République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de 8 Mai 1945-Guelma-
Faculté des Mathématiques, d'Informatique et des Sciences de la matière
Département d'Informatique

Mémoire de fin d'études Master

Filière : Informatique

Option :

Sciences et Technologies de l'Information et de la Communication

Thème

Automatisation de l'analyse des dossiers médicaux pour une prise de décision clinique optimisée

Encadré par :
Dr. BOUGHIDA ADIL

Présenté par :
*K*ouadria Amine

Année Universitaire 2024/2025

Remerciements

On dit souvent que le chemin parcouru compte autant que la destination atteinte. Les années passées à l'université nous ont permis de donner tout son sens à cette maxime. Ce parcours n'a pas été sans embûches, ni sans interrogations profondes, dont les réponses ont souvent exigé de longues heures de travail, de réflexion et de persévérance.

Avant toute chose, je rends grâce à Dieu pour m'avoir accordé la force, le savoir, la patience et l'opportunité d'accomplir ce travail, de surmonter les difficultés rencontrées et d'en voir aujourd'hui l'aboutissement.

Je tiens à exprimer ma profonde gratitude à mon encadreur, le Docteur ***BOUGHIDA Adil****, pour sa disponibilité, ses conseils avisés, ses encouragements constants ainsi que sa patience tout au long de cette aventure académique. Merci de m'avoir accordé votre confiance et d'avoir contribué, par votre accompagnement, à la concrétisation de ce travail.*

Mes plus sincères remerciements vont également à mes parents, pour leur amour inconditionnel, leurs sacrifices et leur soutien indéfectible tout au long de mon parcours universitaire. Sans eux, rien n'aurait été possible.

Enfin, je remercie chaleureusement l'ensemble des enseignants du département d'informatique de l'Université du 8 Mai 1945 de Guelma, pour la qualité de l'enseignement dispensé et pour avoir contribué à ma formation, chacun à sa manière.

Dédicace

Je dédie ce travail

À ma chère mère

Pour ton amour inconditionnel, ton soutien constant, tes sacrifices silencieux et ta foi en moi, je te dédie ce travail avec toute ma reconnaissance et mon affection profonde. Que Dieu te préserve.

À mon père bien-aimé

Pour ta patience, tes efforts et ta force dans les moments difficiles, merci d'avoir toujours été là. Ce travail est aussi le fruit de ta persévérance et de ton amour.

À mon grand frère Ramy

Ta présence et ton soutien ont été essentiels dans mon parcours. Merci pour tes encouragements et ta confiance. Je te dédie ce travail avec respect et admiration.

À mon frère Billel (Allah yarhamou)

Tu restes à jamais vivant dans mon cœur. Ton souvenir m'accompagne à chaque étape. Que Dieu t'accorde Sa miséricorde et le paradis éternel. Ce travail t'est dédié avec tout mon amour.

À mes chers amis

Sincères remerciements à ***IBRAHIM , NIDAL , SAMI , MOHAMED , SEIF , ACHERAF , WASSIM , ABDERRAHMANE , MOHAMED, ISLEM , ALA*** *et tous ceux qui m'ont soutenu et encouragé tout au long de ce parcours. Votre présence a fait toute la différence.*

AMINE

Résumé

La présente étude s'inscrit dans le contexte de la transformation numérique des dossiers médicaux et vise à automatiser le résumé des notes cliniques pour soutenir la prise de décision. L'objectif principal est de développer un modèle capable de générer des résumés cohérents et factuels à partir des dossiers de sortie du *dataset* **MIMIC-IV**.

La méthodologie adoptée repose sur le *fine-tuning* du modèle **LongT5**, choisi pour sa capacité à traiter de très longues séquences textuelles. Une attention particulière a été portée au prétraitement des données, notamment à la génération de résumés cibles cohérents et précis, afin d'assurer une supervision de haute qualité. L'adaptation du modèle pré-entraîné a ensuite été réalisée efficacement grâce à la méthode **LoRA** (Low-Rank Adaptation).

Notre modèle final, nommé `MedSum-LongT5`, atteint des performances solides avec des scores de **52,6% en ROUGE-1**, **35,3% en ROUGE-2** et **42,9% en ROUGE-L**. Ces résultats surpassent nettement les modèles de référence non spécialisés, validant l'efficacité de l'approche.

L'analyse qualitative confirme que le modèle extrait avec fiabilité les informations cliniques clés et génère des résumés bien structurés. Toutefois, des défis subsistent, notamment en ce qui concerne **la généralisation aux cas rares**, **la fidélité factuelle** et **le coût de calcul**. En conclusion, ce travail atteint son objectif principal et fournit une base robuste pour le développement futur d'outils d'IA d'aide à la synthèse clinique, en soulignant l'importance d'une supervision humaine. Les perspectives d'amélioration se concentrent sur l'optimisation du modèle (quantification, distillation) et sur l'intégration de retours d'experts pour en accroître la fiabilité.

Mots-clés : *Résumé automatique de texte,* Large Language Model *(LLM), LongT5, Notes cliniques, MIMIC-IV, Fine-tuning, Métriques ROUGE.*

Abstract

This study is situated within the context of the digital transformation of medical records and aims to automate the summarization of clinical notes to support clinical decision-making. The main objective is to develop a model capable of generating coherent and factual summaries from the discharge summaries of the **MIMIC-IV** *dataset*.

The adopted methodology is based on the *fine-tuning* of the **LongT5** model, chosen for its ability to process very long text sequences. Particular attention was given to data preprocessing, notably the generation of coherent and precise target summaries to ensure high-quality supervision. The adaptation of the pre-trained model was then efficiently performed using the **LoRA** (Low-Rank Adaptation) method.

Our final model, named `MedSum-LongT5`, achieves solid performance with scores of **52.6% for ROUGE-1**, **35.3% for ROUGE-2**, and **42.9% for ROUGE-L**. These results significantly outperform non-specialized baseline models, validating the effectiveness of the approach.

Qualitative analysis confirms that the model reliably extracts key clinical information and generates well-structured summaries. However, challenges remain, particularly regarding **generalization to rare cases**, **factual fidelity**, and **computational cost**. In conclusion, this work achieves its main objective and provides a robust foundation for the future development of AI-powered clinical summarization tools, emphasizing the importance of human supervision. Future work will focus on model optimization (quantization, distillation) and the integration of expert feedback to increase reliability.

Keywords : *Automatic Text Summarization, Large Language Model (LLM), LongT5, Clinical Notes, MIMIC-IV, Fine-tuning, ROUGE Metrics.*

Table des matières

Table des figures

Liste des tableaux

Introduction Générale

Contexte

Le domaine de la santé fait face à une explosion de données numériques. Chaque jour, les établissements hospitaliers produisent un volume considérable d'informations, allant de l'imagerie aux résultats de laboratoire [1]. Cependant, une part majeure de cette connaissance reste "verrouillée" dans des documents textuels non structurés : comptes-rendus de consultation, notes de suivi, protocoles opératoires, etc [2].

Cette situation crée un goulot d'étranglement informationnel. Pour les professionnels de santé, l'analyse manuelle de ces longs documents est une tâche chronophage et fastidieuse [3]. Plus grave encore, elle augmente le risque d'omettre une information cruciale ou de faire une erreur d'interprétation, avec des conséquences potentielles sur la qualité et la sécurité des soins [4]. L'automatisation de l'analyse de ces textes est donc devenue un enjeu majeur pour optimiser la pratique clinique [5].

Motivation et Problématique

Face à ce défi, l'intelligence artificielle, et plus particulièrement les modèles de langage de grande taille nommés *Large Language Models* (LLM), offrent des perspectives prometteuses. Ces technologies ont démontré une capacité sans précédent à comprendre, analyser et synthétiser des textes complexes [6]. Dans le domaine médical, des modèles spécialisés ont déjà prouvé leur potentiel pour des tâches comme l'extraction d'informations ou la réponse à des questions cliniques [7].

Cependant, **le résumé automatique de dossiers médicaux** présente un défi de taille : la fidélité factuelle. Un résumé, même s'il est fluide et grammaticalement correct, **perd toute sa valeur s'il déforme ou omet des informations médicales critiques**. La confiance est la clé de l'adoption de ces outils par les cliniciens [8].

Notre travail s'inscrit au cœur de cet enjeu. La problématique centrale de ce mémoire est donc la suivante : ***Comment développer et adapter une méthode basée sur un LLM pour générer des résumés de notes cliniques qui soient non seulement concis et pertinents, mais surtout factuellement fiables et directement exploitables dans un contexte de prise de décision ?***

Pour répondre à cette question, nous proposons une approche complète allant d'un prétraitement rigoureux des données à un fine-tuning efficace du modèle **LongT5** [9], spécifiquement conçu pour traiter de longs documents.

Pour répondre à cette question, nous proposons une approche complète allant d'un prétraitement rigoureux des données à un fine-tuning efficace du modèle **LongT5**, spécifiquement conçu pour traiter de longs documents. Le modèle ainsi obtenu, que nous appelons `MedSum-LongT5`, est optimisé pour la génération de résumés médicaux précis et cohérents.

Structure du Mémoire

Ce mémoire est organisé en trois chapitres principaux.

- Le **premier chapitre** dresse un état de l'art sur les modèles de langage de grande taille (LLM) et leurs applications dans le domaine biomédical. Il présente les architectures fondamentales, les modèles spécialisés existants et les défis liés à leur utilisation en médecine.
- Le **deuxième chapitre** détaille notre méthodologie. Nous y décrivons le corpus **MIMIC-IV** [10] utilisé, le pipeline complet de prétraitement des données, le choix du modèle **LongT5** et la stratégie d'adaptation par **LoRA** [11].
- Enfin, le **troisième chapitre** est consacré à la présentation et à l'analyse de nos résultats. Il expose le suivi de l'apprentissage du modèle, évalue ses performances quantitatives (scores **ROUGE** [12]) et qualitatives (analyse d'exemples), le compare à d'autres modèles de référence, et se conclut par une discussion sur la portée et les limites de nos travaux.

Chapitre 1

LLM dans le domaine médical

1 Introduction

L'intelligence artificielle a permis des avancées significatives dans de nombreux secteurs, y compris celui de la santé. Parmi les technologies émergentes, les *Large Language Models* (LLM) occupent aujourd'hui une place centrale en permettant l'interprétation et la génération automatique de textes complexes, tels que les documents médicaux. En effet, ces modèles avancés, entraînés sur d'énormes volumes de données textuelles, sont capables d'améliorer divers processus médicaux, notamment la prise de décision clinique, la gestion des dossiers médicaux et la recherche biomédicale [7, 13].

Dans ce chapitre, nous présenterons tout d'abord une vue générale des concepts essentiels liés aux LLM. Nous aborderons ensuite plus précisément leurs différentes architectures, les données sur lesquelles ils sont entraînés, ainsi que leurs capacités et limites identifiées dans la littérature. Nous présenterons leurs principales applications en médecine et discuterons des défis liés à leur intégration, avec une attention particulière portée à la tâche de **résumé de documents médicaux**, qui sera examinée en détail.

2 Fondements, architectures et applications des LLM

2.1 Définition et évolutions des LLMs

Les *Large Language Models* (LLM) sont des systèmes d'intelligence artificielle avancés, construits à l'aide de techniques de Deep Learning. Considérez-les comme de très grands réseaux de neurones, souvent basés sur **l'architecture Transformer** [14], qui sont entraînés sur d'énormes quantités de données textuelles. Cet entraînement approfondi leur permet de comprendre et de générer du texte de manière similaire à un humain [15].

Pour atteindre les capacités des LLM actuels, le domaine du traitement automatique du langage a exploré plusieurs approches et générations de modèles :

1. ***Statistical Language Models* (SLM)** : ces modèles [16, 17], issus de l'apprentissage statistique (années 1990), exploitent la propriété de Markov [18], notamment via les **modèles n-grammes** (prédiction sur n mots de contexte). Largement utilisés en **recherche**

d'information (*Information Retrieval*, IR) [19] et **traitement du langage naturel (NLP)** [20, 21], leur principale limite est la difficulté à estimer les probabilités pour des séquences de mots plus longues (quand n est grand), car beaucoup de ces séquences sont rares ou absentes des données d'entraînement.

2. ***Neural Language Models* (NLM)** : ces modèles, s'appuyant sur des réseaux de neurones comme RNN et LSTM [22] qui modélisent le langage en apprenant des représentations à valeurs vectorielles (embeddings) des mots [23]. Cette approche permet une capture du contexte sémantique plus riche que les SLM. Des outils comme **Word2Vec** [24] ont popularisé leur apprentissage efficace, améliorant significativement les performances par une meilleure généralisation des dépendances contextuelles [25].
3. ***Pre-trained Language Models* (PLM)** : ces modèles apprennent des représentations de mots riches et contextuelles. Initiés par des approches comme les modèles basés sur l'architecture Transformer (comme **BERT** [26]), ils ont établi le paradigme influent de "pré-entraînement puis *fine-tuning*". Cette méthode consiste à d'abord entraîner un modèle sur de vastes datasets non étiquetés, puis à l'adapter (fine-tuning) pour des tâches spécifiques, ce qui a permis d'améliorer considérablement les performances en traitement automatique des langues [25].
4. **Large Language Models (LLM)** : Nés de la mise à l'échelle massive des PLM Transformer (milliards de paramètres), les LLM développent des "capacités émergentes" (en anglais, *emergent abilities*[1] [27] qui surpassent les modèles classiques. Ces nouvelles aptitudes – allant de la génération de texte fluide à des capacités particulièrement distinctives, comme celle illustrée par GPT-3 qui peut résoudre des tâches complexes via l'apprentissage en contexte avec peu d'exemples (*few-shot in-context learning*), surpassant nettement les PLM antérieurs tels que GPT-2 [28] – ont conduit la communauté scientifique à forger le terme spécifique de **"*Large Language Model* (LLM)"** pour désigner ces PLM de très grande taille . Cette puissance s'accompagne de défis en termes de biais, coûts de calcul et interprétabilité [25].

La Figure 1.1 ci-après schématise ce processus d'évolution des modèles de langage à travers les quatre générations distinctes, en mettant en perspective leur capacité croissante à résoudre des tâches.

2.2 Fondations des LLMs : Mécanisme d'Attention et Transformer

L'architecture qui sous-tend la grande majorité des LLM performants actuels est le **Transformer**. Introduit par Vaswani et al. (2017) dans leur article fondateur "*Attention Is All You Need*" [14], le Transformer a constitué une révolution dans le traitement du langage naturel. Son innovation clé est le mécanisme d'**auto-attention (*self-attention*)**, et plus spécifiquement sa variante **multi-têtes (*multi-head self-attention*)**. Ce dernier permet au modèle de peser l'importance relative de tous les éléments (tokens) d'une séquence d'entrée simultanément,

1. Notez qu'un LLM n'est pas nécessairement plus performant qu'un petit PLM, et certaines capacités émergentes peuvent ne pas se manifester dans certains LLM.

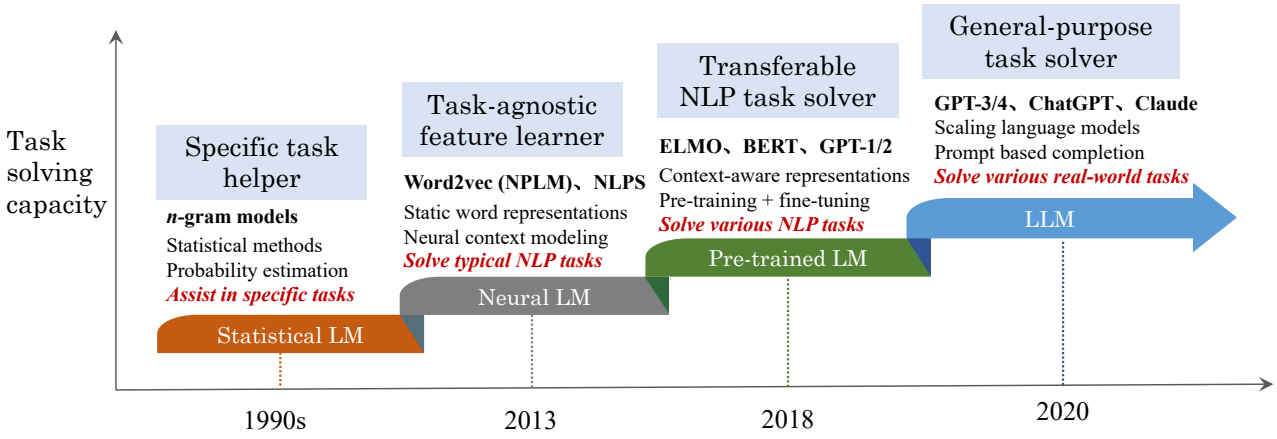


FIGURE 1.1 – Processus d'évolution des modèles de langage [25].

en considérant différentes projections de l'information pour capturer diverses relations. Cela assure une saisie efficace des dépendances contextuelles, y compris celles à longue distance, et favorise une parallélisation poussée des calculs, surpassant ainsi les approches séquentielles des architectures RNN ou LSTM.

La robustesse de l'architecture Transformer et sa capacité à supporter des modèles de très grande profondeur, comme l'illustre la Figure 1.2, découlent de plusieurs choix de conception structurelle essentiels qui assurent la stabilité de l'apprentissage et un bon flux du gradient :

- **L'empilement de couches (Stacked Layers)** : L'encodeur et le décodeur sont typiquement formés par l'empilement de N couches identiques (visibles avec l'indication "$N\times$" sur la Figure 1.2). Chaque couche, contenant des sous-couches d'auto-attention et des réseaux de neurones feed-forward, permet au modèle d'apprendre des représentations des caractéristiques de plus en plus abstraites et contextuelles. (Il est à noter que certaines architectures de LLM spécifiques peuvent n'utiliser que la partie encodeur ou décodeur).
- **Les connexions résiduelles (Residual Connections)** : Autour de chaque sous-couche (c'est-à-dire l'auto-attention et le réseau feed-forward), une connexion résiduelle est systématiquement employée (indiquée par "Add" dans le schéma de la Figure 1.2). Elle consiste à additionner l'entrée de la sous-couche à sa sortie, ce qui aide à atténuer le problème de la disparition du gradient et facilite l'entraînement de réseaux particulièrement profonds.
- **La normalisation de couche (Layer Normalization)** : Chaque connexion résiduelle est immédiatement suivie d'une étape de normalisation de couche ("Norm" dans le schéma de la Figure 1.2). Cette technique stabilise les activations neuronales au sein de chaque couche, contribuant ainsi à accélérer la convergence et à améliorer la généralisation du modèle.

Cette conception architecturale modulaire, combinée à ces mécanismes de stabilisation, a posé les fondations essentielles au développement ultérieur de modèles comptant plusieurs milliards de paramètres et aux avancées majeures qui en ont découlé en Traitement Automatique des Langues [14].

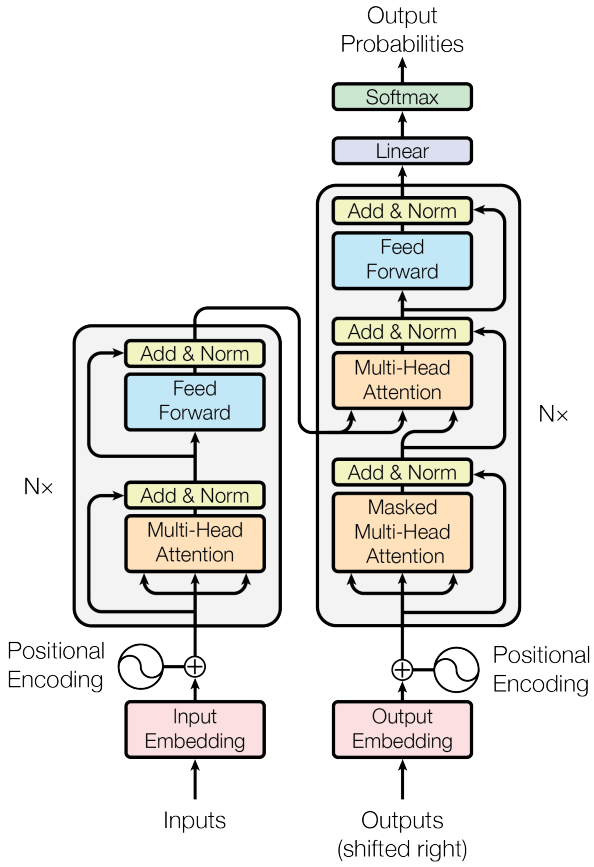


FIGURE 1.2 – Le Transformer – architecture du modèle original [14].

Principales Approches Architecturales des LLM

Les architectures des LLM peuvent être globalement classées en trois catégories principales, chacune optimisée pour des types de tâches spécifiques grâce à des stratégies d'entraînement distinctes :

- **Modèles à Auto-Encodage (*Autoencoding Models*)** (ex. BERT, RoBERTa) : Ces modèles utilisent le *masked language modeling (MLM)*, où des tokens aléatoires dans la séquence d'entrée sont masqués, et le modèle est entraîné à prédire ces tokens masqués pour reconstruire la phrase originale. Cela permet une compréhension contextuelle *bidirectional* (dans les deux sens), rendant ces modèles particulièrement adaptés pour des tâches telles que l'analyse de sentiments et la reconnaissance d'entités nommées (en anglais, *named entity recognition*) [29].
- **Modèles Auto-Régressifs (*Autoregressive Models*)** (ex. GPT-3, BLOOM) : Ces modèles prédisent le token suivant dans une séquence en se basant uniquement sur les tokens précédents. En tant que sous-ensemble, les *Causal Language Models (CLMs)* exploitent un contexte *unidirectional* (unidirectionnel) pour traiter les données séquentielles, garantissant que les prédictions ne dépendent que des tokens antérieurs. Grâce à ce contexte *unidirectional*, ils excellent dans des tâches comme la *text generation* (génération de texte) et l'inférence *zero-shot* [30].
- **Modèles Séquence-à-Séquence (*Sequence-to-Sequence Models*)** (ex. T5, BART) : Ceux-ci emploient à la fois un *encoder* (encodeur) pour traiter la séquence d'entrée et

un *decoder* (décodeur) pour générer la séquence de sortie. L'*encoder* peut appliquer une stratégie de *span corruption*, où des segments continus de texte sont masqués et remplacés par des *sentinel tokens* (tokens sentinelles) uniques, tandis que le *decoder* apprend à reconstruire ces segments masqués. Cette technique améliore la compréhension du contexte et des dépendances séquentielles par le modèle, rendant ces modèles polyvalents pour des tâches comme la traduction, le résumé de texte*summarization*), et le *question answering* [31].

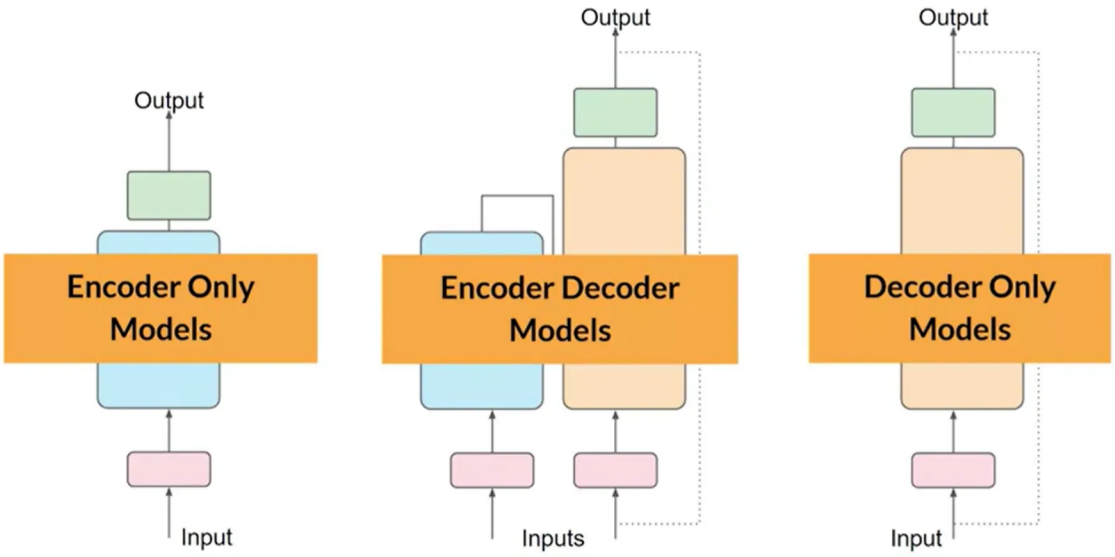


FIGURE 1.3 – Principales Approches Architecturales des LLM [32].

