place*) positively contributed to the *Anamnese* class,

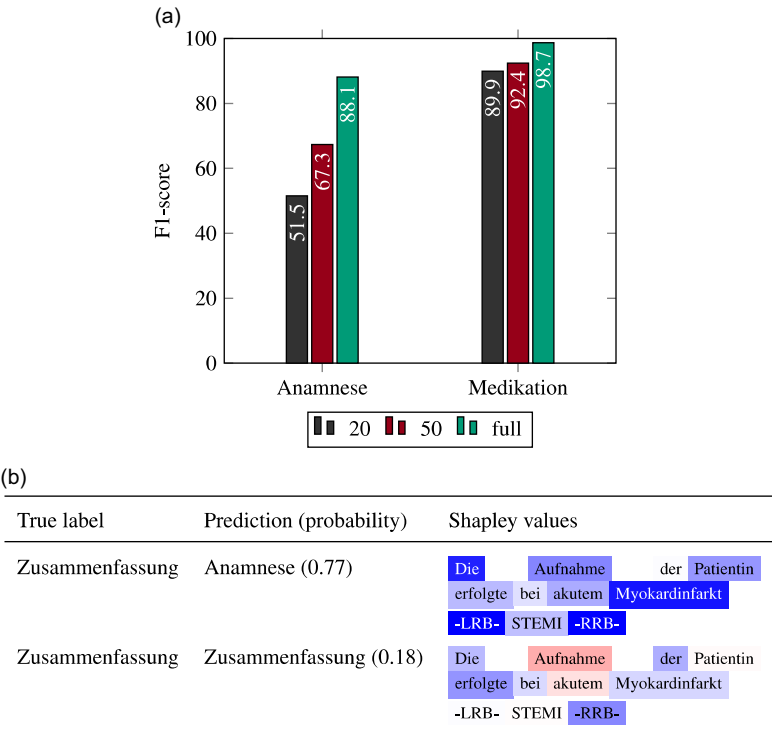


Figure 6. Core experiments: primary class *F1*-score in percentage and selected Shapley values: (a) *F1*-score scores per few-shot sizes for primary classes with using *gbert-base-comb nocontext*. (b) Shapley value analysis for *gbert-base-comb nocontext* with respect to *Anamnese* and *Zusammenfassung* prediction. First column: true label of the sample, second column: predicted label including label probability, third column: selected Shapley values. We used 20 training shots. For readability reasons, we grouped some token sequences. Further details, see Suppl. Fig. S7. Legend: Blue: positive contribution, Red: negative contribution.

while the tokens *Aufnahme* and *Patient* negatively contributed to the correct *Zusammenfassung* class. Analyzing the 20-shot training dataset, we observe that these keywords occur more frequently in samples for *Anamnese* (*Die* (13x), *Aufnahme* (6x), *Patient* (7x), *erfolgte* (8x)) than in samples from *Zusammenfassung* (*Die* (5x), *Aufnahme* (2x), *Patient* (5x), *erfolgte* (6x)). The token *Myokardinfarkt* (acute myocardia) positively contributes to both section classes, and to a higher extent to *Anamnese*, even though we only observe this token in instances from *Zusammenfassung*. The token sequences representing brackets *-LRB-* and *-RRB-* contribute strongly positively to *Anamnese*. Analyzing the training data showed a higher frequency of these tokens in *Anamnese* samples (11x) compared to *Zusammenfassung* (5x).

Note on interpreting Shapley values. Shapley values are additive: they sum up all token contributions along with the base value to yield the prediction probability. Shapley values toward different classes and of different models cannot be compared by absolute value, but only relative to other tokens for the same prediction and the same model.

3.3 Additional experiments

3.3.1 Model size

Given the limited computational resources in clinical infrastructures, we investigated how model size affects performance and investigate its impact with finer-grained analyses. Since there is no *medbertde-large* model available, we compared *gbert-large* and *gbert-base* models.

