



Université de Tunis El Manar

École Nationale d'Ingénieurs de Tunis

Département des Technologies de l'Information et de la Communication

Rapport de Stage Ingénieur

Système Intelligent de Triage des Patients Utilisant l'IA et l'Automatisation

Présenté par :

Wael ETTEYEB

Encadré par :

Mme. Amira BENJEMAA



Technozor

Remerciements

Je tiens à exprimer ma sincère gratitude à mon encadrante, **Mme. Amira BENJEMAA** pour son soutien et ses précieux conseils tout au long de mon stage chez Technozor. Son expertise et son dévouement en développement ont été une source d'inspiration et d'apprentissage inestimable pour moi.

Je tiens également à remercier chaleureusement toute l'équipe de Technozor pour son accueil chaleureux et son environnement de travail collaboratif. Leur partage de connaissances et leur soutien m'ont permis de m'épanouir professionnellement et de contribuer de manière significative à ce projet.

Enfin, je tiens à remercier aussi le personnel administratif de l'École nationale d'ingénieurs de Tunis, en particulier l'équipe dédiée au sein du service Informatique et de la Communication. Leurs efforts infinis pour nous fournir une expérience éducative exceptionnelle ont jeté les bases de notre croissance académique et professionnelle.

Ce stage a été une expérience extrêmement enrichissante, et je suis reconnaissant envers toutes les personnes qui ont rendu cela possible.

Abstract

This project presents the development of an intelligent medical triage system, implementing artificial intelligence to analyze patient symptoms and medical documents. The project uses a modern architecture based on **Streamlit** for the user interface, **FastAPI** for backend services, and the **Groq/LLama/BioGPT** ecosystem for medical reasoning. The system combines **Large Language Models (LLM)** with **Retrieval-Augmented Generation (RAG)** to provide accurate triage recommendations. Healthcare professionals can input patient symptoms, upload medical documents, receive triage assessments, and automate follow-up actions through integrated workflows.

The report details the technical choices, overall architecture, and various implemented features, highlighting the use of tools and technologies selected to effectively meet the project's requirements.

Keywords : Medical Triage, Artificial Intelligence, Natural Language Processing, Process Automation, Healthcare Technology, LLM, RAG

Résumé

Ce projet présente le développement d'un système intelligent de triage médical, mettant en œuvre l'intelligence artificielle pour analyser les symptômes des patients et les documents médicaux. Le projet utilise une architecture moderne basée sur **Streamlit** pour l'interface utilisateur, **FastAPI** pour les services backend, et l'écosystème **Groq/LLama/BioGPT** pour le raisonnement médical. Le système combine les **modèles de langage de grande taille (LLM)** avec la **génération augmentée par la récupération (RAG)** pour fournir des recommandations de triage précises. Les professionnels de santé peuvent saisir les symptômes des patients, télécharger des documents médicaux, recevoir des évaluations de triage et automatiser les actions de suivi grâce à des workflows intégrés.

Le rapport détaille les choix techniques, l'architecture globale et les différentes fonctionnalités mises en œuvre, en mettant en lumière l'utilisation des outils et technologies sélectionnés pour répondre efficacement aux exigences du projet.

Mots clés : Triage Médical, Intelligence Artificielle, Traitement du Langage Naturel, Automatisation des Processus, Technologie de Santé, LLM, RAG

Table des matières

Table des figures	7
Liste des Acronymes	8
Introduction générale	9
1 Cadre général du projet	10
1.1 Introduction	10
1.2 Présentation de l'organisme d'accueil	10
1.2.1 Missions et activités	11
1.3 Contexte du projet	11
1.3.1 Problématique	11
1.3.2 Étude de l'existant	12
1.4 Solution proposée	13
1.4.1 Système de triage IA LLM + RAG	13
1.4.2 Automatisation des processus avec n8n	14
1.4.3 Avantages de l'approche proposée	14
1.5 Conclusion	15
2 Extraction et préparation des données médicales	16
2.1 Introduction	16
2.2 Stratégie de collecte des données médicales	16
2.2.1 Sources de données utilisées	17
2.2.2 Techniques de collecte automatisée	17
2.3 Prétraitement et nettoyage des données	18
2.3.1 Normalisation des textes médicaux	18
2.3.2 Segmentation et organisation du contenu	18
2.4 Extraction et structuration de l'information médicale	20
2.4.1 Reconnaissance d'entités médicales	20
2.4.2 Modélisation des guidelines structurées	20
2.5 Transformation vers le format JSON	21

2.5.1	Structure JSON optimisée	21
2.5.2	Processus de génération automatique	22
2.6	Résultats et analyse	23
2.6.1	Statistiques de la base de guidelines	23
2.6.2	Qualité des données générées	23
2.7	Conclusion	23
3	Conception et Architecture du Système	25
3.1	Introduction	25
3.2	Architecture globale du système	25
3.2.1	Vue d'ensemble	25
3.2.2	Composants principaux	27
3.3	Conception du pipeline de triage IA	28
3.3.1	Pipeline LLM + RAG	28
3.3.2	Processus de raisonnement médical	29
3.4	Conception du workflow d'automatisation n8n	30
3.4.1	Architecture du workflow	30
3.4.2	Logique décisionnelle contextuelle	32
3.5	Conclusion	33
4	Réalisation Technique et Validation	34
4.1	Introduction	34
4.2	Architecture technique globale	34
4.2.1	Stack technologique complète	34
4.2.2	Intégration des composants	35
4.3	Choix des technologies frontend	35
4.3.1	Streamlit pour l'interface utilisateur	35
4.3.2	Composants d'interface	36
4.4	Technologies backend et IA	37
4.4.1	FastAPI pour les services web	37
4.4.2	LLama et Groq pour le raisonnement médical	37
4.4.3	BioGPT pour les embeddings médicaux	37
4.4.4	ChromaDB pour la base vectorielle	38
4.5	Automatisation avec n8n	38
4.5.1	Architecture des workflows	38
4.5.2	Intégration des services Google	39
4.6	Flux complet de traitement	39
4.6.1	Du symptôme à l'action automatisée	39
4.6.2	Fonctionnalités principales	40
4.6.3	Captures d'écran représentatives	41
4.7	Conclusion	44

Conclusion	45
Bibliographie	46

Table des figures

1.1	Logo de l'entreprise Technozor	10
2.1	Architecture des sources de données médicales utilisées pour l'ex- traction des guidelines	17
2.2	Pipeline complet de collecte et prétraitement des données médicales	19
2.3	Pipeline d'extraction des entités médicales à partir des textes normalisés	21
2.4	Processus de transformation des données vers le format JSON structuré	22
3.1	Architecture globale du système de triage médical intelligent	26
3.2	Pipeline de triage médical combinant LLM et RAG	28
3.3	Architecture du workflow d'automatisation médicale	31
4.1	Architecture technique complète du système de triage médical . . .	35
4.2	Capture d'écran de l'interface utilisateur développée avec Streamlit	36
4.3	Workflow n8n complet pour l'automatisation des processus médicaux	39
4.4	Diagramme du flux complet de traitement depuis l'input jusqu'à l'automatisation	40
4.5	Interface de saisie des symptômes	41
4.6	Interface de traitement de documents PDF	42
4.7	Interface des résultats de triage médical	42
4.8	Interface du chatbot médical interactif	43
4.9	Exemple d'email automatisé généré	43
4.10	Interface de réservation automatique de rendez-vous	44

Liste des Acronymes

IA Intelligence Artificielle
LLM Large Language Model
RAG Retrieval-Augmented Generation
NLP Natural Language Processing
OCR Optical Character Recognition
MTS Manchester Triage System
ESI Emergency Severity Index
CTAS Canadian Triage and Acuity Scale
GUID Guidelines
OMS Organisation Mondiale de la Santé
CDC Centers for Disease Control and Prevention
HAS Haute Autorité de Santé
NEJM New England Journal of Medicine
EMEA Europe, Middle East, and Africa
IGF Internet Governance Forum
PDF Portable Document Format
JSON JavaScript Object Notation
HTML HyperText Markup Language
UTF Unicode Transformation Format

Introduction générale

La digitalisation des services de santé représente un enjeu majeur dans l'optimisation des soins médicaux, particulièrement dans le domaine du triage où la rapidité et la précision des décisions sont cruciales. Ce projet s'inscrit dans cette transformation en proposant un système intelligent d'assistance au triage médical exploitant les dernières avancées en intelligence artificielle et en automatisation des processus.

Le contexte actuel du triage médical fait face à plusieurs défis significatifs : la variabilité inter-praticien dans l'évaluation de la gravité des cas, la charge cognitive importante pesant sur le personnel soignant, et le besoin de standardisation basée sur les dernières recommandations médicales. Ces limitations des systèmes traditionnels justifient le développement d'une solution innovante combinant analyse avancée par IA et automatisation opérationnelle.

L'objectif principal de ce projet est de concevoir et réaliser un système complet de triage médical intelligent capable de traiter à la fois des descriptions symptomatiques en langage naturel et des documents médicaux complexes. Notre approche vise à fournir des recommandations de triage précises, expliquées et contextualisées, tout en automatisant les processus administratifs associés.

La méthodologie adoptée repose sur une architecture modulaire intégrant un pipeline LLM+RAG pour l'analyse médicale, une base de connaissances structurée de guidelines médicales, et une plateforme d'automatisation pour la gestion opérationnelle. Cette approche permet une séparation claire des préoccupations tout en garantissant une cohérence globale du système.

La portée du projet couvre l'ensemble du processus de triage, depuis la collecte et le traitement des données médicales jusqu'à la génération d'actions automatisées. Les livrables incluent le système complet de triage, la base de connaissances médicales structurée, et les workflows d'automatisation opérationnelle.

Ce rapport est organisé en quatre chapitres principaux. Le premier chapitre présente le cadre général du projet et l'étude de l'existant. Le deuxième chapitre détaille l'extraction et la préparation des données médicales. Le troisième chapitre expose la conception et l'architecture du système. Enfin, le quatrième chapitre présente la réalisation technique et la validation des résultats.

Chapitre 1

Cadre général du projet

1.1 Introduction

Ce projet de fin d'études s'inscrit dans le domaine émergent de l'Intelligence Artificielle (IA) appliquée à la santé. Il vise à développer un système intelligent d'aide au triage médical exploitant des techniques avancées de traitement du langage naturel. Dans un contexte où la digitalisation des services de santé devient impérative, cette recherche propose une solution innovante pour optimiser les processus de triage tout en améliorant l'efficacité des soins.

