REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Université Saad DAHLAB - Blida 1

Faculté des sciences

Département d'Informatique

Mémoire présenté par :

M. REZAL Sami et M. KEBOUCI Mohamed

Pour l'obtention du diplôme de Master

Domaine : Mathématique et Informatique

Filière : Informatique

Spécialité : Traitement Automatique de la Langue

Sujet :

Développement d'un pipeline TAL pour l'extraction et le résumé automatique des données issues de sources web Algériennes

Soutenu le : 29/06/2025, Devant le jury composé de:

Mme. MEZZI M.	Promotrice	Université de Blida 1
Mme. OUKID S.	Président	Université de Blida 1
Mr. FERFARA	Examinateur	Université de Blida 1

Remerciement

Nous remercions avant tout Allah tout puissant de nous avoir donné la force, la patience et la persévérance nécessaires pour mener à bien ce mémoire.

Nous exprimons notre profonde gratitude à notre promotrice, Melyara MEZZI, pour sa disponibilité, ses conseils pertinents, sa rigueur scientifique et son accompagnement tout au long de ce travail. Son soutien constant a grandement contribué à l'aboutissement de ce projet.

Nous remercions également l'ensemble des enseignants du département d'Informatique pour les connaissances qu'ils nous ont transmises et pour la qualité de leur enseignement tout au long de notre parcours universitaire.

Nos remerciements vont aussi à nos camarades de promotion et à tous ceux qui, de près ou de loin, ont contribué au bon déroulement de ce travail, notamment par leur aide, leurs suggestions ou leur soutien moral.

Enfin, nous adressons nos plus sincères remerciements à nos familles, pour leur compréhension, leur encouragement permanent et leur présence à chaque étape de notre vie académique.

Sami & Mohamed

Résumé

Ce mémoire présente la conception et la réalisation d'une plateforme numérique dédiée à la valorisation du patrimoine culturel algérien. L'objectif principal est de centraliser, structurer et présenter des informations culturelles fiables à travers un système automatisé basé sur le web scraping et le résumé automatique.

Pour surmonter le manque de bases de données accessibles et centralisées, des techniques de collecte automatique de données ont été mises en œuvre pour extraire du contenu depuis des sites web spécialisés. Ces données sont ensuite traitées, nettoyées et résumées automatiquement à l'aide d'algorithmes de Traitement Automatique du Langage Naturel (TALN), afin d'en faciliter l'accès et la compréhension pour le grand public. Le projet combine ainsi des aspects de l'ingénierie logicielle et de l'Intelligence Artificielle (IA), contribuant à la préservation culturelle et à la valorisation numérique du patrimoine algérien..

Mots-clés : *Patrimoine Culturel Algérien, Web Scrapping, Résumé Automatique, Développemen d'une Plateforme Web, Modèle de Langue.*

الملخص

يعرض هذا البحث تصميم وتطوير منصة رقمية مخصصة للتعريف بالتراث الثقافي الجزائري. الهدف الرئيسي هو جمع وتنسيق وتقديم معلومات ثقافية موثوقة عبر نظام آلي يعتمد على تقنيات استخراج البيانات من الويب (Web Scraping) والتلخيص التلقائي للنصوص.

لمواجهة غياب قواعد بيانات مركزية وسهلة الوصول، تم اعتماد تقنيات جمع بيانات آلية لاستخلاص المحتوى من مواقع متخصصة. تُعالَج هذه البيانات وتُنقّى ثم تُلخَّص باستخدام خوارزميات المعالجة الآلية للغة الطبيعية، مما يسهل الوصول إلى المحتوى وفهمه من قبل المستخدم العادي. يجمع هذا المشروع بين هندسة البرمجيات والذكاء الاصطناعي من أجل الإسهام في الحفاظ على التراث الجزائري وتعزيزه رقمياً.

الكلمات المفتاحية: التراث الثقافي الجزائري، استخراج البيانات من الويب، التلخيص الآلي، تطوير منصة ويب، النماذج الغوية.

Abstract

This thesis presents the design and development of a digital platform dedicated to the promotion of Algerian cultural heritage. The main objective is to centralize, structure, and present reliable cultural information through an automated system based on web scraping and automatic text summarization.

