Detecting Omissions in LLM-Generated Medical Summaries

Achir Oukelmoun[1], Nasredine Semmar[2], Gaël de Chalendar[2], Clément Cormi[4], Mariame Oukelmoun[3], Eric Vibert[4], Marc-Antoine Allard[4]

[1]Université Paris-Saclay, 3 Rue Joliot Curie, 91190 Gif-sur-Yvette, France
[2]CEA List, 2 Boulevard Thomas Gobert, 91120 Palaiseau, France
[3]Faculté de médecine et de pharmacie de Rabat, Impasse Souissi, Rabat 10100, Maroc
[4]Hôpital Paul-Brousse AP-HP, 12 Avenue Paul Vaillant Couturier, 94800 Villejuif, France
achir.oukelmoun@outlook.fr, nasredine.semmar@cea.fr, gael.de-chalendar@cea.fr,
clement.cormi-ext@aphp.fr, oukelmoun@gmail.com, eric.vibert@aphp.fr, marcantoine.allard@aphp.fr

Abstract

With the emergence of Large Language Models (LLMs), numerous use cases have arisen in the medical field, particularly in generating summaries for consultation transcriptions and extensive medical reports. A major concern is that these summaries may omit critical information from the original input, potentially jeopardizing the decision-making process. This issue of omission is distinct from hallucination, which involves generating incorrect or fabricated facts. To address omissions, this paper introduces a dataset designed to evaluate such issues and proposes a frugal approach called EmbedKDECheck for detecting omissions in LLM-generated texts. The dataset, created in French, has been validated by medical experts to ensure it accurately represents real-world scenarios in the medical field. The objective is to develop a reference-free (black-box) method that can evaluate the reliability of summaries or reports without requiring significant computational resources, relying only on input and output. Unlike methods that rely on embeddings derived from the LLM itself, our approach uses embeddings generated by a third-party, lightweight NLP model based on a combination of FastText and Word2Vec. These embeddings are then combined with anomaly detection models to identify omissions effectively, making the method well-suited for resource-constrained environments. EmbedKDECheck was benchmarked against black-box state-of-the-art frameworks and models, including SelfCheckGPT, ChainPoll, and G-Eval, which leverage GPT. Results demonstrated its satisfactory performance in detecting omissions in LLM-generated summaries. This work advances frugal methodologies for evaluating the reliability of LLM-generated texts, with significant potential to improve the safety and accuracy of medical decision support systems in surgery and other healthcare domains.

1 Introduction

The advent of Large Language Models (LLMs) such as OpenAI's GPT (Achiam et al., 2023) has revolutionized various fields by enabling generation of coherent and contextually relevant text. These models have found applications ranging from conversational agents to creation of detailed textual summaries and reports. In the medical domain, LLMs have shown particular promise in generating comprehensive reports that assist healthcare professionals in making informed decisions (Alberts et al., 2023).

Despite these advancements, a significant challenge remains unaddressed: the detection of omissions in LLM-generated texts. Existing datasets and evaluation frameworks predominantly focus on hallucinations—instances where the generated text includes incorrect or fabricated information (Li et al., 2024). While hallucination detection is crucial, the issue of omissions, where critical information from the original input is missing, poses a unique and severe risk, especially in the medical field. Omissions can lead to incomplete medical records, potentially jeopardizing patient care and treatment outcomes.

In the context of the French healthcare sector, the use of external providers or public cloud solutions is often impractical due to stringent privacy regulations (AP-HP, 2024). Consequently, LLM implementations typically rely on smaller models hosted on-premise or in hybrid architectures utilizing API microservices (Nabla, 2024). These constraints can exacerbate the problem of omissions, as access to the most advanced LLMs is limited. Therefore, there is a pressing need for quality control mechanisms that can operate efficiently within these resource-constrained environments.

This paper makes two primary contributions. First, it introduces a novel dataset specifically designed to evaluate omission detection methods.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 325–337
November 4-9, 2025 ©2025 Association for Computational Linguistics

This data-set, which is in French, has been meticulously developed and validated by medical experts, including surgeons specializing in hepatic and general surgery. The validation by medical professionals ensures that the dataset accurately reflects real-world scenarios encountered in the healthcare sector, making it a valuable resource for developing and testing omission detection algorithms.

Second, the paper proposes a frugal, LLM-agnostic approach called EmbedKDECheck to detect omissions in generated texts. This method leverages embeddings in conjunction with anomaly detection models to identify missing information. The emphasis on frugality is crucial, as it allows the approach to function effectively within resource-constrained environments without imposing significant computational demands. This is particularly important in the medical field, where computational resources may be limited, and the timely detection of omissions is critical for ensuring the accuracy and reliability of medical documentation.

To clarify, we define "omission" as a situation where the generated text lacks important information or fails to include necessary details that are expected based on the input. This can lead to incomplete or misleading summaries, which are particularly detrimental in medical documentation.