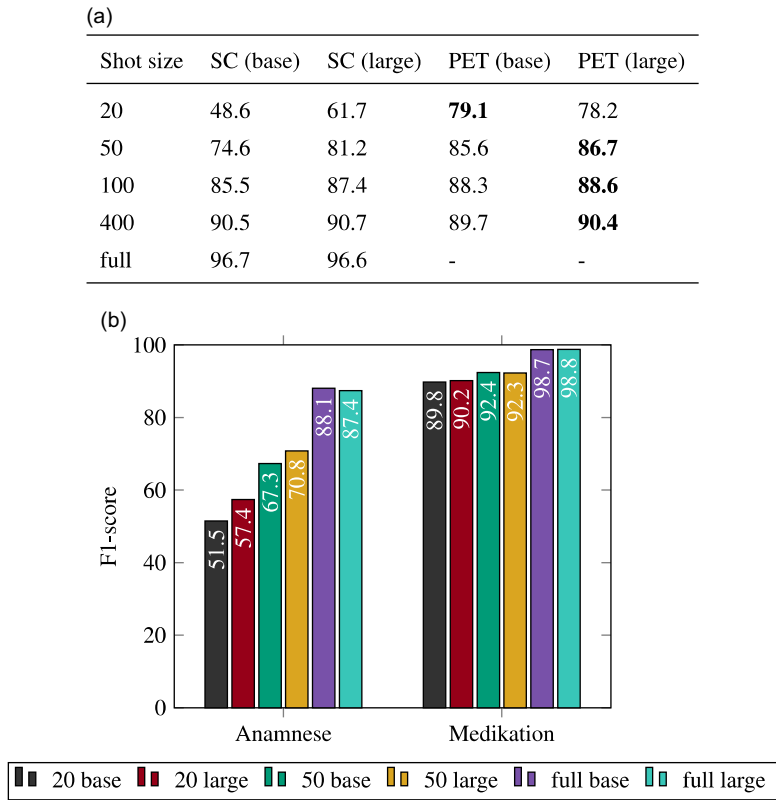


Figure 7. Model size: (a) Accuracy scores in percentage for *gbert-comb nocontext* PLMs using all templates on four few-shot sizes. (b) *F1*-scores in percentage for primary classes for *gbert-comb no context* PLMs using all templates on various few-shot sizes.

Larger model size increases accuracy significantly, by an average of 7.2 pp. for SC models ≤ 100 . PET models, by contrast, benefit less from larger model size than SC models. We even observe a slight performance decrease for shot size 20 (Table 7a). The only significant increase, of 1.1 points accuracy, we observed for shot size 50.

Primary classes: *Gbert-large* yields an increased *F1*-score for *Anamnese* with both shot sizes (20, 50), by an average of +4.7 pp. but this is only significant for shot size 50. By contrast, the difference in *F1*-score (0.1% – 0.4%) for *Medikation* is not statistically significant (Fig. 7b).

Shapley values: Both models, *gbert-base-comb* and *gbert-large-comb* incorrectly classify our running example belonging to *Zusammenfassung* as *Anamnese*. We do not observe significant differences in the respective token contributions (Suppl. Fig. S8).

3.3.2 Null prompts

Inspired by insights of Logan *et al.* (2022) – who removed all tokens from prompt templates, using null prompts instead, with comparable classification results – we evaluated the *gbert-base-comb* model using only three null prompt templates (cf. Section 2.1.1).

Null prompts slightly decrease accuracy scores for shot sizes ≤ 50 by approximately one percentage point. For shot sizes 100 and 400, we note a slight accuracy increase. We only observed statistically significant differences in accuracy for shot-size 50 (template-based model: 85.6%, null-prompt model: 84.6%) (Suppl. Tab. S3).