To address the lack of accessible and centralized heritage databases, automated data collection techniques were implemented to extract content from specialized websites. The collected data is then cleaned, processed, and summarized using Natural Language Processing (NLP) algorithms, making it easier for the general public to access and understand. The project combines Software Engineering and Artificial Intelligence contributing to the cultural preservation and digital valorization of the Algerian heritage.

Keywords: *Algerian Cultural Heritage, Web Scraping, Automatic Summarization, Web Platform Development, Language Models.*

Sommaire

Table des matières

Liste des figures

Liste des tabeaux

INTRODUCTION GÉNÉRALE

1. Contexte Général

Le traitement automatique de la langue (TAL) et l'intelligence artificielle (IA) offrent des solutions puissantes pour traiter le volume croissant d'informations disponibles sur Internet. Dans le contexte algérien, où les données relatives à l'histoire, la culture, la géographie et les personnalités sont dispersées et multilingues (arabe, français, anglais), il est essentiel de développer des outils pour structurer et synthétiser ces informations. Ce projet s'appuie sur des techniques avancées de TAL, telles que le web scraping et le résumé automatique, pour collecter et transformer des données en entrées encyclopédiques. Ces technologies permettront de traiter des données issues de sources ouvertes (réseaux sociaux, blogs, sites web) afin de valoriser et préserver le patrimoine algérien sous une forme accessible et exploitable.

2. Problématique et Objectifs

L'accès aux informations historiques, culturelles et géographiques relatives à l'Algérie demeure complexe en raison de la dispersion des contenus sur une multitude de plates-formes non uniformes, telles que les réseaux sociaux, les blogs, ou encore les sites spécialisés. Cette fragmentation est aggravée par la diversité linguistique du pays — arabe, anglais et français — et par la nature non structurée des données publiées. Ces contraintes limitent la valorisation numérique et la préservation du patrimoine algérien.

Dans ce contexte, une problématique centrale se dégage :
Comment exploiter les techniques de traitement automatique des langues (TAL) et d'intelligence artificielle (IA) pour centraliser, structurer et valoriser efficacement des contenus culturels multilingues issus de sources ouvertes, tout en respectant les spécificités linguistiques et culturelles propres à l'Algérie ?

Afin de répondre à cette problématique, plusieurs questions de recherche peuvent être formulées :

- Quelles stratégies permettent de collecter de manière efficace des données hétérogènes depuis des sources web non structurées ?

- Comment concevoir un système de résumé automatique capable de restituer fidèlement l'information, tout en préservant la richesse contextuelle et culturelle du contenu original ?

- **Q3** : Quels critères et métriques sont les plus adaptés pour évaluer la qualité des résumés générés, en particulier dans un contexte où peu de références ou de modèles informatiques existent ?

Sur la base de cette problématique et des questions de recherche associées, les objectifs suivants sont définis pour ce projet de fin d'études :

1. Mettre en place un pipeline de collecte automatique de données culturelles algériennes à partir de sources ouvertes (web scraping, APIs, etc.), tout en assurant leur classification et leur prétraitement.

2. Développer un système de résumé automatique capable de produire des synthèses fidèles, concises et culturellement cohérentes, en exploitant des modèles récents de TAL et d'IA (par ex. modèles pré-entraînés multilingues).

3. Proposer un cadre d'évaluation qualitatif et quantitatif pour mesurer la pertinence, la concision et la fidélité des résumés générés, en intégrant des considérations linguistiques.

3. Organisation du mémoire

Ce mémoire s'articule autour de trois chapitres principaux qui suivent une progression logique, de l'analyse de l'état de l'art à la mise en œuvre et l'évaluation du système proposé, jusqu'à son développement.

- Chapitre 1 : État de l'art sur les techniques de Web Scrapping et des Modèles de Langue Larges. Ce chapitre présente une revue des principales techniques de traitement automatique du langage naturel (TAL) et d'intelligence artificielle (IA) appliquées à la collecte, à la structuration et à la génération automatique de

résumés. Une attention particulière est portée : aux approches de web scraping; aux modèles de résumé automatique extractif et abstractive ; et des modèles de langue Large.