In this study, we present algorithms that combine embeddings with anomaly detection techniques to identify omissions. We rigorously evaluate these algorithms using the newly proposed dataset, allowing us to assess the effectiveness of our approach in detecting omissions specific to the medical domain. By addressing the critical issue of omissions, this work contributes to the development of robust quality control mechanisms for LLM-generated texts, ultimately enhancing the safety and accuracy of medical decision support systems.

2 Related Works

Evaluating LLM-generated summaries is crucial, especially for omissions. Automated metrics help assess fluency, coherence, relevance, and factual consistency (van Schaik and Pugh, 2024). Reference-free metrics like BLANC (Vasilyev et al., 2020) and SUPERT (Gao et al., 2020) are particularly relevant as they evaluate summaries without requiring reference texts.

Evaluation methods fall into three categories: Black-box methods analyze outputs without accessing internal model states. SelfCheckGPT (Manakul et al., 2023) detects hallucinations by comparing sampled responses. White-box methods require full model access to analyze weights and activations (Azaria and Mitchell, 2023). Grey-box methods use partial access, such as token-level probabilities. Our method, EmbedKDECheck, is a black-box approach that combines embeddings with anomaly detection to detect omissions.

Recent advances in black-box evaluation offer insights for omission detection. ChainPoll (Friel and Sanyal, 2023) outperforms SelfCheckGPT (Manakul et al., 2023) and GPTScore (Fu et al., 2024) on the RealHall (Friel and Sanyal, 2023) benchmark, which closely reflects real LLM usage. G-Eval (Liu et al., 2023) integrates chain-of-thought reasoning (Wei et al., 2022) to enhance summarization quality assessment.

Existing datasets primarily target hallucinations rather than omissions. QAGS (Wang et al., 2020) focuses on factual consistency but not missing content. DROP (Dua et al., 2019) emphasizes discrete reasoning. SummEval (Fabbri et al., 2021) assesses summary coherence and relevance but lacks omission detection. RealHall (Friel and Sanyal, 2023) benchmarks hallucination detection but does not address missing content. These datasets often focus on detecting hallucinations or factual inconsistencies, leaving a gap in the evaluation of omission detection. Our proposed dataset specifically addresses this gap by providing a framework for evaluating omission detection in LLM-generated summaries. Unlike other datasets, ours is tailored to the medical field and has been meticulously constructed to represent actual conditions encountered in medical practice. This ensures that the dataset is highly relevant and practical for real-world applications in the medical domain. Furthermore, the dataset has been validated by medical experts, including surgeons specializing in hepatic and general surgery, to ensure its accuracy and reliability. This level of expert validation, to the best of our knowledge, is not present in other datasets, making our dataset uniquely suited for evaluating omission detection in medical contexts in French.

In summary, the evaluation of LLM-generated summaries is a multifaceted challenge that requires a combination of reference-based, reference-free, and LLM-based metrics. Our approach, which belongs to reference-free metrics, focuses on factual consistency evaluation and seeks to overcome the shortcomings of other methods by offering a clearer and more thorough assessment of omissions in sum-

maries. The proposed method, EmbedKDECheck, is a black-box approach that leverages embeddings combined with anomaly detection models to detect omissions, ensuring reliability without significant computational demands. By using a dataset specifically designed for the medical field and validated by medical experts, our approach provides a robust and practical solution for improving the quality and reliability of LLM-generated medical documentation.

3 Approach and Experiments

3.1 EmbedKDECheck: A Frugal Omission Assessment Method

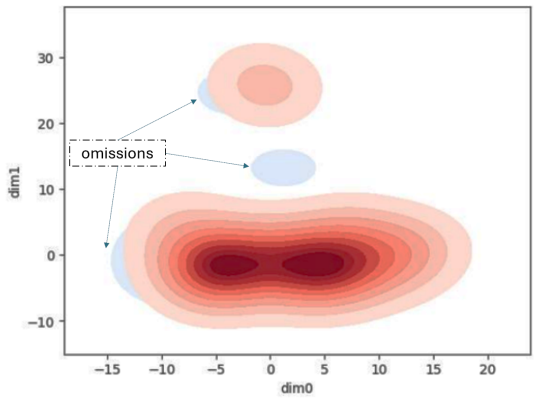

Figure 1: Coverage of input (blue) and output (red) embeddings after PCA. Blue regions correspond to input content, while red regions represent the output. Uncovered blue areas indicate missing or insufficiently represented information, signaling omissions.

EmbedKDECheck evaluates factual consistency by detecting omissions in summaries or reformulations without requiring references or LLM intermediate states. This black-box, reference-free method operates locally, making it infrastructure-independent and computationally frugal.

Given an input (e.g., report) and an output (e.g., summary), EmbedKDECheck assigns:

- A global omission score.
- Local indicators of missing critical content and omitted topics.

The approach analyzes embedding distributions of text segments using Kernel Density Estimation (KDE) (Węglarczyk, 2018). By modeling probability densities, it identifies input content not sufficiently covered in the output, flagging omissions (Figure 1).