2.3 Applications des LLMs

La polyvalence des LLM a permis leur adoption généralisée dans un large éventail d'industries, chacune exploitant leurs capacités de génération et de résolution de problèmes de manière unique :

- **Service Client et Chatbots :** Les LLM améliorent le support client en alimentant des chatbots qui traitent les demandes et le dépannage avec des capacités conversationnelles semblables à celles des humains [30].
- **Création de Contenu et Marketing Numérique :** Les LLM automatisent la génération de contenu, améliorant la productivité et assurant une communication de marque cohérente sur toutes les plateformes [33].
- **Santé et Diagnostics :** Les LLM assistent dans les diagnostics et le résumé des dossiers médicaux, transformant la prise de décision et le reporting dans le secteur de la santé [34, 7, 35].
- **Programmation et Génération de Code :** Des LLM comme Codex génèrent, déboguent et optimisent le code à partir d'instructions en langage naturel, accélérant ainsi le développement logiciel [36].

- **Éducation et E-Learning :** Les LLM offrent un tutorat personnalisé, des retours instantanés et des supports d'étude, améliorant les expériences d'apprentissage pour les étudiants et les éducateurs [33].
- **Analyse Juridique et Financière :** Les LLM rationalisent l'analyse de documents, garantissant l'exactitude et la conformité tout en économisant du temps sur les tâches répétitives [30].
- **Industries Créatives :** Les LLM contribuent à la création de contenu artistique et enrichissent les expériences utilisateur dans les jeux vidéo grâce à des dialogues et des narrations dynamiques [33].
- **Recherche et Gestion des Connaissances :** Les LLM synthétisent des ensembles de données et résument le contenu académique, aidant les chercheurs à extraire des informations exploitables [37].

2.4 Modèles généraux des LLM

Au fil du temps, plusieurs modèles de LLM ont émergé, chacun apportant des avancées significatives et des fonctionnalités spécifiques adaptées aux tâches de traitement du langage naturel. La Figure 1.4 ci-dessous offre une cartographie visuelle de cette évolution.

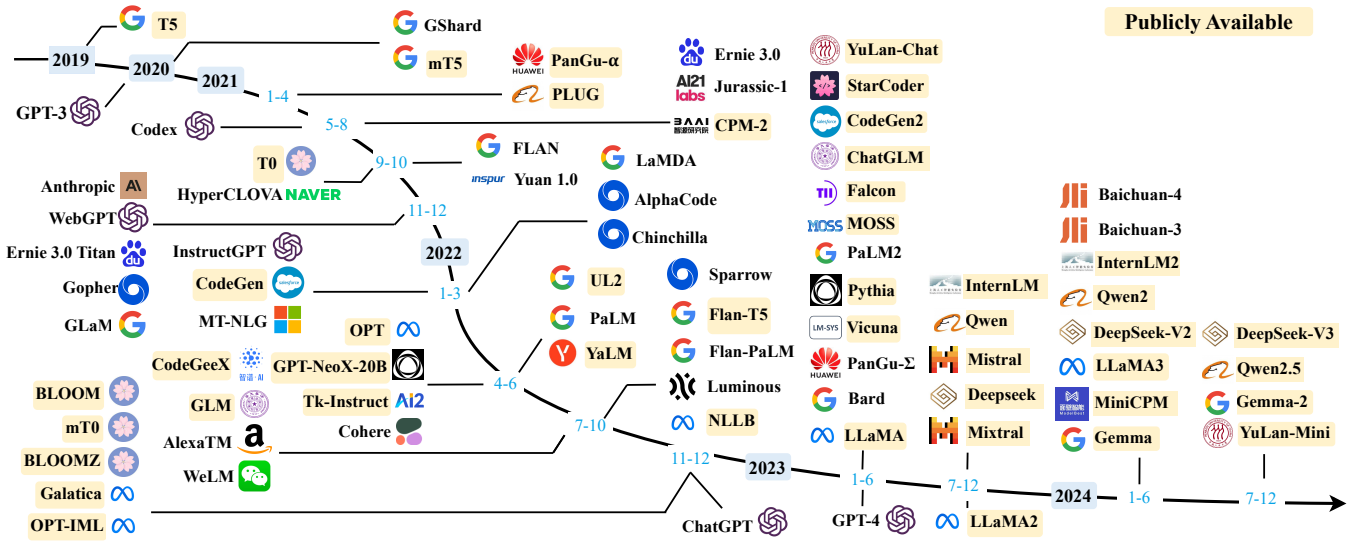


FIGURE 1.4 – Un diagramme montrant l'évolution des modèles de langage de grande taille (LLM) accessibles au public[13].

Parmi les modèles les plus influents, on retrouve :

- **BERT (Bidirectional Encoder Representations from Transformers)** : Lancé par Google en 2018, BERT [29] a marqué une rupture en conditionnant la compréhension du contexte d'un mot à la fois par les tokens précédents et suivants, grâce à son architecture d'encodeur et sa tâche de pré-entraînement par le *Masked Language Modeling* (MLM). Cette approche profondément bidirectionnelle le rend très performant pour l'analyse de sentiment, la recherche d'informations et la réponse aux questions [29]. Le tableau 1.1 ci-dessous compare plusieurs versions et dérivés notables de BERT, en détaillant leurs

spécificités architecturales et leurs caractéristiques.

Modèle (Année)	**Taille**	**Paramètres**	***Dataset***	**Caractéristiques**
BERT-Base (2018) [29]	12 couches, 768 dimensions	110M	BooksCorpus + English Wikipedia	Encodage bidirectionnel, amélioration du contexte des mots
BERT-Large (2018) [29]	24 couches, 1024 dimensions	340M	BooksCorpus + English Wikipedia	Meilleures performances que BERT-Base, mais plus coûteux en calcul
RoBERTa (2019) [38]	24 couches, 1024 dimensions	355M	Common Crawl + OpenWebText + BooksCorpus	Entraînement plus long, plus de données, suppression du NSP (Next Sentence Prediction)
DistilBERT (2019) [39]	6 couches, 768 dimensions	66M	Même jeu de données que BERT	Version allégée, plus rapide, moins de calcul tout en conservant 97% des performances de BERT-Base
ALBERT (2019) [40]	12 couches partagées, 128 dimensions	18M	BooksCorpus + Wikipedia	Compression des poids, performances similaires à BERT-Large avec moins de calcul

TABLE 1.1 – Comparaison des différentes versions du modèle BERT

- **Generative Pre-trained Transformer (GPT)** est une famille de modèles de langage développée par OpenAI. Il s'agit d'un modèle **auto-régressif** qui génère du texte en prédisant le mot suivant dans une séquence, en se basant sur un apprentissage préalable sur de vastes corpus de données textuelles. Il est largement utilisé pour des applications telles que la génération de texte, la réponse aux questions et l'assistance conversationnelle [41]. Le Tableau 1.2 ci-dessous compare plusieurs versions et dérivés notables du modèle GPT.

Modèle (Année)	**Taille (Contexte Max.)**	**Params.**	***Dataset***	**Caractéristiques Notables**	**Multi modal**
GPT-1 (2018) [42]	Décodeur 12 couches (512 tokens)	117M	BooksCorpus [2]	Pré-entraînement génératif sur Transformer pour modélisation du langage et *fine-tuning*.	Non

Suite à la page suivante

2. Le dataset BooksCorpus est une collection de textes de livres non publiés.

Table 1.2 – Suite de la page précédente

Modèle (Année)	**Taille (Contexte Max.)**	**Params**	***Dataset***	**Caractéristiques Notables**	**Multi modal**
GPT-2 (2019) [28]	Décodeur jusqu'à 48 couches (1024 tokens)	1.5B (version max.)	WebText [3]	Capacités de génération de texte améliorées, performance *zero-shot* [4] notable.	Non
GPT-3 (2020) [30]	Décodeur jusqu'à 96 couches (2048 tokens)	175B (version max.)	Mix : Common Crawl (filtré), WebText2, Books1, Books2, Wikipedia.	Excellentes capacités *few-shot* [5]*/zero-shot*, génération de texte de haute qualité, traduction, résumé.	Non
GPT-3.5 (2022)	Contexte jusqu'à 4k-16k tokens (selon version)	175B	Non spécifié	Précision et pertinence contextuelle améliorées, meilleur suivi des instructions, alignement.	Non
GPT-4 / GPT-4o (2023-2024) [43]	Contexte 8k, 32k, ou 128k tokens (selon version)	Non spécifié officiellement (> GPT-3)	Non spécifié	Raisonnement complexe, multimodalité, codage avancé, haute performance sur benchmarks.	Oui (Texte, Image, Audio pour GPT-4o)

Table 1.2 – Comparaison des différentes versions des modèles GPT

- **T5 (Text-to-Text Transfer Transformer)** : développé par Google en 2019 [37], reformule toutes les tâches de NLP sous un cadre unifié texte-à-texte, où l'entrée et la sortie sont toujours des chaînes de texte. Basé sur le modèle Transformer, il utilise le *self-attention* pour traiter les séquences de données. Il est pré-entraîné sur un large corpus de données appelé C4 [6][37]. Cependant, son entraînement est coûteux en ressources et ses résultats peuvent contenir des biais nécessitant un contrôle [13]. Le tableau 1.3 dresse une comparaison entre plusieurs versions et dérivés importants du modèle T5, en mettant en lumière leurs architectures, tailles, jeux de données et caractéristiques spécifiques.

3. WebText est un vaste corpus créé par OpenAI à partir de liens sortants de Reddit.
4. *zero-shot* : Capacité du modèle à effectuer une tâche pour laquelle il n'a reçu aucun exemple d'entraînement spécifique au préalable.
5. *few-shot* : Capacité du modèle à apprendre et performer sur une nouvelle tâche à partir de seulement quelques exemples illustratifs (typiquement de 1 à quelques dizaines).
6. `C4` dataset available on Hugging Face : `https://huggingface.co/datasets/allenai/c4` (accessed June 5, 2025).

Modèle (Année)	**Taille (Contexte Max.)**	**Params.**	***Dataset***	**Caractéristiques Notables**
T5-Base (2020) [37]	12 couches enc-déc, (pré-entraînement sur séquences de 512 tokens)	220M	C4 (Colossal Clean Crawled Corpus)	Cadre unifié text-to-text, pré-entraîné sur C4.
T5-Large (2020) [37]	24 couches enc-déc, (pré-entraînement sur séquences de 512 tokens)	770M	C4	Performances supérieures à T5-Base, plus coûteux en calcul.
T5-XL (2020) [37]	24 couches enc-déc, (pré-entraînement sur séquences de 512 tokens)	3B	C4	Version plus grande, plus performante sur tâches complexes.
T5-XXL (2020) [37]	24 couches enc-déc, (pré-entraînement sur séquences de 512 tokens)	11B	C4	Version T5 la plus puissante, importantes ressources computationnelles.
LongT5 (2022) [9]	Basé sur T5 (ex. Base, Large), contexte jusqu'à 16k tokens	Similaires aux versions T5	C4 (pré-entraînement)	Optimisé pour séquences longues (attention efficace ex : TGlobal), maintient performances sur tâches courtes.
mT5 (2021) [44]	Variable (ex. XXL : 24 couches enc-déc, contexte 512-1024 tokens	300M à 13B	mC4 (multilingual C4, 101 langues)	Version multilingue de T5, performances solides sur tâches multilingues.
FLAN-T5 (2022) [45]	Basé sur les tailles T5 (ex. XL, XXL), contexte variable selon tâche	80M à 11B	Modèles T5 (sur C4) *fine-tunés* sur un large éventail de tâches formulées en instructions.	Amélioration significative des capacités *zero-shot* grâce à *instruction fine-tuning*

TABLE 1.3 – Comparaison des différentes versions et dérivés du modèle T5

- **LLaMA (Large Language Model Meta AI)** : est un modèle de langage développé par Meta AI. Il repose sur une architecture de type Transformer avec une taille variant de 7 à 65 milliards de paramètres [46, 47]. Les principales différences entre Llama et l'architecture Transformer d'origine sont décrites dans le tableau 1.4.

Modèle (Année)	Taille (Contexte Max.)	Params.	*Dataset*	Caractéristiques Notables
Llama 1				
Llama-7B (2023) [46]	32 couches, dim. 4096 (2048 tokens)	7B	Mix de données publiques (1.4T tokens)	Modèle de base efficace, a prouvé l'importance du volume de données pour les "petits" modèles.
Llama-65B (2023) [46]	80 couches, dim. 8192 (2048 tokens)	65B	Mix de données publiques (1.4T tokens)	Version la plus puissante, compétitive avec des modèles plus grands comme GPT-3 à l'époque.
Llama 2				
Llama-2-7B (2023) [47]	32 couches, dim. 4096 (4096 tokens)	7B	Nouveau mix de données publiques (2T tokens)	Version populaire avec contexte doublé et un modèle "Chat" affiné (RLHF)[7] très performant.
Llama-2-70B (2023) [47]	80 couches, dim. 8192 (4096 tokens)	70B	Nouveau mix de données publiques (2T tokens)	Modèle le plus performant de sa génération, optimisé avec *Grouped-Query Attention (GQA)*[8].
Llama 3				
Llama-3-8B (2024) [50]	32 couches, dim. 4096 (contexte 8k tokens)	8B	Nouveau mix de données (>15T tokens)	Performances SOTA[9] pour sa taille. Utilise un nouveau tokenizer (128k vocab) et GQA.
Llama-3-70B (2024) [50]	80 couches, dim. 8192 (contexte 8k tokens)	70B	Nouveau mix de données (>15T tokens)	Compétitif avec les meilleurs modèles fermés de l'époque (ex : GPT-4). Raisonnement et codage améliorés.

TABLE 1.4 – Comparaison des versions clés des modèles Llama 1, 2 & 3

7. **RLHF (Reinforcement Learning from Human Feedback)** : Apprentissage par Renforcement à partir du Feedback Humain. C'est une technique où un modèle est affiné pour mieux s'aligner sur les préférences humaines (utilité, sécurité) en utilisant les évaluations et les corrections fournies par des évaluateurs humains [48].

8. **GQA (Grouped-Query Attention)** : Attention à Requêtes Groupées. Il s'agit d'une optimisation du mécanisme d'attention standard qui accélère la vitesse de génération du modèle (inférence) en réduisant l'utilisation de la mémoire, ce qui le rend plus efficace [49].

9. **SOTA** : Acronyme pour *State-of-the-Art*, qui signifie "État de l'art" en français.

3 Les LLM dans le Contexte Biomédical : Un État de l'Art

Le domaine médical constitue un champ d'application privilégié pour les modèles de langage de grande taille (LLM), en raison de la richesse, de la complexité et de la sensibilité des données qu'il génère. Entre les articles scientifiques, les dossiers médicaux électroniques, les protocoles cliniques ou encore les dialogues patients-médecins, le volume d'informations textuelles est considérable et souvent rédigé dans un langage technique et spécialisé. Les modèles génériques, bien qu'efficaces sur des tâches linguistiques générales, montrent leurs limites face à ce type de contenu. Cela a conduit à l'émergence de modèles spécialisés, pré-entraînés ou affinés spécifiquement sur des corpus biomédicaux, afin de mieux capturer les subtilités terminologiques, les relations cliniques et les structures narratives propres au domaine de la santé.

3.1 Les Modèles Spécialisés pour le Domaine Médical

De nombreux modèles de langage ont été adaptés au domaine médical à partir d'architectures généralistes comme BERT, T5 ou GPT. Cette spécialisation repose sur un pré-entraînement ou un affinement (fine-tuning) sur des corpus biomédicaux (PubMed, MIMIC-III, dialogues cliniques, etc.). Parmi ces modèles, on retrouve BioBERT, SciBERT, ClinicalT5, ou encore BioGPT, chacun étant optimisé pour des tâches telles que la reconnaissance d'entités nommées (NER), les questions-réponses (QA), ou l'analyse de textes cliniques. Le tableau ci-dessous compare les modèles les plus influents dans ce domaine.

Modèle (Année)	**Base**	**Param.**	**Corpus d'entraînement**	**Tâches**
BioBERT (2019)[34]	BERT	110M	18B tokens (PubMed + PMC)	NER, QA, Extraction de relations
PubMedBERT (2020)[35]	BERT	110M / 340M	3.2B tokens (PubMed + PMC)	NER, QA, Classification
ClinicalBERT (2019)[51]	BERT	110M	112k notes cliniques (MIMIC-III)	NER, QA, Analyse clinique
BioMed-RoBERTa (2020)[52]	RoBERTa	125M	7.55B tokens (S2ORC)	NER, QA, Classification
SciFive (2021)[53]	T5	220M / 770M	PubMed + PMC	QA, Résumé, Génération de texte
ClinicalT5 (2021)[54]	T5	220M / 770M	2M notes cliniques (MIMIC-III)	QA, Résumé, Génération de texte
BioGPT (2022)[55]	GPT	1.5B	15M articles (PubMed)	Génération de texte, QA
BioMedLM (2022)[56]	GPT	2.7B	110GB (Pile)	Génération de texte, QA

Modèle (Année)	Base	Param.	Corpus d'entraînement	Tâches
GatorTron (2022)[57]	GPT	8.9B	>82B tokens + EHRs	NER, QA, Analyse clinique
Med42 (2022)[58]	GPT	7B / 70B	250M tokens (PubMed + MedQA)	QA, Génération de texte
MEDITRON (2022)[59]	LLaMA2	7B / 70B	48.1B tokens (PubMed + guidelines)	QA, Génération de texte

TABLE 1.5 – Comparatif de modèles biomédicaux spécialisés

3.2 Les Datasets Médicaux de Référence

Les Datasets pour les modèles de langage médical (LLM) proviennent de diverses sources. Les plateformes open-source telles que Hugging Face et GitHub offrent un accès immédiat à des jeux de données pré-curés, tandis que des revues de littérature et des recherches sur Google permettent d'identifier des ressources supplémentaires. Ces stratégies garantissent une collecte large et complète couvrant un large éventail d'applications médicales. Un résumé de ces jeux de données est présenté dans le Tableau 1.6 [60].

Dataset	Type	Langue	Échelle	Description
MIMIC-III[61]	EHR	Anglais	58 K admissions	Données hospitalières détaillées pour 46 520 patients
MIMIC-IV[10]	EHR	Anglais	504 K admissions	Admissions de 2008 à 2019 avec une organisation modulaire
CPRD[62]	EHR	Anglais	2 K cabinets médicaux	Données de 60 millions de patients
PubMed[63]	Littérature	Anglais	36 M citations	Articles biomédicaux avec résumés
PMC[64]	Littérature	Anglais	8 M articles	Articles en texte intégral
RCT[65]	Littérature	Anglais	4 K résumés	Revues systématiques Cochrane
MS^2[66]	Littérature	Anglais	470 K résumés	Grand jeu de données de résumés multi-documents
SumPubMed[67]	Littérature	Anglais	33 K résumés	33 772 résumés d'articles biomédicaux
CORD-19[68]	Littérature	Anglais	1 M articles	Publications sur la COVID-19
OpenWebText[69]	Web	Anglais	38 Go texte	Alternative open-source à WebText
C4[70]	Web	Anglais	750 Go texte	750 Go de texte extrait du web

Dataset	Type	Langue	Échelle	Description
UMLS[71]	Base de connaissances	Anglais	2 M entités	900 K concepts médicaux normalisés
cMeKG[72]	Base de connaissances	Chinois	10 K maladies	Base de données chinoise sur les maladies et médicaments
DrugBank[73]	Base de connaissances	Anglais	16 K médicaments	Données sur les médicaments et leurs interactions
PubMedQA[74]	QA	Anglais	273 K paires QA	Inclut des annotations d'experts et des réponses générées
MedDialog-EN[75]	Dialogue	Anglais	260 K dialogues	Conversations patient-médecin
CheXpert[76]	Multimodal	Anglais	224 K radiographies	Radiographies thoraciques avec annotations
PathVQA[77]	Multimodal	Anglais	32 K paires QA	Images pathologiques avec questions-réponses

TABLE 1.6 – Un aperçu des Dataset les plus utilisés dans le domaine médical pour les LLMs[60].

3.3 Panorama des tâches des LLM en médecine

Les modèles de langage de grande taille (LLM) sont employés dans une variété de tâches en traitement automatique du langage appliqué à la médecine. Les principales catégories sont les suivantes :

- **Reconnaissance d'entités nommées (NER)** : il s'agit de l'extraction automatique d'entités biomédicales telles que les maladies, les médicaments ou les procédures à partir de textes cliniques ou scientifiques. Des systèmes basés sur BERT, par exemple, sont entraînés pour annoter les termes médicaux dans les dossiers patients [78].
- **Question-réponse (QA)** : cette tâche consiste à répondre à des questions cliniques ou scientifiques en s'appuyant sur la littérature médicale ou les dossiers de santé. Les modèles sont évalués sur des jeux de données comme PubMedQA [79] ou MedQA, qui nécessitent la recherche d'informations pertinentes dans des articles biomédicaux.
- **Classification de textes** : elle inclut la catégorisation de documents médicaux (ex. : type de pathologie, spécialité clinique, pronostic). Des modèles comme BERT peuvent être ajustés (fine-tuned) sur des corpus de type EHR pour prédire un diagnostic ou détecter une situation critique [80].
- **Génération de texte** : les LLM peuvent produire automatiquement du texte médical, tel que des rapports d'hospitalisation, des lettres de sortie, ou des prescriptions. Bien que ces modèles soient capables de générer un texte fluide et structuré, la garantie de véracité médicale reste un défi majeur.

- **Dialogue médical** : les LLM permettent également de concevoir des systèmes de dialogue interactif, utilisés comme chatbots médicaux pour assister les patients ou les professionnels de santé. Ces agents sont capables de répondre à des questions simples ou d'effectuer un triage initial.

En conclusion, la tâche de **résumé automatique (summarization)** émerge comme particulièrement prometteuse mais complexe. Par exemple, la génération de résumés à partir de rapports radiologiques a récemment été explorée [81]. Ce type de tâche exige de condenser des informations médicales critiques tout en préservant leur exactitude, ce qui en fait un véritable défi pour les LLM biomédicaux.

4 Le Résumé de Documents Médicaux par les LLM : Méthodes et Performances

Le résumé automatique de textes médicaux vise à aider les professionnels de santé en condensant des informations volumineuses (notes de service, résultats d'examens, articles de recherche) en synthèses concises. Dans un contexte de surcharge d'informations, cette technologie peut considérablement réduire la charge documentaire des cliniciens [82]. Par exemple, Van Veen et al. (2024) soulignent qu'analyser et résumer de vastes dossiers électroniques de santé (« EHR ») impose un fardeau important sur les cliniciens [82]. En fournissant aux médecins des synthèses automatiques, on peut libérer du temps consacré à la rédaction et minimiser les risques d'erreur lors de la transcription d'informations cruciales [83]. De plus, ces résumés peuvent améliorer la communication avec les patients : des études récentes montrent que des rapports médicaux **« orientés patient »**, rédigés dans un langage clair, favorisent l'adhésion au traitement et la compréhension de leur état de santé [84]. En radiologie par exemple, la génération de versions « patient-friendly » des comptes-rendus augmente la compréhension patient et la satisfaction, alors que les médecins manquent souvent de temps pour rédiger de tels résumés [84].

Il existe deux grandes familles de méthodes de résumé automatique : **extractif** et **abstractive**. Le résumé extractif sélectionne et assemble des fragments (phrases ou segments) du document original, tandis que le résumé abstractive génère de nouveaux énoncés qui paraphrasent le contenu source [12]. Les méthodes extractives garantissent la fidélité lexicale car elles recopient strictement le texte source, mais produisent souvent des résumés plus longs ou moins cohérents. Les méthodes abstractives, facilitées par les LLM, synthétisent l'information en reformulant ou en condensant les idées clés, ce qui tend à produire des résumés plus concis et fluides, au prix d'un risque accru d'hallucinations. Enfin, on distingue aussi le résumé de dossier patient (résumer plusieurs notes cliniques relatives à un même patient) du résumé **d'article scientifique** (condensation de la littérature biomédicale). Les premiers traitent des données narrative/historique (antécédents, bilans, prescriptions), tandis que les seconds traitent de textes formels structurés en sections (objectifs, méthodes, résultats). Ces tâches présentent des enjeux différents en termes de contenu, de style et d'audience, et les travaux en NLP médical s'attachent à développer des approches spécifiques pour chaque cas.

4.1 Méthodologies et Approches existantes

Les approches modernes exploitent massivement des modèles de langage pré-entraînés de grande taille. On utilise couramment des architectures *transformer* de type encodeur-décodeur (seq2seq) comme TS ou BART, ou des modèles auto-régressifs (comme GPT). Par exemple, Berg & Dalianis (2024) ont fine-tuné plusieurs variantes du modèle pré-entraîné BART sur des notes cliniques suédoises pour générer automatiquement des synthèses de rapports de sortie d'hôpital (discharge summaries) [85]. De même, Pal et al. (2023) ont comparé BART, TS et leurs variantes fine-tunées (dont Longformer et FLAN-T5) sur les résumés de dossiers de sortie de l'hôpital MIMIC-III, constatant que le modèle **FLAN-T5** spécialisé offrait les scores ROUGE les plus élevés (≈ 0.456) sur les tâches de résumé [83]. D'autres travaux ont développé des modèles pré-entraînés spécifiquement sur le domaine biomédical ou clinique : par exemple, le modèle ClinicalTS (Lu et al., 2022) est un TS générique pré-entraîné sur un large corpus de texte clinique [86]. De même, des variantes comme **BioGPT** ou **BioMedLM** ont été proposées pour le domaine biomédical. Ces modèles sont ensuite domain-adaptés (par fine-tuning) sur des données annotées de résumés médicaux afin d'améliorer leur pertinence sur ce type de texte.