- Chapitre 2 : Conception du pipeline TAL pour la valorisation des données culturelles algérienne. Ce chapitre décrit en détail l'approche proposée. Il inclut : l'architecture globale du pipeline ; les méthodes de collecte et de nettoyage de données web de patrimoine ; les techniques de prétraitement linguistique adaptées au contexte algérien ; la mise en œuvre d'un système de résumé automatique à l'aide de modèles de LLM ; les choix techniques retenus pour l'adaptation culturelle et linguistique des résumés.

- Chapitre 3 : Implémentation, tests et évaluation du système développé. Ce chapitre expose l'environnement de développement et les outils utilisés (bibliothèques TAL, frameworks IA, plates-formes de déploiement). Il détaille : les expériences de validation (jeux de données, scénarios de test) ; les résultats obtenus selon différentes métriques (ROUGE, BERTScore, évaluation humaine) ; et enfin, une présentation de la plateforme développée dans le cadre du projet et des pistes d'amélioration futures.

CHAPITRE I: ÉTAT DE L'ART

1. Introduction

L'exploitation des données issues de sources web algériennes nécessite des outils capables de les extraire, structurer et résumer de manière automatique. *Le web scrapping* constitue la première étape de ce processus, en permettant la collecte de contenus souvent non structurés et épars.

Pour valoriser ces données, des techniques de *Résumé Automatique*, appuyées par les *Modèles de Langue de grande taille* (LLM), offrent des solutions performantes de synthèse et de génération. Ce chapitre présente un aperçu des approches existantes dans ces trois domaines, en lien avec le pipeline développé dans ce travail.

2. Recours au Web Scraping

La gigantesque quantité de données liée au patrimoine culturel algérien, qu'il soit matériel ou immatériel, représente aujourd'hui une richesse documentaire considérable, mais encore sous-exploitée. Ces données sont partiellement diffusées par des institutions culturelles, des musées, ou des plates-formes administratives locales et nationales, sous forme de fiches descriptives, de galeries d'images, ou de bases d'archives en ligne.

Pour combler ces lacunes, une solution complémentaire se trouve sur le web public. De nombreux acteurs — institutions culturelles, passionnés d'histoire, chercheurs, ou collectivités locales — publient en ligne des informations précieuses, bien que souvent présentées de manière non structurée et hétérogène. Ces contenus, répartis sur des dizaines de sites web, peuvent inclure des descriptions détaillées de monuments, des témoignages, des photos, ou des calendriers d'événements liés au patrimoine immatériel. Cependant, certaines informations essentielles sur les sites, les objets ou les pratiques culturelles, comme leur localisation précise, leur état de conservation, ou encore leur statut juridique, sont souvent imprécises, obsolètes ou tout simplement absentes de ces bases officielles. Parfois, elles sont dispersées sur des blogs spécialisés, des articles de presse, ou des pages d'associations locales, rendant leur exploitation difficile [1].

Ainsi, dans un contexte où les données culturelles sont principalement disponibles sur Internet, mais dispersées, non structurées et souvent difficilement exploitables par les outils classiques, le besoin d'automatiser leur collecte devient crucial. Les bases de données officielles sont souvent incomplètes et ne répondent pas à toutes les exigences des projets de valorisation du patrimoine. C'est dans cette optique que le web scraping s'impose comme une alternative efficace pour extraire des informations à partir de pages web.

2.1. Extraction de Données à partir des Documents Web

L'extraction d'information à partir de pages Web constitue une extension des techniques classiques d'extraction appliquées aux ressources textuelles [2]. Les documents Web sont majoritairement semi-structurés, bien que certains soient entièrement structurés ou totalement non-structurés. Les approches traditionnelles, centrées sur les textes non-structurés ou les bases de données relationnelles, s'avèrent souvent inadéquates pour ce type de contenu. Les méthodes d'extraction de données exploitent les métadonnées présentes dans le HTML, telles que les balises, les délimiteurs ou les unités syntaxiques, afin de localiser l'information pertinente. Cela a mené à l'émergence d'outils spécialisés appelés extracteurs ou adaptateurs, capables de transformer les données extraites selon un format prédéfini. Enfin, différents courants méthodologiques ont été identifiés, reposant sur l'analyse structurelle des documents Web, l'induction, les langages de description, ou encore l'exploitation d'ontologies pour guider l'extraction automatique.