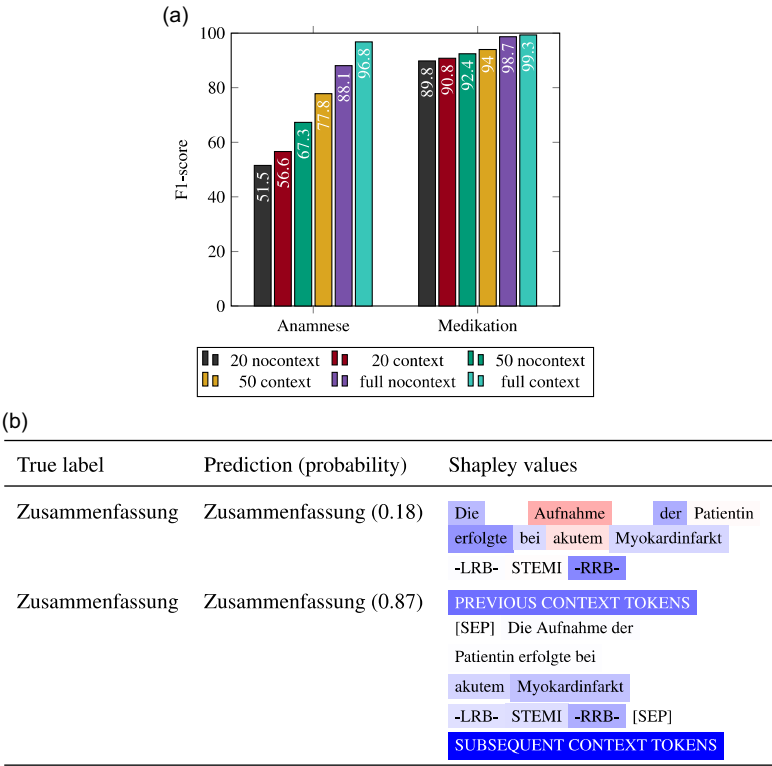


Figure 8. Additional experiments (context) – primary classes *F1*-scores and selected Shapley values: (a) *F1*-scores in percentage per few-shot sizes for primary classes with *nocontext* and *context* using *gbert-base-comb*. Comparing to *gbert-base-comb* trained on full training data with *nocontext* and *context*. (b) Shapley value analysis for *gbert-base-comb nocontext* and *gbert-base-comb context*. First column: true label of the sample, second column: predicted label including label probability, third column: selected Shapley values. We used 20 training shots. For readability reasons, we grouped some token sequences. Further details, see Suppl. Fig. S10. Legend: Blue: positive contribution, Red: negative contribution.

Primary classes: For our primary classes, we did not observe a consistent pattern. Null prompts have a slightly negative impact on *F1*-scores for *Anamnese* and *Medikation* with 20 shots. By contrast, with 50 shots, accuracy significantly decreases for *Anamnese*, but slightly increases for *Medikation* (92.4% vs. 95.9%).

3.3.3 Adding context

Predicting section classes is difficult for tokens that frequently occur in different classes, as discussed for the example in Fig. 6. To reduce the degree of ambiguity of individual tokens, we experimented with two types of *contextualization* of classification instances: Adding (1) the previous and subsequent paragraph (*context*) and (2) only the previous paragraph (*prevcontext*). Suppl. Fig. S9 shows that across all few-shot sizes, (1) *context* (with mean +2.4 accuracy points) and (2) *prevcontext* (with mean +1.6 accuracy points) both achieve significantly higher accuracy than *nocontext* models (cf. Section 2.5.4).

Primary classes: *Context* models improve the *F1*-scores for both primary classes (by mean +7.8 points for *Anamnese* and +1.3 for *Medikation*) (Fig. 8a). For *Anamnese*, statistically significant improvement is only reached using 50 shots.

Shapley values: *gbert-base-comb context* correctly classifies our running example with 86.6% probability (Table 8b). Most highly contributing tokens belong to the context (previous or following, with Shapley values: 0.057 + 0.596), while the main paragraph has an accumulated Shapley

Table 3. Combining and evaluating best-performing methods: Accuracy scores in percentage for *gbert-large-comb context* evaluated on few-shot sizes [20, 50, 100, 400] with *base* vs. *large* model sizes in *context* vs. *nocontext* settings using PET. Comparison to corresponding SC model fine-tuned on full training set.

Shot size	Base <i>nocontext</i>	Large <i>nocontext</i>	Base <i>context</i>	Large <i>context</i>
20	79.1	78.2	80.5	84.3
50	85.6	86.7	89.2	89.4
100	88.3	88.6	90.9	91.3
400	90	90.4	92.8	93.4
full (SC)	96.7	96.6	98.6	98.6

value of 0.106. The previous context contains the sequence: *Zusammenfassende Beurteilung*, a frequent section-specific title. The subsequent paragraph is the longest paragraph (37 tokens). Previously negatively contributing tokens (*Aufnahme* and *Patient*) are now positively contributing to the correct class: *Zusammenfassung*.

3.3.4 Combining best-performing methods

Our *core experiments* indicated that the *gbert-base-comb* model performed best of all tested models. The *additional experiments* showed that models using all five templates (cf. Section 2.1.1), a BERT-large architecture and contextualization often achieved the best performance. Hence, we investigated whether this combination (*gbert-large-comb context* trained with all templates) could further close the performance gap to a model trained on full training set.

Table 3 shows that *gbert-large-comb context* significantly outperforms both *gbert-base-comb* and *gbert-large-comb* without context. Moreover, *gbert-large-comb context* statistically significantly outperforms *gbert-base-comb context* for 20, 100 and 400 shots. Overall, the *gbert-large-comb context* outperforms *nocontext* and *base* models over all shot-sizes, yielding best results with 400 shots. Yet, PET still lags behind the *full* SC setting, with a minimal gap of −5.2 points accuracy.