KDE (Węglarczyk, 2018), a non-parametric density estimation method (Parzen, 1962), assigns probabilities to embedding distributions:

$$\hat{f}_h(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} K_h(\mathbf{x} - \mathbf{x}_i),$$

where $K_h(\mathbf{x})$ is a Gaussian kernel and h the bandwidth parameter. KDE provides adaptive density estimation, helping detect omissions effectively. Larger h results in a smoother, more generalized density estimation, while smaller h increases sensitivity to local variations, enhancing detection of small omissions but potentially introducing noise.

Figure 2 outlines EmbedKDECheck's main steps:

- Segment input/output text and extract embeddings.
- Construct KDE distributions over the embeddings.
- Compute omission scores using probability ratios:

$$\text{OM}_{\text{score}} = \frac{1}{\frac{\text{KDE probability density}}{\text{min density over output words}}}$$

- High scores indicate likely omissions.

The global omission score is computed as the maximum token-level score across the source text. Tokens with scores near the global value highlight which parts of the source text contributed most to the detected omission, providing interpretability and pinpointing omitted topics.

In our framework, the "batch" consists of the tokens from a source text and its corresponding summary. The length of the source text is not limiting since KDE is fitted on the output embeddings, not the full source. However, the output text (summary) should remain reasonably sized, which aligns with typical summarization settings.

PCA is applied before KDE to reduce embedding dimensionality, which is critical because KDE estimates multivariate Gaussian densities, and both the computational cost and the reliability of density estimation degrade in high-dimensional spaces. While OM scores could theoretically be computed without PCA, dimensionality reduction improves stability and efficiency.

To provide a more intuitive understanding of the KDE-based omission scoring process, we introduce a worked example illustrating how token-level embeddings from the source report are projected into the learned density space, and how unusually "sparse" tokens contribute to omission detection.

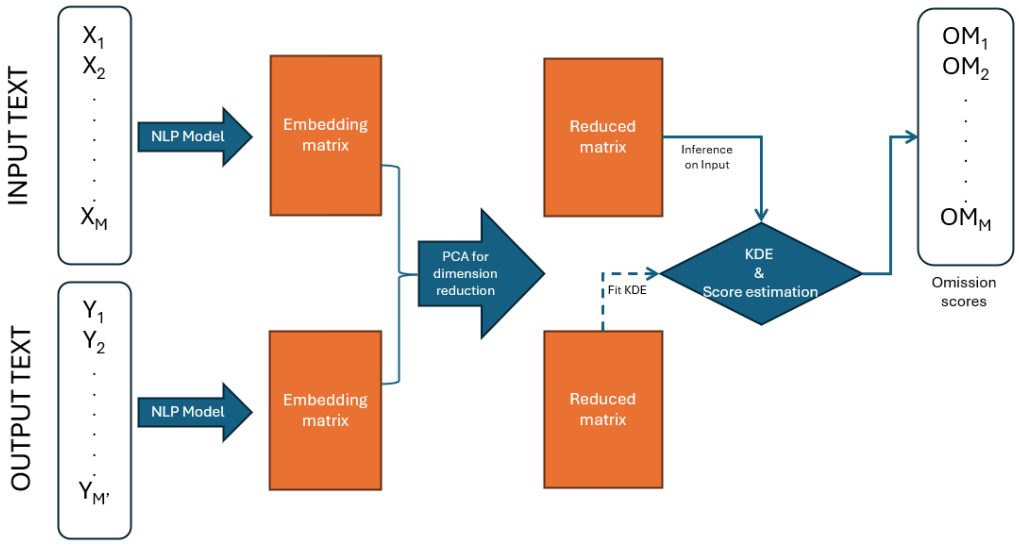

Figure 2: Overview of omission detection using KDE.

Example. Consider a short source excerpt: *"The patient underwent hepatectomy and showed no postoperative infection."* and its generated summary: *"The patient underwent surgery."* Each token from the source (e.g., "hepatectomy", "infection") and from the summary is embedded into a shared vector space. A Kernel Density Estimator (KDE) is then fitted over the summary embeddings to model the expected semantic distribution of the expressed content. When evaluating the source embeddings within this learned density space, tokens such as "hepatectomy" and "infection" fall in low-density regions, indicating that their corresponding information was not reflected in the summary. These tokens receive lower local density values and therefore higher omission scores. Aggregating across all tokens, the global omission score corresponds to the maximum token-level omission score, capturing the most significant missing concept.

In simple terms, KDE estimates the "normal density" of token embeddings from the summary; if a token (or group of tokens) from the source lies in a region of unusually low density, it signals a potential omission. This provides actionable interpretability. A schematic figure (Figure 2) in the paper illustrates this process step-by-step.

For efficiency, EmbedKDECheck utilizes lightweight embeddings FTW2V (Oukelmoun et al., 2023) combining FastText (Bojanowski et al., 2017) and Word2Vec (Mikolov et al., 2013), fine-tuned on a 32M-word independent medical corpus from the collaborating hospital. The system runs on CPU only, requiring 2.4GB of RAM and 1 hour 35 minutes of training on an Intel i7-10750H CPU.