La figure 1 ci-dessous, présente une vue d'ensemble synthétique du processus d'extraction d'informations sur le Web.

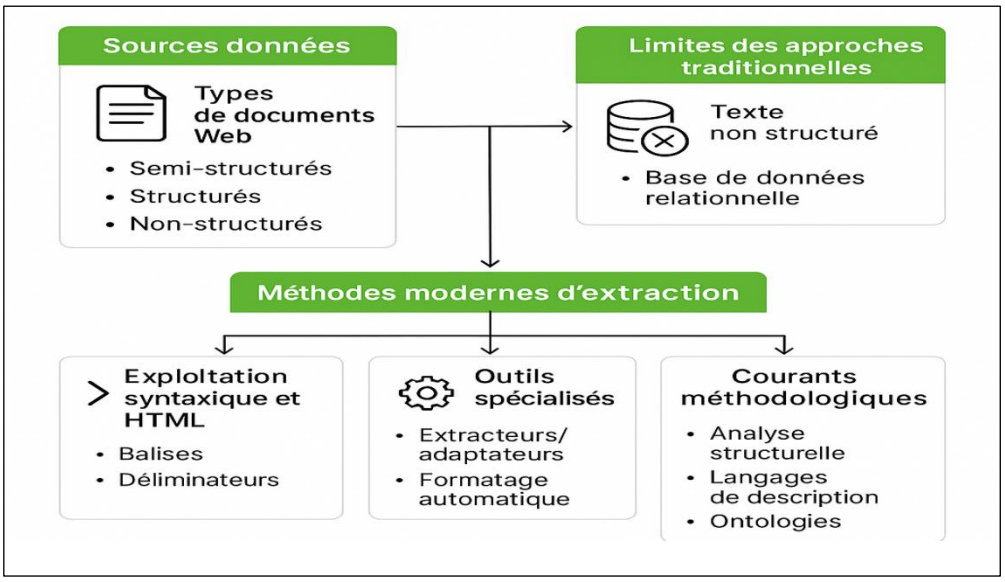

Figure 1 : Processus de l'Extraction de Donnée sur le Web.

2.2. Techniques d'Extraction Automatisée

La croissance rapide des contenus sur Internet a rendu nécessaires des méthodes d'extraction automatisée. Parmi elles, le *Web Scraping* et le *Web Crawling* occupent une place centrale. Complémentaires, ces deux techniques permettent de collecter et structurer des données issues du web. Elles sont devenues essentielles dans des domaines comme la Veille Stratégique, la Recherche d'Information, l'Analyse de la Concurrence ou encore l'Intelligence Artificielle. Cette section propose un aperçu de ces deux techniques d'extraction d'informations sur le Web, en mettant en lumière leurs caractéristiques, leurs limites, ainsi que les approches modernes développées pour y remédier.

2.2.1. Web Scraping

Le Web Scrapping est une technique d'Extraction automatisée et structuration de données spécifiques. Le web scraping (ou extraction de données web) est une technique ou un processus permettant de collecter ou d'extraire automatiquement une grande quantité de données à partir d'un ou plusieurs sites web en un temps court. Il est largement reconnu comme une technique efficace et puissante pour la collecte de données hétérogènes et volumineuses via un programme, un autre site web ou un script [3, 4].