Primary classes: For our primary classes, *gbert-large-comb context* now outperforms *gbert-large-comb nocontext* by large margin (Fig. 9a). Only the 50-shot results for *Anamnese* are not statistically significant (*F1*-score of all shot-sizes cf. Suppl. Fig. S11).

We also compared the *large* and *base* versions of *gbert^{*}-comb context*. The *F1*-score for *Anamnese* is significantly increased by +14.6 points with 20 shots and by +2.6 points with 50 shots. Performance for *Medikation* is significantly increased by +2.2 points with 20 shots, but insignificantly decreased with 50 shots. (Suppl. Fig. S12)

Shapley values: We tested whether the token contributions differ between the *large* and *base gbert^{*}-comb context* models (Table 9b). The large model predicts the true class *Zusammenfassung* with 99.2% probability, +12.7 points above the *base* model. The *large context* model now also places greater emphasis on the main paragraph, as opposed to the context. The ratio of the accumulated Shapley values ($\frac{\text{classified instance}}{\text{context paragraphs}}$, higher is better) is 0.36 for *gbert-large-comb context* and 0.16 for *gbert-base-comb context*.

4. Discussion

In this section, we discuss our empirical findings in light of the challenges and proposed solutions outlined in Section 1.

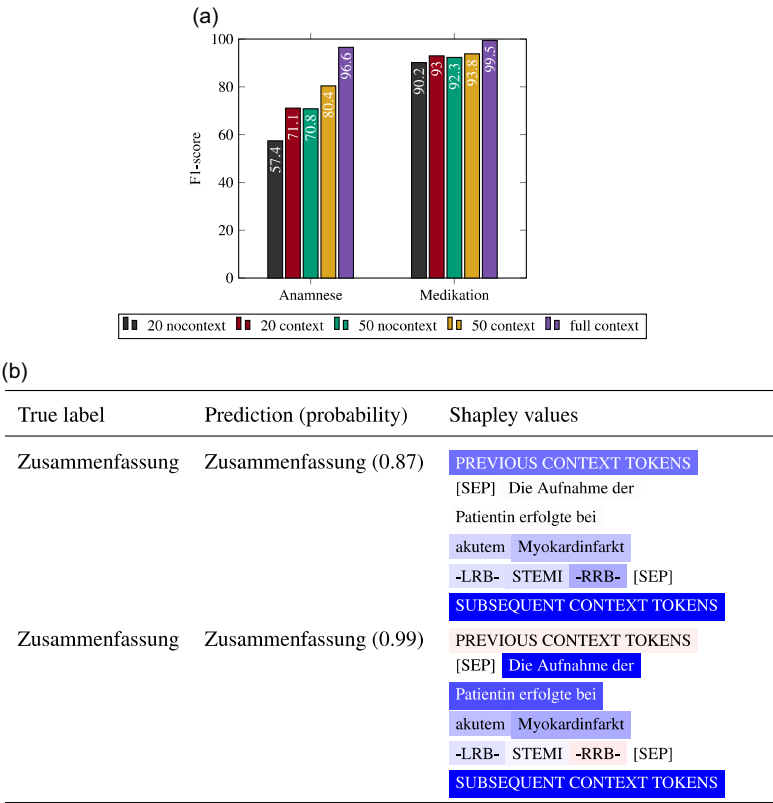


Figure 9. Additional experiments (combined methods) – primary classes *F1*-scores and selected Shapley values: (a) *F1*-scores in percentage per few-shot sizes for primary classes with *nocontext* and *context* using *gbert-large-comb*. Comparing to *gbert-large-comb* trained on full training data with *context*. (b) Shapley value analysis for *gbert-base-comb context* and *gbert-large-comb context*. First column: true label of the sample, second column: predicted label including label probability, third column: selected Shapley values. We used 20 training shots. For readability reasons, we grouped some token sequences. More detailed results, see Suppl. Fig. S13. Legend: Blue: positive contribution, Red: negative contribution.