La phase d'entraînement implique diverses stratégies. Classiquement, les LLM sont fine-tunés sur des paires texte original – résumé (par ex. notes cliniques et résumés d'examen) en minimisant l'erreur de prédiction séquence-à-séquence. Pour alléger le coût, on peut recourir à des techniques de fine-tuning *paramétrique efficace* : ainsi Hu et al. (2021) proposent l'approche **LoRA** (Low-Rank Adaptation) qui insère des couches de faible rang dans le modèle pré-entraîné pour l'adapter sans mettre à jour tous les paramètres [11]. Les plateformes modernes offrent également des méthodes de fine-tuning efficaces (par ex. Hugging Face PEFT). En complément, on explore les méta-approches : un pré-entraînement **supplémentaire** sur des corpus biomédicaux (« continued pre-training » [87]) avant fine-tuning peut améliorer les performances.

Enfin, les techniques de prompting ou d'apprentissage en contexte (« in-context learning ») sont de plus en plus utilisées avec les grands modèles non spécialisés (GPT-3, GPT-4, etc.). Dans ce cadre, on guide directement le LLM avec des instructions ou des exemples de résumé : par exemple, ChatGPT peut être sollicité en zero-shot pour générer un résumé à partir d'un prompt tel que « Based on the above report, summarize the key findings in plain language ». Tariq et al. (2024) ont ainsi utilisé GPT-3.5-turbo pour produire, en zero-shot, plusieurs formats (résumé succinct, rapport « facile-patient », recommandations) à partir de rapports radiologiques anonymisés [84]. Ces méthodes évitent la phase coûteuse de fine-tuning, mais leur qualité dépend de la formulation du prompt et du contexte fourni.

4.2 Évaluation des Performances

L'évaluation quantitative des résumés générés est classiquement fondée sur des métriques de similarité par chevauchement lexical. Les plus répandues sont **ROUGE** (Recall-Oriented Understudy for Gisting Evaluation) et **BLEU**. Par exemple, la métrique ROUGE-L évalue la longueur de la plus longue sous-séquence commune entre le résumé généré et la référence, tandis que BLEU calcule la précision des n-grammes [12, 88]. Ces scores varient de 0 (aucune

similarité) à 1 (identiques). Dans la pratique, un ROUGE-L élevé indique que le résumé contient la plupart des informations clés du texte source, même si elles sont reformulées différemment. Tang et al. (2023) observent ainsi que ChatGPT obtient des scores ROUGE-L relativement élevés sur des résumés de revues Cochrane, traduisant une bonne couverture du contenu, alors que son score BLEU reste faible (ce qui reflète un vocabulaire différent du texte de référence) [89].

Les autres métriques de surface incluent METEOR (qui pondère précision et rappel de mots-clés et de synonymes) et **CIDEr** (destiné aux résumés d'images, mais parfois adapté) ou encore la **similitude cosinus** entre vecteurs d'empreintes textuelles. Cependant, ces mesures n'évaluent pas la **fidélité factuelle** du résumé. Par exemple, un score ROUGE élevé ne garantit pas que les faits relatés sont corrects ou véridiques. Des travaux récents insistent sur la nécessité de métriques « factuelles » spécifiques : c'est le cas de FactCC (Kryściński et al., 2020) qui entraîne un classifier pour détecter les incohérences factuelles entre résumé et source [90]. D'autres approches comme **BERTScore** (Zhang et al., 2019, non cité ici) comparent les embeddings contextuels pour juger de la qualité sémantique.

4.2.1 Comparaison des performances

La littérature rapporte des résultats variés selon les tâches et les modèles. Par exemple, Pal et al. (2023) montrent qu'un modèle FLAN-T5 finement ajusté sur des sections de résumé de sortie d'hôpital (MIMIC-III) atteint un ROUGE d'environ 0.456 [83]. Dans un contexte radiologique, Nishio et al. (2023) rapportent qu'un LLM spécialisé en IRM neurologique, pré-entraîné et affiné pour résumer les conclusions d'IRM, obtient ROUGE-L ≈ 0.52 (BLEU-1=0.46, METEOR=0.28) [91]. Sur des dossiers radiologiques plus généraux, Van Veen et al. (2024) ont évalué une version adaptée de Llama2-13B sur des rapports de radiologie (jeu de données Open-I) et obtenu ROUGE-L ≈ 0.355 lorsque le modèle était invité à se comporter en « clinicien expert » par prompt [82].

Table 1.7 – Résultats de performances de résumé médical

Modèle (approche)	**Données / Tâche**	**ROUGE-L (≈)**	**Source**
FLAN-T5 (fine-tuned)	Section « Historique » des notes de sortie (MIMIC-III)	0.456	[83]
LLaMA2-13B (prompt « expert médical » avec LoRA)	Synthèse de rapports radiologiques (Open-I)	0.355	[82]
T5/BART fine-tuné (neuroradiologie)	Conclusions d'IRM neurologique (données radiologiques)	0.520	[91]

(Dans tous les cas, des scores ROUGE-L autour de 0.3–0.5 reflètent une concordance partielle du contenu, soulignant la difficulté du résumé automatique dans le domaine médical.)

4.3 Défis et Limites Actuels du Résumé Médical

Les principaux défis du résumé automatique en milieu médical sont liés à la **fidélité factuelle**, aux **longues séquences contextuelles** et aux contraintes de confidentialité.

- **Fidélité factuelle et hallucinations.** Les modèles génératifs, surtout en mode *abstractive*, peuvent introduire des informations erronées (« hallucinations ») non présentes dans le texte source. Ce risque est particulièrement critique en médecine, où une fausse information peut avoir des conséquences graves. Les LLMs sont en effet « très habiles » pour produire des énoncés plausibles mais non fondés [82]. Kryściński et al. (2020) soulignent que les mesures traditionnelles (ROUGE, BLEU) ne capturent pas ces erreurs, d'où l'importance d'évaluations orientées factualité [90]. Des études récentes montrent néanmoins que la confiance interne du modèle ou des techniques de vérification automatique peuvent aider à identifier les hallucinations. Néanmoins, garantir qu'un résumé reste véridique et complet constitue un défi ouvert du domaine.
- **Longueur et complexité des dossiers.** Les **dossiers médicaux électroniques (EHR)** sont souvent très volumineux (centaines de pages), comprenant de multiples notes de différents spécialistes et formats (SOAP notes, résultats d'examens, prescriptions). Exiger qu'un LLM traite l'intégralité de ce contexte dépasse généralement sa fenêtre contextuelle native. Les approches hybrides (par ex. résumé hiérarchique, extraction préalable des informations clés, RAG – *Retrieval-Augmented Generation*) sont explorées pour gérer ces contextes étendus. Par ailleurs, la richesse sémantique et la complexité linguistique du domaine médical – abréviations, termes techniques, formulations standardisées – requiert souvent des modèles pré-entraînés ou adaptés au contexte biomédical. Un modèle généraliste peut avoir du mal à interpréter correctement les acronymes ou à relier des termes cliniques.
- **Confidentialité et anonymisation.** Les données médicales contiennent des informations personnelles sensibles. Toute application de résumé automatique doit impérativement protéger la confidentialité des patients. En pratique, les données utilisées pour l'entraînement ou l'évaluation doivent être dé-identifiées (suppression des noms, numéros de sécurité sociale, etc.) [84]. Par exemple, Tariq et al. (2024) précisent avoir anonymisé les rapports radiologiques (ôtant les identifiants, codes patients et dates) pour se conformer aux régulations HIPAA [84]. Cette contrainte limite l'accès aux données médicales et augmente la complexité de la recherche (les corpus doivent être soigneusement nettoyés avant d'être partagés).
- **Limites des métriques d'évaluation.** Enfin, les métriques automatiques elles-mêmes montrent leurs limites pour juger la qualité du résumé. Outre leur aveuglement aux erreurs factuelles, elles ne mesurent pas l'utilité clinique du résumé ni la satisfaction du médecin. Des études cliniques comparant LLM et professionnels ont révélé que, même lorsque les scores ROUGE sont proches, les médecins évaluent la concision et la sécurité d'emploi d'une manière plus nuancée. Ainsi, de futures recherches doivent proposer des indicateurs plus adaptés (p. ex. évaluation par experts, métriques cliniquement informées) pour vraiment évaluer l'efficacité des résumés en milieu de soins.

En résumé, malgré les progrès spectaculaires des LLM, le résumé automatique en médecine reste un domaine où rigueur et prudence sont de mise. Les modèles offrent un potentiel pour soulager la charge de travail et améliorer la communication, mais ils doivent être utilisés en complément de l'expertise humaine, avec une attention particulière aux erreurs factuelles, au traitement des longs dossiers médicaux et à la préservation de la confidentialité des données.

5 Conclusion

En synthèse, ce chapitre a établi que si les modèles de langage de grande taille (LLM) offrent un potentiel immense pour le résumé de textes médicaux, leur application se heurte à des défis majeurs. L'état de l'art montre que la **fidélité factuelle**, la gestion des **contextes longs et complexes** des dossiers patients, et la **spécialisation au domaine médical** sont les principaux verrous à lever pour une application clinique fiable.

Face à ces constats, notre travail propose une approche pragmatique et ciblée pour répondre à ces défis. Pour adresser le problème des textes longs, nous avons sélectionné le modèle **LongT5**, spécifiquement conçu pour cette tâche. Pour garantir la pertinence clinique et la qualité de la supervision, nous avons mis en œuvre une méthodologie de prétraitement de données rigoureuse sur le corpus **MIMIC-IV**. Enfin, pour spécialiser le modèle de manière efficace, nous avons eu recours à la technique de fine-tuning optimisé **LoRA**.

Le chapitre suivant détaillera chacune de ces étapes méthodologiques, de la préparation des données à la configuration précise du modèle en vue de son entraînement.

Chapitre 2

Méthodologie pour le résumé automatique de notes cliniques

1 Introduction

Ce chapitre présente l'ensemble des étapes méthodologiques mises en œuvre pour développer un système de résumé automatique à partir des notes de sortie hospitalières issues du jeu de données MIMIC-IV. L'objectif principal est de générer automatiquement des résumés cohérents et informatifs de la section *Brief Hospital Course*, en s'appuyant uniquement sur les sections cliniques pertinentes contenues dans chaque document médical.

La méthodologie repose sur une chaîne de traitement structurée, depuis l'extraction des données brutes jusqu'à l'utilisation effective du modèle entraîné. Elle se compose des phases suivantes :

- **Prétraitement des textes** : cette première phase inclut le nettoyage des notes cliniques, l'extraction des sections médicales essentielles, le filtrage des documents trop longs ou incomplets, ainsi qu'un reformatage contrôlé des résumés cibles, afin de construire un corpus supervisé adapté à l'apprentissage automatique.
- **Préparation des données et adaptation du modèle** : une fois le corpus textuel prétraité, il est transformé en représentations numériques via la tokenisation (SentencePiece), puis utilisé pour ajuster un modèle pré-entraîné de type LongT5 à l'aide de la méthode LoRA (Low-Rank Adaptation). Une configuration spécifique de fine-tuning est appliquée pour optimiser les performances.
- **Évaluation et interaction utilisateur** : une fois le modèle fine-tuné, il est capable de générer des résumés cliniques à partir de nouvelles entrées, dans un cadre d'utilisation réelle où un utilisateur peut interagir avec le système via des requêtes.

La Figure 2.1 ci-dessous offre une vue d'ensemble du pipeline complet, de la préparation des données à l'inférence finale.

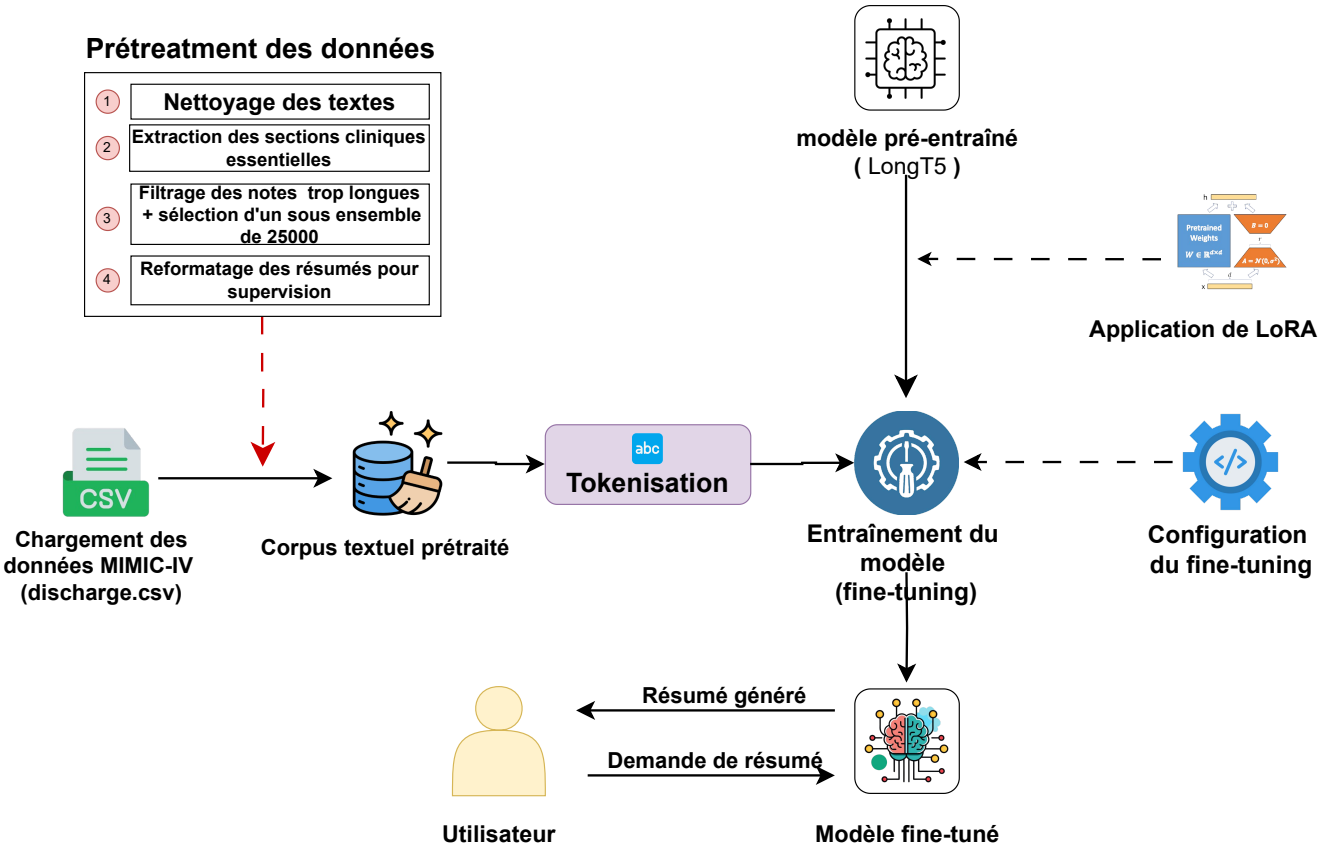


FIGURE 2.1 – Schéma global de la méthodologie de résumé automatique des notes cliniques.

Ce schéma illustre le processus méthodologique adopté. À partir des fichiers `discharge.csv` du dataset MIMIC-IV, un prétraitement structuré est appliqué pour nettoyer les textes, extraire les sections pertinentes et générer les résumés cibles. Le corpus ainsi constitué est ensuite tokenisé afin de produire des séquences numériques compatibles avec le modèle. L'entraînement est réalisé à l'aide d'un modèle LongT5 pré-entraîné, adapté à la tâche via LoRA et une configuration fine-tuning spécifique. Une fois le modèle entraîné, celui-ci peut être utilisé pour générer des résumés à la demande dans un cadre d'interaction utilisateur.

2 Présentation du *dataset* utilisé

Dans ce mémoire, nous utilisons la base **MIMIC-IV** comme source principale de données cliniques afin de proposer notre système de résumé des textes médicaux. La *dataset* **MIMIC-IV** (*Medical Information Mart for Intensive Care*) est une base de données cliniques librement accessible contenant des données réelles de patients hospitalisés, recueillies au *BIDMC* entre 2008 et 2022 [10]. Elle est disponible sur la plateforme **PhysioNet** [1].

La version actuelle (v3.1, octobre 2024) comprend environ 364 627 patients, 546 028 hospitalisations et 94 458 séjours en soins intensifs. Toutes les données identifiantes ont été supprimées

1. **MIMIC-IV :** `https://physionet.org/content/mimiciv/3.1/`

conformément à la législation américaine HIPAA (Health Insurance Portability and Accountability Act) [92].

Il faut noter que l'accès à cette base est libre mais soumis à une procédure de validation éthique. Il est nécessaire d'obtenir une accréditation via la plateforme **PhysioNet**, incluant la validation d'une formation à la recherche sur données de santé et la signature d'un accord d'utilisation. Une fois ces étapes validées, les données peuvent être téléchargées directement depuis le portail PhysioNet.

2.1 Processus de développement et structure du *dataset* MIMIC-IV

Processus de développement La figure 2.2 illustre le processus de développement de MIMIC-IV. Les données brutes sont d'abord acquises à partir de plusieurs systèmes hospitaliers : l'entrepôt de données du BIDMC [2], le système d'information des soins intensifs (MetaVision [3]) et d'autres sources externes. Ensuite, lors d'une étape de transformation, des scripts en langage SQL sont utilisés pour fusionner ces sources hétérogènes en un schéma relationnel unique et cohérent [92].

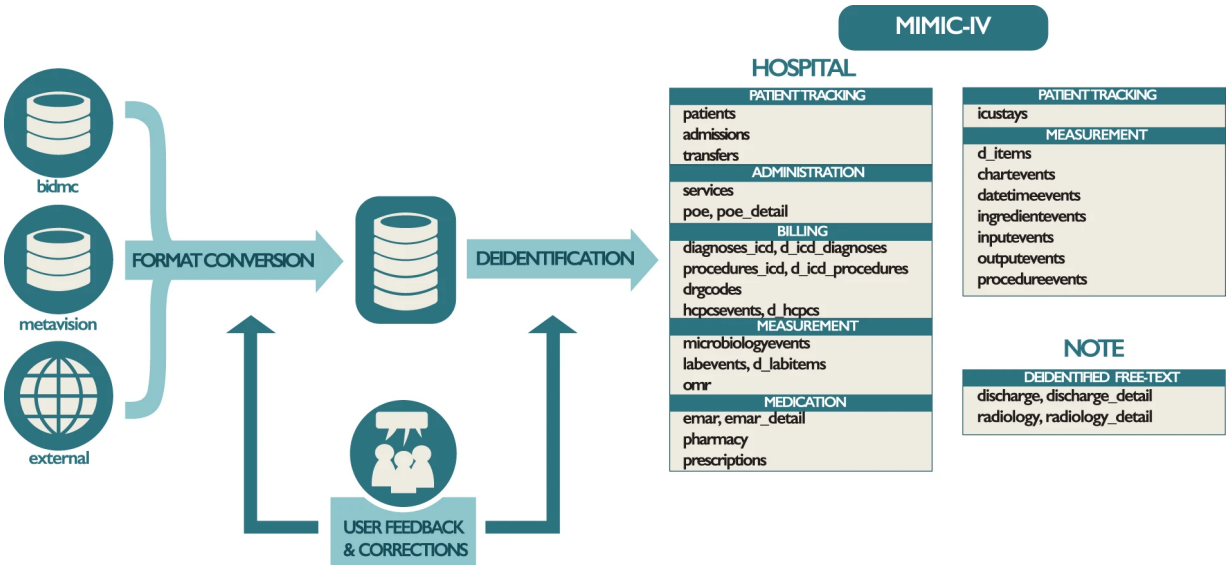


FIGURE 2.2 – Processus de développement de la base MIMIC-IV depuis les données hospitalières jusqu'à la structuration finale [92].

Un processus rigoureux de dé-identification est appliqué à MIMIC-IV afin de respecter la réglementation HIPAA, qui impose la suppression de 18 types d'identifiants personnels. Les informations sensibles sont remplacées par des codes alphanumériques (par exemple, subject_d pour les patients et hadm_id pour les séjours), les dates sont décalées pour masquer l'année réelle tout en conservant les intervalles, et les âges des patients de plus de 89 ans sont regroupés. Dans les notes cliniques, les éléments sensibles sont remplacés par "_____", garantissant l'anonymisation des textes. Enfin, un accord d'utilisation (DUA) interdit toute tentative de ré-identification [92].

2. **BIDMC :** C'est le centre médical de Boston où les données de la base MIMIC-IV ont été collectées.
3. **MetaVision :** Le système d'information clinique utilisé dans les unités de soins intensifs (ICU) du BIDMC pour enregistrer les données des patients en temps réel.

Les enregistrements ainsi anonymisés sont répartis dans des modules liés par clé, et des corrections peuvent être appliquées au fur et à mesure des *feedbacks* d'expérience utilisateur [92].

Structure du *dataset* Le *dataset* MIMIC-IV est structuré en plusieurs modules thématiques (Hospital, ICU, Note, etc.). Cette architecture modulaire et relationnelle permet de lier facilement les données d'une même hospitalisation à travers les différentes tables grâce à des clés communes comme subject_id et hadm_id [93, 92]. Les principaux modules sont :

- Le module **Hospital (HOSP)** contient les informations administratives et hospitalières générales, notamment les tables `patients`, `admissions`, `transfers`, ainsi que les analyses de laboratoire (`labevents`), les cultures microbiologiques, les prescriptions, les diagnostics et procédures codées [93, 94].
- Le module **ICU** regroupe les données collectées au lit du patient en soins intensifs : observations physiologiques chronométrées, perfusions intraveineuses, bilans sortants, traitements en cours [93, 94].
- Le module **Note** regroupe les textes cliniques dé-identifiés, incluant les résumés de sortie ('discharge'), les comptes rendus de radiologie ('radiology') et leurs tables de détail associées ('discharge_detail', 'radiology_detail') [93, 95].

D'autres modules spécialisés complètent la base comme le module **Chest X-ray (MIMIC-CXR)**, qui contient plus de 370 000 images radiographiques thoraciques associées à des comptes rendus en texte libre [92, 96]. Deux autres modules notables sont également disponibles : le module **Emergency Department**, qui regroupe les signes vitaux, les diagnostics et les données de triage, et le module **Diagnostic Electrocardiogram**, qui fournit des enregistrements ECG de 10 secondes sur 12 dérivations [92].

Le schéma de la figure 2.3 présente les principaux modules de MIMIC-IV et leur contenu. Grâce à son organisation relationnelle, il est possible de relier efficacement les différents types de données cliniques d'un patient (par exemple, associer un dosage biologique à une hospitalisation puis à une note de sortie), ce qui constitue un atout majeur pour les applications en **intelligence artificielle médicale** [92].

2.2 Sélection des données depuis MIMIC-IV

Pour cette étude, notre corpus textuel repose exclusivement sur le fichier `discharge.csv`, qui constitue une composante centrale du module `note` de la base MIMIC-IV (mimic-iv-note). Ce fichier contient **331 794 notes de sortie** (*discharge summary notes*) rédigés entre 2008 et 2019 pour **145 915 patients** hospitalisés ou passés par les urgences. Chaque document est un texte narratif long, entièrement dé-identifié, et contenant en moyenne 2 267 tokens [93, 97, 98].

Ces notes sont précieuses pour les tâches de traitement du langage naturel (NLP) en contexte clinique, car elles offrent une source riche de données textuelles réelles, jusque-là peu accessibles dans ce domaine. Le fichier est structuré en plusieurs colonnes, dont les plus pertinentes pour notre analyse sont résumées dans le tableau 2.1.