2.2.1.1. Historique du Web Scrapping

Les étapes marquantes de l'évolution du web scrapping se résume en [3, 5]:

- **Début des années 1990** : Les premières recherches sur l'extraction de données web débutent, alors que le terme "Big Data" commence également à émerger pour décrire l'explosion des volumes de données.
- **Avant 1997 (Construction Manuelle)** : Les premiers adaptateurs sont construits manuellement par des experts en programmation (Java, Perl), une approche fastidieuse et peu flexible face aux changements fréquents des formats web.
- **Milieu des années 1990 (Robots de prix)** : Les "robots de prix" ou moteurs de comparaison de prix, tels que BargainFinder en 1995, voient le jour, illustrant une application précoce du web scraping pour le commerce en ligne.
- **1997 (WIEN)** : Le système WIEN est le premier à introduire la notion d'adaptateurs semi-automatique, capable d'apprendre des règles d'extraction à partir d'exemples de pages web étiquetés.
- **1997-1999 (STALKER)** : Le système STALKER est développé pour traiter les pages web à structure hiérarchique en utilisant un "arbre ECT" (Embedded Catalog Tree), ce qui le rend moins sensible à l'ordre ou à l'absence d'attributs dans les données.
- **1998 (SOFTMEALY)** : SOFTMEALY affine la notion de délimiteur en introduisant des "séparateurs" qui intègrent le contenu même de l'attribut à extraire, améliorant la précision, notamment pour les fragments de texte moins denses.
- **Début des années 2000 (Vers l'Automatisation)** : Des systèmes comme CO-TESTING (2000), IEPAD (2001), ROADRUNNER (2001), DELA (2002) et OLERA (2004) sont créés, réduisant progressivement la nécessité d'intervention humaine pour la génération des règles d'extraction.
- **2004 (Contexte Sémantique)** : Les recherches explorent l'annotation sémantique et la généralisation de contexte pour une extraction plus robuste qui s'adapte aux changements de format et aux textes non structurés, en se basant sur le sens et le type des données.

- **Années 2010 - Présent (Applications et Outils Modernes)** : Le web scraping se développe avec des outils et frameworks Python performants comme Scrapy[1], BeautifulSoup[2], et Selenium[3], permettant des applications complexes telles que la surveillance des prix ou la collecte de données maritimes.

2.2.1.2. Processus de Scraping

L'approche du Web Scrapping se basée principalement sur la structure des pages. Le processus suit plusieurs étapes [3, 6]:

a) **Accéder au site web :** Le programme se connecte au site et demande la page désirée. Il reçoit le contenu de la page en format HTML.

b) **Analyser et extraire les données :** Une fois le code HTML obtenu, le programme l'examine pour trouver et récupérer les informations spécifiques recherchées (par exemple, des titres, des liens, des paragraphes). Des outils spéciaux aident à transformer ce code en une structure que l'ordinateur peut comprendre.

c) **Formater et sauvegarder :** Les données collectées sont ensuite organisées et enregistrées, souvent dans des formats comme CSV ou JSON, ou dans une base de données, pour pouvoir être utilisées plus tard .

Ce mécanisme transforme les pages conçues pour l'affichage humain en jeux de données utilisables pour des traitements automatisés.

2.2.1.3. Techniques de Scraping Avancées

Plusieurs techniques permettent d'aller plus loin [6]:

- **Analyse structurelle** : repérage de schémas récurrents dans des pages similaires ou exploitation de structures HTML répétitives pour automatiser l'extraction sur des pages ayant une mise en page similaire.

[1] https://www.scrapy.org/

[2] https://pypi.org/project/beautifulsoup4/

[3] https://www.selenium.dev/

- Traitement des contenus dynamiques générés par JavaScript.
- Utilisation d'expressions régulières pour extraire des chaînes ciblées.
- Adaptation des extracteurs aux modifications fréquentes des pages web.

Les outils avancés permettent également la navigation simulée, la gestion des sessions, et le passage de vérifications automatiques comme les CAPTCHA.

2.2.1.4. Défis du Web scrapping

Malgré son potentiel, le web scraping présente plusieurs défis importants [3, 7]:

- **Évolution fréquente des sites web:** Les modifications régulières des structures de pages rendent les scripts de scraping rapidement obsolètes, nécessitant une adaptation continue.
- **Mécanismes de protection anti-scraping:** Les sites intègrent des techniques comme les honeypots ou des limites de profondeur qui compliquent l'extraction automatisée des données.
- Qualité et fiabilité des données extraites: Les informations collectées peuvent être incomplètes, erronées ou imprécises, ce qui limite leur exploitation fiable.
- Contraintes légales et éthiques: La pratique du web scraping soulève des questions légales importantes, nécessitant de se conformer aux directives d'accès fixées par les modérateurs des sites web ainsi qu'aux réglementations relatives à l'utilisation des données.