S₁ Domain- and Expert-dependent. In in-depth evaluations, we compared four pretraining approaches using PET and SC for two public German-language models (Gururangan et al. 2020): (1) *initial pretraining* using general German texts with *gbert* versus exclusively medical and clinical data with *medbertde* (Fig. 3); and *further-pretraining* of these PLMs for (2) *task-adaption*, (3) *domain-adaptation*, and (4) combined *task and domain-adaptation*. **Finding.** *Gbert* overall accuracy gradually improved with further-pretraining. The task- and domain-adapted *gbert-base-comb* performs best compared to all models, and with only 20 shots outperforms *gbert-base* by +11.5 accuracy points. Also, the positive effect of further-pretraining was more consistent for PET compared to SC models. By contrast, further-pretrained *medbertde*-based SC and PET models did not achieve consistent performance improvements. **Finding.** Pretraining from scratch with sufficient clinical and medical data can benefit various MIE tasks. However, when pretraining data limited and/or concentrated on a narrow domain, for example oncology, as in the case of *medbertde*, further-pretraining was found not to enhance performance. **Finding.** While *medbertde-base* without further-pretraining outperformed *gbert-base* in all shot sizes, and similarly when trained on the *full* dataset (Fig. 5), it did not improve performance if further pretrained and was outperformed by further-pretrained *gbert-base*.

- S₂ **Resource-constraints.** Prompt-based fine-tuning with PET produces superior classification results in few-shot learning scenarios. **Finding.** We observed a steady increase in the performance of PET compared to SC models With decreasing few-shot training set sizes (400-10 shots). Using 20 shots, the PET *gbert-base-comb nocontext* model outperforms the corresponding SC model by +30.5 pp. The same *gbert-base-comb nocontext* PET model with 50 shots even rivals the SC model trained on *full* data, leaving a gap of −11.1 pp. Especially semi-structured section classes, such as *Medikation*, perform close to the full model by −6.3 pp. (Fig. 6a). Our few-shot models are also *robust* as measured by standard deviation. **Finding.** *Null prompts* exhibit comparable results with no significant difference in performance, especially with few-shot sizes exceeding 100. **Finding.** Contextualize data with surrounding *context* paragraphs improved classification results for most section classes, especially primary classes. It allowed our base models to correctly predict our running false-positive sample as *Zusammenfassung*. However, compared to the base models interpretability analysis using SHAP revealed that the large model places greater emphasis on main paragraph tokens rather than on context paragraphs. Contextualization further reduced the accuracy gap between *gbert*-*-*comb context*-based PET models trained on 50 shots to the *full* SC model to −9 to −9.5pp; for classes such as *Medikation* even to −5 to −6pp. Contextualization does not require complex preprocessing or manual annotation.
- S₃ **On-premise:** Using smaller models saves computational resources. We therefore compared classification performances of *base* and *large* BERT PLMs. **Finding.** Large PLMs achieve better classification results. However, model size has a lower impact on the performance of PET compared to SC models (Fig. 7a). For classes such as *Medikation* the further-pretrained *gbert-base-comb* PLM performs almost on par with *gbert-large-comb* (Fig. 7b) **Finding.** For complex sections with free text such as *Anamnese*, *gbert-large* PLMs achieved better performance. They also better recognize contextualized instances (Table 9b and Suppl. Sect. S1.2).
- S₄ **Transparency.** Shapley values (Lundberg *et al.* 2017) an interpretability method based on saliency features and helped identify problems in *training data quality* and *model decisions*. We identified tokens that frequently occur in false-positive classes by analyzing model predictions (Fig. 8). **Finding.** The use of Shapley features is especially beneficial in few-shot scenarios, as it enables data engineers to select few-shot samples with high precision. Shapley values also proved instrumental for identifying problems with contextualization: It became clear that with very small shot sizes, and for section classes with short spans, the model prioritized the context over the instance to be classified. They also provided evidence that our *gbert-large-comb* model outperforms its base counterpart by focusing on key parts of contextualized samples. **Finding.** Our analysis of Shapley values showed that *gbert-large-comb* makes more reliable predictions than *gbert-base-comb*, by prioritizing features of instances to be classified over context (Table 9b).

5. Conclusions and recommendations

In this work, we have presented best-practice strategies to identify an ideal setup to address the multifaceted challenges of conducting a MIE task, such as clinical section classification, in a lower-resource domain and language such as the German clinical domain. In summary, our best-performing setup used a task- and domain-adapted BERT-large architecture trained with PET on contextualized samples using all five template types.