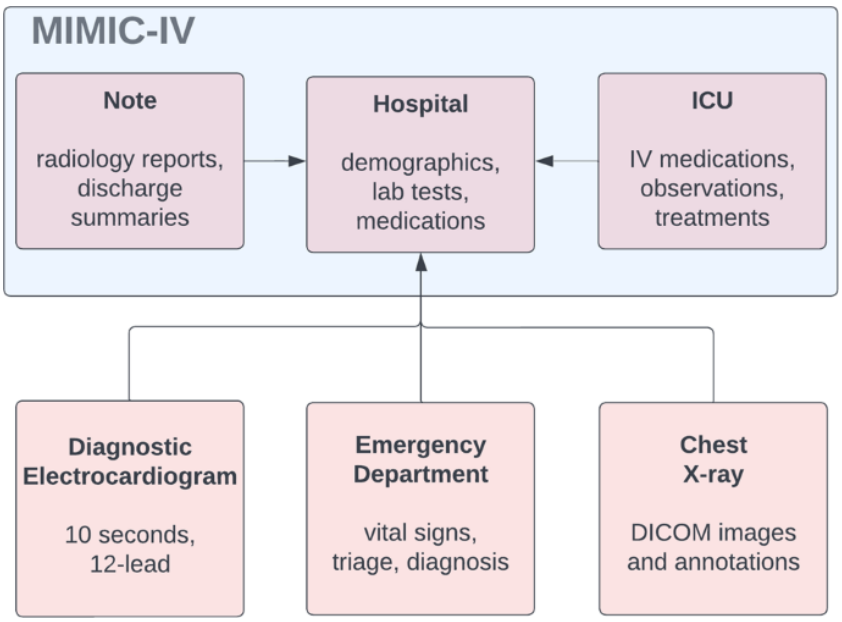


FIGURE 2.3 – Structure modulaire de la base MIMIC-IV avec les principaux modules cliniques et administratifs [92].

TABLE 2.1 – Description des colonnes principales du fichier `discharge.csv`

Nom de la Colonne	Description
`note_id`	Identifiant unique pour chaque note clinique.
`subject_id`	Identifiant unique du patient, servant de clé étrangère vers la table des patients.
`hadm_id`	Identifiant unique de l'hospitalisation, servant de clé étrangère vers la table des admissions.
`note_type`	Type de la note. Pour ce fichier, la valeur est toujours 'DS' (Discharge Summary).
`note_seq`	Numéro de séquence de la note pour un patient lors d'un même séjour.
`charttime`	Date et heure associées au contenu de la note (généralement la date de rédaction).
`storetime`	Date et heure de l'enregistrement de la note dans le système informatique.
`text`	Contenu textuel intégral du résumé de sortie. C'est la colonne centrale de notre étude.

La colonne la plus importante pour notre travail est `text`, car elle contient le contenu narratif complet du résumé de sortie. Ce texte n'est pas un bloc monolithique ; il est lui-même organisé en plusieurs sections cliniques, bien que leur présence et leur ordre puissent varier. Les sections les plus fréquemment retrouvées sont décrites dans le tableau 2.2 :

TABLE 2.2 – Sections pertinentes extraites des notes de sortie MIMIC-IV

Nom de la section (FR)	**Nom dans la base (MIMIC-IV)**	**Description**
Nom du patient	`Name`	Identifiant patient (anonymisé, remplacé par un code).
Numéro d'unité	`Unit No`	Code d'identification de l'unité de soins.
Date d'admission	`Admission Date`	Date d'entrée à l'hôpital.
Date de sortie	`Discharge Date`	Date officielle de sortie du patient.
Sexe	`Sex`	Sexe administratif du patient.
Service hospitalier	`Service`	Spécialité médicale ayant géré le séjour (ex. MEDICINE).
Motif de consultation	`Chief Complaint`	Raison principale ayant motivé l'admission.
Histoire de la maladie actuelle	`History of Present Illness`	Détails cliniques sur la progression récente des symptômes.
Procédures chirurgicales	`Major Surgical or Invasive Procedure`	Actes chirurgicaux ou invasifs effectués pendant l'hospitalisation.
Antécédents médicaux	`Past Medical History`	Résumé des conditions médicales chroniques.
Antécédents familiaux	`Family History`	Données héréditaires ou antécédents familiaux.
Antécédents sociaux	`Social History`	Comportements de vie, addiction, statut social.
Examen clinique à l'admission	`Admission Physical Exam`	Bilan clinique à l'entrée du patient.
Examen clinique de sortie	`Discharge Physical Exam`	Résumé de l'examen clinique final avant la sortie.
Résultats pertinents	`Pertinent Results`	Résultats biologiques ou radiologiques significatifs.
Examens complémentaires	`Imaging/Studies`	Compte-rendus d'imagerie ou autres examens diagnostiques.
Cours hospitalier bref	`Brief Hospital Course`	Description synthétique de l'évolution pendant le séjour.
Diagnostic de sortie	`Discharge Diagnosis`	Diagnoses retenues au moment de la sortie.
Médicaments à la sortie	`Discharge Medications`	Liste des traitements prescrits après la sortie.
Instructions de sortie	`Discharge Instructions`	Recommandations de suivi post-hospitalier.

2.3 *Brief Hospital Course* comme cible du résumé automatique

Dans les résumés de sortie hospitaliers, la section `Brief Hospital Course` (BHC) joue un rôle essentiel. Elle résume de manière concise le déroulement de l'hospitalisation, les décisions médicales prises, les réponses aux traitements, et les complications éventuelles. Cette section permet aux professionnels de santé de comprendre rapidement l'évolution du patient sans lire l'intégralité du dossier.

Grâce à sa structure synthétique, cette section est très utile pour :

- comprendre les faits marquants de l'hospitalisation,
- suivre la réponse du patient aux traitements,
- guider le suivi médical après la sortie.

Dans ce travail, nous avons choisi de nous concentrer sur la section *Brief Hospital Course* comme cible principale (*target*) pour l'entraînement du modèle de résumé automatique.

Exemple de section *Brief Hospital Course* :

```
Brief Hospital Course: HCV cirrhosis c/b ascites, HIV on ART, h/o IVDU,
COPD, bipolar, PTSD, presented from OSH ED with worsening abd distension
over past week and confusion. # Ascites - worsening abd distension and
discomfort. Likely due to portal HTN. Non-compliance with meds and diet.
SBP negative. Diuretics: furosemide 40 mg PO daily, spironolactone 50 mg PO
daily. Good response. Scheduled PCP and liver clinic follow-up.
```

Analyse rapide :

- Décrit le motif principal d'hospitalisation (ascite, confusion).
- Résume la cause probable (hypertension portale) et le traitement.
- Mentionne les médicaments utilisés et la réponse du patient.
- Termine par les recommandations de suivi.

Ce choix de la section *Brief Hospital Course* comme cible du résumé repose sur sa richesse informative et sa structure condensée, qui en font une référence naturelle pour entraîner un modèle de synthèse clinique. Après avoir défini précisément les données d'entrée et de sortie, la prochaine étape consiste à préparer ces textes de manière adéquate. La section suivante détaille ainsi le pipeline de prétraitement mis en place pour normaliser, filtrer et structurer les données en vue de leur exploitation par un modèle de type séquence-à-séquence.

3 Choix et Adaptation du Modèle LongT5

Dans ce travail, nous avons choisi, adapté et entraîné un modèle LLM pour produire des résumés cliniques à partir de notes hospitalières longues ; cette section détaille le modèle utilisé (LongT5), les raisons de ce choix, et la méthode LoRA employée pour un fine-tuning efficace. La figure 2.4 retrace les principales étapes de construction du modèle utilisé, depuis le pré-entraînement de LongT5 sur C4, jusqu'à son adaptation légère via LoRA avant fine-tuning.

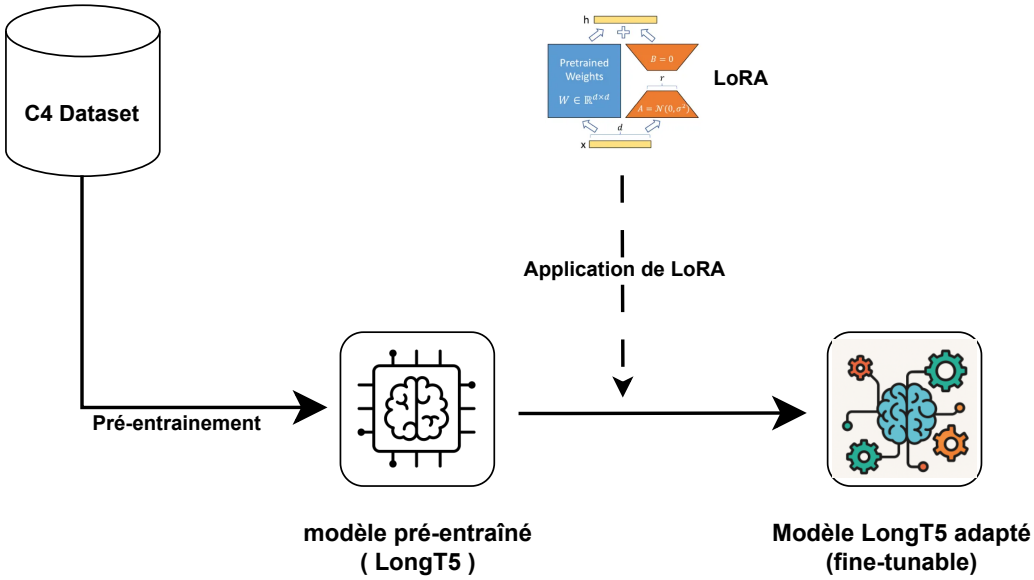


FIGURE 2.4 – Pré-entraînement du modèle LongT5 sur le corpus C4, suivi de son adaptation par LoRA.

3.1 Présentation du modèle LongT5

LongT5 [9] est une extension du modèle **T5** [37], un transformeur encodeur–décodeur pré-entraîné en mode *text-to-text*. Comme T5, il adopte une architecture à double composant : un **encodeur** qui traite le texte d'entrée et un **décodeur** qui génère la sortie (résumé, traduction, etc.). La principale modification apportée par LongT5 réside dans **ses mécanismes d'attention pour gérer de très longs textes** [9].

Pour ce faire, LongT5 combine deux types de mécanismes d'attention (figure 2.5) dans son encodeur pour remplacer l'attention complète et coûteuse du T5 standard :

1. **Attention locale :** Dans l'encodeur de LongT5, chaque token n'« attend » qu'un rayon fixe de voisins (par exemple ±127 tokens) au lieu de tous les tokens comme dans l'attention complète. Ce masque à fenêtres glissantes réduit la complexité en limitant le contexte immédiat de chaque mot [9].
2. **Attention Transient-Global (TGlobal) :** Pour capter un contexte plus large, LongT5 introduit un mécanisme global. On découpe la séquence en blocs de taille fixe (p.ex. 16 tokens) et on calcule un token global par bloc (somme normalisée des embeddings du bloc). Chaque token d'entrée peut alors assister non seulement ses voisins locaux, mais aussi tous les tokens globaux, offrant ainsi une vue d'ensemble du document [9].

pré-entraînement du LongT5 LongT5 conserve les mêmes objectifs de pré-entraînement que T5, à savoir une tâche de **corruption de spans** (*span corruption*), où des segments entiers de texte sont masqués puis reconstruits par le modèle, favorisant ainsi une compréhension profonde du contexte. Il est pré-entraîné sur le **corpus C4 (Colossal Clean Crawled Corpus)** [99], un vaste ensemble de données textuelles web nettoyées [9].

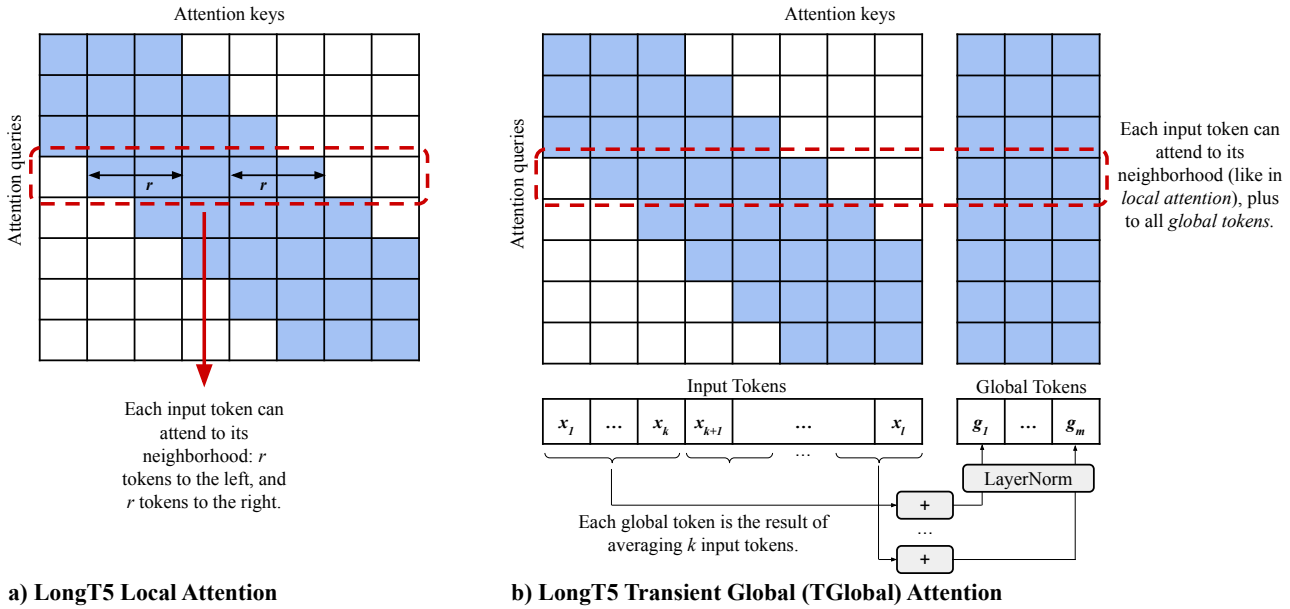


FIGURE 2.5 – Illustration des deux mécanismes d'attention testés dans LongT5 [9].

Capacités de longueur d'entrée et de sortie Dans le cadre de leur expérimentation, les auteurs de LongT5 ont évalué leur modèle avec des longueurs d'entrée allant jusqu'à **16384 tokens**, ce qui constitue une avancée significative par rapport aux modèles T5 classiques, souvent limités à 512 ou 1024 tokens [37]. Cette capacité permet à LongT5 de traiter des documents beaucoup plus longs, tout en maintenant de bonnes performances. Concernant la longueur des sorties, bien que le modèle puisse théoriquement générer jusqu'à **910 tokens**, les tâches de résumé ou de question-réponse s'appuient généralement sur des sorties beaucoup plus courtes qui variant selon le besoin (**512 tokens pour les résumés**, 128 tokens pour la QA) [9].

3.2 Pourquoi LongT5 ?

Le choix du modèle LongT5 pour cette étude repose sur sa capacité à répondre aux défis spécifiques posés par le résumé de documents médicaux, qui sont souvent longs et denses en informations. Plusieurs raisons motivent ce choix :

- **Gestion des séquences longues :** Les notes cliniques, dépassent fréquemment la limite de 512 tokens des transformeurs classiques. Grâce à ses mécanismes d'attention locale et globale, LongT5 est nativement capable de traiter des documents beaucoup plus longs, ce qui est essentiel pour analyser l'intégralité d'un rapport médical sans perte d'information.
- **Architecture spécialisée pour le résumé :** LongT5 hérite du framework *text-to-text* de T5, qui est particulièrement adapté aux tâches de génération comme le résumé. De plus, sa méthode de pré-entraînement, inspirée de PEGASUS [4] [100], consiste à reconstruire des phrases importantes masquées, ce qui l'entraîne directement à identifier et à générer des contenus synthétiques.

4. PEGASUS (Pre-training with Extracted Gap-sentences for Abstractive SUmmarization) est un modèle de type Transformer pré-entraîné pour la génération de résumés de texte. Il apprend à générer les phrases importantes d'un texte après les avoir supprimées du document original [100].

- **Performances de pointe démontrées :** Des études ont montré que LongT5 atteint des performances de l'état de l'art sur plusieurs benchmarks de résumé de textes longs et de question-réponse. Cette efficacité prouvée sur des tâches similaires nous conforte dans son potentiel à générer des résumés cliniques de haute qualité [9].

En somme, LongT5 allie une architecture flexible à des mécanismes d'attention conçus pour les textes longs et à un pré-entraînement orienté résumé. Ces caractéristiques en font un candidat idéal pour notre tâche : générer des résumés concis et pertinents à partir de notes hospitalières complexes.

3.3 Adaptation du LongT5 par LoRA

La méthode **LoRA** (**Lo**w-**R**ank **A**daptation), introduite par Hu et al. (2021), vise à rendre le fine-tuning de grands modèles (comme LongT5) plus efficace et plus rapide [11]. Nous utilisons LoRA pour adapter efficacement le modèle LongT5, en réduisant le nombre de paramètres à entraîner tout en maintenant des performances comparables au fine-tuning complet. De plus, LoRA n'introduit aucun surcoût d'inférence, tout en atteignant une qualité équivalente, voire supérieure, sur des tâches comme le résumé automatique [11].

3.3.1 Fonctionnement de LoRA

Le fonctionnement de LoRA peut être résumé en quatre étapes principales :

1. **Geler les poids d'origine :** Le principe de base est de geler (*to freeze*) la quasi-totalité des poids (*weights*) du modèle. Ces millions de paramètres d'origine ne sont donc pas modifiés pendant le processus d'entraînement.
2. **Injecter de petites matrices :** Pour certaines couches stratégiques du modèle, LoRA injecte deux petites low-rank adaptation matrices, généralement nommées A et B.
3. **Entraîner uniquement l'adaptation :** Durant le *training*, seules ces nouvelles et petites matrices A et B sont entraînées. Comme elles contiennent très peu de paramètres par rapport au modèle entier, cette étape demande beaucoup moins de mémoire et de puissance de calcul.
4. **Appliquer une mise à jour légère :** La correction ou l'adaptation (*update*) apportée par LoRA est le résultat de la multiplication de ces deux petites matrices ($B * A$). Cet ajustement est ensuite simplement ajouté au résultat de la couche d'origine, qui était restée gelée.

La figure 2.6 illustre bien cette différence : à gauche, une mise à jour complète et coûteuse de tous les poids (*full fine-tuning*), et à droite, l'ajout d'une petite adaptation ciblée et efficace avec LoRA.

3.3.2 Configuration légère de LoRA pour LongT5

Pour adapter le modèle LongT5 à la tâche de génération de résumés médicaux, nous avons utilisé LoRA avec une configuration simple et efficace. Elle permet de réduire drastiquement le nombre de paramètres à entraîner tout en conservant de bonnes performances.

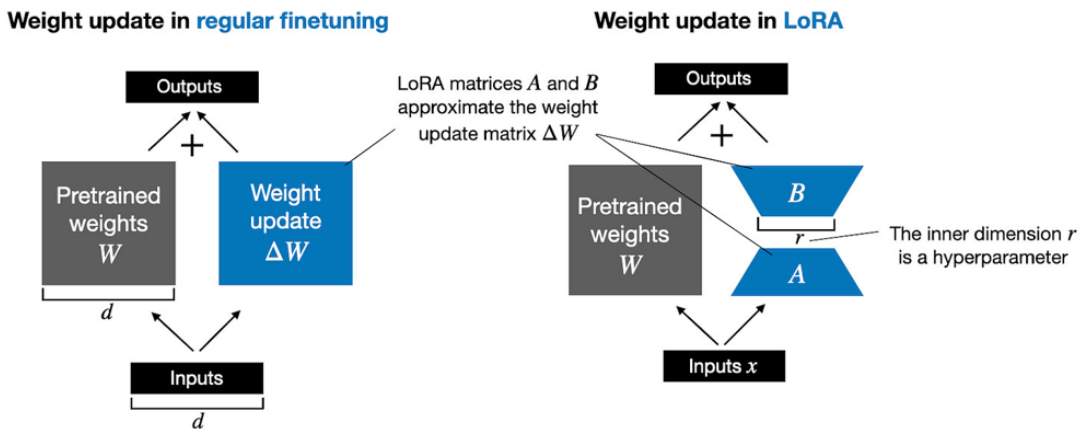


FIGURE 2.6 – Mécanisme LoRA – comparaison entre une mise à jour complète des poids (*full fine-tuning*, à gauche) et l'adaptation par matrices de faible rang (LoRA, à droite) [101].

Les choix d'hyperparamètres sont les suivants :

- **Rang** $r = 8$: ce petit rang contrôle la capacité d'adaptation. Une valeur faible permet de limiter le nombre de paramètres tout en capturant des mises à jour utiles.
- **Facteur d'échelle** $\alpha = 16$: il amplifie les mises à jour produites par LoRA, compensant la faible dimension du rang.
- **Dropout = 0.1** : introduit une régularisation pour éviter le surapprentissage.
- **Cibles : matrices Q et V** : appliquer LoRA uniquement aux matrices de requêtes (Q) [5] et de valeurs (V) [6] est un bon compromis entre efficacité et impact, selon les recommandations de [11].

La configuration LoRA utilisée dans ce travail permet une réduction drastique du nombre de paramètres mis à jour pendant l'entraînement. Voici la répartition des paramètres :

- **Paramètres totaux du modèle :** 297 820 800
- **Paramètres entraînables avec LoRA :** 884 736
- **Proportion de paramètres entraînables :** 0,30 %

Cette approche permet d'entraîner seulement **0.3% des paramètres** du modèle, sans impact notable sur la qualité finale. Elle représente donc un excellent compromis entre coût de calcul et performance.

4 Prétraitement Textuel et Structuration des Données

Le pipeline de prétraitement vise à transformer les notes cliniques brutes du fichier `discharge.csv` (MIMIC-IV) en un corpus structuré de paires *input-target*, optimisé pour la génération automatisée de résumés de type *Brief Hospital Course* avec un modèle *LongT5*.

5. La matrice Q (*Query*) encode ce que chaque token cherche à extraire du contexte.
6. La matrice V (*Value*) contient les représentations informatives que chaque token peut fournir aux autres.

Un aperçu global du pipeline complet de prétraitement est illustré dans la figure 2.7. Cette figure décrit de manière séquentielle les six étapes principales menant de la note clinique brute à un corpus structuré et prêt à l'entraînement.

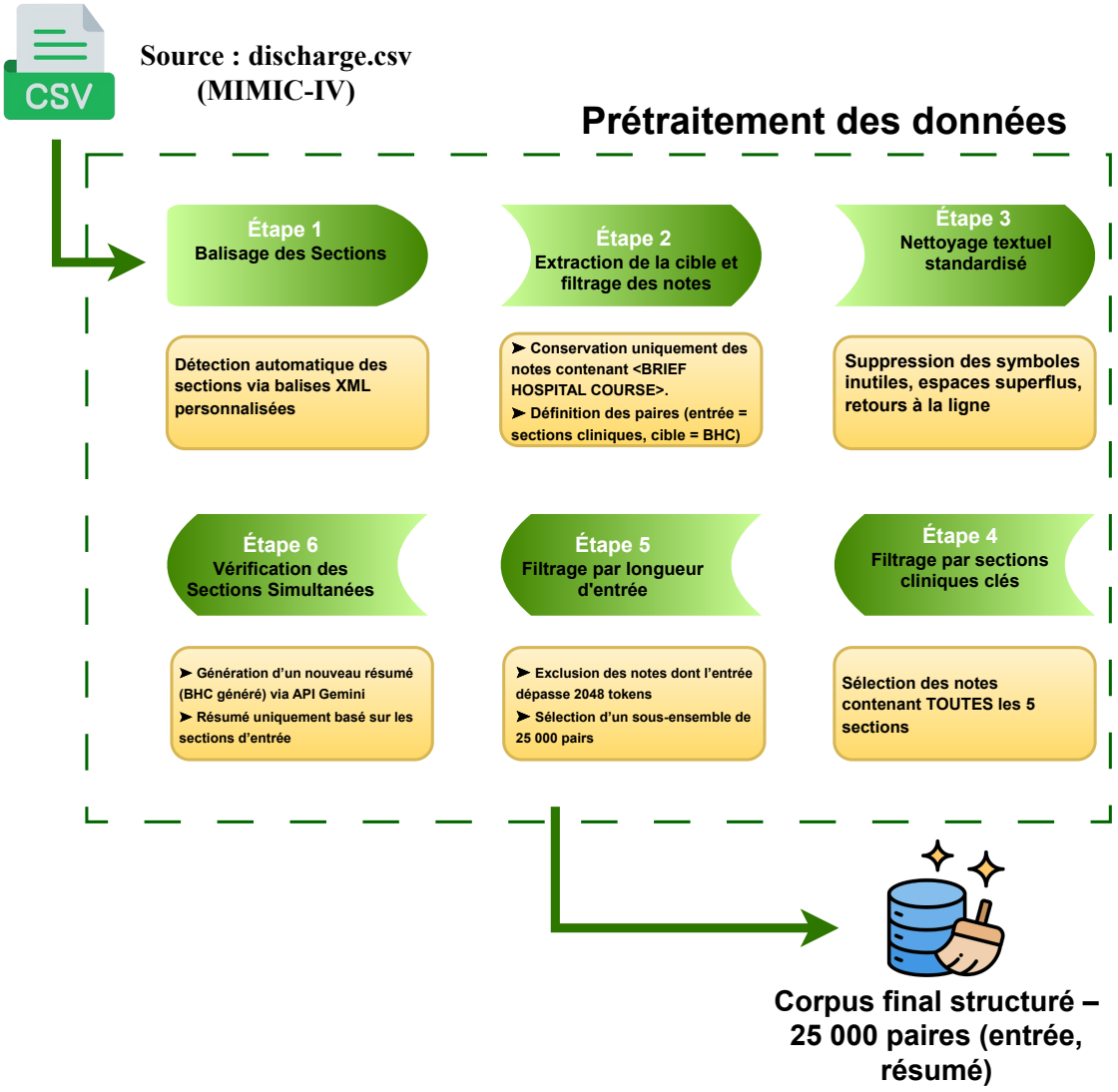


FIGURE 2.7 – Schéma du pipeline de prétraitement des notes cliniques (MIMIC-IV).