2.2.2. Web Crawling

Le web crawling désigne le processus d'exploration automatique du web afin d'en extraire des données. Cette technique est notamment utilisée par les moteurs de recherche pour analyser et indexer les pages disponibles en ligne. Un robot d'exploration, appelé web crawler, commence par visiter une liste d'URLs connues, puis suit les liens hypertextes pour découvrir de nouvelles pages. À la manière d'une araignée parcourant sa toile, il construit progressivement une carte du web. Lors de l'exploration, le crawler analyse divers éléments comme le titre, la méta-description ou encore la structure interne de la page [8]. Ce processus permet d'organiser l'information et de faciliter son accès, bien que seule une partie des pages web soit réellement indexée.

2.2.2.1. Fonctionnement Général d'un Crawler

Pour assurer une exploration efficace et optimiser l'indexation du contenu, le fonctionnement général d'un crawler comprend les étapes suivantes [9] :

a. Définition des URLs de départ (seeds) telles que le crawler commence par une liste d'URLs préalablement sélectionner pour la tâche d'extraction.
b. Le crawler parcourt et télécharge le contenu des pages web.
c. Il extrait les liens présents sur chaque page visitée.
d. Les liens trouvés sont ajoutés à la file d'attente pour exploration.
e. Il procède à un filtrage pour éviter les doublons et les boucles infinies.
f. Selon les objectifs, les données sont soit indexées, soit stockées.

2.2.2.2. Techniques de Crawling Optimisées

Pour optimiser ce processus, plusieurs techniques peuvent être utilisées [3-5, 7, 9] :

- **Utilisation de Sitemaps** : les crawlers identifient les liens hypertextes, qu'ils soient en HTML ou en XML, pour faciliter l'exploration ou l'indexation par des moteurs comme Google.
- **Méthodes de crawling spécifiques** (Cell text et Follow mode) : Les crawlers analysent la structure HTML et le Document Object Model (DOM) des pages pour extraire des données, y compris sur des pages web dynamiques utilisant JavaScript.
- **Exclusion de pages inutiles** : Pour éviter d'indexer des pages sans intérêt ou dupliquées, il est possible de spécifier des sections du site web à exclure via le fichier robots.txt. Il est également recommandé de limiter le nombre de pages récupérées ou la profondeur de parcours pour contourner les mécanismes anti-scraping ou les "HoneyPots" .
- **Ciblage des bonnes URLs** : Les crawlers commencent à partir d'une URL initiale et suivent les liens hypertextes pour découvrir de nouvelles pages. Une stratégie d'optimisation consiste à privilégier la recherche de "sources de données primaires" plutôt que des sites agrégateurs, en sélectionnant les URL

basées sur des critères comme la disponibilité de données et l'importance du trafic commercial.

2.2.2.3. Défis du Web Crawling

Le web crawling remplit quatre fonctions majeures [9]:

- L'indexation des contenus pour les moteurs de recherche
- Le suivi automatisé des mises à jour,
- L'archivage des pages web,
- L'analyse des réseaux hypertextes.

Malgré son importance, plusieurs défis majeurs se présentent [3-5]:

- **Volume et Complexité des Données et des Structures** : Les crawlers doivent gérer des volumes massifs de données, souvent non structurées ou semi-structurées, ce qui complique leur traitement et leur exploitation dans les applications. De plus, l'hétérogénéité des technologies web (HTML, JavaScript, AJAX) et des structures de documents rend l'extraction difficile, car les méthodes peuvent échouer si les pages ne correspondent pas aux modèles attendus.
- **Modifications Fréquentes de la Structure des Sites Web** : Les modifications fréquentes des structures de données sur les sites web augmentent les coûts de développement et de maintenance des outils d'extraction. Un simple changement peut rendre un crawler inopérant, entraînant l'extraction de données incomplètes ou incorrectes et rendant la construction des règles fastidieuse.
- **Qualité et Complétude des Données** : La fiabilité et la crédibilité des informations collectées sont difficiles à garantir, car les données peuvent inclure des erreurs, des omissions ou même de la fraude. L'absence fréquente de source et le manque de partage public de certaines informations limitent l'accès aux données primaires et peuvent conduire à des résultats d'extraction incomplets ou invalides.
- **Problématiques Linguistiques et de Normalisation** : La diversité des langues sur le Web pose un défi pour la recherche et la normalisation des

données. Cette pluralité linguistique implique également une variété d'abréviations et d'unités de mesure, rendant nécessaire l'uniformisation des valeurs extraites.