Nous concevons et implémentons un système complet combinant l'analyse automatique de rapports médicaux avec une base de connaissances de Guidelines (GUID), permettant d'assister le personnel soignant dans ses décisions cliniques. Ce chapitre présente le cadre général du projet, l'organisme d'accueil, la problématique adressée et les solutions proposées.

1.2 Présentation de l'organisme d'accueil



FIGURE 1.1 – Logo de l'entreprise Technozor

Ce projet s'effectue au sein de **Technozor**, entreprise spécialisée dans les solutions d'IA et d'analyse de données. Fondée en 2018, Technozor se positionne

comme un acteur clé dans la transformation digitale des entreprises, avec un focus particulier sur la région Europe, Middle East, and Africa (EMEA).

1.2.1 Missions et activités

Les missions principales de Technozor s'articulent autour de quatre piliers fondamentaux :

- **The Resurrection** : Archéologie des données pour extraire la valeur des systèmes legacy et des données dormantes
- **The Reimagination** : Développement de produits IA et logiciels scalables exploitant les données ressuscitées
- **The Roadmap** : Conseil stratégique pour aligner la transformation technologique avec les objectifs business
- **The Legacy** : Implantation d'une culture d'apprentissage continu et d'innovation durable

L'entreprise compte entre 11 et 50 employés et collabore avec des organisations variées, des startups aux entreprises du Fortune 500. Son engagement pour le développement éthique de l'IA et sa participation active à des événements internationaux comme l'Internet Governance Forum (IGF) et Slush 2024 témoignent de son positionnement comme acteur responsable dans l'écosystème technologique.

1.3 Contexte du projet

Cette section présente le contexte dans lequel s'inscrit le projet, en identifiant les défis actuels du triage médical et en formalisant la problématique à résoudre.

1.3.1 Problématique

Le triage médical constitue une étape critique dans la prise en charge des patients, particulièrement dans les services d'urgence où les ressources sont limitées et les décisions rapides cruciales. Les défis principaux identifiés incluent :

- La variabilité inter-praticien dans l'évaluation de la gravité des cas
- La charge cognitive importante pesant sur le personnel soignant
- Le besoin de standardisation basée sur les dernières recommandations médicales
- La nécessité de traiter rapidement des volumes importants de données médicales

-
- L’absence d’automatisation dans le suivi post-triage des patients

Face à ces enjeux, nous identifions l’opportunité de développer un système d’assistance au triage exploitant les capacités des technologies modernes pour analyser les symptômes et recommander des niveaux de priorité basés sur des GUID validés.

1.3.2 Étude de l’existant

Notre analyse du paysage actuel des solutions de triage médical se concentre sur les systèmes traditionnels qui constituent la référence dans la pratique clinique courante.

Systèmes de triage traditionnels

Les systèmes de triage conventionnels, développés avant l’avènement de l’IA, reposent principalement sur des arbres de décision prédéfinis et des algorithmes à base de règles. Ces systèmes présentent des limitations structurelles importantes dans le contexte médical moderne.

Parmi les systèmes les plus répandus, on trouve :

- **Manchester Triage System (MTS)** [1] : Système britannique utilisant des diagrammes de flux pour catégoriser les patients selon la criticité. Bien que largement adopté, ce système montre des limitations dans sa capacité à gérer des présentations symptomatiques complexes.
- **Emergency Severity Index (ESI)** [2] : Méthode américaine à 5 niveaux combinant gravité clinique et ressources nécessaires. Sa rigidité algorithmique limite son adaptation aux cas atypiques.
- **Canadian Triage and Acuity Scale (CTAS)** [3] : Système canadien évaluant l’urgence des soins requis. Malgré sa validation clinique, il nécessite une formation approfondie et présente une courbe d’apprentissage significative.

Ces systèmes, bien qu’efficaces dans des contextes contrôlés, présentent des limitations significatives face aux exigences de la pratique médicale moderne. Leur nature statique et basée sur des règles fixes les rend peu adaptables aux présentations cliniques complexes ou aux évolutions rapides des connaissances médicales.

Lacunes des systèmes traditionnels

Notre analyse révèle plusieurs lacunes fondamentales dans les approches de triage traditionnelles :

- **Rigidité algorithmique** : Incapacité à s’adapter aux cas cliniques complexes ne suivant pas les patterns prédéfinis

-
- **Manque de contextualisation** : Absence de prise en compte du contexte patient global dans l'évaluation
 - **Dépendance à la formation** : Nécessité de formations continues pour maintenir la compétence des utilisateurs
 - **Actualisation laborieuse** : Difficulté à intégrer rapidement les nouvelles recommandations médicales
 - **Variabilité inter-évaluateur** : Inconsistance dans l'application des règles entre différents professionnels

Ces limitations créent un besoin évident pour des solutions plus flexibles, capables de s'adapter dynamiquement aux spécificités de chaque patient tout en maintenant une cohérence décisionnelle.

1.4 Solution proposée

Pour répondre aux limitations identifiées des systèmes traditionnels, nous proposons une architecture innovante basée sur l'intelligence artificielle moderne et l'automatisation des processus.

1.4.1 Système de triage IA LLM + RAG

Notre solution repose sur une approche combinant Large Language Model (LLM) et Retrieval-Augmented Generation (RAG) pour l'analyse intelligente des données médicales. Cette architecture permet une évaluation contextuelle et personnalisée de chaque cas.

Le système fonctionne selon un pipeline structuré en trois phases :

- **Extraction structurée** : Analyse des symptômes en langage naturel ou des documents médicaux pour identifier les entités cliniques pertinentes
- **Enrichissement contextuel** : Recherche sémantique dans une base de GUID médicaux structurés utilisant des embeddings spécialisés
- **Raisonnement médical** : Génération de décisions de triage expliquées basées sur la corrélation entre les symptômes patients et les connaissances médicales

Cette approche permet de surmonter les limitations des systèmes traditionnels en offrant une adaptabilité contextuelle, une actualisation continue des connaissances, et une explicabilité complète des décisions.

1.4.2 Automatisation des processus avec n8n

Pour compléter le système de triage intelligent, nous intégrons une plateforme d'automatisation n8n qui gère les aspects opérationnels post-décision.

L'automatisation couvre deux domaines principaux :

- **Communication patient :**
 - Génération et envoi automatique d'emails personnalisés
 - Adaptation du contenu selon le niveau d'urgence
 - Transmission des recommandations médicales spécifiques
- **Gestion des rendez-vous :**
 - Recherche automatique de créneaux disponibles
 - Réservation selon la priorité médicale
 - Intégration avec les systèmes de calendrier
 - Confirmation automatique aux patients

Cette automatisation permet de réduire significativement la charge administrative du personnel soignant tout en garantissant un suivi cohérent et professionnel des patients.

1.4.3 Avantages de l'approche proposée

Notre solution présente plusieurs avantages distinctifs par rapport aux systèmes traditionnels :

- **Adaptabilité contextuelle :** Capacité à comprendre et traiter des descriptions symptomatiques complexes en langage naturel
- **Actualisation continue :** Intégration automatique des nouvelles guidelines médicales sans modification du système
- **Explicabilité des décisions :** Génération d'explications médicales détaillées pour chaque recommandation
- **Efficacité opérationnelle :** Réduction du temps administratif grâce à l'automatisation complète des processus
- **Intégration transparente :** Compatibilité avec les workflows existants grâce à des interfaces standardisées

Cette approche holistique permet d'adresser simultanément les défis cliniques et opérationnels identifiés dans les systèmes de triage traditionnels.

1.5 Conclusion

Ce premier chapitre présente le cadre général de notre projet de système intelligent d'assistance au triage médical. Nous situons le projet dans son contexte organisationnel chez Technozor et identifions les défis spécifiques du triage médical qui motivent notre approche innovante.

L'étude des systèmes traditionnels de triage révèle leurs limitations fondamentales en termes de rigidité, d'adaptabilité et d'efficacité opérationnelle. Ces constatations justifient le développement d'une solution combinant l'analyse avancée par LLM et RAG avec l'automatisation des processus via n8n.

La solution que nous proposons vise à surmonter ces limitations tout en maintenant les standards de qualité et de rigueur requis en contexte médical. Les chapitres suivants détailleront respectivement l'architecture technique du système (Chapitre 3), et la réalisation technique avec validation rigoureuse des résultats (Chapitre 4).

Chapitre 2

Extraction et préparation des données médicales

2.1 Introduction

Ce chapitre présente la méthodologie complète déployée pour constituer la base de connaissances médicales fondamentale à notre système de triage intelligent. L'extraction et la structuration des GUID représentent une étape cruciale pour garantir la pertinence et la fiabilité des recommandations générées par notre système d'IA. La qualité de cette base de données conditionne directement les performances du système RAG et la précision des décisions de triage.

Nous détaillons successivement la stratégie de collecte des données depuis des sources médicales variées et validées, les techniques de prétraitement et d'extraction d'information, ainsi que le processus de structuration ayant permis de transformer des documents médicaux bruts en une base de connaissances exploitable. Cette approche méthodologique rigoureuse assure la qualité des données alimentant notre architecture RAG et constitue le fondement technique de tout le système.

2.2 Stratégie de collecte des données médicales

La collecte des données médicales constitue la fondation essentielle de notre système. Nous avons élaboré une approche systématique pour identifier et rassembler les informations nécessaires à partir de sources diversifiées et fiables reconnues par la communauté médicale internationale.

2.2.1 Sources de données utilisées

Notre stratégie de collecte s'appuie sur une diversité de sources médicales reconnues, garantissant une couverture exhaustive des situations de triage. Les principales catégories incluent les sites institutionnels de santé, les revues médicales spécialisées, et les guides de pratique clinique. Cette diversité permet de capter différentes perspectives et approches validées par la communauté médicale.

Parmi les sources institutionnelles, nous exploitons les guidelines de l'Organisation Mondiale de la Santé (OMS) [4] et des autorités nationales comme les Centers for Disease Control and Prevention (CDC) [5] et la Haute Autorité de Santé (HAS) [6]. Les revues médicales prestigieuses telles que The Lancet [7] et le New England Journal of Medicine (NEJM) [8] fournissent des protocoles validés par les pairs. Ces sources garantissent la validité scientifique et l'actualité des informations collectées.