Les étapes ci-dessous opèrent exclusivement sur le texte brut, sans conversion numérique préalable.

Étape 1 – Balisage des Sections

Afin de structurer les résumés de sortie hospitaliers, chaque texte brut a été segmenté en sections cliniques explicites à l'aide de balises personnalisées, inspirées du format XML. Ces balises, écrites en majuscules (par exemple : `<CHIEF COMPLAINT>`, `<HISTORY OF PRESENT ILLNESS>`), permettent de marquer clairement le début de chaque section. Ce balisage rend la structure logique du document explicite et exploitable automatiquement pour les étapes ultérieures du pipeline de traitement.

Extrait avant balisage :

```
Sex: F Service: MEDICINE Allergies: No Known Allergies
Chief Complaint: Worsening ABD distension and pain
Major Surgical or Invasive Procedure: Paracentesis
History of Present Illness: Patient with HCV cirrhosis...
```

Extrait après balisage :

```
<SEX> F
<SERVICE> MEDICINE
<ALLERGIES> No Known Allergies
<CHIEF COMPLAINT> Worsening ABD distension and pain
<MAJOR SURGICAL OR INVASIVE PROCEDURE> Paracentesis
<HISTORY OF PRESENT ILLNESS> Patient with HCV cirrhosis...
```

Cette structuration permet d'isoler les sections d'intérêt de façon robuste. Dans notre cas, cinq sections sont extraites comme entrée du modèle : `CHIEF COMPLAINT`, `HISTORY OF PRESENT ILLNESS`, `PHYSICAL EXAM`, `PERTINENT RESULTS`, et `DISCHARGE DIAGNOSIS`. La section `BRIEF HOSPITAL COURSE` est utilisée comme cible du résumé automatique.

Étape 2 – Extraction de la Cible et Filtrage des Notes

L'ensemble des notes de sortie hospitaliers a été filtré pour ne conserver que les documents contenant explicitement la section `<BRIEF HOSPITAL COURSE>`. Cette étape garantit que chaque exemple du corpus dispose d'un résumé clinique cohérent, qui sera utilisé comme cible (`target`) pour l'entraînement du modèle.

Le tableau 2.3 présente l'impact du filtrage appliqué aux résumés de sortie hospitaliers. Le nombre total de notes passe de 331,794 à 270,033, en supprimant toutes celles ne contenant pas la section `<BRIEF HOSPITAL COURSE>`.

TABLE 2.3 – Comparaison des statistiques avant et après filtrage des notes sans `<BRIEF HOSPITAL COURSE>`.

Critère	**Avant filtrage**	**Après filtrage**
Nombre total de notes	331 794	270 033
Présence garantie de `<BHC>`	Non	Oui
Taux de couverture des cibles	Incomplet	100%

Le contenu de cette section a été extrait automatiquement à l'aide d'expressions régulières (*RegEx*) appliquées à la colonne `text` de chaque note. Cette opération isole précisément le sous-texte situé sous l'en-tête `Brief Hospital Course`. Les notes ne comportant pas cette section ont été exclues, assurant ainsi une homogénéité dans les données.

Le reste du document — c'est-à-dire toutes les sections cliniques autres que `<BRIEF HOSPITAL COURSE>` — constitue l'entrée (`input`) du modèle. Cette séparation explicite entre l'entrée et la cible est essentielle pour le cadre d'apprentissage supervisé du résumé automatique.

Étape 3 – Nettoyage Textuel Standardisé

Un nettoyage rigoureux du texte a été réalisé afin de garantir une cohérence dans le format des documents. Cette étape comprend la **suppression des espaces superflus**, des **retours à la ligne inutiles**, et des **symboles non pertinents**. L'objectif est d'obtenir un format homogène facilitant le traitement automatique.

Exemple illustratif :

- **Avant nettoyage :**

```
Pt   reports   self-discontinuing  Lasix \n\n because she
"doesn't want more chemicals."
```

- **Après nettoyage :**

```
Pt reports self-discontinuing Lasix because she "doesn't want more chemicals."
```

Ce nettoyage permet d'éviter les erreurs lors de la tokenisation ou du parsing du texte, et d'assurer une meilleure qualité des entrées pour le modèle de résumé.

Étape 4 – Filtrage par Sections Clés

Afin d'améliorer la qualité des données d'entrée, un filtrage supplémentaire a été appliqué pour ne conserver que les notes contenant un ensemble minimal de sections cliniques jugées essentielles. Seules les notes intégrant toutes les sections suivantes ont été retenues :

- `<HISTORY OF PRESENT ILLNESS>` : cette section décrit en détail l'évolution récente de l'état du patient et constitue une source centrale pour comprendre le contexte clinique de l'hospitalisation.
- `<MAJOR SURGICAL OR INVASIVE PROCEDURE>` : elle informe sur les interventions majeures réalisées durant le séjour, ce qui est crucial pour évaluer la trajectoire thérapeutique.
- `<PERTINENT RESULTS>` : elle regroupe les résultats d'examens biologiques ou d'imagerie jugés significatifs, utiles pour appuyer les décisions médicales.
- `<DISCHARGE DIAGNOSIS>` : elle fournit le diagnostic final posé à la sortie, servant de synthèse décisionnelle de l'épisode de soins.
- `<DISCHARGE MEDICATIONS>` : cette section liste les traitements prescrits à la sortie, indicateur direct du plan thérapeutique retenu.

Ensuite, seules **les notes contenant simultanément les cinq sections clés** mentionnées ci-dessus **ont été retenues**, afin de garantir un contenu clinique riche pour la tâche de résumé.

La figure 2.8 présente la distribution des longueurs de notes (en nombre de tokens) avant et après l'application du filtrage basé sur la présence des cinq sections cliniques clés. Avant filtrage, les notes étaient nettement plus longues, avec une moyenne de **2646 tokens**, un maximum atteignant **14794 tokens**, et une grande dispersion (écart-type de **1064**). La médiane était de **2521 tokens**, ce qui indique que la majorité des documents dépassaient largement la limite acceptable pour un modèle classique.

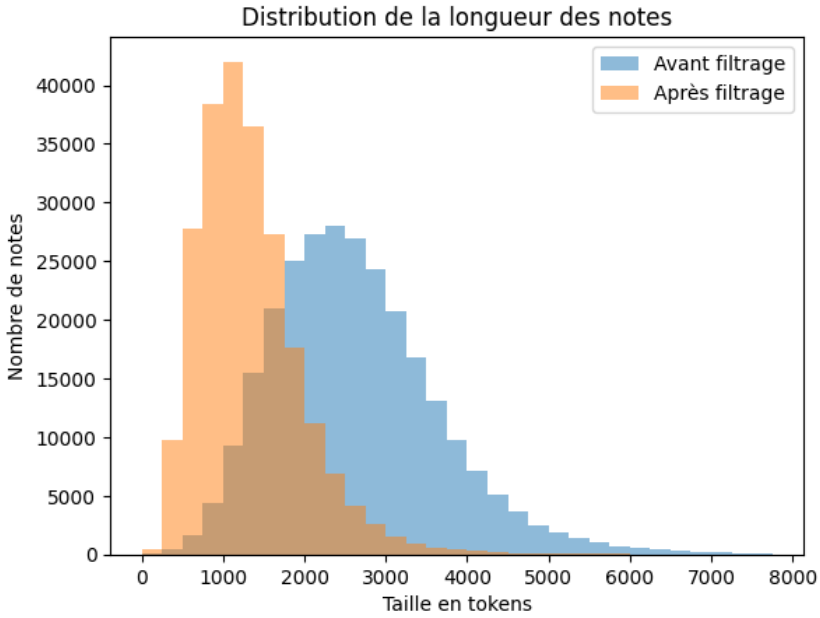


FIGURE 2.8 – Histogramme des longueurs de notes (en tokens) avant et après filtrage des sections cliniques.

Après filtrage, la longueur moyenne chute à **1275 tokens**, avec un maximum ramené à **12938 tokens** et un écart interquartile beaucoup plus resserré (entre **818** et **1605 tokens**). La médiane passe à **1179 tokens**, ce qui confirme une compression efficace de la séquence d'entrée tout en conservant un contenu clinique structuré. Cette réduction est essentielle pour rendre les données compatibles avec les contraintes de longueur imposées par les modèles de type *encoder-decoder*, tout en assurant une homogénéité favorable à l'entraînement.

Après filtrage, le nombre total de notes est réduit à **229037**, et la longueur moyenne chute à **1328 tokens**, avec un maximum ramené à **12938 tokens** et un écart interquartile beaucoup plus resserré (entre **881** et **1643 tokens**). La médiane passe à **1226 tokens**, ce qui confirme une compression efficace de la séquence d'entrée tout en conservant un contenu clinique structuré. Cette réduction est essentielle pour rendre les données compatibles avec les contraintes de longueur imposées par les modèles de type *encoder-decoder*, tout en assurant une homogénéité favorable à l'entraînement.

Le tableau 2.4 résume l'impact du filtrage sur le corpus. On note une réduction de **15.1,%** du nombre de notes, une chute de la longueur moyenne de **2646** à **1328 tokens**, et une meilleure homogénéité des documents. Ces ajustements assurent une compatibilité avec les modèles encoder-decoder tout en préservant la richesse clinique.

TABLE 2.4 – Comparaison des statistiques du corpus avant et après le filtrage

Statistique	**Avant filtrage**	**Après filtrage**
Nombre de notes	270 033	229 037
Réduction du corpus	–	**-15.1%**
Longueur moyenne en tokens (entrée)	2646	1328
Médiane (Q2)	2521	1226
Écart interquartile (Q1–Q3)	1905–3223	881–1643
Longueur maximale	14 794	12 938

Étape 5 – Filtrage par Longueur de l'Entrée

Cette étape a été appliquée pour garantir la compatibilité des données avec les modèles de type *encoder-decoder*, notamment Long-T5, dont la capacité maximale atteint **16,384 tokens** [9]. Toutefois, pour des raisons de contraintes matérielles (mémoire GPU, temps de calcul), nous avons volontairement fixé une limite à **2048 tokens** pour les séquences d'entrée. Seules les notes dont la concaténation des cinq sections sélectionnées respecte cette limite ont été conservées.

Ce seuil permet d'éviter toute troncature lors de l'entraînement, tout en maintenant un niveau suffisant d'information clinique. La figure 2.10 illustre visuellement la distribution cumulative des longueurs avant et après cette étape.

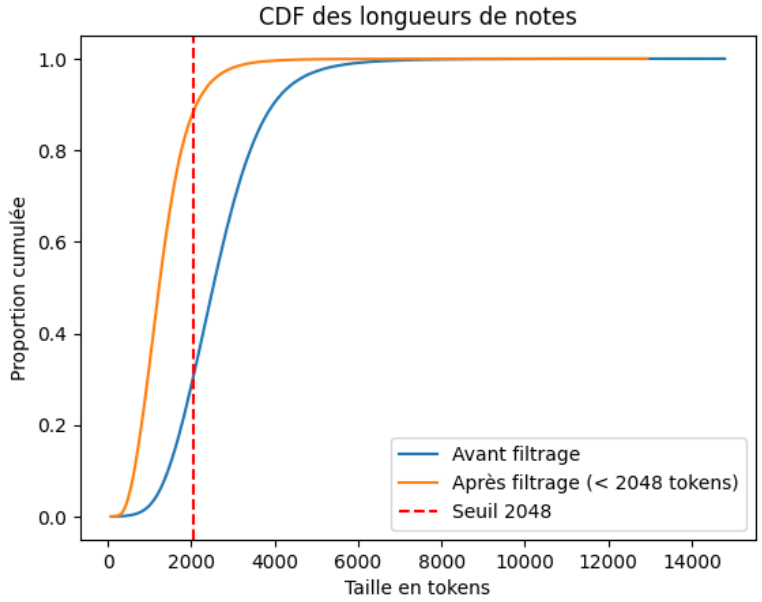


FIGURE 2.9 – Courbe cumulative (CDF) des longueurs de notes (en tokens) avant et après le filtrage. Le seuil de 2048 tokens, représenté en rouge, marque la limite maximale retenue pour les entrées.

Le tableau 2.5 montre qu'après application du seuil de 2048 tokens, **11,6,%** des notes ont été exclues, ramenant le corpus de **229037 à 202517** documents. La longueur moyenne des séquences est passée de **1328 à 1161 tokens**.

TABLE 2.5 – Comparaison des statistiques des longueurs de séquences avant et après filtrage par longueur.

Statistique	**Avant filtrage (5 sections)**	**Après filtrage (2048 tokens)**
Nombre de notes	229 037	202 517
Réduction du corpus	–	**-11.6%**
Longueur moyenne en tokens (entrée)	1328	1161
Médiane (Q2)	1226	1146

Suite à la page suivante

Table 2.5 – Suite de la page précédente

Statistique	**Avant filtrage (5 sections)**	**Après filtrage (2048 tokens)**
Écart interquartile (Q1–Q3)	881–1643	838–1475
Longueur maximale	12 938	2048 (seuil)

Étape 6 – Reformatage qualitatif des résumés pour supervision

L'objectif de cette étape est de générer automatiquement une version résumée et réécrite de la section `<BRIEF HOSPITAL COURSE>` à partir de la seule section d'entrée (`input`), afin de créer une cible de supervision plus adaptée à l'apprentissage automatique. Pour cela nous utilisons **API Gemini 2.0 Flash** [7]. Cette modification est motivée par deux raisons principales :

- la section `<BRIEF HOSPITAL COURSE>` d'origine est souvent trop longue pour un résumé standardisé,
- elle peut contenir des informations issues d'autres sections non retenues dans l'entrée, ce qui risque d'introduire un biais de génération lors de l'entraînement.

Ainsi, la nouvelle version du BHC est obtenue par réduction contrôlée du résumé initial, en ne conservant que les informations présentes dans les cinq sections sélectionnées pour l'entrée, à savoir : `HISTORY OF PRESENT ILLNESS`, `MAJOR SURGICAL OR INVASIVE PROCEDURE`, `PERTINENT RESULTS`, `DISCHARGE DIAGNOSIS`, et `DISCHARGE MEDICATIONS`. Cette approche garantit une supervision fidèle, alignée avec les données disponibles en entrée, et améliore la cohérence de l'apprentissage supervisé.

Compte tenu des limites de temps de génération, du coût d'inférence avec Gemini 2.0 Flash (environ \$0.15 pour 1M tokens en entrée, \$0.60 pour 1M tokens en sortie), et du nombre maximal de requêtes possibles, un sous-échantillon de **25 000 notes cliniques** a été sélectionné parmi les **229 037** disponibles après filtrage. Cette taille offre un compromis acceptable entre qualité de supervision, couverture sémantique et ressources disponibles.

Configuration etu utilisation du API Gemini Flash 2.0

Modèle utilisé : `gemini-2.0-flash` via l'API Google Generative AI, avec les paramètres suivants :

- `temperature = 0.2` : pour limiter la créativité et maximiser la précision,
- `max_output_tokens = 640` : limite stricte sur la taille du résumé généré.

Instructions strictes imposées au modèle : Chaque appel à l'API est guidé par un *prompt* construit dynamiquement, contraignant le modèle à :

- générer uniquement à partir du contenu de la section `input` (<= 2048 tokens),
- ignorer activement toute information contenue uniquement dans `<BRIEF HOSPITAL COURSE>`,

7. Gemini Flash 2.0 : `https://cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-0-flash`

- ne faire aucune inférence ni ajout implicite,
- adopter un style clinique professionnel,
- respecter une limite stricte de tokens (ex. 215 tokens).

Un exemple d'instruction passée au modèle est illustré ci-dessous (schéma général du prompt) :

```
You are a medical summarization assistant. Your SOLE TASK is to create a concise
hospital course summary.

Read the following sections carefully:
- ‘input‘: This is the ONLY permissible source of information for your summary.
- ‘brief_hospital_course‘: DO NOT extract facts unless also in ‘input‘.

STRICT RULES:
1. 100% of content must come from ‘input‘
2. DO NOT use ‘brief_hospital_course‘ for facts
3. NO inference or additions
4. Formal clinical style
5. Max token count = 215
```

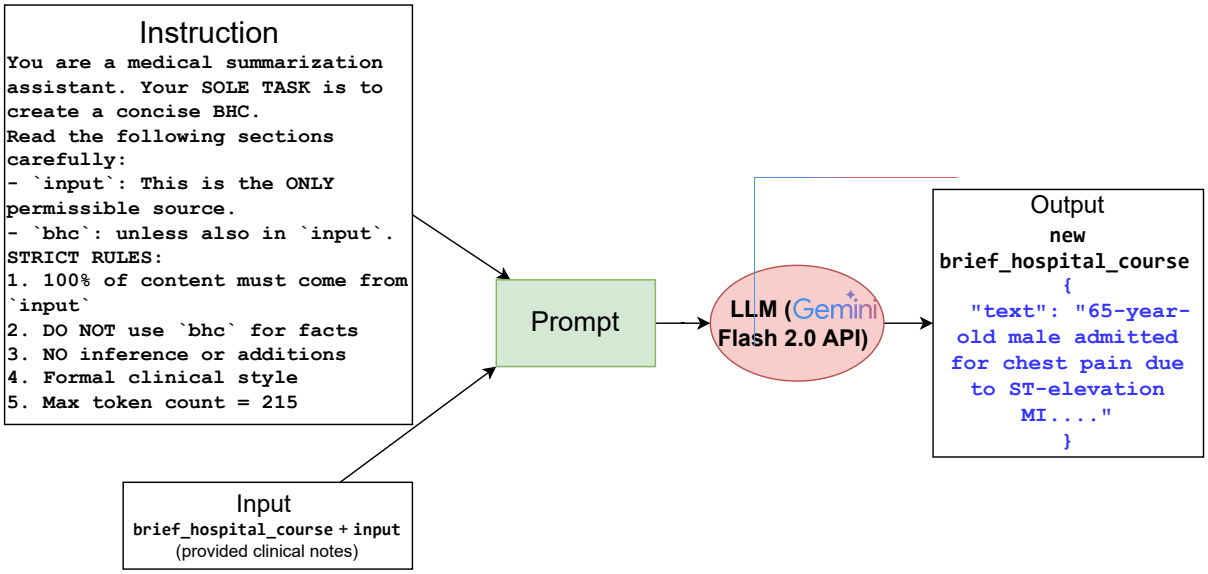


FIGURE 2.10 – Schéma de génération contrôlée des résumés BHC avec API Gemini Flash 2.0

Chaque exemple est traité indépendamment afin de garantir une génération reproductible, conforme aux exigences de qualité du domaine médical.

Statistiques descriptives post-prétraitement

Après génération automatique des nouveaux résumés à l'aide de Gemini Flash 2.0, les longueurs moyennes des entrées sont restées proches de celles observées avant cette étape (moyenne

de **1157 tokens** contre **1161**), ce qui confirme que le sous-échantillon est représentatif (Tableau 2.6). En revanche, les longueurs des cibles ont significativement diminué : la moyenne passe de **598 tokens** (BHC original) à seulement **256 tokens** (BHC généré), avec un maximum réduit de **6848** à **910 tokens**. Cette réduction garantit une meilleure compacité et cohérence des résumés, tout en respectant les contraintes fixées par l'instruction.

TABLE 2.6 – Comparaison des statistiques des longueurs des entrées et des cibles avant et après génération automatique des résumés.

Statistique	**Avant génération (202 517)**	**Après génération / post-prétreatment (25 000)**
Séquences d'entrée (input)		
Longueur moyenne	1161	**1157**
Écart-type	419	419
Médiane (Q2)	1146	1143
Écart interquartile (Q1–Q3)	838–1475	837–1470
Longueur maximale	2048	2048
Cibles (résumé BHC généré)		
Longueur moyenne	598	**256**
Écart-type	406	102
Médiane (Q2)	495	244
Écart interquartile (Q1–Q3)	313–777	182–319
Longueur maximale	6848	910

Après les étapes de prétraitement, le corpus nettoyé se compose de **25,000 paires** *(input, target)*. Comme illustré par la figure 2.11, les séquences d'entrée (`input`) sont généralement plus longues, avec des valeurs réparties jusqu'à la limite maximale de **2048 tokens**. En comparaison, les résumés générés (`target`) présentés dans la figure 2.12 sont plus compacts, avec une concentration majoritaire entre **150 et 350 tokens**. Cette différence de longueur est attendue dans le cadre d'une tâche de résumé automatique et reflète une structuration efficace des données pour l'entraînement.

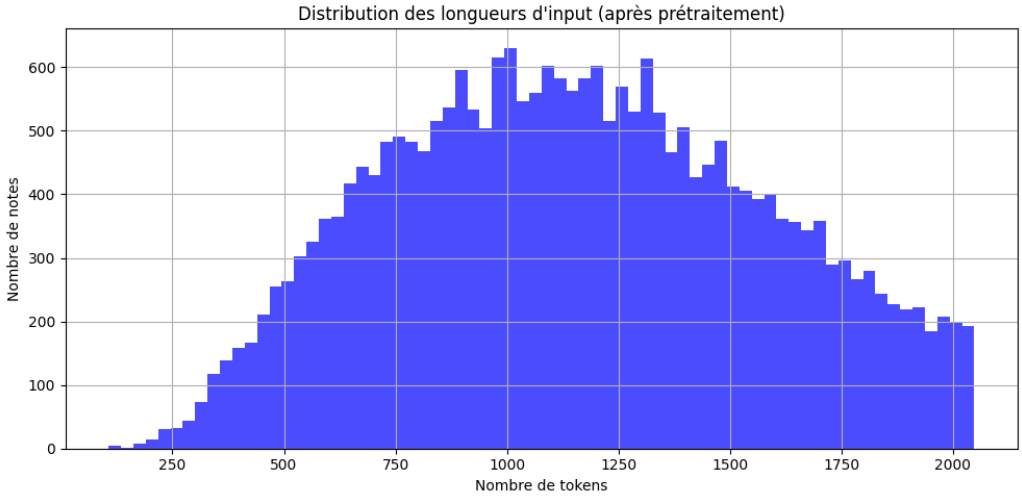


FIGURE 2.11 – Distribution des longueurs des séquences d'entrée (*input tokens*) (post-prétraitement).

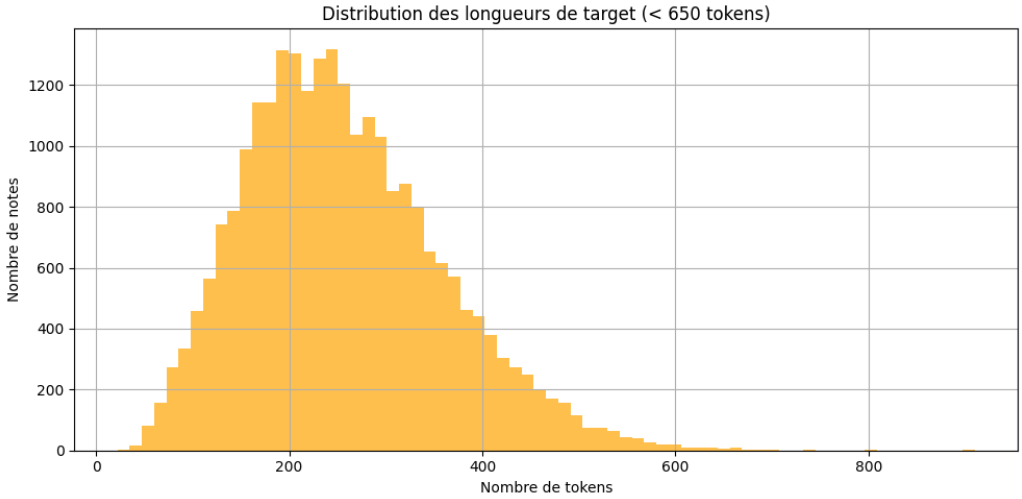


FIGURE 2.12 – Distribution des longueurs des séquences de sortie (*summary tokens*) (post-prétraitement).