Our method is language-agnostic, requiring only a word-level embedding provider. For efficiency, we use lightweight models such as Word2Vec and FastText combined under the FTW2V framework, suitable for KDE-based anomaly detection without relying on large LLMs. While the current dataset is French, future work will explore a translated English version and conduct a brief error analysis on non-medical text (e.g., legal abstracts) to assess broader cross-domain and cross-lingual applicability.

Since EmbedKDECheck requires no training, the dataset was split into validation (80%) and test (20%) sets with balanced labels. The detection threshold, number of PCA components, and KDE bandwidth were optimized on the validation set via random search. Final performance was measured on the test set, and experiments indicate robustness to moderate variations in threshold and bandwidth, while validation tuning ensures an optimal precision–recall balance.

3.2 Benchmarking models and metrics

This section introduces the reference-free and black-box models used for comparison with the EmbedKDECheck.

- **SelfCheckGPT** (Manakul et al., 2023): This approach evaluates a given summary by comparing it to multiple alternative summaries

generated for the same input. Specifically, for the benchmark, we sampled eight summaries per input. The embeddings were then computed using the openai model *text-embedding-ada-002*, and the final score was obtained as the mean of $1-$ cosine similarity between the assessed summary and the sampled ones. The method was tested with both *GPT-3.5 Turbo* and *GPT-4* for summary generation. The prompt used is provided in Appendix C. A temperature of 0.4 was applied to generate sufficient variability in the sampled summaries.

- **ChainPoll** (Friel and Sanyal, 2023): This approach is primarily based on prompting. The models tested here are *GPT-3.5 Turbo* and *GPT-4*. The prompt asks the LLM to develop a chain of thought before providing the final prediction on whether the summary contains an omission. The prompt used is provided in Appendix B.
- **G-Eval** (Liu et al., 2023): This approach also utilizes Chain of Thought (COT) reasoning but asks the LLM to provide a score, which is defined in detail, instead of giving a direct prediction. Both *GPT-4* and *GPT-3.5 Turbo* were tested. The prompt used is provided in Appendix A.
- **GPTScore** (Fu et al., 2024): This method uses the OpenAI model *text-embedding-ada-002* to obtain the input and output embeddings. The omission score is then calculated as $1-$ cosine similarity between the embeddings.

The scores were calculated for each pair, and recall and precision were then estimated based on the score threshold that maximizes the F1 score when the model generates a floating-point score instead of a direct prediction.

3.3 Evaluation Dataset

3.3.1 Dataset Description

Using real or anonymized medical reports in France was not feasible due to strict privacy regulations and ethical considerations. Data protection laws, such as the General Data Protection Regulation (GDPR), impose significant restrictions on the use of personal data, particularly in sensitive domains like healthcare. Complete anonymization is challenging because detailed medical reports can still risk re-identification (anonymization, 2024; re identification, 2024).

Medical reports, often manually written by surgeons or medical assistants, may contain inconsistencies and variations in writing style, level of detail, and adherence to structured templates. These irregularities can impact both clarity and completeness, and complicate automated NLP operations. Moreover, publicly available datasets for factual consistency in medical reports are limited (Luo et al., 2024), motivating the creation of a new synthetic dataset in French, which will be released as open-source.

To build the dataset, 50 anonymized medical reports were first collected from experts. Each report was slightly modified (names, dates, locations) to generate 50 fictitious seed reports. GPT-4-32K was then used to produce 15 synthetic variants per seed report, with instructions to maintain report structure while altering content such as family history, symptoms, and complications. Reports shorter than 200 words were discarded, leaving 674 reports with an average length of 353 words.

Each report contains standard sections, including *Motif d'Hospitalisation* (Reason for Hospitalization), *Antécédents* (Medical History), *Histoire de la Maladie* (History of Present Illness), *Clinique* (Clinical Examination), and *Évolution dans le Service* (In-Hospital Course). For each report, GPT-4-32K was used to generate two summaries: a complete summary capturing all key information, and a summary with intentional omissions created by randomly removing approximately 50% of the report's sentences prior to generation.

This process resulted in a dataset of 1,348 triplets, each consisting of a report, a corresponding summary, and a binary label indicating whether the summary contains omissions, providing a comprehensive benchmark for evaluating omission detection methods.

3.3.2 Quality Assessment

The synthetic dataset was evaluated on three main criteria: similarity to real reports, linguistic diversity, and content accuracy. The dataset consists of triplets (report, summary, label). To validate the synthetic labels, 90 report-summary pairs were blindly annotated by two expert surgeons in three stages, and the resulting annotations were then compared to the synthetic dataset labels to assess consistency. The stages were:

1. Determining whether the report was human- or LLM-generated.

2. Labeling the summary as complete or contain-

ing omissions.

3. Rating content quality (good, average, bad) if the summary was complete.

For the completeness/omission task, annotators achieved 95% accuracy and Cohen's $\kappa = 0.82$, indicating strong agreement. In contrast, distinguishing human vs. LLM-generated reports resulted in $\kappa = 0.10$ and F1 ≈ 0.5, showing that synthetic reports are highly realistic and nearly indistinguishable from human-written reports. Omissions were generated by removing 50% of sentences, and evaluation metrics were computed on this manually validated subset.