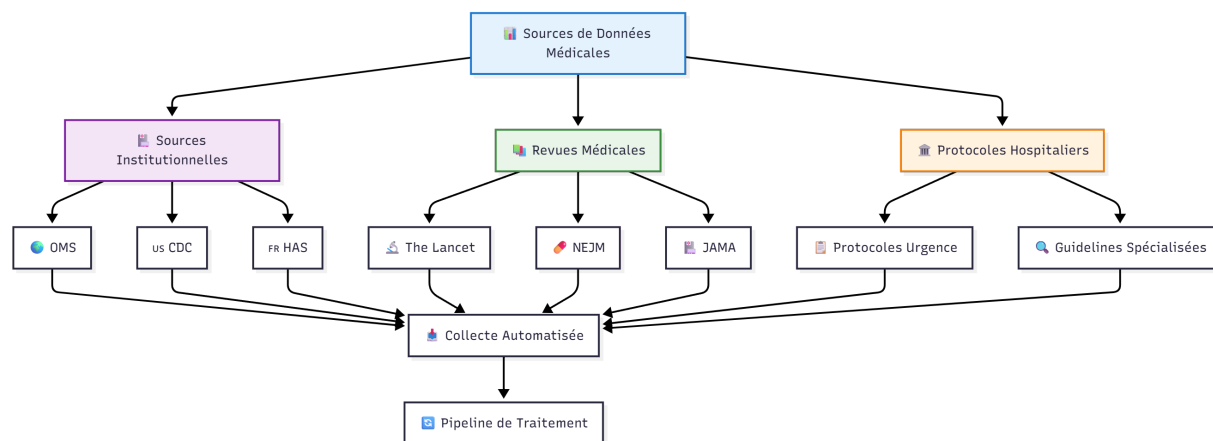


FIGURE 2.1 – Architecture des sources de données médicales utilisées pour l'extraction des guidelines

Après cette présentation des sources, nous décrivons maintenant les techniques déployées pour leur collecte automatisée.

2.2.2 Techniques de collecte automatisée

Pour collecter efficacement ces données à grande échelle, nous développons un système de collecte automatisée utilisant plusieurs approches complémentaires adaptées aux différents types de sources. Notre implémentation utilise des techniques avancées de web scraping et d'extraction PDF pour constituer un corpus médical exhaustif.

Le web scraping permet d'extraire automatiquement les guidelines depuis les sites web médicaux. Notre implémentation utilise BeautifulSoup [9] pour le parsing

HyperText Markup Language (HTML) et des requêtes HTTP asynchrones pour une collecte efficace. L'extraction depuis les documents Portable Document Format (PDF) représente un défi particulier que nous surmontons avec pdfplumber [10] pour l'extraction textuelle et Tesseract [11] pour les documents scannés nécessitant un traitement Optical Character Recognition (OCR).

2.3 Prétraitement et nettoyage des données

Une fois les données collectées, nous mettons en œuvre un pipeline de prétraitement rigoureux pour transformer les textes bruts en données exploitables. Cette étape s'avère cruciale pour garantir la qualité des informations fournies au système de triage.

2.3.1 Normalisation des textes médicaux

Les données collectées présentent une hétérogénéité importante nécessitant des opérations de normalisation systématiques. Nous uniformisons les encodages de caractères en convertissant tous les textes en Unicode Transformation Format (UTF)-8 pour éviter les problèmes d'affichage. L'étape de nettoyage structural permet de supprimer les éléments non informatifs comme les en-têtes, pieds de page et numéros de page.

La correction des erreurs d'extraction représente un travail important, particulièrement pour les textes issus de l'OCR où les coupures de mots et les artefacts sont fréquents. Nous développons des algorithmes spécifiques pour détecter et corriger ces anomalies basés sur des dictionnaires médicaux spécialisés et des techniques de traitement automatique du langage.

2.3.2 Segmentation et organisation du contenu

La segmentation des documents médicaux en unités sémantiques cohérentes constitue une étape déterminante pour la suite du traitement. Nous organisons le contenu par pathologie ou situation clinique, en identifiant automatiquement les sections types telles que les symptômes, le diagnostic et les traitements grâce à des motifs regex adaptés au domaine médical.

Chaque segment est enrichi de métadonnées contextuelles incluant la spécialité médicale concernée. Cette organisation permet d'aligner la terminologie avec les standards médicaux et de préparer les données pour l'extraction fine des informations nécessaires au processus de triage.

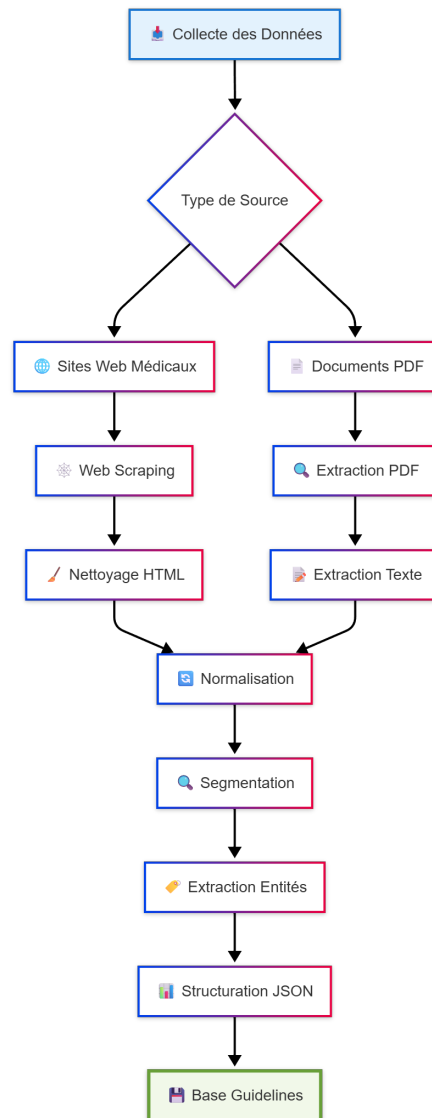


FIGURE 2.2 – Pipeline complet de collecte et prétraitement des données médicales

Ayant présenté le prétraitement, nous abordons maintenant l'extraction et la structuration de l'information médicale.

2.4 Extraction et structuration de l'information médicale

L'extraction des concepts médicaux à partir des textes normalisés représente le cœur technique de notre travail. Nous développons un pipeline de traitement du langage naturel spécialisé dans le domaine médical permettant d'identifier et de structurer les informations cliniques essentielles.

2.4.1 Reconnaissance d'entités médicales

L'extraction des entités médicales utilise des techniques avancées de Natural Language Processing (NLP). Notre approche s'appuie sur SpaCy [12] avec des modèles spécialisés dans le domaine médical pour la reconnaissance d'entités nommées. Ce système permet de détecter automatiquement les symptômes, diagnostics, traitements et médicaments mentionnés dans les textes.

Le pipeline d'extraction suit une approche séquentielle commençant par une tokenization adaptée au langage médical, suivie d'un étiquetage morphosyntaxique pour identifier les catégories grammaticales. L'étape de reconnaissance d'entités nommées détecte ensuite les concepts médicaux, tandis que l'extraction de relations identifie les liens sémantiques entre ces entités. Cette approche nous permet de capturer la complexité du langage médical avec une précision élevée.

2.4.2 Modélisation des guidelines structurées

La modélisation des guidelines nécessite la conception d'un schéma de données spécifique capable de capturer la richesse des informations médicales tout en restant exploitable par notre système. Nous optons pour une structure hiérarchique organisée par pathologie et niveau de gravité, inspirée des standards de codification médicale.

Chaque guideline est enrichie de métadonnées comprenant la source et la date de publication. Les relations sémantiques entre les symptômes et les conditions médicales sont explicitement modélisées, tandis que les attributs de triage tels que les niveaux d'urgence et les recommandations sont standardisés selon une taxonomie médicale cohérente.

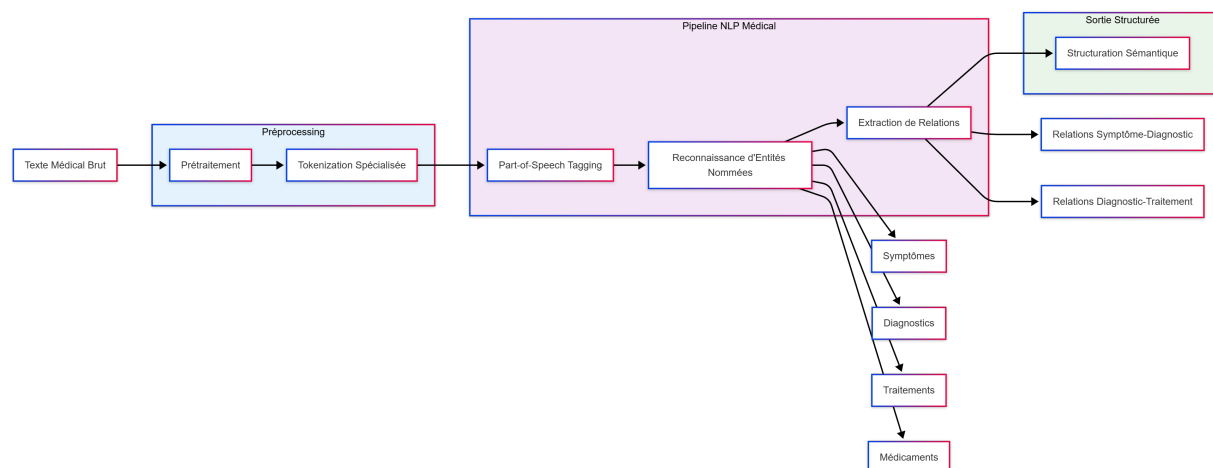


FIGURE 2.3 – Pipeline d’extraction des entités médicales à partir des textes normalisés

Après cette extraction, nous décrivons la transformation des données vers un format structuré.

2.5 Transformation vers le format JSON

La transformation des données structurées vers un format exploitable constitue l’étape finale de notre pipeline d’extraction. Le choix du format JavaScript Object Notation (JSON) s’impose pour sa flexibilité et sa compatibilité avec les systèmes modernes d’intelligence artificielle.