5 Tokenisation et Formatage pour LongT5

5.1 Tokenisation des Données

La transformation des rapports médicaux bruts et de leurs résumés associés en représentations numériques exploitables par un modèle de langage repose sur un processus de tokenisation rigoureux. Dans ce travail, nous utilisons le tokenizer `T5Tokenizer` (commun aux architectures T5 et LongT5 [9]), basé sur l'algorithme **SentencePiece** (mode Unigram) [102]. Ce choix est particulièrement adapté aux textes médicaux grâce à sa capacité à gérer des termes spécialisés et des constructions linguistiques complexes.

Le tokenizer employé présente les propriétés suivantes :

- **Vocabulaire** : Fixe à **32 000 tokens** (identique à T5 [37]), couvrant à la fois les séquences d'entrée et de sortie.

- **Segmentation sub-lexicale** : Contrairement aux approches word-level, SentencePiece opère directement sur les caractères Unicode, apprenant des sous-unités (*subwords*) optimisées pour le domaine médical (p.ex., "hypertension" → `"hyper"`, `"tension"`).
- **Spécialisation** : Capacité à encoder des termes techniques (comme "tachycardie ventriculaire") sans recourir systématiquement à des tokens inconnus (`<unk>`).

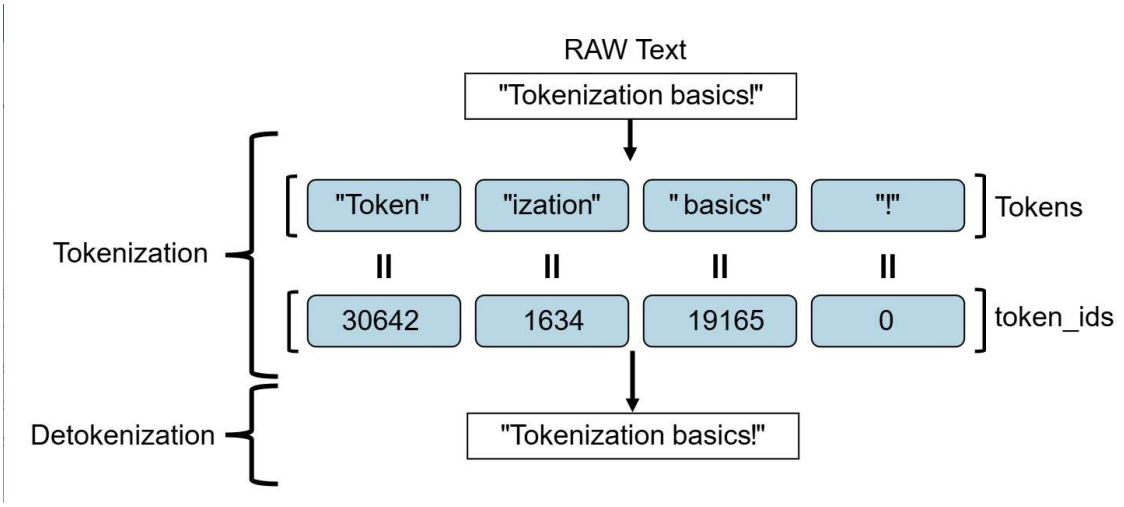


FIGURE 2.13 – Processus complet de tokenisation des données textuelles (Source : [103])

5.2 Gestion des Longueurs de Texte

Pour adapter les textes aux contraintes de LongT5, les stratégies suivantes sont appliquées :

- **Troncation (*Truncation*)** :
  - **Entrées** : dans notre *dataset* (après prétraitement, voir section 4), aucune note ne dépasse 2048 tokens (la limite de LongT5). Ainsi, la troncation n'a aucun effet sur les données du *dataset*. En revanche, lors de l'inférence sur de nouvelles données, certaines notes cliniques peuvent dépasser cette limite, impliquant une troncation automatique pouvant entraîner une perte d'information contextuelle. **La perte d'information est donc négligeable et n'a pas d'impact significatif sur l'entraînement du modèle.**
  - **Sorties** : Seuls 363 résumés (sur 25 000) dépassent 512 tokens (voir figure 2.12. Ces cas sont tronqués tout en conservant la structure syntaxique.
  - **Remplissage (*Padding*)** :
    - Les séquences courtes sont complétées avec `[PAD]` (ID=0). Exemple pour une entrée de 100 tokens :

      `["Le", "patient", ..., "[PAD]", "[PAD]"] (total 2048 tokens)`
    - Pour les cibles, les `[PAD]` sont remplacés par `-100` (ignorés par la loss). Exemple :

      `["Résumé", "médical", -100, -100] (total 512 tokens)`

6 Configuration, Entraînement et Génération du Modèle LongT5

Cette section décrit en détail l'ensemble du processus expérimental lié à l'entraînement du modèle LongT5 sur notre corpus de résumés cliniques. Elle est structurée en plusieurs sous-parties complémentaires. Nous commençons par présenter les caractéristiques matérielles utilisées pour l'entraînement, puis nous détaillons la répartition du jeu de données, les hyperparamètres choisis, le protocole d'évaluation, ainsi que la configuration de la génération lors de la validation et du test. L'objectif est d'offrir une vue complète et reproductible du cadre expérimental dans lequel les résultats du chapitre suivant ont été obtenus.

6.1 Configuration Matérielle

L'entraînement a été réalisé sur une instance **Google Colab Pro+** avec les spécifications matérielles suivantes :

- **GPU** : NVIDIA A100 avec 40 Go de mémoire VRAM
- **CPU** : 2 cores
- **RAM** : 83,5 Go de mémoire vive
- **Stockage** : 235,7 Go d'espace disque SSD

L'architecture Ampere de NVIDIA A100 tire parti des cœurs *Tensor Core* pour accélérer les calculs en précision mixte (*mixed precision*), notamment le format *bfloat16* (BF16) [104]. Contrairement au FP16 traditionnel, le BF16 préserve une plage dynamique étendue (8 bits d'exposant), réduisant ainsi les risques de saturation numérique lors de l'entraînement des LLMs [105].

6.2 Répartition du *Dataset*

Le corpus final, constitué de **25 000 paires (input, target)**, a été divisé aléatoirement en trois *subsets* :

- **Entraînement (Train)** : **20 000 exemples** (80 %)
- **Validation** : **2 500 exemples** (10 %)
- **Test** : **2 500 exemples** (10 %)

Cette répartition garantit une séparation claire entre les phases d'apprentissage, d'ajustement des hyperparamètres, et d'évaluation finale. Les résultats obtenus sur l'ensemble de test seront analysés en détail dans le chapitre 3.

6.3 Paramètres d'Entraînement et d'Optimisation

La configuration des hyperparamètres d'Entraînement et d'Optimisation a été déterminée comme suit :

- **Taille de lot (*Batch Size*)** : **32 échantillons**, aussi bien pour l'entraînement que pour l'évaluation. Cette valeur représente un compromis entre l'efficacité mémoire (Car on a 40GB VRAM GPU) et la stabilité des gradients. Un *batch* de 32 permet de traiter efficacement les longs documents cliniques tout en maintenant une bonne variabilité des échantillons.
- **Nombre d'époques (*Epoch Number*)** : **Fixé à 4**, suffisant pour observer une convergence sans surapprentissage significatif, comme le démontrent les courbes d'apprentissage (See section 4.1, chapitre 3).
- **Algorithme d'optimisation (*Optimizer*)** : AdamW [106], variant modernisée d'Adam [107] intégrant une décroissance de poids (*weight decay*) pour améliorer la généralisation [108]. Formellement, la mise à jour des poids w_t à l'étape t suit :

$$w_t = w_{t-1} - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} - \eta \cdot \lambda \cdot w_{t-1} \tag{2.1}$$

  où η est le taux d'apprentissage, $\hat{m}_t$ et $\hat{v}_t$ les estimateurs des premier et second moments, et λ le coefficient de régularisation [106]. Cet optimiseur est particulièrement adapté aux tâches de génération de texte comme notre résumé de notes cliniques.
- **Taux d'apprentissage (*Learning Rate*)** : Initialisé à 5×10^{-4} avec :
  - Une phase de *warm-up* linéaire sur 5% (*warmup ratio = 0.05*) des itérations pour stabiliser les gradients initiaux.
  - Une décroissance cosinusoïdale ultérieure, adoucissant progressivement les mises à jour (*learning rate scheduler type = "cosine"*) [109].
- **Régularisation** : Coefficient $\lambda = 0.01$ pour la décroissance de poids (***weight decay***), limitant la croissance des paramètres sans compromettre la convergence.

6.4 Évaluation et Sélection du Modèle

Le protocole d'évaluation repose sur :

- **Fréquence d'évaluation** : Validation intermédiaire toutes les 100 itérations, permettant un suivi granulaire des performances. Cette fréquence offre un bon compromis entre suivi précis et efficacité computationnelle.
- **Métrique principale** : ROUGE-L (*ROUGE Longest Common Subsequence*), mesure la longueur de la plus longue sous-séquence commune (LCS) entre entre les résumés générés et les références cliniques [12]. ROUGE-L est particulièrement adapté pour évaluer la qualité des résumés médicaux [110].
- **Sélection finale** : Conservation du modèle optimal qui a une grande valeur de ROUGE-L.

6.5 Configuration de la Génération lors de la Validation

Lors de la validation, les sorties du modèle LongT5 sont générées à l'aide d'une configuration de décodage contrôlée, afin de produire des résumés cohérents, compacts et non répétitifs.

Cette configuration influence directement les métriques d'évaluation (comme ROUGE) et vise à simuler les conditions d'inférence réelles.

Les paramètres de génération utilisés sont les suivants :

- `num_beams = 4` : active une recherche par faisceaux (*beam search*) pour améliorer la diversité et la qualité des résumés générés.
- `length_penalty = 0.8` : pénalise les séquences longues afin de favoriser des résumés concis.
- `no_repeat_ngram_size = 3` : interdit la répétition de trigrammes pour éviter les redondances.
- `generation_max_length = 215` : impose une limite stricte à la longueur des sorties, cohérente avec le format attendu des résumés cliniques.

Cette configuration a été utilisée de manière systématique pendant **toutes les étapes de validation**, ainsi que **lors de l'évaluation finale sur l'ensemble de test**, garantissant ainsi une génération cohérente et une comparaison fidèle des performances du modèle.

7 Conclusion

À l'issue de ce chapitre, nous disposons d'un corpus cliniquement pertinent, normalisé et aligné avec les contraintes des modèles de génération. Le modèle LongT5, adapté via LoRA, a été entraîné et évalué à l'aide d'une stratégie rigoureuse d'optimisation et de validation.

Les configurations de génération ont été soigneusement calibrées pour produire des résumés informatifs, non redondants et adaptés à un usage médical réel. Le pipeline mis en œuvre est donc prêt à être évalué empiriquement.

Le chapitre suivant présente les résultats quantitatifs et qualitatifs de l'approche proposée. Nous y analyserons les performances du modèle à travers des métriques standards de résumé automatique (ROUGE), des comparaisons avec les résumés originaux, et des exemples illustratifs issus du corpus. Cette analyse permettra de valider la robustesse et la pertinence clinique de notre méthode.

Chapitre 3

Mise en oeuvre et résultats de l'approche proposée

1 Introduction

Ce chapitre est consacré à l'analyse des résultats expérimentaux obtenus suite au fine-tuning du modèle **LongT5** sur les rapports de sortie médicaux extraits de la base de données **MIMIC-IV**. L'objectif de cette évaluation est de mesurer la qualité des résumés générés automatiquement en s'appuyant sur la **métrique ROUGE** dans ses différentes variantes (ROUGE-1, ROUGE-2, ROUGE-L), reconnue pour sa capacité à estimer la couverture du contenu de référence.

Pour ce faire, ce chapitre s'articule en plusieurs étapes clés. Nous débutons par une analyse du **suivi de l'apprentissage**, en examinant l'évolution de la fonction de perte (*loss*) et des scores ROUGE pour évaluer la stabilité et la convergence du modèle. Nous présentons ensuite les **performances quantitatives finales** sur l'ensemble de test, en les comparant à plusieurs modèles de référence. Une **analyse qualitative** d'exemples concrets viendra illustrer les forces et les faiblesses du modèle en pratique. Enfin, une **discussion générale** synthétisera l'ensemble de ces résultats, identifiera les limites de l'approche et proposera des perspectives d'amélioration.

2 Présentation de l'environnement de développement et des outils

Cette section détaille l'environnement matériel et logiciel utilisé pour l'ensemble de nos expérimentations. Nous y présentons d'abord la configuration matérielle employée pour l'entraînement du modèle, suivie des plateformes logicielles, du langage de programmation et des bibliothèques essentielles qui ont permis de mener à bien notre tâche de résumé de notes cliniques avec LongT5. Ces outils ont été sélectionnés pour leur adéquation avec les exigences du traitement de modèles de langage de grande taille et pour leur efficacité dans un contexte de recherche.

2.1 Environnements et Matériel Utilisé

Nos travaux ont été menés en exploitant principalement un environnement cloud pour les phases d'entraînement exigeantes en ressources, complété par un environnement local pour le développement et les tests.

- **Pour l'entraînement et l'évaluation finale :** L'entraînement, l'affinage (fine-tuning) ainsi que l'évaluation finale de notre modèle sur le jeu de données de test ont été réalisés sur la plateforme **Google Colab Pro+**. Comme mentionné dans le chapitre 2, nous avons bénéficié d'un accès à un **GPU NVIDIA A100** avec **40 Go de VRAM** et **83 Go de RAM**. Le recours à cette configuration matérielle robuste pour l'ensemble du cycle d'expérimentation garantit la reproductibilité et la fiabilité des métriques de performance obtenues.
- **Pour le déploiement et les tests applicatifs :** Une fois le modèle affiné, il a été intégré dans une application de bureau pour être testé sur un ordinateur personnel disposant des caractéristiques suivantes :
  - **Processeur (CPU) :** AMD Ryzen 5 4500U
  - **Mémoire vive (RAM) :** 16 Go
  - **Carte graphique (GPU) :** Radeon Graphics
  - **Stockage :** 500 Go

  Cette étape visait à évaluer les performances du modèle, notamment sa latence d'inférence et sa consommation de ressources, dans un environnement d'utilisation grand public doté d'un GPU moderne. L'environnement local, basé sur **Anaconda** et **Jupyter Notebook**, a également servi pour les phases initiales de développement.

2.2 Langage de programmation et Bibliothèques

Le langage **Python** a été le pilier de notre développement, choisi pour sa syntaxe claire, son vaste écosystème de bibliothèques dédiées à la science des données et à l'intelligence artificielle, et son intégration native avec les frameworks de deep learning.

Les principales bibliothèques utilisées pour la mise en œuvre de notre système de résumé sont les suivantes :

1. **PyTorch :** Framework de deep learning principal, utilisé pour définir, entraîner et optimiser le modèle LongT5 [111].
2. **Transformers (Hugging Face) :** Bibliothèque centrale pour télécharger, configurer et utiliser le modèle pré-entraîné `LongT5` et son tokenizer associé [112].
3. **PEFT (Hugging Face) :** Utilisée pour le fine-tuning efficace du modèle via la méthode LoRA, réduisant ainsi les coûts de calcul et de mémoire [113].
4. **Datasets (Hugging Face) :** Employée pour le chargement, le prétraitement et la gestion optimisée de notre corpus de notes cliniques [114].
5. **Pandas :** Outil essentiel pour le nettoyage, la manipulation et la structuration des données textuelles brutes avant leur traitement [115].

6. **re (Regular Expression) :** Module natif de Python, utilisé pour filtrer et extraire des sections de texte via des expressions régulières lors du prétraitement [116].
7. **google-generativeai :** Interface pour l'API **Gemini 2.0 Flash**, employée pour la génération contrôlée des résumés cibles durant le prétraitement [117].
8. **Evaluate (Hugging Face) :** Utilisée pour l'évaluation des résumés générés via le calcul de métriques standards comme ROUGE [12].
9. **Tkinter :** A servi au développement d'une interface graphique simple pour la démonstration du système de résumé [118].

3 Pertinence des métriques ROUGE dans l'évaluation de résumés médicaux

3.1 Définitions des métriques ROUGE

Les métriques **ROUGE** (*Recall-Oriented Understudy for Gisting Evaluation*) mesurent le chevauchement lexical entre un résumé automatique et un ou plusieurs résumés de référence [12]. Autrement dit, elles comparent les *n-grammes*[1] du résumé généré à ceux du résumé humain. Les principales variantes utilisées sont :

- **ROUGE-1** : recense le chevauchement des unigrammes (mots simples) [12].
  Exemple : Référence = « le patient est stable » ; Généré = « le patient est conscient »
  Unigrammes de la référence = {le, patient, est, stable} ; du résumé = {le, patient, est, conscient}
  Unigrammes communs = {le, patient, est} $\Rightarrow$ ROUGE-1 = $\frac{3}{4} = 0{,}75$
- **ROUGE-2** : mesure le chevauchement des bigrammes (paires de mots consécutifs) [12].
  Exemple : Référence = « le patient est stable » ; Généré = « le patient est conscient »
  Bigrams de la référence = {le patient, patient est, est stable} ; du résumé = {le patient, patient est, est conscient}
  Bigrams communs = {le patient, patient est} $\Rightarrow$ ROUGE-2 = $\frac{2}{3} \approx 0{,}67$
- **ROUGE-L** : repose sur la plus longue sous-séquence commune (Longest Common Subsequence – LCS) partagée dans le même ordre [12].
  Exemple : Référence = « le patient est stable » ; Généré = « le patient est conscient »
  LCS = « le patient est » (longueur 3) $\Rightarrow$ ROUGE-L = $\frac{3}{4} = 0{,}75$

Ces scores sont normalisés entre 0 et 1 : un score plus élevé indique une plus grande similarité lexicale avec le résumé de référence [12].

3.2 Utilité de la Métrique ROUGE pour l'Évaluation

Les métriques ROUGE (Recall-Oriented Understudy for Gisting Evaluation) sont largement adoptées pour évaluer la qualité des résumés automatiques, y compris dans le domaine médical

1. Un n-gramme est une séquence continue de n mots dans un texte. Par exemple, les bigrammes (n = 2) de « le patient est stable » sont : « le patient », « patient est », « est stable ».

[66, 119]. Elles fournissent une mesure quantitative qui est à la fois objective et reproductible, permettant de comparer différents systèmes de résumé sans dépendre d'évaluations humaines coûteuses [119].

Leur pertinence dans le contexte du résumé médical repose sur plusieurs avantages clés :

- **Standardisation et Simplicité :** ROUGE est une métrique standard, facile à mettre en œuvre et rapide à calculer, ce qui permet une évaluation automatisée à grande échelle [12].
- **Objectivité et Reproductibilité :** Les scores ROUGE sont numériques et déterministes, ce qui garantit des évaluations objectives et des comparaisons expérimentales fiables entre différents modèles [119].
- **Accent sur la Couverture de l'Information :** La métrique est principalement orientée vers le rappel (*recall*), mesurant la proportion d'informations du résumé de référence qui sont présentes dans le résumé généré [12]. Cette orientation est cruciale dans le domaine médical, où capturer un maximum d'informations pertinentes est primordial [66]. Les variantes comme ROUGE-1 et ROUGE-2 évaluent spécifiquement cette couverture lexicale en se basant sur les unigrammes et bigrammes communs [12].

Ainsi, dans le cadre de notre travail de fine-tuning de LongT5 sur les dossiers cliniques MIMIC-IV, un score ROUGE élevé est interprété comme une bonne capacité du modèle à couvrir les informations médicales importantes présentes dans les résumés de référence.

4 Suivi de l'apprentissage du modèle proposé

Dans cette section, nous examinons l'évolution de l'apprentissage du modèle LongT5 lors du fine-tuning sur les rapports médicaux MIMIC-IV. Nous analysons les courbes de *loss* et les scores ROUGE pendant l'entraînement, afin d'évaluer la stabilité de l'apprentissage, sa convergence et tout signe éventuel de sur-apprentissage. Les Figures 3.1 et 3.2 ainsi que le Tableau 3.1 résument ces résultats.

4.1 Évolution des courbes de *loss*

La figure 3.1 montre que le *training loss* du modèle chute rapidement au début du fine-tuning, puis se stabilise progressivement autour de 0,6–0,7 vers 12 000 étapes. De manière concomitante, la *validation loss* diminue continûment, passant d'environ 0,589 à 0,458 sur la même plage d'étapes.

La tendance générale à la baisse des deux courbes indique que le modèle apprend correctement et de manière stable, **sans signe de surapprentissage (*overfiting*)**.

L'écart modéré entre les courbes, et surtout le fait que la *validation loss* soit inférieure à celle de l'entraînement, confirme que le modèle **ne surapprend pas et généralise bien**. Ce phénomène s'explique principalement par la régularisation **weight_decay** utilisée lors de l'entraînement :

- **Pendant l'entraînement :** le *train loss* inclut l'erreur de prédiction ainsi qu'une pénalité de régularisation (le *weight decay*).

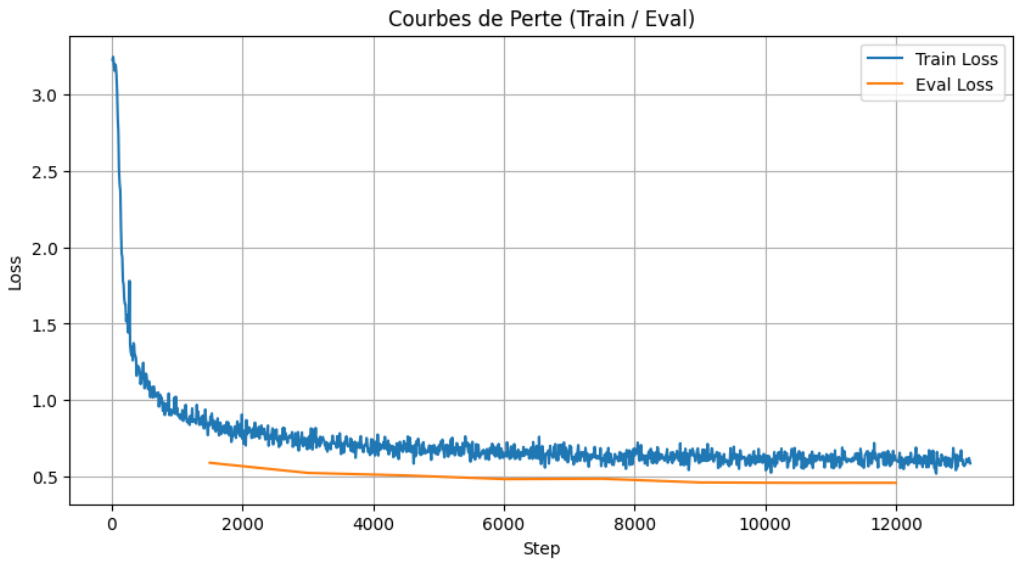


FIGURE 3.1 – Courbes de *loss* du modèle pendant le fine-tuning (entraînement en bleu, validation en orange).

- **Pendant la validation :** La *validation loss* mesure uniquement l'erreur de prédiction, sans cette pénalité.

Cette pénalité, ajoutée uniquement à le *training loss*, explique pourquoi cette dernière est artificiellement plus élevée. La convergence du modèle s'amorce visiblement vers 6 000 étapes, où les courbes commencent à se stabiliser, témoignant du succès de la procédure de *fine-tuning*.

4.2 Convergence des scores ROUGE

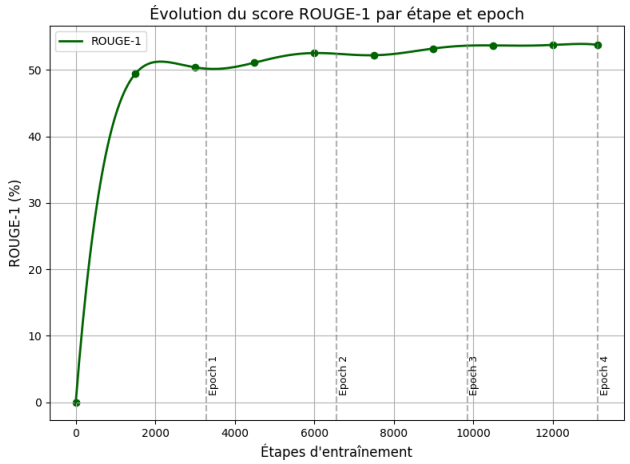
La figure 3.2 illustre l'évolution des scores ROUGE obtenus sur le jeu de validation au cours de l'entraînement. **On observe une progression spectaculaire dès les premières 1 500 étapes**, où les trois métriques connaissent une croissance très rapide. Cette phase initiale démontre que le modèle **s'adapte très vite à la tâche de résumé clinique, en apprenant les patrons fondamentaux des données**.