Linguistic diversity was assessed by analyzing CamemBERT (Martin et al., 2020) embeddings of real and synthetic reports via PCA (Figure 3). The close clustering of embeddings confirms that the synthetic dataset mimics the linguistic characteristics of real reports.

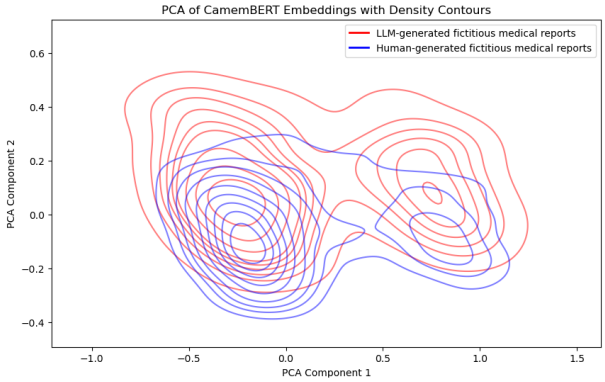

Figure 3: PCA of CamemBERT embeddings with density contours for synthetic and real reports.

For quality evaluation, summaries labeled as complete were further rated: 93% were considered good, 5% average, and 2% bad (with "bad" indicating at least one critical omission). In the omission detection task, annotators achieved 95% accuracy. Critical omissions are defined as missing information essential to clinical decisions or the overall understanding of the patient's situation. This confirms that synthetic labels are reliable and that the global omission score accurately identifies the source tokens responsible for omissions.

4 Results

The performance of the Embed2KDE model in detecting omissions in LLM-generated summaries is evaluated, with results presented in Tables 1 and 2.

The code and dataset are available on GitHub. The EmbedKDECheck code can be accessed at this repository[1]. The dataset is available at this repository[2].

If these links do not work for any reason, please do not hesitate to contact the main author via email for access to the code or dataset.

4.1 Omission Detection

EmbedKDECheck was tested for omission detection in French medical summaries and compared with several algorithms (Table 1). It achieved the highest F1-Score of 0.91, with recall at 0.88 and precision at 0.93, demonstrating its robustness in identifying omissions. While models such as ChainPoll and G-Eval show near-perfect recall, their tendency to predict almost every instance as an omission results in lower precision and F1 scores. In contrast, EmbedKDECheck strikes a better balance, offering more reliable omission detection. The method also identifies omitted topics, offering insights into summary completeness. Its approach is generalizable to other fields, making it a versatile tool for omission detection where frugality is required.

Algorithm	Recall	Precision	F1-Score
SelfCheck-gpt3-turbo	0.79	0.77	0.78
SelfCheck-gpt4	0.86	0.84	0.85
ChainPoll-gpt3-turbo	0.97	0.67	0.51
ChainPoll-gpt4	**0.99**	0.68	0.80
G-Eval-gpt3-turbo	0.98	0.46	0.63
G-Eval-gpt4	0.99	0.68	0.80
GPTScore	0.80	0.79	0.80
EmbedKDECheck	0.88	**0.93**	**0.91**

Table 1: Omission Detection Scores

4.2 Frugality Assessment

Frugality was evaluated using estimated FLOPS (Floating Point Operations Per Second) for each model (Table 2). For GPT-based models, FLOPS were calculated by multiplying the number of tokens by the model size. In contrast, for EmbedKDECheck, FLOPS were derived from the product of the running time and peak hardware FLOPS.

These results highlight EmbedKDECheck's effectiveness and efficiency in omission detection, making it a strong candidate for summarization tasks that require both performance and frugality.

[1] https://github.com/achok7893/EmbedKDECheck_hallucination_detection

[2] https://github.com/achok7893/EmbedKDECheck_OmissionsDection_Dataset_Fr_Healthcare

Model	Estimated Tera FLOPS
SelfCheck-gpt3-turbo	1545.40
SelfCheck-gpt4	8804.03
ChainPoll-gpt3-turbo	770.73
ChainPoll-gpt4	4404.16
G-Eval-gpt3-turbo	513.81
G-Eval-gpt4	2936.04
GPTScore	2.14
EmbedKDECheck	0.11

Table 2: Estimated FLOPS for Different Models

5 Deployment

EmbedKDECheck is designed for efficient and privacy-compliant deployment in medical environments. Its lightweight architecture enables real-time omission detection using only commodity CPUs, without the need for large-scale GPUs or external API calls. The system relies exclusively on word-level embeddings (FastText and Word2Vec combined under the FTW2V framework) and a Kernel Density Estimation (KDE) module for anomaly detection. This frugal design minimizes computational cost and energy usage while ensuring full compatibility with hospital privacy and security requirements, as all computations can be executed locally.