To reduce the demand for clinical knowledge in MIE we showed in S₁ that few-shot prompting performed particularly well with further-pretrained general-domain PLMs and helped to reduce the demand of clinical expert knowledge for manual data annotation. Our experiments revealed that pretraining data have a strong impact on few-shot learning results (see S₂), especially if

training data are limited. Specifically, *general domain PLMs* such as *gbert*, pretrained on massive amounts of general language, can be effectively domain- and task-adapted by *further-pretraining* on clinical routine data. In contrast, PLMs pretrained on domain-specific data from scratch, such as *medbertde* may outperform *gbert* if not further-pretrained, but may not benefit from further-pretraining. Therefore, if further-pretraining for domain adaptation is not feasible due to IT constraints, we recommend choosing clinical PLMs like *medbertde* over non-adapted general PLMs.

Our study indicated that prompt-based learning methods improve classification results if annotated data are rare, and effectively reduces time investment and costs of manual data annotation. The larger the amount of annotated data, the higher the efficiency of *null prompts*, which further save engineering time (see *S₂*). Moreover, contextualizing classification instances improves performance, especially for the primary classes, and further closes the gap to *full* models.

We found in *S₃* that in case of limited computing resources, prompting methods allow practitioners to employ smaller PLMs in a few-shot scenario, while achieving classification results comparable to larger models. However, free-text sections, such as *Anamnese*, may still benefit from larger model architectures (Fig. 7).

Finally, in *S₄*, we addressed the need for transparent and trustworthy model predictions in low-resource German clinical NLP, and possible use cases for *interpretability* methods. Our study demonstrates that the analysis of Shapley values can help improve training data quality, which is especially important with small shot sizes. Examining Shapley values, or similar interpretability methods, can also inform model selection, by revealing tokens that contribute to classification errors in specific model types. Finally, model interpretability is crucial in safety-critical domains such as clinical routine, to enhance the trustworthiness of model predictions.

Our study presents strategies and best-practice approaches for optimizing MIE in lower-resource clinical language settings. It highlights the benefits of few-shot prompting with further-pretrained PLMs as a measure to reduce the demand for manual annotation by clinicians. We further demonstrate that prompt-based learning and contextualization significantly enhance classification accuracy, especially in low-resource scenarios, while keeping demands on computing resources low. We are certain that these insights help to advance MIE tasks in clinical settings in the context of lower-resource languages such as German.

6. Declarations

6.1 Ethics approval and consent to participate

The authors state that this study complies with the Declaration of Helsinki. Our task has been performed with respect to Section 46 Abs.2 Nr.2a (LKHG) and Section 13 Abs.1 Landesdatenschutzgesetz BW. In this context, we had the possibility to use the data for the purpose of optimizing internal clinical procedures.

6.2 Availability of data and materials

We used CARDIO:DE, a distributable German corpus containing 500 cardiovascular doctor's letters from the clinical routine, for all our experiments (available with a signed DUA: <https://doi.org/10.11588/data/AFYQDY>). Annotations of the held-out datasets are not publicly available as authors of CARDIO:DE use it for shared task competitions. But they are available from the corresponding author on reasonable request. For more details about the dataset, preprocessing steps, data annotation, and data distribution, cf: Richter-Pechanski *et al.* (2023).

Supplementary material. The supplementary material for this article can be found at <https://doi.org/10.1017/nlp.2024.52>.

Competing interests. The author(s) declare none.