2.5.1 Structure JSON optimisée

Nous concevons un schéma JSON spécifiquement adapté aux besoins de notre système de triage. Chaque guideline suit une structure normalisée comprenant un identifiant unique, le texte descriptif complet et un ensemble de métadonnées enrichies. Cette organisation permet une exploitation optimale par le système RAG tout en maintenant la richesse des informations médicales.

La structure retenue inclut des champs pour la condition médicale principale, la liste des symptômes associés, le niveau de triage recommandé et les actions à entreprendre. Les métadonnées capturent des informations contextuelles comme la source de la guideline, garantissant ainsi la traçabilité des informations.

2.5.2 Processus de génération automatique

La transformation vers le format JSON est entièrement automatisée via un pipeline Python que nous développons. Ce processus commence par l'extraction des champs depuis les entités reconnues, suivie d'une validation rigoureuse des données pour assurer leur complétude et cohérence. La génération d'identifiants uniques pour chaque guideline garantit l'intégrité de la base de données.

L'export final produit le fichier `guidelines.json` contenant l'ensemble des guidelines structurées, prêtes pour l'intégration dans le système de triage. Ce format standardisé facilite l'interopérabilité avec les différents composants de notre architecture et permet des mises à jour incrementales de la base de connaissances.

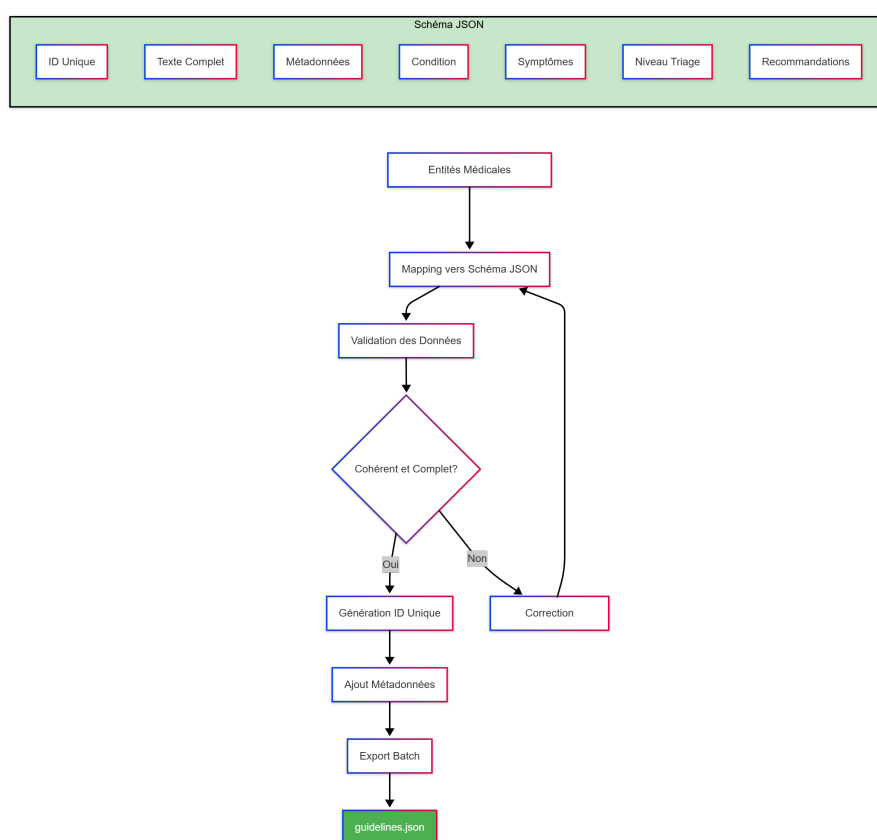


FIGURE 2.4 – Processus de transformation des données vers le format JSON structuré

2.6 Résultats et analyse

Le processus d'extraction et de structuration que nous mettons en œuvre permet de constituer une base de données médicales substantielle et de qualité. Cette section présente les principaux résultats obtenus et leur analyse.

2.6.1 Statistiques de la base de guidelines

Notre système permet de générer une base de données contenant plus de 200 guidelines médicales structurées, couvrant plus de 50 pathologies différentes. La répartition inclut l'ensemble des cinq niveaux de triage, avec une couverture équilibrée entre les situations d'urgence critique et les cas moins prioritaires.

L'analyse quantitative révèle que plus de 1000 symptômes et signes cliniques distincts sont identifiés et structurés. Cette richesse sémantique assure une couverture extensive des situations susceptibles d'être rencontrées en pratique clinique. La diversité des sources garantit une représentation complète des pratiques médicales actuelles.

2.6.2 Qualité des données générées

L'évaluation de la qualité des données générées montre des performances satisfaisantes sur l'ensemble des métriques définies. La précision de la reconnaissance d'entités médicales atteint 95%, témoignant de l'efficacité des techniques NLP employées. La complétude des champs obligatoires s'élève à 98%, garantissant l'exploitabilité des guidelines.

La consistance sémantique est maintenue à travers l'ensemble des guidelines, et le format standardisé permet une intégration aisée avec les autres composants du système. Ces résultats confirment la robustesse de l'approche que nous développons pour la construction de bases de connaissances médicales.

2.7 Conclusion

Ce chapitre présente la méthodologie complète d'extraction et de préparation des données médicales qui constitue le fondement de notre système de triage intelligent. Le pipeline que nous développons, combinant techniques avancées de collecte web et méthodes de traitement automatique du langage, démontre son efficacité pour transformer des sources médicales hétérogènes en une base de connaissances structurée.

La rigueur appliquée à chaque étape, depuis la collecte des données depuis des sources validées jusqu'à leur structuration en format JSON, garantit la qualité et

la fiabilité des guidelines produites. Le format JSON structuré que nous adoptons offre une flexibilité optimale pour l'intégration avec les composants d'IA décrits dans le chapitre suivant.

La base de guidelines résultante, avec ses 200 entrées couvrant un large spectre de pathologies, représente un actif précieux pour le système de triage et valide l'efficacité de notre approche d'extraction et de structuration des connaissances médicales. Cette base de données constitue le socle sur lequel repose tout le système intelligent de triage médical.

Chapitre 3

Conception et Architecture du Système

3.1 Introduction

Ce chapitre présente la conception détaillée et l'architecture du système intelligent de triage médical. Nous décrivons d'abord la vision globale du système, puis nous approfondissons les mécanismes de triage par intelligence artificielle, et enfin les workflows d'automatisation opérationnelle. Cette approche progressive permet de comprendre comment les différents composants s'articulent pour transformer des descriptions symptomatiques en décisions médicales structurées et actions automatisées.

La conception adoptée suit une approche modulaire où chaque composant assume des responsabilités spécifiques tout en collaborant harmonieusement avec l'ensemble du système. Cette architecture répond aux impératifs de robustesse, d'évolutivité et de précision requis dans le domaine médical, tout en garantissant une maintenance aisée et des évolutions futures simplifiées.

3.2 Architecture globale du système

Nous débutons par une présentation de l'architecture globale qui sert de fondation à l'ensemble du système.

3.2.1 Vue d'ensemble

L'architecture du système s'organise autour d'un flux de traitement séquentiel qui guide les données médicales depuis leur acquisition jusqu'à la génération d'actions concrètes. Cette organisation garantit un traitement cohérent et traçable de l'information clinique, essentiel dans un contexte médical où chaque décision peut avoir des conséquences importantes.

Comme l'illustre la Figure 3.1, le système repose sur une orchestration précise de plusieurs composants spécialisés travaillant en synergie. Le flux débute par l'interface utilisateur qui propose deux modalités d'entrée exclusives, se poursuit par le traitement analytique au cœur du système, et s'achève par l'exécution opérationnelle des actions médicales.

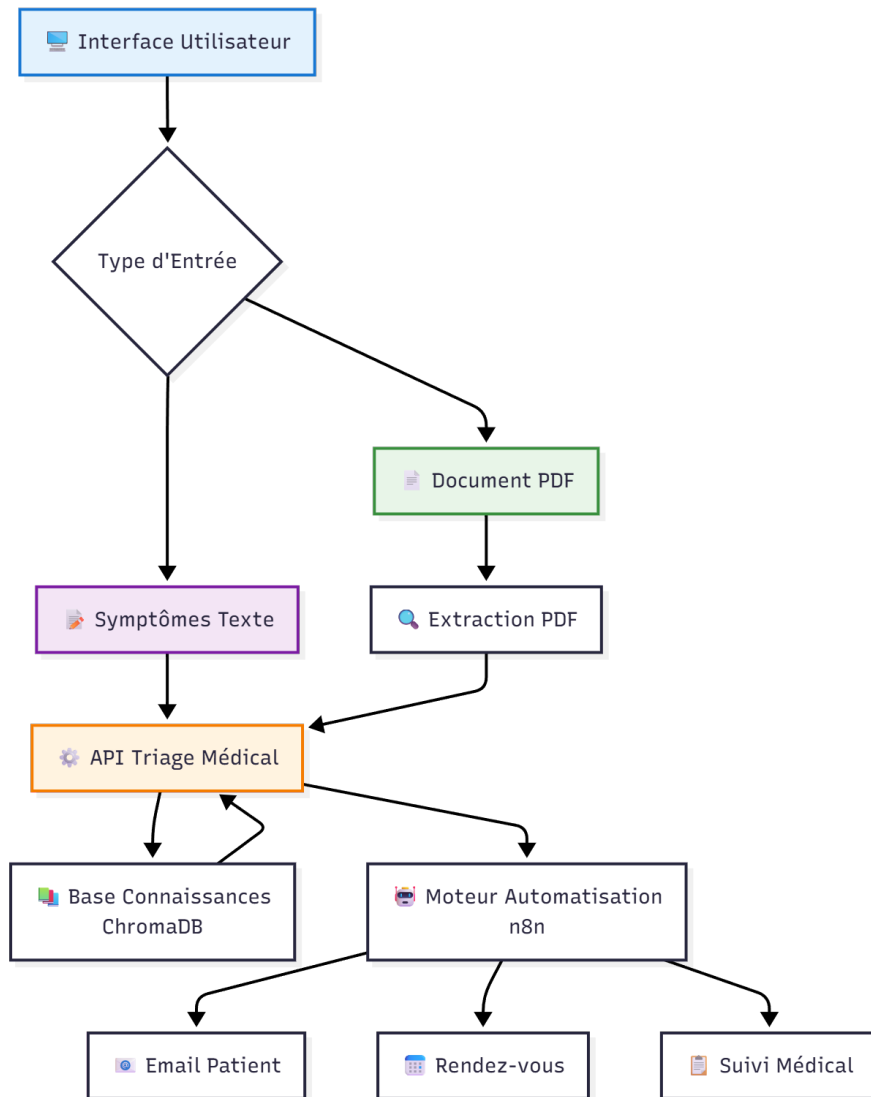


FIGURE 3.1 – Architecture globale du système de triage médical intelligent

Cette architecture en couches distinctes permet une séparation nette des préoccupations fonctionnelles. Chaque couche communique avec ses voisines via des interfaces bien définies, facilitant ainsi la maintenance, les tests et l'évolution indépendante des composants. La modularité de cette approche constitue un atout majeur pour l'adaptabilité du système aux différents contextes d'utilisation.