The module is intended to be deployed in an **experimental configuration** at an incubator affiliated with a major hospital in the Paris area. This deployment aims to evaluate the system's effectiveness and integration feasibility within a controlled environment replicating clinical data flows. In this setup, EmbedKDECheck will operate as a validation layer for LLM-generated summaries and surgical reports, identifying potential omissions in critical information without interfering with real medical workflows.

Because the method requires only lightweight embeddings and a KDE-based scoring function, its computational footprint is several orders of magnitude lower than GPT-based evaluators such as SelfCheckGPT or G-Eval. This enables continuous experimentation and large-scale benchmarking on standard CPU workstations. Furthermore, its language-agnostic design facilitates adaptation to multilingual and cross-domain contexts, including English medical datasets and other specialized domains such as legal or industrial documentation.

The experimental architecture integrates EmbedKDECheck within the hospital-affiliated incubator's LLM monitoring pipeline, providing a safe, interpretable, and resource-efficient validation layer to enhance the reliability of AI-assisted medical reporting systems.

6 Discussion

The results demonstrate that EmbedKDECheck achieves a strong balance between performance and computational efficiency in omission detection. As shown in Tables 1 and 2, the model reaches an F1-score of 0.91 while remaining cost-effective, making it particularly suitable for deployment in resource-constrained environments. By leveraging a compact kernel density estimation framework over embeddings, EmbedKDECheck effectively captures semantic deviations that indicate missing information, without requiring access to the underlying generative model. This frugal and reference-free design makes it a practical solution for large-scale auditing of LLM-generated summaries.

A key contribution of this work is the construction of a **synthetic yet expert-validated dataset** for omission detection in medical summaries. To address this, we generated 674 synthetic reports with GPT-4-32K, designed to closely resemble authentic clinical narratives while ensuring privacy compliance. We removed 50% of the sentences in each report to guarantee the presence of substantial and medically relevant omissions, thereby strengthening the reliability of the ground-truth labels. Lower removal rates often lead to minor or absent omissions, reducing the dataset's discriminative power. The dataset was rigorously validated for similarity, diversity, and factual accuracy by medical experts and is released openly to encourage reproducibility and further research.

Beyond the medical domain, **EmbedKDECheck** shows strong potential for generalization. Its architecture can be adapted to other high-stakes fields such as law or finance, where preserving critical information in summaries is equally vital. The same principles can extend to related tasks like hallucination detection, factual consistency estimation, or bias identification in generated content. By remaining reference-free, the approach aligns with growing needs for transparency and accountability in large language model evaluation, particularly when gold-standard references are unavailable. Future work will focus on extending the method to other forms of hallucinations while preserving both frugality and interpretability.

7 Limitations

Despite its promising performance, **EmbedKDECheck** presents some limitations that open avenues for further investigation. First, its effectiveness depends heavily on the **quality and domain relevance of the embeddings**. Although pretrained embedding models offer strong general representations, they may fail to capture fine-grained medical or contextual nuances without domain-specific adaptation. Fine-tuning or hybrid embedding strategies could improve sensitivity to subtle omissions in specialized domains.

Second, while EmbedKDECheck is computationally efficient, its performance may still vary depending on available resources and chosen hyperparameters, Achieving an optimal balance between precision and cost remains a critical challenge for deployment in large institutional pipelines.

Third, the construction of a fully reliable ground truth dataset with manually verified omission counts remains **prohibitively expensive and time-consuming**. Such a process requires extensive input from domain experts, particularly surgeons or specialists, making it difficult to scale validation efforts. The synthetic dataset proposed here represents a necessary compromise between realism and feasibility, though further efforts are needed to expand and diversify human-validated corpora.

Finally, as a locally based, reference-free method, EmbedKDECheck is by construction robust to the omission rate used during dataset generation, but may still face challenges in settings where omissions are subtle, stylistic, or semantically diffuse. Addressing these limitations will require future research into interpretable, hybrid systems that combine statistical robustness with semantic reasoning. Despite these challenges, EmbedKDECheck significantly outperforms existing methods in omission detection and provides a scalable foundation for reliable evaluation of LLM-generated summaries across sensitive domains.

Ethical considerations

Due to legal (GDPR[3]) and ethical (privacy) concerns, real or simply anonymized medical reports couldn't be used. Instead, we created a synthetic dataset by generating fictitious reports from anonymized ones and using GPT-4-32K to produce similar reports. This approach ensured compliance with ethical standards and privacy regulations.

[3] https://en.wikipedia.org/wiki/General_Data_Protection_Regulation

Acknowledgments

We would like to express our sincere gratitude to the surgeons specializing in hepatic and general surgery for their invaluable contribution to the annotation and validation of the dataset used in this study. Their clinical expertise was essential in ensuring the quality and medical relevance of the annotations, making our dataset particularly robust for evaluating omission detection.

We also thank the research teams at **CEA List** for their scientific guidance, methodological support, and insightful technical discussions, which greatly contributed to the development of this work.

Finally, we acknowledge the **Chaire BOPA (Bloc Opératoire Augmenté)** for its institutional support, strategic guidance, and commitment to promoting interdisciplinary research at the interface of medicine and artificial intelligence.