Passée cette phase, l'amélioration se poursuit de manière plus graduelle et continue, comme le montrent les gains entre 1 500 et 12 000 étapes :

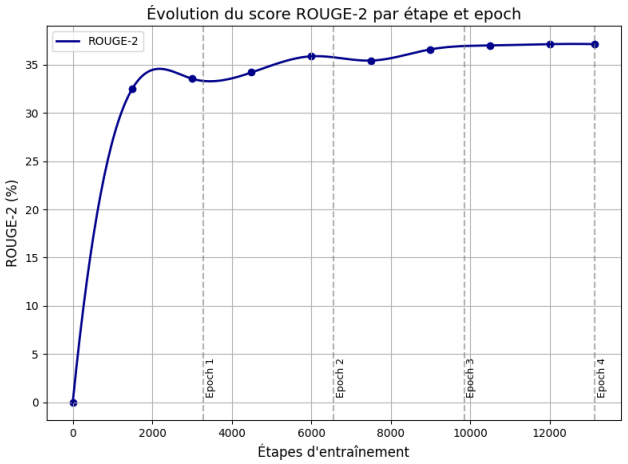
- Le score ROUGE-1 progresse d'environ 49,4 % à 53,8 % (+4,3 points).
- ROUGE-2 passe de ∼32,5 % à ∼37,1 % (+4,6 points).
- ROUGE-L de ∼41,0 % à ∼45,1 % (+4,1 points).

Cette croissance continue reflète **un gain constant en qualité des résumés produits** par le modèle. On note un léger ralentissement de la progression vers 6 000–7 500 étapes, où les courbes semblent amorcer un plateau temporaire avant de reprendre leur ascension. Vers la fin de l'entraînement, la pente s'atténue, indiquant une stabilisation des performances.

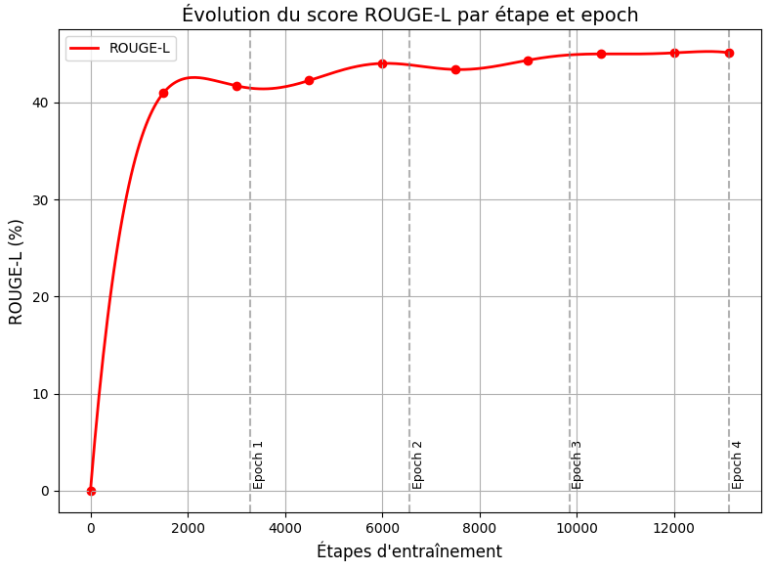
En tout état de cause, l'absence de baisse des scores ROUGE confirme qu'il n'y a **pas de dégradation liée à un sur-apprentissage** : le modèle continue d'apprendre tout en conservant une bonne généralisation sur les données de validation.



(a) ROUGE-1



(b) ROUGE-2



(c) ROUGE-L

FIGURE 3.2 – Évolution des scores ROUGE (1, 2 et L) du modèle pendant le fine-tuning.

Le Tableau 3.1 récapitule numériquement ces métriques clés. On y retrouve la baisse régulière des *training* et *validation loss* ainsi que la progression continue des scores ROUGE. Ce tableau confirme la tendance constatée visuellement : aucune oscillation brutale n'apparaît, et l'évolution est globalement monotone.

En particulier, l'écart entre le *training loss* et la *validation loss* reste limité, attestant de la stabilité du processus d'apprentissage. De plus, les gains en ROUGE soulignent l'amélioration de la performance du modèle sur la tâche de résumé. En somme, ces résultats indiquent que l'apprentissage converge correctement : la performance du modèle s'améliore de manière constante sans signe manifeste de sur-apprentissage.

TABLE 3.1 – Valeurs des *loss* et scores ROUGE à différentes étapes du fine-tuning

Step	Training Loss	Validation Loss	ROUGE-1	ROUGE-2	ROUGE-L
1 500	0.8334	0.58946	49.43	32.51	40.97
3 000	0.7497	0.52332	50.40	33.54	41.71
4 500	0.7148	0.50685	51.10	34.19	42.26
6 000	0.6451	0.48278	52.54	35.87	44.02
7 500	0.5930	0.48510	52.21	35.42	43.39
9 000	0.6397	0.46088	53.23	36.58	44.35
10 500	0.6105	0.45820	53.69	36.99	45.00
12 000	0.6069	0.45841	53.76	37.12	45.10

5 Performances du Modèle

5.1 Évaluation Finale sur l'Ensemble de Test

Après la phase de fine-tuning, la performance finale du modèle, que nous nommons `MedSum-LongT5`, a été mesurée sur l'ensemble de test. Cet ensemble est composé de 2500 exemples jamais vus durant l'entraînement, garantissant une évaluation objective de la capacité de généralisation du modèle. L'évaluation s'est concentrée sur la qualité des résumés générés à travers les métriques ROUGE, ainsi que sur le *loss* du modèle.

Il est particulièrement notable que ces mesures de performance ont été obtenues sur l'infrastructure haut de gamme utilisée pour l'entraînement, à savoir un GPU **NVIDIA A100 (40 Go VRAM)**. Sur cette configuration, le traitement des 2500 notes de l'ensemble de test a pris **2,185,47 secondes**, soit une durée moyenne de seulement **0,87 seconde par résumé**.

Les résultats finaux obtenus sur l'ensemble de test sont synthétisés dans le tableau 3.2.

TABLE 3.2 – Scores et métriques de performance du modèle `MedSum-LongT5` sur l'ensemble de test.

Métrique de Qualité (ROUGE)	**Score Obtenu**
ROUGE-1	52,59 %
ROUGE-2	35,31 %
ROUGE-L	42,92 %
Métrique de *loss*	**Valeur**
Test Loss	0,4605
Métriques de Performance	**Valeur**
Temps d'inférence total	2185,47 s
Temps d'inférence moyen	0,87 s

L'analyse de ces métriques permet de tirer plusieurs conclusions sur la performance du modèle :

- Un score **ROUGE-1** de **52,59 %** indique une excellente superposition lexicale. Cela signifie que plus de la moitié des mots présents dans les résumés de référence se retrouvent

dans les résumés générés par le modèle. Ce score élevé suggère que le modèle est capable de sélectionner et de réutiliser le vocabulaire médical pertinent et les termes clés.

- Le score **ROUGE-2**, qui s'élève à **35,31 %**, est particulièrement significatif. En mesurant la correspondance des paires de mots (bigrammes), il montre que le modèle ne se contente pas de reprendre des mots isolés, mais qu'il parvient à préserver des segments de phrases et des collocations courtes, ce qui est un indicateur de cohérence locale.
- Le score **ROUGE-L** atteint **42,92 %**. Cette métrique évalue la plus longue sous-séquence commune, ce qui reflète la capacité du modèle à maintenir une structure de phrase similaire à celle des résumés de référence. Un bon score ROUGE-L indique que les résumés générés sont non seulement pertinents en termes de mots, mais aussi fluides et structurellement cohérents.

Enfin, Le *test loss* est de **0,46**, une valeur très proche de la *validation loss* finale (vue à la section précédente). Cela confirme l'absence de sur-apprentissage et la bonne capacité de généralisation du modèle sur des données entièrement nouvelles.

En synthèse, ces résultats démontrent la robustesse du modèle `MedSum-LongT5`. Il produit des résumés qui sont non seulement fidèles au contenu lexical (ROUGE-1), mais aussi cohérents au niveau des phrases (ROUGE-2) et structurellement bien formés (ROUGE-L), validant ainsi l'efficacité de notre approche de fine-tuning.

5.2 Comparaison avec des modèles non fine-tunés (baselines)

Avant de comparer les performances, nous présentons brièvement les modèles inclus dans cette évaluation (qui sont tous des modèles *baseline* :

- `MedSum-LongT5` : Il s'agit de notre modèle principal, basé sur LongT5, ayant subi un fine-tuning sur les rapports de sortie médicaux extraits de la base MIMIC-IV.
- `MedSum-T5` : Un second modèle que nous avons nous-mêmes fine-tuné, basé sur T5-base, entraîné sur les mêmes données médicales que `MedSum-LongT5`, mais uniquement sur les exemples dont la longueur d'entrée est inférieure à 512 tokens. Cette limitation impose une troncature du contexte pour les documents plus longs, ce qui peut impacter la qualité du résumé.
- `LongT5-base` [2] : Variante de LongT5 pré-entraînée sur des données générales, sans spécialisation médicale ni fine-tuning sur le corpus cible [9].
- `BART-large-CNN` [3] : Modèle de génération de texte pré-entraîné sur les jeux de données journalistiques CNN/DailyMail, conçu pour le résumé automatique en langue naturelle [31].

Le tableau 3.3 présente les métriques d'évaluation (*test loss* et scores ROUGE) pour chacun des modèles. On observe que le modèle **`MedSum-LongT5` (LongT5-base adapté au domaine médical) obtient la *test loss* la plus faible ainsi que les scores ROUGE les plus**

2. `https://huggingface.co/google/long-t5-tglobal-base`
3. `https://huggingface.co/facebook/bart-large-cnn`

élevés. Ces résultats indiquent une qualité de résumé nettement améliorée par rapport aux modèles non adaptés ou aux baselines génériques. Par exemple, `MedSum-LongT5` dépasse le modèle `LongT5-base` non fine-tuné, ce qui illustre l'intérêt du fine-tuning dans le domaine médical et l'efficacité de la variante LongT5 pour gérer des textes longs.

Modèle	Test Loss	ROUGE-1	ROUGE-2	ROUGE-L
`BART-large-CNN`	1.6944	35.9139	18.3085	25.0999
`LongT5-base`	1.2479	26.6719	13.8447	18.5348
`MedSum-T5`(ours)	0.9117	52.6865	34.9804	43.8119
`MedSum-LongT5`(ours)	**0.4605**	**52.5931**	**35.3102**	**42.9214**

TABLE 3.3 – Performances comparatives des quatre modèles sur le jeu de test (2500 exemples).

`MedSum-LongT5` vs `LongT5-base` Le modèle de base `LongT5-base`, n'ayant pas bénéficié d'un entraînement sur des données médicales spécifiques, affiche des performances inférieures (*loss* plus élevée, ROUGE plus bas). En particulier, l'écart en ROUGE-2 est notable, ce qui traduit une moindre précision dans la capture des informations clés bivalentes. Le fine-tuning médical de `MedSum-LongT5` lui permet d'apprendre le vocabulaire et les tournures spécifiques au domaine (meilleure qualité sémantique) et d'optimiser les pondérations sur des exemples cliniques. De plus, `LongT5-base` profite intrinsèquement de sa capacité à traiter de longues séquences d'entrée sans tronquer le texte [120]. Ainsi, `MedSum-LongT5` bénéficie à la fois de l'architecture adaptée aux longues entrées et du fine-tuning spécialisé, ce qui justifie son gain de performance par rapport à `LongT5-base`.

`MedSum-LongT5` vs `BART-large-CNN` `BART-large-CNN` atteint de bonnes performances de base en résumé généraliste, mais il n'a pas été conçu pour le domaine médical ni pour des séquences excessivement longues. `MedSum-LongT5` le devance en ROUGE-1 et ROUGE-2, ce qui reflète une meilleure adéquation au contenu du domaine de la santé et à la complexité sémantique des rapports médicaux. Le fine-tuning sur un corpus médical confère à `MedSum-LongT5` une précision supérieure dans la retranscription des faits cliniques, comme le soulignent les récents travaux montrant qu'un LLM adapté sur des données cliniques obtient des résumés souvent au moins aussi bons que ceux d'experts humains.

`MedSum-LongT5` vs `MedSum-T5` Les deux modèles ont bénéficié d'un fine-tuning médical, mais `MedSum-T5` utilise l'architecture `T5-base` d'origine, limitée à une longueur d'entrée maximale (généralement 512 tokens). En pratique, si un document médical dépasse cette longueur, le reste est tronqué, entraînant une perte d'information potentiellement critique. Cette contrainte réduit la qualité finale du résumé, surtout pour le ROUGE-2 et le ROUGE-L, qui dépendent de la continuité sémantique du texte. `MedSum-LongT5`, au contraire, peut ingérer de plus larges extraits (grâce à le *global attention*) et éviter les pertes de données. En résumé, `MedSum-LongT5` combine la pertinence du fine-tuning avec la puissance d'un modèle capable de traiter de longues entrées, ce qui se traduit par des gains significatifs en précision et en qualité sémantique par rapport à `MedSum-T5`.

Ces observations confirment que les améliorations observées (***loss* réduit et scores ROUGE supérieurs) sont dues à la spécialisation du modèle et à son architecture**. D'une part, **le fine-tuning sur des textes médicaux permet au modèle d'apprendre le jargon et les concepts propres au domaine.** D'autre part, le bénéfice du traitement de **séquences longues** se traduit par une meilleure cohérence globale, expliquant les performances supérieures de `MedSum-LongT5` sur l'ensemble des métriques.

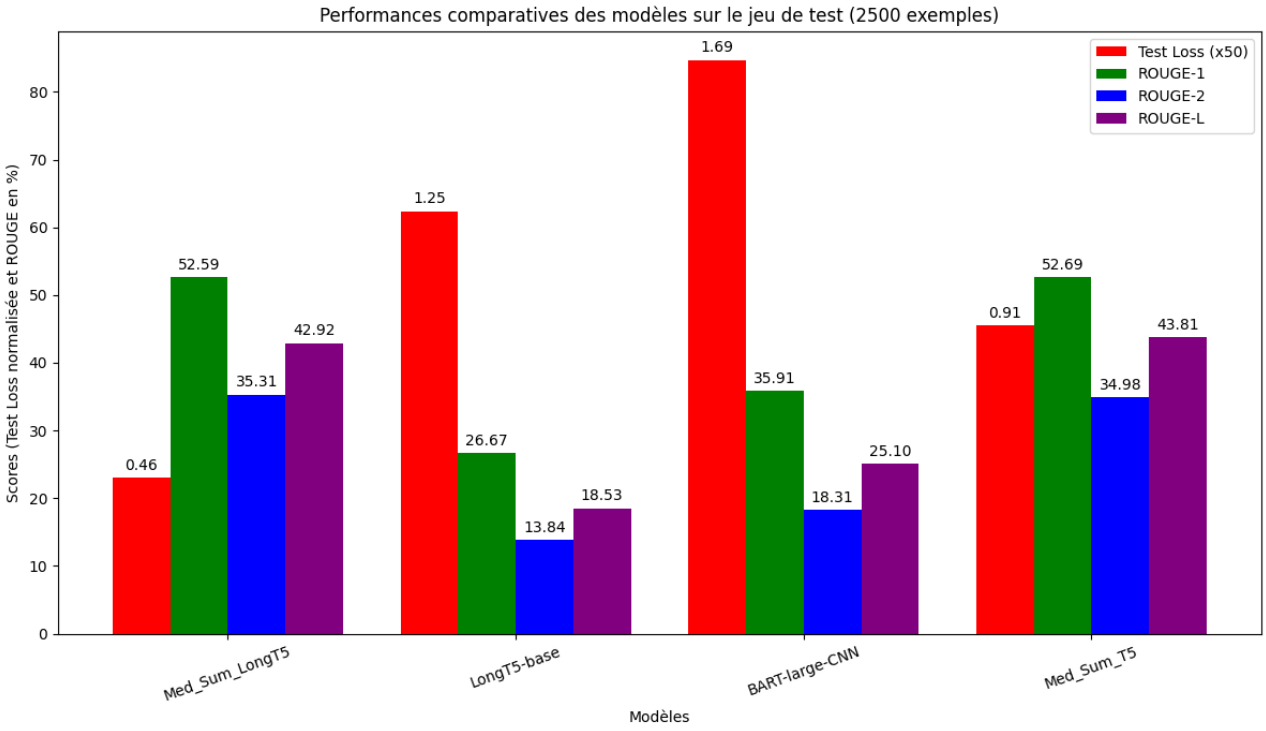


FIGURE 3.3 – Comparaison des performances (Test Loss, ROUGE-1, ROUGE-2, ROUGE-L) pour les quatre modèles évalués.

6 Analyse Qualitative et Expérimentation Interactive

Après avoir évalué les performances quantitatives du modèle, cette section propose une analyse qualitative détaillée pour illustrer concrètement ses forces et ses faiblesses. Nous présentons d'abord un exemple de génération issu d'un cas clinique du jeu de test, avant de discuter des tendances générales observées et du temps de génération sur un ordinateur personnel.

6.1 Exemple Illustratif de Génération sur un Cas Clinique

Le tableau 3.4 présente un cas typique soumis au modèle `MedSum-LongT5` via notre interface d'expérimentation.

Entrée (input)	<MAJOR SURGICAL OR INVASIVE PROCEDURE>___: Left craniotomy for aneurysm clipping <HISTORY OF PRESENT ILLNESS>___ year old male with several episodes of nonspecific feeling unwell over the past several months; evaluation of this revealed a 4-5mm left MCA aneurysm He presents today for elective left craniotomy for aneurysm clipping. <PERTINENT RESULTS>Please see OMR for pertinent lab and imaging results. <DISCHARGE MEDICATIONS>1. Docusate Sodium 100 mg PO BID 2. LevETIRAcetam 500 mg PO BID Duration: 7 Days RX *levetiracetam 500 mg 1 tablet(s) by mouth BID. Disp #*10 Tablet Refills: *0 3. OxyCODONE (Immediate Release) 5 mg PO Q6H: PRN pain Hold for sedation. Do not drive while taking. RX *oxycodone 5 mg 1 tablet(s) by mouth every six (6) hours Disp #*15 Tablet Refills: *0 4. Acetaminophen 325-650 mg PO Q6H: PRN fever or pain 5. Diltiazem Extended-Release 240 mg PO DAILY 6. Pravastatin 20 mg PO QPM 7. Sertraline 50 mg PO DAILY <DISCHARGE DIAGNOSIS>Cerebral aneurysm
Résumé humain	The patient presented for elective left craniotomy for aneurysm clipping. Postoperative medications included docusate sodium, levetiracetam, oxycodone, acetaminophen, diltiazem, pravastatin, and sertraline. Discharge diagnosis : Cerebral aneurysm.
Résumé généré	The patient was admitted for elective left craniotomy for aneurysm clipping. Discharge medications included Docusate Sodium, LevETIRAcetam, OxyCODONE, Acetaminophen, Diltiazem Extended-Release, Pravastatin, and Sertraline.

TABLE 3.4 – Exemple de génération de résumé par le modèle à partir d'un cas clinique réel.

Analyse comparative de l'exemple. Le résumé généré est globalement fidèle. Il reprend correctement la procédure principale (craniotomie) et la liste complète des médicaments. Cependant, il omet le diagnostic de sortie explicite ("Cerebral aneurysm"), bien que l'information soit implicite. Cette observation ponctuelle illustre une tendance générale : le modèle excelle à extraire des listes et des faits, mais peut parfois manquer des éléments de contexte ou de conclusion.

6.2 Synthèse des Forces et Limites Observées

En élargissant l'analyse à plusieurs exemples via l'interface (figures 3.4a, 3.4b), nous pouvons synthétiser les points forts et les faiblesses récurrentes du modèle.

6.2.0.1 Points forts Le modèle démontre une excellente capacité à identifier les informations structurées comme les **diagnostics principaux**, les **interventions chirurgicales** et

les **listes de médicaments**. De plus, les résumés suivent souvent une **structure logique** (historique → traitement → sortie), ce qui les rend cohérents et faciles à lire.

6.2.0.2 Faiblesses fréquentes Trois types d'erreurs principaux ont été identifiés : les **hallucinations factuelles** (ajout d'éléments absents de la source), l'**omission d'informations critiques** (comme des résultats biologiques pertinents), et des **problèmes de syntaxe** mineurs (phrases incomplètes ou confuses).

6.3 Performance en Déploiement Local

Les tests interactifs ont également permis de mesurer la performance du modèle sur l'ordinateur personnel (avec GPU RTX 2050), simulant une condition d'utilisation réelle. Sur des textes d'entrée de longueur variable (400 à 2000 tokens), le **temps moyen pour générer un résumé est de 98,93 secondes**.

Cette durée, supérieure aux 0,87 seconde mesurées sur le GPU A100, illustre le compromis entre la performance sur une infrastructure de recherche et celle sur du matériel grand public. Ce temps reste néanmoins acceptable pour une application où le résumé est généré en tâche de fond, sans nécessiter une réponse instantanée.

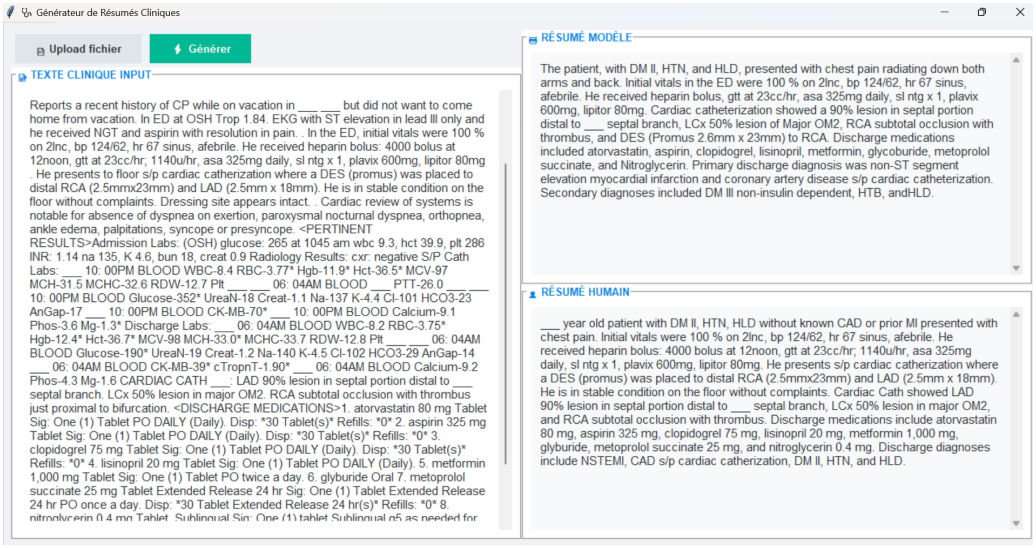
7 Discussion Générale et Perspectives

L'évaluation de notre modèle `MedSum-LongT5` a démontré des performances quantitatives et qualitatives solides. Cette section vise à synthétiser ces résultats, à discuter des forces et des limites de l'approche, et à tracer des pistes d'amélioration directes pour les travaux futurs.

Synthèse et interprétation des performances Notre approche, centrée sur le fine-tuning d'un modèle spécialisé pour les textes longs, s'est avérée être une stratégie gagnante. Les performances finales sur l'ensemble de test, avec des scores atteignant **52,59 % pour ROUGE-1** et **35,31 % pour ROUGE-2**, attestent de la capacité du modèle à générer des résumés fidèles et de haute qualité. Ce succès repose sur deux piliers : premièrement, le **fine-tuning sur des données médicales** a permis au modèle d'acquérir le vocabulaire et la structure syntaxique des rapports cliniques ; deuxièmement, **l'architecture de LongT5** a été cruciale pour assurer la cohérence et la couverture des informations en gérant efficacement le contexte sur de longues séquences.

Limites identifiées et pistes d'amélioration techniques Malgré ces succès, notre analyse met en évidence des limites concrètes qui constituent des axes d'amélioration immédiats :

- **Biais et généralisation des données :** Le corpus MIMIC-IV, bien que riche, est géographiquement et institutionnellement limité, ce qui peut affecter la robustesse du modèle sur des données différentes et sur des cas cliniques rares.
- **Coût computationnel :** L'architecture de LongT5 reste gourmande en ressources GPU, ce qui peut être un frein à son déploiement ou à des ré-entraînements fréquents.



(a) Exemple Patient A

(b) Exemple Patient B

FIGURE 3.4 – Interface de test interactif développée.

- **Fiabilité factuelle :** L'analyse qualitative a révélé des cas, bien que modérés, **d'hallucinations ou d'omissions d'informations**, ce qui est une barrière pour une utilisation clinique non supervisée.

Face à ces constats, plusieurs perspectives techniques peuvent être envisagées pour renforcer le modèle :

1. **Enrichissement et augmentation du corpus :** Intégrer des données multi-centriques pour améliorer la généralisation et appliquer des stratégies d'augmentation de données pour mieux traiter les cas rares.
2. **Optimisation de l'architecture et de l'inférence :** Explorer des techniques de com-

pression de modèle, comme **la quantification** [4] ou la distillation [5], pour réduire l'empreinte mémoire et le coût computationnel sans sacrifier significativement la performance.