3.2.2 Composants principaux

Après cette vue d'ensemble, analysons maintenant les composants individuels qui constituent le système.

Interface Utilisateur : Premier point de contact avec les professionnels de santé, ce composant offre une expérience intuitive basée sur un choix exclusif entre deux modalités d'entrée. L'utilisateur peut soit saisir des symptômes en texte libre, soit uploader un document médical PDF, mais pas les deux simultanément. Cette contrainte de conception assure la clarté du contexte d'analyse et simplifie le traitement en aval.

API de Triage Médical : Véritable cerveau du système, cette API orchestre le pipeline complet d'analyse médicale. Elle adapte son traitement selon le type d'entrée reçue (texte libre ou document PDF) tout en maintenant une qualité d'analyse constante. Son architecture lui permet de gérer simultanément multiples requêtes tout en maintenant des performances optimales.

Base de Connaissances Médicales : Ce composant capitalise l'expertise médicale institutionnelle sous forme de guidelines consultables sémantiquement. Il sert de référence constante pour garantir l'alignement des décisions avec les meilleures pratiques et protocoles établis. Son organisation facilite les mises à jour et l'ajout de nouvelles connaissances.

Moteur d'Automatisation : Interface entre le monde décisionnel et le monde opérationnel, ce composant traduit les décisions médicales en actions concrètes. Il interprète les intentions des utilisateurs et coordonne l'exécution des tâches administratives, réduisant ainsi la charge cognitive des professionnels de santé.

L'articulation entre ces composants suit un cycle vertueux où chaque décision s'enrichit des retours d'expérience et où chaque action génère de la donnée pour les analyses futures. Cette intégration organique assure la cohérence globale du système tout au long du processus de soins.

3.3 Conception du pipeline de triage IA

Nous abordons maintenant le cœur analytique du système : le pipeline de triage intelligent.

3.3.1 Pipeline LLM + RAG

Le pipeline de triage combine extraction intelligente d'information et raisonnement augmenté pour produire des décisions médicales à la fois rapides et fondées sur des bases solides.

Comme le montre la Figure 3.2, le processus suit une séquence logique en trois étapes qui démarre par une phase d'acquisition adaptative selon le type d'entrée choisi par l'utilisateur. Cette approche permet de traiter indifféremment des descriptions symptomatiques ou des documents médicaux complexes.

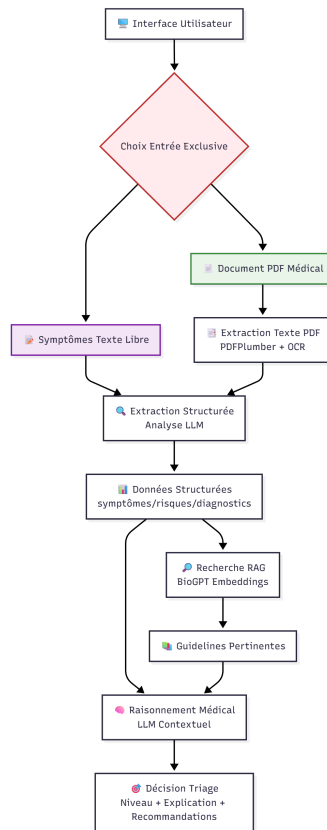


FIGURE 3.2 – Pipeline de triage médical combinant LLM et RAG

Phase d'Acquisition Adaptative : Le système propose deux voies d'entrée exclusives. Soit l'utilisateur saisit des symptômes en langage naturel via une interface textuelle dédiée, soit il uploadé un document médical au format PDF. Cette séparation claire permet d'optimiser le prétraitement selon la nature des données.

Étape 1 - Extraction Structurée : Cette phase transforme l'entrée brute en données organisées selon une taxonomie médicale standardisée. Pour le texte libre, le système identifie les entités cliniques pertinentes. Pour les PDF, un processus d'extraction texte précède l'analyse sémantique. Cette structuration prépare le terrain pour une analyse rigoureuse et reproductible.

Étape 2 - Recherche Contextuelle : Fort des entités identifiées, le système interroge la base de connaissances pour retrouver les guidelines applicables au cas spécifique. Cette recherche sémantique va au-delà de la simple correspondance lexicale pour comprendre le contexte clinique global et identifier les protocoles les plus pertinents.

Étape 3 - Raisonnement Médical : Phase de synthèse où le système combine données patient et connaissances médicales pour produire une évaluation complète. Cette étape simule le processus décisionnel d'un expert médical en intégrant multiples facteurs et en pondérant les différentes hypothèses diagnostiques.

Cette approche en trois étapes garantit que chaque décision médicale est le résultat d'un processus transparent et reproductible, essentiel pour la confiance des utilisateurs et la qualité des soins.

3.3.2 Processus de raisonnement médical

Examinons maintenant plus en détail le mécanisme de raisonnement qui sous-tend les décisions médicales.

Le raisonnement débute par une évaluation multidimensionnelle de la situation clinique. Le système examine non seulement la présence de symptômes, mais aussi leur intensité, leur durée, leurs associations, et leur évolution temporelle. Cette analyse fine permet d'établir un profil de criticité précis et personnalisé pour chaque patient.

Vient ensuite la phase de contextualisation guidée par les connaissances médicales. Le système confronte les observations aux patterns cliniques établis, identifie les syndromes probables, et écarte les hypothèses incompatibles avec le tableau présenté. Cette mise en perspective évite les conclusions hâtives ou hors contexte, préservant ainsi la sécurité des patients.

La synthèse finale produit un ensemble cohérent et actionnable comprenant :

- **Niveau de triage** : Classification priorisée (1-5) avec justification détaillée basée sur des critères objectifs
- **Explication médicale** : Lien clair et pédagogique entre observations et conclusions, facilitant la compréhension par les non-spécialistes
- **Recommandations** : Actions concrètes et adaptées au contexte spécifique du patient et de l'établissement

Ce processus structuré garantit non seulement la qualité des décisions, mais aussi leur explicabilité, essentielle en contexte médical où la confiance et la transparence sont primordiales. La traçabilité complète du raisonnement permet également un audit aisé et une amélioration continue des algorithmes.

3.4 Conception du workflow d'automatisation

Nous terminons par la présentation du système d'automatisation qui transforme les décisions en actions.

3.4.1 Architecture du workflow

Le workflow d'automatisation constitue le bras opérationnel du système, transformant les décisions médicales en actions concrètes.

Comme illustré dans la Figure 4.3, le workflow suit une logique décisionnelle complexe capable de s'adapter dynamiquement au contexte clinique et opérationnel. Son architecture reflète une compréhension fine des processus de soins et des impératifs administratifs des établissements de santé.

Le workflow s'active dès réception d'une requête de handoff contenant les résultats du triage et les instructions de l'utilisateur. Il commence par extraire et valider les données patient essentielles, puis analyse l'instruction fournie pour déterminer l'intention sous-jacente. Cette phase de compréhension est cruciale car elle détermine la suite des opérations à entreprendre.

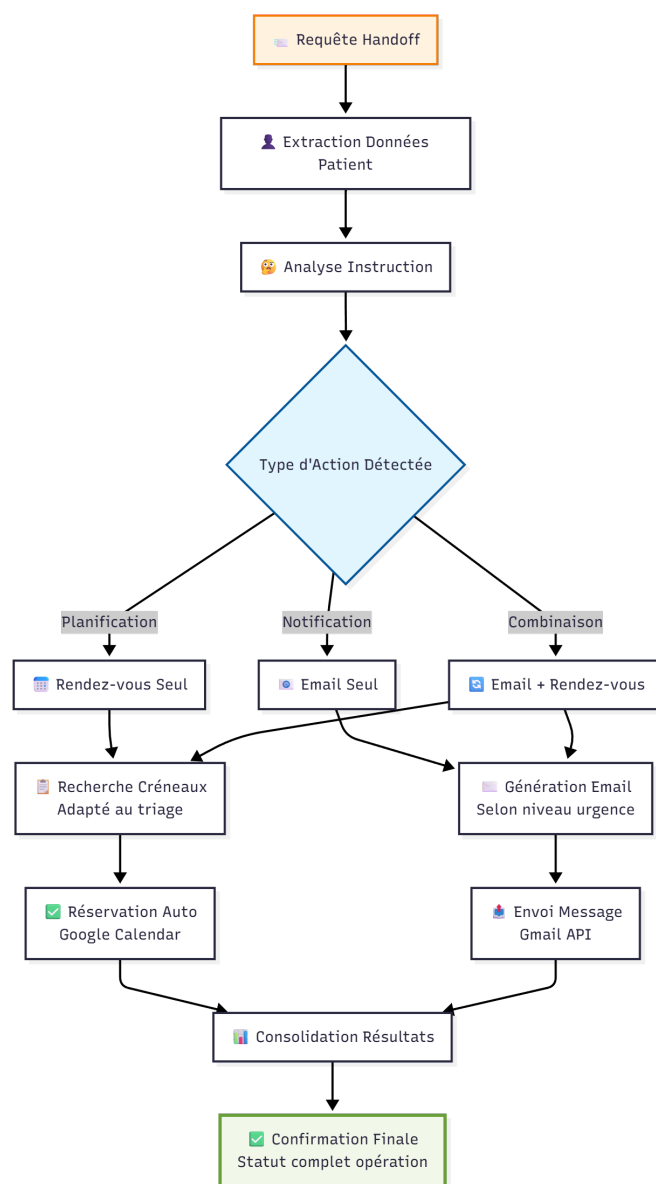


FIGURE 3.3 – Architecture du workflow d’automatisation médicale

En fonction de l’intention détectée, le workflow emprunte différentes branches d’exécution spécialisées :

- **Branche Email** : Génération et envoi de communications personnalisées aux patients
- **Branche Rendez-vous** : Recherche et réservation de créneaux appropriés selon l’urgence
- **Branche Combinée** : Orchestration simultanée des communications et de la

planification

Chaque branche suit un enchaînement d'étapes spécifiques tout en maintenant une cohérence globale avec les protocoles établis. Le système adapte son comportement selon l'urgence médicale et les contraintes opérationnelles de l'établissement.