References

- Attanasio G., *et al.* (2023). ferret: a framework for benchmarking explainers on transformers. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, pp. 256–266.
- Borchert F., *et al.* (2022). Ggponc 2.0-the german clinical guideline corpus for oncology: curation workflow, annotation policy, baseline ner taggers. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pp. 3650–3660.
- Bressem K.K., *et al.* (2024). medbert.de: a comprehensive german bert model for the medical domain. *Expert Systems with Applications* **237**, 121598.
- Brown T., *et al.* (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems* **33**, pp. 1877–1901.
- Buitinck L., *et al.* (2013). API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122.
- Chan B., Schweter S. and Möller T. (2020). German’s next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain (Online). International Committee on Computational Linguistics, pp. 6788–6796.
- Chang Y., *et al.* (2024). A survey on evaluation of large language models.
- Chowdhery A., *et al.* (2023). PaLM: scaling language modeling with pathways. *Journal of Machine Learning Research* **24**(240), 1–113.
- Clusmann J., *et al.* (2023). The future landscape of large language models in medicine. *Communications Medicine* **3**(1), 141.
- Denny J.C., *et al.* (2008). Development and evaluation of a clinical note section header terminology. In *AMIA Annual Symposium Proceedings*. American Medical Informatics Association, vol. **2008**, p. 156.
- Devlin J., *et al.* (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota. Association for Computational Linguistics, pp. 4171–4186.
- Edinger T., *et al.* (2017). Evaluation of clinical text segmentation to facilitate cohort retrieval. In *AMIA Annual Symposium Proceedings*. American Medical Informatics Association, vol. **2017**, p. 660.
- Fan F.-L., *et al.* (2021). On interpretability of artificial neural networks: a survey. *IEEE Transactions on Radiation and Plasma Medical Sciences* **5**(6), 741–760.
- Gao T., Fisch A. and Chen D. (2021). Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Online. Association for Computational Linguistics, pp. 3816–3830.
- Ge Y., *et al.* (2023). Few-shot learning for medical text: a review of advances, trends, and opportunities. *Journal of Biomedical Informatics* **144**, 104458.
- Guo Z., *et al.* (2023). Evaluating large language models: a comprehensive survey.
- Gururangan S., *et al.* (2020). Don’t stop pretraining: adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics, pp. 8342–8360.
- Hahn U. and Oleynik M. (2020). Medical information extraction in the age of deep learning. *Yearbook of Medical Informatics* **29**(01), 208–220.
- Idrissi-Yaghir A., *et al.* (2024). Comprehensive study on German language models for clinical and biomedical text understanding. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, Torino, Italia. ELRA and ICCL, pp. 3654–3665.
- Jacovi A. and Goldberg Y. (2020). Towards faithfully interpretable NLP systems: how should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics, pp. 4198–4205.
- Jantscher M., *et al.* (2023). Information extraction from german radiological reports for general clinical text and language understanding. *Scientific Reports* **13**(1), 2353.
- Jiang A.Q., *et al.* (2023). Mistral 7b.
- Kojima T., *et al.* (2022). Large language models are zero-shot reasoners. In Koyejo S., *et al.* (ed), *Advances in Neural Information Processing Systems* **35**. Curran Associates, Inc., pp. 22199–22213.
- Lake B.M., Salakhutdinov R. and Tenenbaum J.B. (2015). Human-level concept learning through probabilistic program induction. *Science* **350**(6266), 1332–1338.
- Landolsi M.Y., Hlaoua L. and Ben Romdhane L. (2023). Information extraction from electronic medical documents: state of the art and future research directions. *Knowledge and Information Systems* **65**(2), 463–516.
- Leaman R., Khare R. and Lu Z. (2015). Challenges in clinical natural language processing for automated disorder normalization. *Journal of Biomedical Informatics* **57**, 28–37.
- Lee J., *et al.* (2020). Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* **36**(4), 1234–1240.