3. **Amélioration de la fidélité et du pilotage :** Expérimenter des techniques de *prompt engineering* plus avancées pour mieux contraindre la génération et réduire les hallucinations. Un cycle d'apprentissage continu, basé sur les retours d'experts, pourrait également corriger les biais de manière itérative.

Implications et conclusion de l'étude En conclusion, ce travail valide l'efficacité d'un modèle de type Transformer, spécialisé et adapté aux textes longs, pour la synthèse de l'information médicale. En dépit des limites identifiées, `MedSum-LongT5` représente une avancée significative. Les résultats confirment que notre approche répond à l'objectif principal de ce travail : construire un système performant pour le résumé automatique de rapports médicaux complexes. Le modèle constitue ainsi une base solide, prometteuse pour des applications cliniques assistées par l'IA, à condition de maintenir une supervision humaine pour valider les cas critiques.

8 Conclusion

Ce chapitre a présenté une évaluation exhaustive du modèle **MedSum-LongT5**. L'analyse confirme que le fine-tuning spécialisé sur le corpus MIMIC-IV a permis d'atteindre des performances robustes, comme en témoignent les scores ROUGE élevés qui surpassent les modèles de référence. Sur le plan qualitatif, le modèle a démontré sa capacité à extraire l'information clinique essentielle avec une structure cohérente, bien que des limites subsistent, notamment en matière de généralisation, de couverture des cas rares et de coût computationnel.

Dans l'ensemble, ces observations confirment que **l'objectif principal de ce mémoire — construire un modèle performant pour le résumé automatique de rapports médicaux complexes — a été atteint**. Le modèle proposé se révèle prometteur pour des applications cliniques assistées par l'IA, à condition de prévoir une supervision humaine pour valider les cas critiques.

Enfin, cette évaluation ouvre la voie vers des pistes d'amélioration et d'intégration, qui seront discutées dans le chapitre suivant dédié à la conclusion générale et aux perspectives futures.

4. La quantification consiste à réduire la précision des poids et des activations du modèle (par exemple de 32 bits à 8 bits), afin de diminuer la taille mémoire et d'accélérer l'inférence sur des appareils à ressources limitées.

5. La distillation de connaissances (*knowledge distillation*) est une méthode d'apprentissage supervisé qui consiste à entraîner un modèle plus petit (étudiant) à imiter les prédictions d'un modèle plus grand (enseignant), introduite par Hinton et al. [121].

Conclusion Générale

Dans ce mémoire, nous nous sommes intéressés à la problématique de l'automatisation de l'analyse des dossiers médicaux, un enjeu majeur pour l'optimisation de la prise de décision clinique. Nous avons proposé et validé une approche basée sur le fine-tuning d'un modèle de langage spécialisé pour générer des résumés de notes cliniques. Nous clôturons ce document par une synthèse des contributions apportées, une discussion des perspectives de recherche futures, et un résumé des connaissances acquises durant ce travail.

Contributions

L'apport principal de ce travail réside dans le développement d'un système pour le résumé de textes médicaux longs. Nos contributions peuvent être résumées en trois points :

- **Une méthodologie de traitement robuste :** Nous avons mis en place un pipeline complet, allant d'un prétraitement rigoureux des données du corpus **MIMIC-IV** à une stratégie de fine-tuning efficace du modèle **LongT5** via la méthode **LoRA**. Cette méthodologie inclut une étape originale de génération de résumés cibles par une API LLM pour garantir une supervision de haute qualité.
- **Un modèle performant et validé :** Nous avons produit et évalué le modèle `MedSum-LongT5` qui atteint des performances solides sur des données non vues. Avec des scores de **52,59 % en ROUGE-1**, **35,31 % en ROUGE-2 42,92 % en ROUGE-L**, notre modèle surpasse significativement les approches de référence, démontrant l'efficacité de la spécialisation pour les textes longs du domaine médical.
- **Une démonstration applicative :** Au-delà des métriques, nous avons intégré le modèle final dans une interface graphique fonctionnelle. Cette application démontre la faisabilité d'un déploiement local et permet une évaluation interactive, illustrant le potentiel pratique de notre solution dans un environnement clinique simulé.

Perspectives

Ce travail ouvre la voie à plusieurs perspectives d'amélioration et de recherche, à court et long terme.

Améliorations techniques directes. Pour répondre aux limites identifiées, les prochaines étapes pourraient se concentrer sur l'optimisation du modèle. Des techniques comme la **quanti-**

fication ou la **distillation de connaissances** permettraient de réduire son coût computationnel. Par ailleurs, des stratégies de ***prompt engineering*** avancées et l'intégration de boucles de **retour d'experts** pourraient améliorer la fidélité factuelle et diminuer les *hallucinations*.

Axes de recherche futurs. À plus long terme, plusieurs directions de recherche passionnantes peuvent être explorées. On pourrait étendre le système à un contexte **multimodal**, en lui permettant d'analyser à la fois le texte des notes et les données structurées associées (résultats biologiques, données démographiques). Enfin, une autre perspective serait de l'adapter pour générer des résumés **orientés patient**, dans un langage simplifié, afin d'améliorer la communication médecin-patient.

Acquis Personnels

La réalisation de ce mémoire a été une expérience d'apprentissage riche et formatrice. Sur le plan technique, ce projet m'a permis de maîtriser l'ensemble de la chaîne de développement d'un projet en NLP, depuis la manipulation de grands corpus textuels jusqu'au déploiement d'un modèle de deep learning. J'ai pu acquérir des compétences solides sur des technologies de pointe comme les architectures **Transformer**, les techniques de fine-tuning efficace (**PEFT/LoRA**) et l'utilisation des librairies de l'écosystème **Hugging Face** (`transformers`, `datasets`, `evaluate`).

Au-delà des aspects techniques, ce travail a renforcé mes capacités en gestion de projet, et en résolution de problèmes complexes. La nécessité de comprendre les enjeux du domaine médical m'a également appris à dialoguer avec un champ d'expertise différent et à adapter la technologie pour répondre à des besoins concrets et critiques.

Bibliographie

[1] M.D. Ahmed Hassan. Big data in healthcare : Opportunities and challenges. https://www.mghihp.edu/news-and-more/opinions/data-analytics/big-data-healthcare-opportunities-and-challenges, January 2025. Consulté le 20 juin 2025.

[2] Brian Eastwood. How to navigate structured and unstructured data as a healthcare organization. *HealthTech Magazine*, May 2023. Accessed : 2025-06-20.

[3] Christine Sinsky, Lacey Colligan, Ling Li, Mirela Prgomet, Sam Reynolds, Lindsey Goeders, Johanna Westbrook, Michael Tutty, and George Blike. Allocation of physician time in ambulatory practice : a time and motion study in 4 specialties. *Annals of internal medicine*, 165(11) :753–760, 2016.

[4] Ministère des Solidarités et de la Santé (France). Risques médicaux et sécurité des soins. https://www.sante.fr/risque-medical-et-securite-des-soins, 2025. Consulté le 20 juin 2025.

[5] Kent. 10 benefits of automation in healthcare. https://www.folderit.com/blog/10-benefits-of-automation-in-healthcare/, November 2023. Consulté le 20 juin 2025.

[6] Andrew A Borkowski, Colleen E Jakey, Stephen M Mastorides, Ana L Kraus, Gitanjali Vidyarthi, Narayan Viswanadhan, and Jose L Lezama. Applications of chatgpt and large language models in medicine and health care : benefits and pitfalls. *Federal Practitioner*, 40(6) :170, 2023.

[7] Kai He, Rui Mao, Qika Lin, Yucheng Ruan, Xiang Lan, Mengling Feng, and Erik Cambria. A survey of large language models for healthcare : from data, technology, and applications to accountability and ethics. *Information Fusion*, page 102963, 2025.

[8] Akash Ghosh, Arkadeep Acharya, Raghav Jain, Sriparna Saha, Aman Chadha, and Setu Sinha. Clipsyntel : clip and llm synergy for multimodal question summarization in healthcare. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22031–22039, 2024.

[9] Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. Longt5 : Efficient text-to-text transformer for long sequences. *arXiv preprint arXiv :2112.07916*, 2021.

[10] Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, et al. Mimic-iv, a freely accessible electronic health record dataset. *Scientific data*, 10(1) :1, 2023.

[11] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora : Low-rank adaptation of large language models. *arXiv preprint arXiv :2106.09685*, 2021.

[12] Chin-Yew Lin. Rouge : A package for automatic evaluation of summaries. In *Text summarization branches out*. ACL, 2004.

[13] Andrea Matarazzo and Riccardo Torlone. A survey on large language models with some insights on their capabilities and limitations. *arXiv preprint arXiv :2501.04040*, 2025.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[15] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. *arXiv preprint arXiv :2307.06435*, 2023.

[16] Jianfeng Gao and Chin-Yew Lin. Introduction to the special issue on statistical language modeling, 2004.

[17] Ronald Rosenfeld. Two decades of statistical language modeling : Where do we go from here ? *Proceedings of the IEEE*, 88(8) :1270–1278, 2000.

[18] John D Kalbfleisch and Jerald Franklin Lawless. The analysis of panel data under a markov assumption. *Journal of the american statistical association*, 80(392) :863–871, 1985.

[19] Xiaoyong Liu and W Bruce Croft. Statistical language modeling for information retrieval. *Annu. Rev. Inf. Sci. Technol.*, 39(1) :1–31, 2005.

[20] Lalit R Bahl, Peter F Brown, Peter V De Souza, and Robert L Mercer. A tree-based statistical language model for natural language speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(7) :1001–1008, 1989.

[21] Thorsten Brants, Ashok Popat, Peng Xu, Franz Josef Och, and Jeffrey Dean. Large language models in machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 858–867, 2007.

[22] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8) :1735–1780, 1997.

[23] Felipe Almeida and Geraldo Xexéo. Word embeddings : A survey. *arXiv preprint arXiv :1901.09069*, 2019.

[24] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv :1301.3781*, 2013.

[25] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv :2303.18223*, 1(2), 2023.

[26] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pages 4171–4186, 2019.

[27] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv :2206.07682*, 2022.

[28] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8) :9, 2019.

[29] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert : Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv :1810.04805*, 2018.

[30] Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv :2005.14165*, 2020.

[31] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart : Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv :1910.13461*, 2019.

[32] DeepLearningAI. Generative ai with large language models. `https://www.coursera.org/learn/generative-ai-with-llms`, 2024. Coursera course.

[33] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv :2108.07258*, 2021.

[34] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert : a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4) :1234–1240, 2020.

[35] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1) :1–23, 2021.

[36] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv :2107.03374*, 2021.

[37] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140) :1–67, 2020.

[38] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta : A robustly optimized bert pretraining approach. *arXiv preprint arXiv :1907.11692*, 2019.

[39] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert : smaller, faster, cheaper and lighter. *arXiv preprint arXiv :1910.01108*, 2019.

[40] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert : A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv :1909.11942*, 2019.

[41] Gokul Yenduri, Ramalingam M, Chemmalar Selvi G, Supriya Y, Gautam Srivastava, Praveen Kumar Reddy Maddikunta, Deepti Raj G, Rutvij H Jhaveri, Prabadevi B, Weizheng Wang, Athanasios V Vasilakos, and Thippa Reddy Gadekallu. Gpt (generative pre-trained transformer) – a comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions. *arXiv preprint arXiv :2305.10435*, 2023.

[42] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

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

[44] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5 : A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv :2010.11934*, 2020.

[45] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70) :1–53, 2024.

[46] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama : Open and efficient foundation language models, 2023.

[47] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang,

Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2 : Open foundation and fine-tuned chat models, 2023.

[48] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35 :27730–27744, 2022.

[49] Joshua Ainslie, James Lee-Thorp, Michiel De Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa : Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv :2305.13245*, 2023.

[50] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv :2407.21783*, 2024.

[51] Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew BA McDermott. Publicly available clinical bert embeddings. *arXiv preprint arXiv :1904.03323*, 2019.

[52] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining : Adapt language models to domains and tasks. In *Proceedings of the Association for Computational Linguistics (ACL)*, 2020.

[53] Long N Phan, Hieu T Nguyen, Tuan D Nguyen, and Truong-Son Nguyen. Scifive : a text-to-text transformer model for biomedical literature. *arXiv preprint arXiv :2106.03598*, 2021.

[54] Qingyu Lu, Dejing Dou, and Thien Huu Nguyen. Clinicalt5 : A generative language model for clinical text. In *Findings of the Association for Computational Linguistics : EMNLP 2022*, pages 5436–5443, 2022.

[55] Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. Biogpt : generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6) :bbac409, 2022.

[56] Ankit Venigalla, Jonathan Frankle, and Michael Carbin. Biomedlm : a domain-specific large language model for biomedical text. *MosaicML*, 2022.

[57] Jan Clusmann, Fiona R Kolbinger, Bastian Rieck, Daniel Truhn, and Jakob Nikolas Kather. The future landscape of large language models in medicine. *Communications Medicine*, 3(1) :141, 2023.

[58] Daniel Ferber, Yuxuan Zhang, Yuxuan Zhang, and Yuxuan Zhang. Gpt-4 for information retrieval and comparison of medical oncology guidelines. *NEJM AI*, 1(AIcs2300235), 2024.

[59] Antoine Bosselut, Yuxuan Zhang, Yuxuan Zhang, and Yuxuan Zhang. Meditron : Open medical foundation models adapted for clinical practice. *Preprint*, 2024.

[60] Deshiwei Zhang, Xiaojuan Xue, and Xiayang Ying. A survey of datasets in medicine for large language models. *Intelligence & Robotics*, (27), 2024.

[61] Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, Li wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3(1) :160035, 2016.

[62] Emily Herrett, Arlene M Gallagher, Krishnan Bhaskaran, Harriet Forbes, Rohini Mathur, Tjeerd van Staa, and Liam Smeeth. Data resource profile : Clinical practice research datalink (cprd). *International Journal of Epidemiology*, 44(3) :827–836, 06 2015.

[63] PubMed Data. `https://pubmed.ncbi.nlm.nih.gov/download/`, 2023. Accessed April 12, 2024.

[64] PubMed Central (PMC). `https://www.ncbi.nlm.nih.gov/pmc/`, 2024. Accessed April 12, 2024.

[65] Byron C. Wallace, Sayantan Saha, Frank Soboczenski, and Iain J. Marshall. Generating (factual ?) narrative summaries of rcts : Experiments with neural multi-document summarization. *AMIA Joint Summits on Translational Science Proceedings*, 2021 :605–614, 2021.

[66] John DeYoung, Iz Beltagy, Matt van Zuylen, Ben Kuehl, and Luke S. Wang. MS^2 : Multi-Document Summarization of Medical Studies. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7494–7513, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. Accessed 12 April.

[67] Vishal Gupta, Piyush Bharti, Payal Nokhiz, and Harsh Karnick. SumPubMed : Summarization dataset of PubMed scientific articles. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing : Student Research Workshop*, pages 292–303, 2021. Accessed 15 April.

[68] Luke S. Wang, Kaitao Lo, Yoganand Chandrasekhar, and et al. CORD-19 : The COVID-19 Open Research Dataset. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*, Online, 2020. Association for Computational Linguistics. Accessed 15 April.

[69] Abdulrahman Gokaslan and Vincent Cohen. OpenWebText Corpus. `http://Skylion007.github.io/OpenWebTextCorpus`, 2019. Accessed 15 April.

[70] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140) :1–67, 2020.

[71] Dennis A. Benson, Mark Cavanaugh, Karen Clark, Ilene Karsch-Mizrachi, David J. Lipman, James Ostell, and Eric W. Sayers. GenBank. *Nucleic Acids Research*, 32(suppl_1) :D267–D270, 2004. Accessed 15 April.

[72] Hilla Rosen, Ofer Biran, Nathan Agmon, and Arnon Lotan. Blockchain-based secure data sharing for electronic medical records in cloud environments. *Information*, 11(4) :186, 2020. Accessed 15 April.

[73] Yi Chen, Jingping Liu, Hailin Li, Feng Li, Mark S. Chen, and Cheng-Ju Kuo. Reference sequence (RefSeq) database at NCBI : current status, taxonomic expansion, and functional annotation. *Nucleic Acids Research*, 46(D1) :D1074–D1082, 2018. Accessed 15 April.

[74] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. PubMedQA : A dataset for biomedical research question answering. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577, Hong Kong, China, November 2019. Association for Computational Linguistics.

[75] Xuehai He, Shu Chen, Zeqian Ju, Xiangyu Dong, Hongchao Fang, Sicheng Wang, Yue Yang, Jiaqi Zeng, Ruisi Zhang, Ruoyu Zhang, Meng Zhou, Penghui Zhu, and Pengtao Xie. MedDialog : Two Large-scale Medical Dialogue Datasets. *CoRR*, abs/2004.03329, 2020. Accessed 15 April.

[76] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. Chexpert : A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 590–597, 2019.

[77] Xuehai He, Yichen Zhang, Luntian Mou, Eric Xing, and Pengtao Xie. PathVQA : 30000+ Questions for Medical Visual Question Answering. *CoRR*, abs/2003.10286, 2020. Accessed 15 April.

[78] Xingyao Wang, Yu Zhang, Xiang Ren, Yu Zhang, Marinka Zitnik, Jingbo Shang, and Jiawei Han. Cross-type biomedical named entity recognition with deep multi-task learning. *Bioinformatics*, 35(10) :1745–1752, 2018.

[79] Qiao Jin, Bhuwan Dhingra, Zheng Liu, William W Cohen, and Xinghua Lu. Pubmedqa : A dataset for biomedical research question answering. *arXiv preprint arXiv :1909.06146*, 2019.

[80] Fei Li, Yingjun Jin, Wei Liu, Bhuwan P S Rawat, Ping Cai, and Hong Yu. Fine-tuning bidirectional encoder representations from transformers (bert)–based models on large-scale electronic health record notes : An empirical study. *JMIR Medical Informatics*, 8(3) :e18457, 2020.

[81] M. Nishio et al. Fully automatic summarization of radiology reports using natural language processing with language models. *medRxiv*, 2023.

[82] D. Van Veen et al. Clinical text summarization : Adapting large language models can outperform human experts. *PMC*, 2024.

[83] A. Pal et al. Neural summarization of electronic health records. *arXiv*, 2023.

[84] R. Tariq et al. Patient-centered radiology reports with generative artificial intelligence. *Scientific Reports*, 2024.

[85] H. Berg and H. Dalianis. Using bart to automatically generate discharge summaries from swedish clinical text. *ACL Anthology*, 2024.

[86] Q. Lu et al. Clinicalt5 : A generative language model for clinical text. *EMNLP*, 2022.

[87] S. Gururangan et al. Don't stop pretraining : Adapt language models to domains and tasks. *ACL*, 2020.

[88] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu : a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[89] L. Tang et al. Evaluating large language models on medical evidence summarization. *npj Digital Medicine*, 2023.

[90] W. Kryściński et al. Evaluating the factual consistency of abstractive text summarization. *EMNLP*, 2020.

[91] M. Nishio et al. Fully automatic summarization of radiology reports using natural language processing with language models. *medRxiv*, 2023.

[92] Alistair E. W. Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J. Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, Li-wei H. Lehman, Leo A. Celi, and Roger G. Mark. Mimic-iv, a freely accessible electronic health record dataset. *Scientific Data*, 10(1) :1, 2023.

[93] Ary L. Goldberger, Luis A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, and H. Eugene Stanley. PhysioBank, PhysioToolkit, and PhysioNet : Components of a new research resource for complex physiologic signals. *Circulation [Online]*, 101(23) :e215–e220, 2000. RRID :SCR_007345.

[94] Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Brian Gow, Benjamin Moody, Steven Horng, Leo Anthony Celi, and Roger Mark. MIMIC-IV (version 3.1). `https://doi.org/10.13026/kpb9-mt58`, 2024. PhysioNet, RRID :SCR_007345.

[95] Alistair Johnson, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. MIMIC-IV-Note : Deidentified free-text clinical notes (version 2.2). `https://doi.org/10.13026/1n74-ne17`, 2023. PhysioNet, RRID :SCR_007345.

[96] Alistair Johnson, Tom Pollard, Roger Mark, Seth Berkowitz, and Steven Horng. MIMIC-CXR Database (version 2.1.0). `https://doi.org/10.13026/4jqj-jw95`, 2024. PhysioNet, RRID :SCR_007345.

[97] Amir Aali, David Van Veen, Yasin Arefeen, Jason Hom, Christian Bluethgen, Emily P. Reis, Stavros Gatidis, Nicholas Clifford, James Daws, Ali Tehrani, Jin Kim, and Akshay Chaudhari. MIMIC-IV-Ext-BHC : Labeled Clinical Notes Dataset for Hospital Course Summarization (version 1.2.0). `https://doi.org/10.13026/5gte-bv70`, 2025. PhysioNet, RRID :SCR_007345.

[98] Asad Aali, Dave Van Veen, Yamin Ishraq Arefeen, Jason Hom, Christian Bluethgen, Eduardo Pontes Reis, Sergios Gatidis, Namuun Clifford, Joseph Daws, Arash S Tehrani, et al. A dataset and benchmark for hospital course summarization with adapted large

language models. *Journal of the American Medical Informatics Association*, 32(3) :470–479, 2025.

[99] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv :1910.10683*, 2019.

[100] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. Pegasus : Pre-training with extracted gap-sentences for abstractive summarization. In *International conference on machine learning*, pages 11328–11339. PMLR, 2020.

[101] PhD Raschka, Sebastian. Practical tips for finetuning llms using LoRA (low-rank adaptation). AheadofAI (SebastianRaschka's AIMagazine), November 2023. Accessed via magazine.sebastianraschka.com ; URL : https ://magazine.sebastianraschka.com/p/practical-tips-for-finetuning-llms.

[102] Taku Kudo and John Richardson. Sentencepiece : A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing : System Demonstrations*, pages 66–71, 2018.

[103] Intel Corporation. OpenVINO Tokenizers : Tokenizer and Detokenizer for Generative AI Workflow. `https://docs.openvino.ai/nightly/openvino-workflow-generative/ov-tokenizers.html`, 2025. Accessed : 2025-06-14.

[104] Ronny Krashinsky, Olivier Giroux, Stephen Jones, Nick Stam, and Sridhar Ramaswamy. NVIDIA Ampere Architecture In-Depth. NVIDIA Developer Blog, May 2020. Accessed : 2025-06-14.

[105] Dhiraj Kalamkar, Dheevatsa Mudigere, Naveen Mellempudi, Dipankar Das, Kunal Banerjee, Sasikanth Avancha, Dharma Teja Vooturi, Nataraj Jammalamadaka, Jianyu Huang, Hector Yuen, et al. A study of bfloat16 for deep learning training. *arXiv preprint arXiv :1905.12322*, 2019.

[106] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv :1711.05101*, 2017.

[107] Diederik P Kingma and Jimmy Ba. Adam : A method for stochastic optimization. *arXiv preprint arXiv :1412.6980*, 2014.

[108] Ahmed Yassin. Adam vs. adamw : Understanding weight decay and its impact on model performance, 2022. Article Medium.

[109] Dive into Deep Learning. Learning rate scheduling, 2023. Documentation D2L.

[110] Asad Aali, Dave Van Veen, Yamin Ishraq Arefeen, Jason Hom, Christian Bluethgen, Eduardo Pontes Reis, Sergios Gatidis, Namuun Clifford, Joseph Daws, Arash S Tehrani, et al. A dataset and benchmark for hospital course summarization with adapted large language models. *Journal of the American Medical Informatics Association*, page ocae312, 2024.

[111] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch : An

imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8026–8037. Curran Associates, Inc., 2019.

[112] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, et al. Transformers : State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing : System Demonstrations*, pages 38–45. Association for Computational Linguistics, 2020.

[113] Hugging Face PEFT team. Peft : Parameter-efficient fine-tuning. `https://github.com/huggingface/peft`, 2022.

[114] Quentin Lhoest, Albert Suárez Villanova, Leandro von Werra, Victor Sanh, Lewis Debut, Julien Chaumond, and Thomas Wolf. Datasets : A community library for natural language processing. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 175–186. Association for Computational Linguistics, 2021.

[115] Wes McKinney. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. SciPy, 2010.

[116] Python Software Foundation. *re — Regular expression operations*, 2025. Version 3.12.

[117] Google. Google ai for developers - gemini api. `https://ai.google.dev/`, 2025. Accessed : June 2025.

[118] Python Software Foundation. *tkinter — Python interface to Tcl/Tk*, 2025. Version 3.12.

[119] Federico Moramarco and et al. Evaluation of automatic summarization metrics for medical evidence summarization. *arXiv preprint arXiv :2205.04761*, 2022.

[120] Mandar Gupta, Dheeru Dua, Matt Gardner, Partha Talukdar, and Sameer Singh. Scaling long-form question answering. *arXiv preprint arXiv :2101.10318*, 2021.

[121] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv :1503.02531*, 2015.

.