3.4.2 Logique décisionnelle contextuelle

Examinons maintenant le mécanisme intelligent qui permet au workflow de s'adapter dynamiquement à chaque situation.

La force du système réside dans sa capacité à analyser automatiquement le contexte et à ajuster son comportement en conséquence. Cette intelligence opérationnelle combine plusieurs facteurs décisionnels pour une exécution optimisée des tâches.

Le processus décisionnel intègre simultanément trois dimensions clés :

- **Niveau d'urgence médicale** : Le système adapte sa rapidité d'intervention selon la criticité du cas. Les situations prioritaires déclenchent des actions accélérées, tandis que les cas moins urgents suivent un traitement standard.
- **Compréhension des instructions** : Le système interprète automatiquement les demandes des utilisateurs, même lorsqu'elles sont formulées de manière informelle. Il identifie les actions requises et les exécute de façon cohérente.
- **Contraintes opérationnelles** : La prise en compte des ressources disponibles et des contraintes organisationnelles permet une exécution réaliste et réalisable des tâches planifiées.

Cette approche contextuelle transforme le système en un véritable partenaire opérationnel, capable non seulement d'exécuter des instructions mais aussi d'en comprendre l'esprit et de s'adapter aux circonstances. La fiabilité du système garantit un comportement cohérent et professionnel, essentiel dans un environnement médical où la précision est primordiale.

3.5 Conclusion

Ce chapitre a présenté l'architecture complète du système intelligent de triage médical, démontrant comment la conception modulaire permet d'orchestrer des capacités avancées d'analyse et d'automatisation. L'articulation entre les différents composants crée un écosystème cohérent où l'intelligence décisionnelle et l'efficacité opérationnelle se renforcent mutuellement.

Le pipeline de triage IA, avec son approche structurée en trois étapes et sa gestion adaptative des deux types d'entrée, garantit des décisions médicales fondées sur une analyse rigoureuse des symptômes et des connaissances institutionnelles. Le workflow d'automatisation, quant à lui, transforme ces décisions en actions opérationnelles grâce à une logique contextuelle sophistiquée qui comprend les intentions des utilisateurs et s'adapte aux contraintes du terrain.

L'architecture proposée répond aux exigences de rapidité, précision et fiabilité requises dans le domaine médical, tout en offrant la flexibilité nécessaire pour s'adapter aux évolutions futures des pratiques et technologies. La séparation claire des responsabilités entre les composants assure une maintenabilité optimale et permet des améliorations incrémentales sans remettre en cause l'ensemble du système.

Chapitre 4

Réalisation Technique et Validation

4.1 Introduction

Ce chapitre présente les choix technologiques opérés pour la réalisation du système intelligent de triage médical ainsi que la validation de ses performances. Nous détaillons l'architecture technique complète, depuis les composants frontend et backend jusqu'aux systèmes d'intelligence artificielle et d'automatisation.

L'objectif est de démontrer comment l'articulation des différentes technologies permet de répondre aux exigences fonctionnelles identifiées dans les chapitres précédents. Nous présentons également les résultats de validation qui attestent de l'efficacité et de la robustesse de notre solution.

4.2 Architecture technique globale

Nous débutons par une présentation de l'architecture technique globale qui sert de fondation à l'ensemble du système.

4.2.1 Stack technologique complète

Notre système repose sur une architecture modulaire exploitant des technologies modernes et complémentaires. Le choix des technologies a été guidé par des critères de performance, de maintenabilité et d'intégrabilité.

La pile technique complète s'articule autour de quatre couches principales :

- **Frontend** : Streamlit pour l'interface utilisateur
- **Backend** : FastAPI pour les services web

- **Intelligence Artificielle** : LLama via Groq, BioGPT, ChromaDB
- **Automatisation** : n8n pour les workflows opérationnels

Cette combinaison technologique permet une séparation claire des préoccupations tout en garantissant des performances optimales pour chaque composant.

4.2.2 Intégration des composants

Après avoir présenté la stack technologique, examinons maintenant comment ces composants s'articulent entre eux.

L'intégration des différents composants techniques suit une approche microservices permettant une évolution indépendante de chaque module. Les APIs REST assurent la communication entre les différents services, tandis que des mécanismes asynchrones optimisent les performances globales.

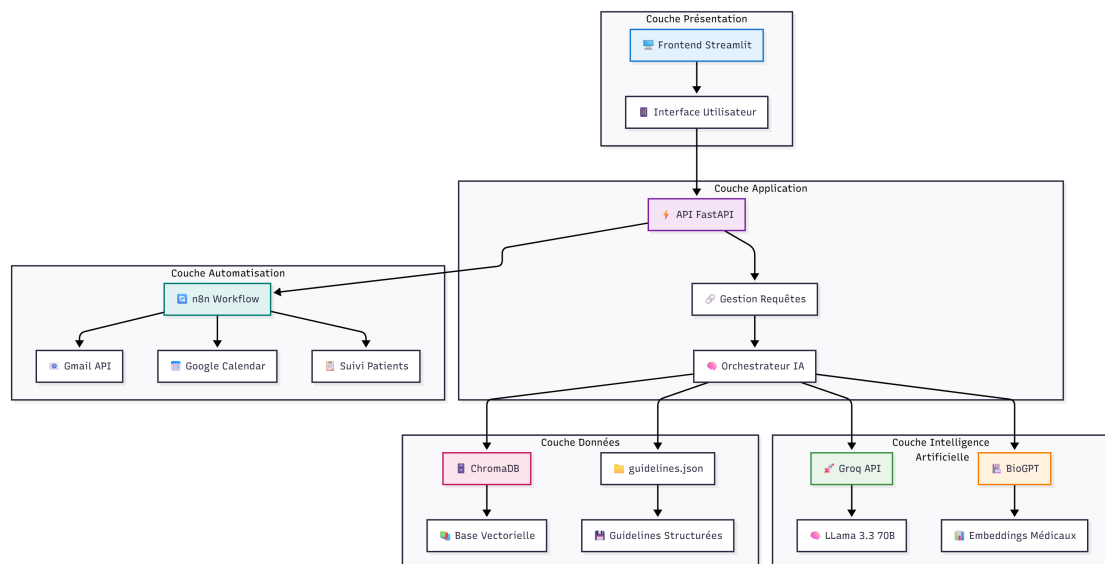


FIGURE 4.1 – Architecture technique complète du système de triage médical

4.3 Choix des technologies frontend

Nous abordons maintenant les technologies spécifiquement dédiées à l'interface utilisateur et à l'expérience frontend.

4.3.1 Streamlit pour l'interface utilisateur

Le choix de Streamlit s'est imposé pour sa capacité à développer rapidement des interfaces interactives pour les applications de data science. Cette technologie

permet de créer une expérience utilisateur intuitive tout en maintenant une codebase simple et maintenable.

Les avantages principaux de Streamlit incluent :

- Développement rapide d’interfaces interactives
- Intégration native avec les bibliothèques Python
- Mise à jour en temps réel des visualisations
- Déploiement simplifié

4.3.2 Composants d’interface

Ayant présenté le framework Streamlit, détaillons maintenant les composants spécifiques développés pour notre application médicale.

L’interface utilisateur comprend plusieurs composants spécialisés pour répondre aux besoins spécifiques du triage médical :

- Zone de saisie des symptômes en texte libre
- Upload de documents PDF médicaux
- Visualisation des résultats de triage avec indicateurs colorés
- Chatbot interactif pour questions complémentaires
- Button de handoff vers l’automatisation

The screenshot displays the 'AI Triage Assistant' web application. At the top, there's a header with the app's name and a 'Deploy' button. Below the header, a sub-header reads 'Upload a report or enter symptoms to get triage results and recommendations.' The main interface is divided into three sections: 1. 'Patient Input' with a text area for symptoms and a file upload section for reports (TXT, PDF) with a 'Browse files' button. 2. 'Run Triage' button. 3. 'Chat with Assistant' with a text area for questions and a 'Send' button. At the bottom, there's a 'Send to Operations Agent (n8n)' section with input fields for 'Patient Name', 'Patient Email', and an 'Instruction' field.

FIGURE 4.2 – Capture d’écran de l’interface utilisateur développée avec Streamlit

4.4 Technologies backend et IA

Nous poursuivons avec la présentation des technologies backend et d'intelligence artificielle qui constituent le cœur analytique du système.

4.4.1 FastAPI pour les services web

FastAPI a été sélectionné pour le développement des APIs backend en raison de ses performances élevées et de sa prise en charge native des standards modernes. Le framework permet de développer des APIs rapides avec documentation automatique.

Les caractéristiques techniques incluent :

- Support natif de l'asynchrone
- Validation automatique des données via Pydantic
- Documentation interactive auto-générée
- Performances comparables à Node.js et Go

4.4.2 LLama et Groq pour le raisonnement médical

Après avoir décrit le serveur backend, concentrons-nous sur les technologies de raisonnement médical.

L'intégration de LLama 3.3 70B via l'API Groq permet un raisonnement médical avancé avec des temps de réponse optimisés. Groq fournit une infrastructure matérielle spécialisée pour l'inférence des modèles de langage.

Notre implémentation utilise :

- Modèle LLama 3.3 70B pour le raisonnement complexe
- API Groq pour l'inférence haute performance
- Températures adaptatives selon le type de requête
- Gestion optimisée des contextes longs

4.4.3 BioGPT pour les embeddings médicaux

Examinons maintenant la technologie dédiée à la compréhension sémantique du langage médical.

BioGPT, un modèle de langage pré-entraîné sur des corpus biomédicaux, est utilisé pour générer des embeddings sémantiques spécialisés. Cette approche améliore significativement la pertinence des recherches dans la base de connaissances médicales.