2.2.3. Complémentarité entre Scraping et Crawling

Il est important de noter que le web scraping vise à extraire des informations spécifiques, tandis que le web crawling (exploration du web) consiste à naviguer entre les pages en suivant des liens pour les indexer [4]. Souvent, le crawling est nécessaire pour le scraping afin de se déplacer d'une page à l'autre [3].

- Le crawling est vu comme une phase de découverte automatisée du web.
- Le scraping, quant à lui, consiste à extraire des données ciblées depuis les pages identifiées par le crawler.
- Ensemble, ils forment un système d'extraction complet, utile pour la veille, l'analyse de marché ou l'indexation.

La figure 2,ci-dessous, représente un schéma simple comparant le fonctionnement du web scrapper et le web crawler.

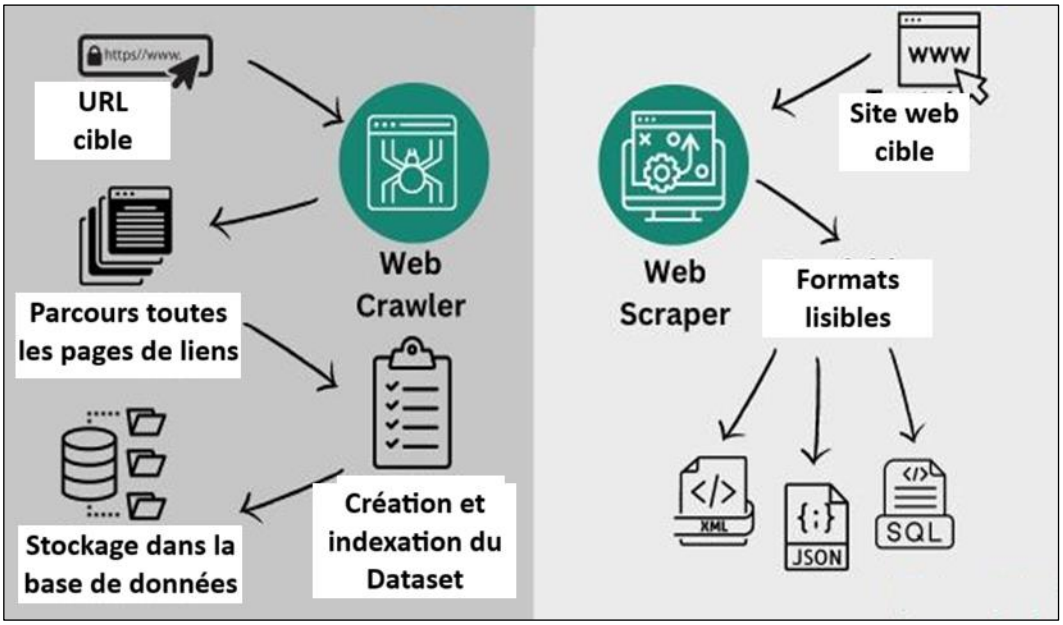

Figure 2: La Différence entre Web Scraping et le Web crawling.

2. Résumé Automatique de Texte

Le résumé automatique de texte (ATS) est un domaine de recherche actif visant à condenser de larges volumes de texte en des versions plus courtes, tout en préservant le contenu informationnel et la signification globale du document original. L'objectif principal de l'ATS est de générer des résumés brefs, cohérents et informatifs, permettant aux utilisateurs de saisir efficacement les idées clés d'un texte et de réduire considérablement le temps et l'effort nécessaires à la lecture et à la compréhension de documents volumineux [10].

L'ATS, en tant qu'application clé du traitement du langage naturel (NLP), est devenu indispensable pour filtrer et organiser cette masse d'informations, offrant un accès rapide aux faits et sujets les plus pertinents.