This work would not have been possible without the dedication and collaboration of all the partners mentioned above.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ian L Alberts, Lorenzo Mercolli, Thomas Pyka, George Prenosil, Kuangyu Shi, Axel Rominger, and Ali Afshar-Oromieh. 2023. Large language models (llm) and chatgpt: what will the impact on nuclear medicine be? *European journal of nuclear medicine and molecular imaging*, 50(6):1549–1552.

CNIL anonymization. 2024. L'anonymisation de données personnelles. https://www.cnil.fr/fr/technologies/lanonymisation-de-donnees-personnelles.

EDS AP-HP. 2024. L'Entrepôt de Données de Santé de l'AP-HP. https://www.aphp.fr/connaitre-lap-hp/recherche-innovation/lentrepot-de-donnees-de-sante-de-lap-hp.

Amos Azaria and Tom Mitchell. 2023. The internal state of an llm knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *arXiv preprint arXiv:1903.00161*.

Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Robert Friel and Atindriyo Sanyal. 2023. Chainpoll: A high efficacy method for llm hallucination detection. *arXiv preprint arXiv:2310.18344*.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2024. GPTScore: Evaluate as You Desire. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6556–6576, Mexico City, Mexico. Association for Computational Linguistics.

Yang Gao, Wei Zhao, and Steffen Eger. 2020. Supert: Towards new frontiers in unsupervised evaluation metrics for multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1347–1354.

Hongzhao Li, Hongyu Wang, Xia Sun, Hua He, and Jun Feng. 2024. Prompt-guided generation of structured chest x-ray report using a pre-trained llm. *arXiv e-prints*, pages arXiv–2404.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522.

Zheheng Luo, Qianqian Xie, and Sophia Ananiadou. 2024. Factual consistency evaluation of summarisation in the era of large language models. *Expert Systems with Applications*, page 124456.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017.

Louis Martin, Benjamin Muller, Pedro Ortiz Suarez, Yoann Dupont, Laurent Romary, Éric Villemonte De La Clergerie, Djamé Seddah, and Benoît Sagot. 2020. Camembert: a tasty french language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Nabla. 2024. All you need to know about Nabla's privacy and security features. https://www.nabla.com/blog/privacy-security/.

Achir Oukelmoun, Nasredine Semmar, Gaël De Chalendar, Enguerrand Habran, Eric Vibert, Emma Goblet, Mariame Oukelmoun, and Marc-Antoine Allard. 2023. A study on the relevance of generic word embeddings for sentence classification in hepatic surgery. In *2023 20th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA)*, pages 1–8. IEEE.

Emanuel Parzen. 1962. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076.

CNIL re inditification. 2024. L'anonymisation de données personnelles. https://www.cnil.fr/fr/technologies/lanonymisation-de-donnees-personnelles.

Tempest A van Schaik and Brittany Pugh. 2024. A field guide to automatic evaluation of llm-generated summaries. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2832–2836.

Oleg Vasilyev, Vedant Dharnidharka, and John Bohannon. 2020. Fill in the blanc: Human-free quality estimation of document summaries. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 11–20.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. *arXiv preprint arXiv:2004.04228*.

Stanisław Węglarczyk. 2018. Kernel density estimation and its application. In *ITM web of conferences*, volume 23, page 00037. EDP Sciences.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, pages 24824–24837, Red Hook, NY, USA. Curran Associates Inc.

A G-Eval prompt

This appendix presents the evaluation prompt used in the study. The original prompt is in French, followed by its English translation to facilitate understanding for reviewers.

A.1 Prompt in French

Vous recevrez un **compte rendu médical** et un **résumé** de ce compte rendu. Votre tâche est d'évaluer le résumé sur la base de sa **complétude** et de sa capacité à inclure toutes les informations critiques issues du compte rendu.

Veuillez suivre attentivement les instructions ci-dessous et vous y référer tout au long de l'évaluation.

A.1.1 Critères d'évaluation

Complétude et pertinence (1-3)

- **Score 1 (Insuffisant)** :
  - Le résumé manque plusieurs informations critiques essentielles à la compréhension du cas.
  - Les omissions pourraient avoir un impact significatif sur la prise de décision médicale ou les soins du patient.
- **Score 2 (Moyen)** :
  - Le résumé inclut certaines informations clés, mais omet une ou deux informations importantes.
  - Bien que les omissions soient notables, elles ne compromettent pas totalement la compréhension du cas.
- **Score 3 (Excellent)** :
  - Le résumé est complet et inclut toutes les informations critiques issues du compte rendu médical.
  - Aucune omission significative n'est présente, et le résumé permet une compréhension totale du cas.

A.1.2 Étapes d'évaluation

1. **Étape 1 :** Lisez attentivement le compte rendu médical et identifiez les détails principaux (diagnostics, traitements, résultats de tests, antécédents, etc.).
2. **Étape 2 :** Comparez le résumé au compte rendu médical. Identifiez les informations manquantes ou incorrectes.
3. **Étape 3 (raisonnement en chaîne) :**
   - Analysez étape par étape si le résumé correspond au compte rendu.
   - Soulignez les divergences ou omissions et évaluez leur importance.
   - Expliquez clairement votre raisonnement pour le score attribué.
4. **Étape 4 :** Attribuez un score de complétude entre 1 et 3, en suivant les critères ci-dessus.