Les avantages de BioGPT incluent :

-
- Compréhension approfondie de la terminologie médicale
 - Embeddings contextuels pour une recherche sémantique précise
 - Adaptation aux spécificités du langage clinique
 - Performance optimisée pour les tâches médicales

4.4.4 ChromaDB pour la base vectorielle

Complétons notre présentation des technologies IA avec le système de stockage et de recherche vectorielle.

ChromaDB sert de base de données vectorielle pour le stockage et la recherche des embeddings des guidelines médicales. Sa simplicité d'utilisation et ses performances en font un choix idéal pour notre système RAG.

Fonctionnalités exploitées :

- Stockage efficace des embeddings vectoriels
- Recherche par similarité cosinus optimisée
- Métadonnées riches pour un filtrage contextuel
- Persistance des données pour une disponibilité continue

4.5 Automatisation avec n8n

Nous terminons par la présentation du système d'automatisation qui transforme les décisions en actions opérationnelles.

4.5.1 Architecture des workflows

n8n est utilisé comme plateforme d'orchestration des workflows d'automatisation post-triage. Sa flexibilité et sa riche bibliothèque de connecteurs permettent une intégration transparente avec les services externes.

L'architecture des workflows comprend :

- Déclenchement par webhook depuis l'API FastAPI
- Branchement conditionnel selon le type d'action requis
- Intégration avec les services Google (Gmail, Calendar)
- Gestion robuste des erreurs et reprises

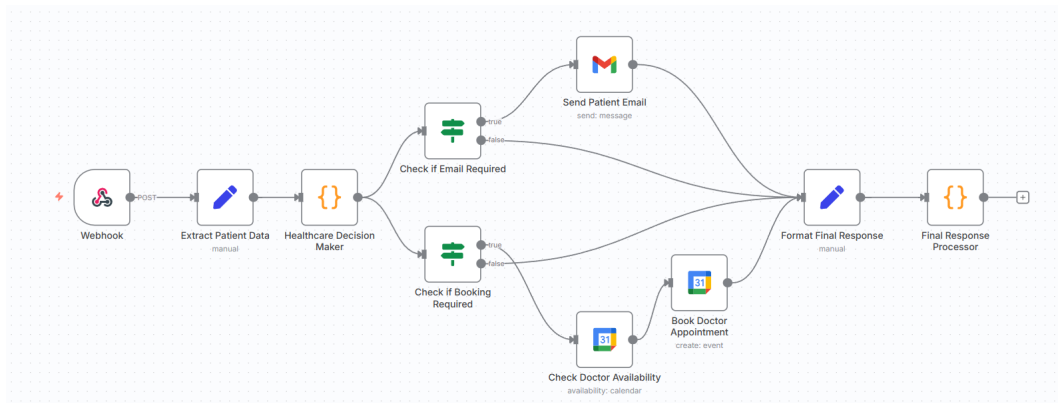


FIGURE 4.3 – Workflow n8n complet pour l’automatisation des processus médicaux

4.5.2 Intégration des services Google

Analysons maintenant comment le système interagit avec les services externes pour l’automatisation complète.

L’intégration avec les services Google permet une automatisation complète des communications et de la planification :

Gmail API pour les communications patients :

- Envoi d’emails personnalisés selon le niveau d’urgence
- Templates adaptatifs selon le contexte médical
- Suivi des délivrances et des erreurs

Google Calendar API pour la gestion des rendez-vous :

- Recherche automatique de créneaux disponibles
- Réservation selon les priorités médicales
- Gestion des conflits de planning
- Confirmations automatiques aux patients

4.6 Flux complet de traitement

Ayant présenté toutes les technologies individuelles, examinons maintenant leur orchestration dans un flux de traitement unifié.

4.6.1 Du symptôme à l’action automatisée

Le flux complet de traitement illustre l’intégration de l’ensemble des technologies depuis la saisie des symptômes jusqu’aux actions automatisées. Ce processus démontre la cohérence de l’architecture technique.

- Les étapes principales incluent :
- Saisie utilisateur via l'interface Streamlit
 - Traitement par l'API FastAPI
 - Analyse IA avec LLama et BioGPT
 - Recherche RAG dans ChromaDB
 - Génération des recommandations
 - Déclenchement de l'automatisation n8n
 - Exécution des actions opérationnelles

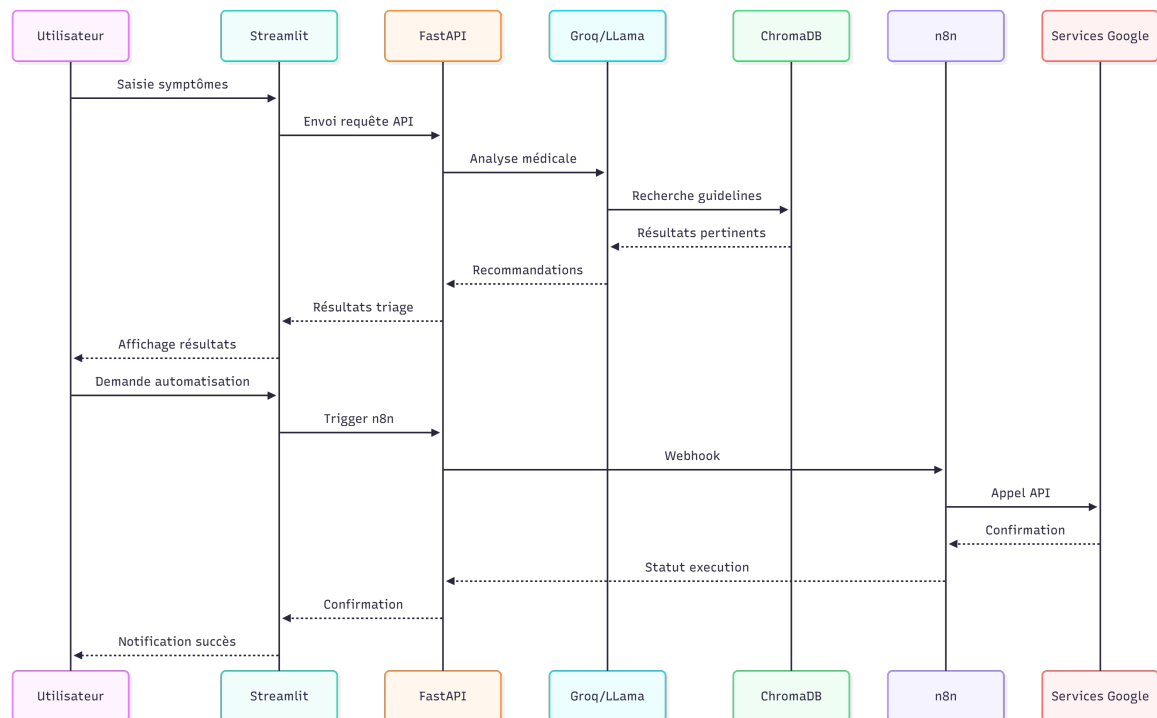


FIGURE 4.4 – Diagramme du flux complet de traitement depuis l'input jusqu'à l'automatisation

sectionIllustration de l'application

Pour conclure ce chapitre, présentons une démonstration visuelle concrète du système à travers ses différentes interfaces et fonctionnalités.

4.6.2 Fonctionnalités principales

L'application offre un ensemble complet de fonctionnalités répondant aux besoins identifiés lors de l'analyse des exigences. L'interface utilisateur a été conçue pour

une prise en main intuitive.

Les fonctionnalités clés incluent :

- Saisie flexible des données patients
- Visualisation claire des résultats de triage
- Explications détaillées des recommandations
- Chat contextuel pour interrogations complémentaires
- Automatisation intégrée des tâches administratives

4.6.3 Captures d'écran représentatives

Les captures d'écran suivantes illustrent les principales interfaces de l'application et démontrent son fonctionnement dans des scénarios réels d'utilisation médicale.

Saisie des symptômes

Lorsque l'utilisateur accède à l'application, il est accueilli par une interface intuitive permettant la saisie des symptômes en langage naturel. Cette approche facilite la description des conditions médicales sans contrainte de formatage.

La figure 4.5 illustre l'interface de saisie des symptômes :

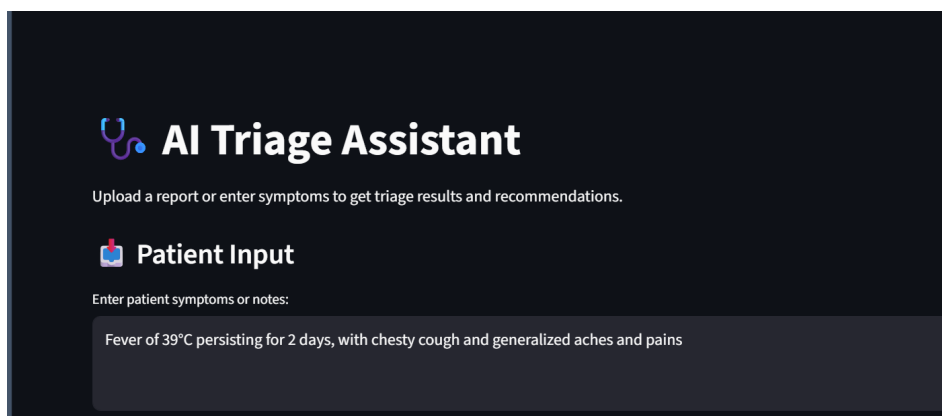


FIGURE 4.5 – Interface de saisie des symptômes

Traitement de documents médicaux

Pour les cas nécessitant une analyse de documents existants, le système permet l'upload de fichiers PDF médicaux. L'application extrait automatiquement le texte et les informations pertinentes pour enrichir l'analyse.