- Li Y., et al. (2023). A comparative study of pretrained language models for long clinical text. *Journal of the American Medical Informatics Association* **30**(2), 340–347.
- Liu P., et al. (2023). Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing. *ACM Computing Surveys* **55**(9), 1–35.
- Logan R.IV, et al. (2022). Cutting down on prompts and parameters: simple few-shot learning with language models. In Muresan S., Nakov P. and Villavicencio A. (eds), *Findings of the Association for Computational Linguistics: ACL, Dublin, Ireland*. Association for Computational Linguistics, pp. 2824–2835.
- Lohr C., Eder E. and Hahn U. (2021). Pseudonymization of phi items in german clinical reports. In *Public Health and Informatics*. IOS Press, pp. 273–277
- Lohr C., et al. (2018). Cda-compliant section annotation of german-language discharge summaries: guideline development, annotation campaign, section classification. In *AMIA Annual Symposium Proceedings*. American Medical Informatics Association, vol. **2018**, p. 770.
- Lundberg S.M. and Lee S.-I. (2017). A unified approach to interpreting model predictions. In Guyon I., et al. (ed), *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., pp. 4765–4774
- Moradi M., et al. (2021). GPT-3 models are poor few-shot learners in the biomedical domain. arXiv preprint arXiv:2109.02555.
- Mosca E., et al. (2022). Shap-based explanation methods: a review for nlp interpretability. In *Proceedings of the 29th International Conference on Computational Linguistics*, pp. 4593–4603.
- Parcalabescu L. and Frank A. (2024). On measuring faithfulness or self-consistency of natural language explanations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*.
- Parnami A. and Lee M. (2022). Learning from few examples: a summary of approaches to few-shot learning. arXiv preprint arXiv:2203.04291.
- Peng C., et al. (2023). A study of generative large language model for medical research and healthcare. *npj Digital Medicine* **6**(1), 210.
- Pomares-Quimbaya A., Kreuzthaler M. and Schulz S. (2019). Current approaches to identify sections within clinical narratives from electronic health records: a systematic review. *BMC Medical Research Methodology* **19**(1), 1–20.
- Reynolds L. and McDonell K. (2021). Prompt programming for large language models: beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–7.
- Ribeiro M. T., Singh S. and Guestrin C. (2016). why should i trust you? explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144.
- Richter-Pechanski P., et al. (2019). Deep learning approaches outperform conventional strategies in de-identification of german medical reports. In *GMDS*, pp. 101–109
- Richter-Pechanski P. and Dieterich C. (2023). CARDIO: DE [V1.01].
- Richter-Pechanski P., et al. (2021). Automatic extraction of 12 cardiovascular concepts from german discharge letters using pre-trained language models. *Digital Health* **7**, 20552076211057662.
- Richter-Pechanski P., et al. (2023). A distributable german clinical corpus containing cardiovascular clinical routine doctor's letters. *Scientific Data* **10**(1), 207.
- Scao T.L., et al. (2022). BLOOM: a 176b-parameter open-access multilingual language model. ArXiv, abs/2211.05100.
- Schick T., Schmid H. and Schütze H. (2020). Automatically identifying words that can serve as labels for few-shot text classification. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, pp. 5569–5578.
- Schick T. and Schütze H. (2021a). Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Online. Association for Computational Linguistics, pp. 255–269.
- Schick T. and Schütze H. (2021b). It's not just size that matters: small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online. Association for Computational Linguistics, pp. 2339–2352.
- Schick T. and Schütze H. (2022). True few-shot learning with prompts—A real-world perspective. *Transactions of the Association for Computational Linguistics* **10**, 716–731.
- Shapley L.S., et al. (1953). A value for n-person games. In *Contributions to the Theory of Games II*
- Shin T., et al. (2020). AutoPrompt: eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online. Association for Computational Linguistics, pp. 4222–4235.
- Singhal K., et al. (2023). Large language models encode clinical knowledge. *Nature* **620**(7972), 172–180.
- Sivarajkumar S. and Wang Y. (2022). Healthprompt: a zero-shot learning paradigm for clinical natural language processing. In *AMIA Annual Symposium Proceedings*. American Medical Informatics Association, vol **2022**, p. 972.
- Sun C., et al. (2019). How to fine-tune bert for text classification? In *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Proceedings 18*, October 18–20, 2019, Kunming, China. Springer, pp. 194–206.

- Sundararajan M., Taly A. and Yan Q.** (2017). Axiomatic attribution for deep networks. In *International Conference on Machine Learning*. PMLR, pp. 3319–3328.
- Taylor N., et al.** (2023). Clinical prompt learning with frozen language models. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–11. <https://ieeexplore.ieee.org/abstract/document/10215061>.
- Thirunavukarasu A.J., et al.** (2023). Large language models in medicine. *Nature Medicine* **29**(8), 1930–1940.
- Tjoa E. and Guan C.** (2020). A survey on explainable artificial intelligence (xai): toward medical xai. *IEEE Transactions on Neural Networks and Learning Systems* **32**(11), 4793–4813.
- Touvron H., et al.** (2023). Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.
- Uzuner Ö., Solti I. and Cadag E.** (2010). Extracting medication information from clinical text. *Journal of the American Medical Informatics Association* **17**(5), 514–518.
- Wang J., et al.** (2022). Towards unified prompt tuning for few-shot text classification. In Goldberg Y. Kozareva Z. and Zhang Y. (eds), *Findings of the Association for Computational Linguistics: EMNLP, Abu Dhabi, United Arab Emirates*. Association for Computational Linguistics, pp. 524–536.
- Wang Y., Zhao Y. and Petzold L.** (2023). Are large language models ready for healthcare? a comparative study on clinical language understanding. In *Proceedings of the 8th Machine Learning for Healthcare Conference*. PMLR, vol. **219**, pp. 804–823, Proceedings of Machine Learning Research.
- Wu C.** (2023). Pmc-llama: further finetuning llama on medical papers. arXiv preprint arXiv:2304.14454.
- Wu S., et al.** (2020). Deep learning in clinical natural language processing: a methodical review. *Journal of the American Medical Informatics Association* **27**(3), 457–470.
- Yeh A.** (2000). More accurate tests for the statistical significance of result differences. In *COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics*.
- Zhu Q., et al.** (2021). When does further pre-training MLM help? an empirical study on task-oriented dialog pre-training. In *Proceedings of the Second Workshop on Insights from Negative Results in NLP*, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics, pp. 54–61.