Compte rendu médical : [INSÉREZ ICI LE COMPTE RENDU MÉDICAL] **Résumé fourni :** [INSÉREZ ICI LE RÉSUMÉ À ANALYSER] **Formulaire d'évaluation (scores UNIQUEMENT) :** - Complétude :

A.2 English Translation of the Prompt

You will receive a **medical report** and a **summary** of this report. Your task is to evaluate the summary based on its **completeness** and its ability to include all critical information from the report. Please carefully follow the instructions below and refer to them throughout the evaluation.

A.2.1 Evaluation Criteria

Completeness and relevance (1-3)

- **Score 1 (Insufficient)**:
  - The summary lacks several critical pieces of information essential for understanding the case.
  - The omissions could significantly impact medical decision-making or patient care.
- **Score 2 (Moderate)**:
  - The summary includes some key information but omits one or two important details.
  - Although the omissions are noticeable, they do not completely compromise understanding of the case.
- **Score 3 (Excellent)**:
  - The summary is complete and includes all critical information from the medical report.
  - No significant omissions are present, and the summary allows for a full understanding of the case.

A.2.2 Evaluation Steps

1. **Step 1:** Carefully read the medical report and identify the main details (diagnoses, treatments, test results, medical history, etc.).
2. **Step 2:** Compare the summary with the medical report. Identify any missing or incorrect information.
3. **Step 3 (chain reasoning):**
   - Analyze step by step whether the summary corresponds to the report.
   - Highlight discrepancies or omissions and assess their importance.
   - Clearly explain your reasoning for the assigned score.
4. **Step 4:** Assign a completeness score between 1 and 3, following the criteria above.

Medical report: [INSERT MEDICAL REPORT HERE] **Provided summary:** [INSERT SUMMARY TO BE ANALYZED HERE] **Evaluation form (scores ONLY):** - Completeness:

B ChainPoll prompt

This appendix presents the prompt used for binary omission detection. The original prompt is in French, followed by its English translation for better clarity.

B.1 Prompt in French

Vous êtes un assistant médical spécialisé dans l'analyse de comptes rendus médicaux. Vous recevrez : 1. Un **compte rendu médical** détaillé. 2. Un **résumé** de ce compte rendu. **Votre tâche :** 1. Identifier et expliquer si le résumé omet des informations médicales importantes qui figurent dans le compte rendu. 2. Indiquer s'il existe des omissions importantes avec une valeur binaire : - **0** : Pas d'omissions importantes. - **1** : Des omissions importantes sont présentes. **IMPORTANT :** À la fin de votre analyse, incluez une ligne au format clair : `OMISSION_RESULT = [0 ou 1]`

B.1.1 Étapes

1. **Étape 1 :** Analysez le compte rendu médical en identifiant les informations essentielles (diagnostics, traitements, antécédents, résultats de tests, etc.).

2. **Étape 2 :** Comparez le résumé fourni au compte rendu original. Relevez les informations importantes manquantes, le cas échéant.

3. **Étape 3 :** Justifiez votre décision en listant les éléments omis ou confirmant qu'aucune omission significative n'est présente.

4. **Étape 4 :** Fournissez le résultat binaire au format clair.

Compte rendu médical : [INSÉREZ ICI LE COMPTE RENDU MÉDICAL] **Résumé fourni :** [INSÉREZ ICI LE RÉSUMÉ À ANALYSER]

B.2 English Translation of the Prompt

You are a medical assistant specializing in the analysis of medical reports. You will receive: 1. A **detailed medical report**. 2. A **summary** of this report. **Your task:** 1. Identify and explain whether the summary omits important medical information present in the report. 2. Indicate whether significant omissions exist using a binary value: - **0**: No significant omissions. - **1**: Significant omissions are present. **IMPORTANT:** At the end of your analysis, include a clearly formatted line: `OMISSION_RESULT = [0 or 1]`

B.2.1 Steps

1. **Step 1:** Analyze the medical report by identifying key information (diagnoses, treatments, history, test results, etc.).

2. **Step 2:** Compare the provided summary to the original report. Note any missing important information, if applicable.

3. **Step 3:** Justify your decision by listing omitted elements or confirming that no significant omission is present.

4. **Step 4:** Provide the binary result in a clear format.

Medical report: [INSERT MEDICAL REPORT HERE] **Provided summary:** [INSERT SUMMARY TO BE ANALYZED HERE]

C SelfCheckGPT System Prompt

The system prompt used in the model is as follows:

> **French:** *Vous êtes un assistant. Je vais vous fournir un compte-rendu médical synthétique et vous allez devoir me fournir un résumé complet.*

English Translation: *You are an assistant. I will provide you with a concise medical report, and you will be required to provide a complete summary.*