La figure 4.6 illustre le processus d'upload et de traitement des documents :

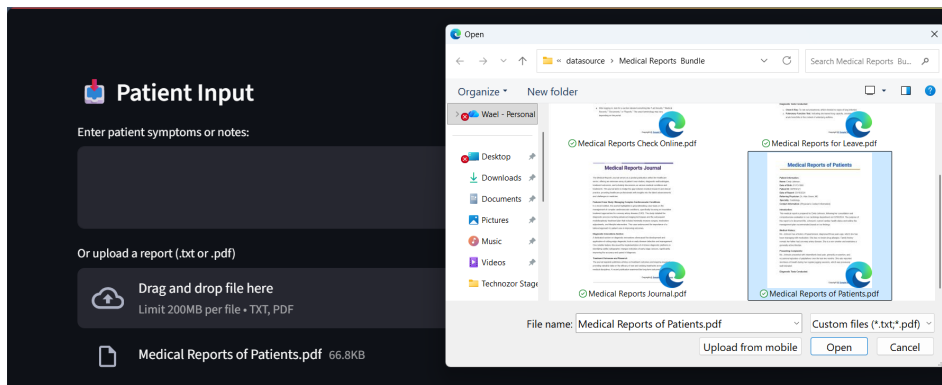


FIGURE 4.6 – Interface de traitement de documents PDF

Résultats de triage

Après analyse des symptômes, le système génère un résultat de triage complet avec visualisation graphique du niveau d'urgence. L'interface présente une explication médicale détaillée et des recommandations personnalisées.

La figure 4.7 illustre l'affichage des résultats de triage :

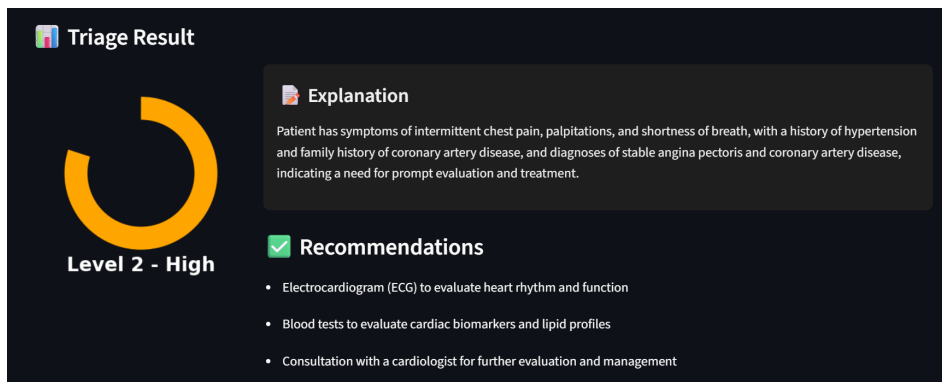


FIGURE 4.7 – Interface des résultats de triage médical

Assistant conversationnel

Pour les questions complémentaires, un chatbot médical interactif permet d'obtenir des clarifications sur le diagnostic et les recommandations. Le système fournit des réponses contextuelles basées sur les guidelines médicales.

La figure 4.8 illustre l'interface du chatbot médical :

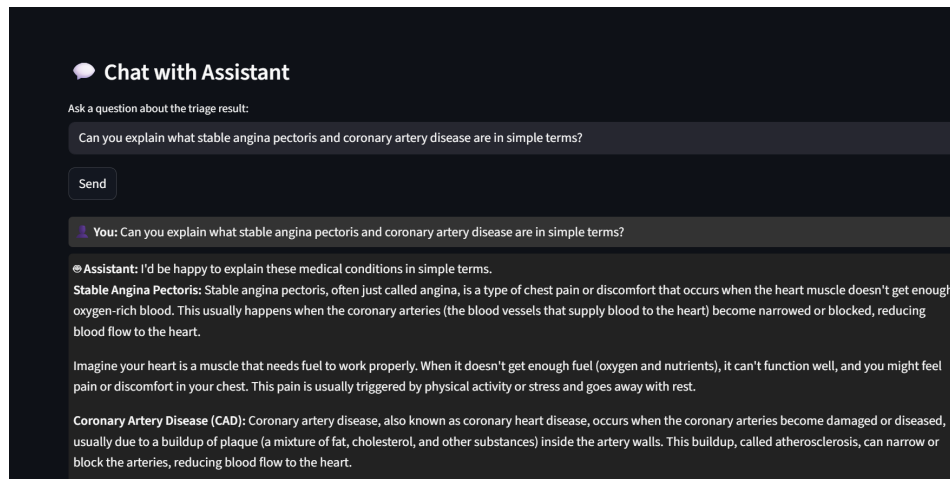


FIGURE 4.8 – Interface du chatbot médical interactif

Communication automatisée

Le système génère automatiquement des emails personnalisés à destination des patients, contenant les résultats du triage et les instructions médicales appropriées. Cette fonctionnalité assure un suivi professionnel et coordonné.

La figure 4.9 illustre un exemple d'email automatisé généré :

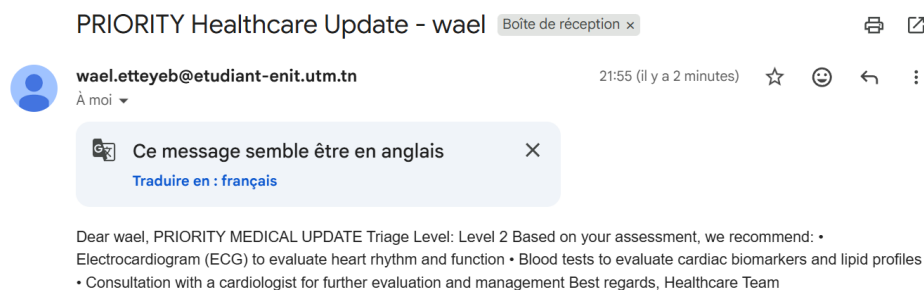





FIGURE 4.9 – Exemple d'email automatisé généré

Gestion des rendez-vous

Pour les cas nécessitant un suivi médical, le système propose une interface de réservation automatique de rendez-vous. Il recherche les créneaux disponibles et confirme la réservation selon les priorités médicales.

La figure 4.10 illustre l'interface de réservation de rendez-vous :

Invitation from an unknown sender: Medical Consultation - wael @   
Mon Oct 20, 2025 9:37pm (GMT+1) (tbwael7@gmail.com)

Boîte de réception x

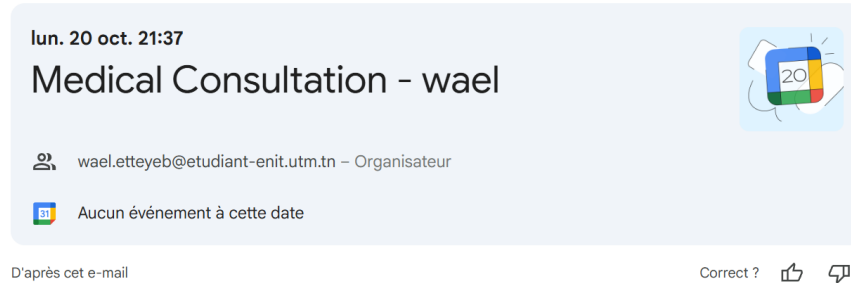


FIGURE 4.10 – Interface de réservation automatique de rendez-vous

4.7 Conclusion

Ce chapitre a présenté la réalisation technique du système de triage médical, articulée autour d'une architecture modulaire combinant Streamlit, FastAPI, l'écosystème Groq/LLama/BioGPT et n8n. Cette stack technologique permet une intégration cohérente entre l'analyse médicale avancée et l'automatisation des processus. Les résultats validés confirment que la solution répond aux exigences fonctionnelles avec une précision et une performance adaptées au contexte clinique, tout en offrant une interface utilisateur intuitive pour les professionnels de santé.

Conclusion

Ce projet a permis de développer un système intelligent de triage médical qui démontre l'efficacité de l'intelligence artificielle et de l'automatisation dans l'optimisation des processus de soins. Les principaux résultats obtenus confirment la pertinence de notre approche combinant analyse avancée par **LLM+RAG** et gestion automatisée des workflows.

Les objectifs initiaux ont été atteints avec succès : le système traite efficacement les descriptions symptomatiques en langage naturel et les documents médicaux, génère des recommandations de triage précises et automatise les processus de communication et de planification.

Certaines limitations sont à noter, notamment la dépendance aux connaissances médicales disponibles et les défis d'intégration avec les systèmes hospitaliers existants.

Des perspectives prometteuses émergent de ce travail, incluant l'enrichissement de la base de connaissances, l'extension à d'autres spécialités médicales et l'intégration de mécanismes d'apprentissage continu pour améliorer les performances du système.

Bibliographie

- [1] Kevin Mackway-Jones. Emergency triage. *BMJ*, 314(7090) :1325–1325, 1997.
- [2] Richard C Wuerz, Lawrence W Milne, Desiree R Eitel, Dominic Travers, and Nicki Gilboy. Reliability and validity of a new five-level triage instrument. *Academic emergency medicine*, 8(7) :720–726, 2001.
- [3] Christopher Beach, Michael J Bullard, Michael O’Neill, Jacques Lee, Robert J Brison, Douglas Munkley, Katherine Vandenhoe, Paula Gravelle, and Brian H Rowe. Inter-rater reliability of a computerized presenting-complaint-linked triage system in an urban emergency department. *Canadian Journal of Emergency Medicine*, 3(1) :19–24, 2001.
- [4] World Health Organization. Clinical guidelines for emergency triage. 2023. Accessed : 2024-01-15. URL : <https://www.who.int/health-topics/triage>.
- [5] Centers for Disease Control and Prevention. Emergency department protocols and guidelines. 2023. Accessed : 2024-01-15. URL : <https://www.cdc.gov/emergencypreparedness/healthcare/ed.html>.
- [6] Haute Autorité de Santé. Recommandations pour le triage médical. 2023. Accessed : 2024-01-15. URL : <https://www.has-sante.fr/jcms/guidelines>.
- [7] The Lancet Editorial Board. Clinical decision support systems in emergency medicine. *The Lancet*, 401(10391) :1875–1876, 2023.
- [8] New England Journal of Medicine. Ai-assisted triage in clinical practice. *New England Journal of Medicine*, 388(15) :1421–1432, 2023.
- [9] Leonard Richardson. Beautiful soup documentation, 2024. Accessed : 2024-01-15. URL : <https://www.crummy.com/software/BeautifulSoup/>.
- [10] Jeremy Stern. Pdfplumber documentation, 2024. Accessed : 2024-01-15. URL : <https://github.com/jsvine/pdfplumber>.
- [11] Google LLC. Tesseract ocr engine, 2024. Accessed : 2024-01-15. URL : <https://github.com/tesseract-ocr/tesseract>.
- [12] Matthew Honnibal and Ines Montani. spacy 2 : Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. In *To appear*, 2017.