3.2. Historique des Techniques de Résumé Automatique

Le résumé automatique a évolué en quatre grandes étapes. D'abord, l'ère statistique (avant 2000) reposait sur des méthodes simples, comme la fréquence des mots ou TF-IDF, introduites notamment par Luhn (1958) [11]. Ces approches extractives étaient efficaces mais limitées en compréhension sémantique.

Dans les années 2000, l'apprentissage automatique classique a permis l'usage de modèles supervisés et de règles linguistiques, mais restait dépendant de caractéristiques manuelles et spécifiques au domaine [12].

L'arrivée du deep learning dans les années 2010, avec les modèles encodeur–décodeur (RNN, LSTM, GRU), a permis une meilleure prise en compte du contexte. Toutefois, ce n'est qu'avec l'architecture Transformer [13] que des modèles comme BERT, GPT ou BART ont pu s'imposer comme standards du résumé, malgré leur besoin en fine-tuning.

Depuis 2020, les grands modèles de langue (LLM) tels que GPT-3 et ChatGPT ont marqué un tournant : ils produisent des résumés cohérents en zero-shot ou few-shot, en combinant paradigmes extractif et abstrait avec une qualité linguistique supérieure [14-15].

La figure 4 ci-dessous résume les grandes ères du Résumé Automatique.

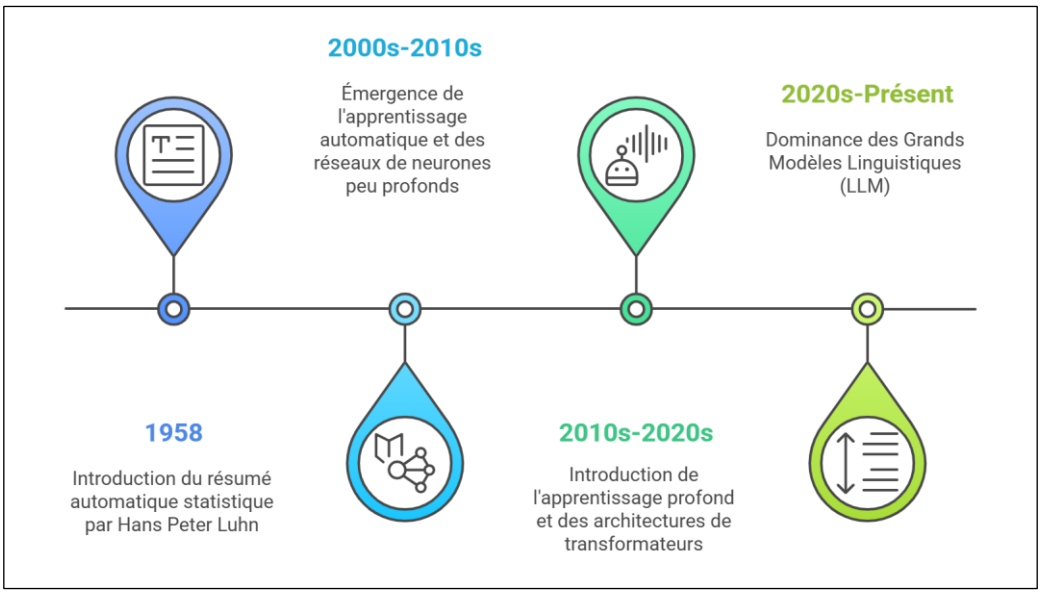

Figure 3: Évolution des Approches de Résumé Automatique de Texte.

3.3. Techniques de Résumé Automatique

Les systèmes ATS sont traditionnellement classés en méthodes extractives, abstractives et hybrides. Avec l'avènement des LLM, une nouvelle catégorie, les méthodes basées sur les LLM, a émergé, offrant une flexibilité accrue.

3.3.1. Méthodes de Résumé Automatique Conventionnelles

Il existe trois grandes familles de méthodes de résumé automatique de texte [16]:

- **Résumé Extractif** Le résumé extractif sélectionne directement les phrases les plus importantes du texte, au moyen de scores statistiques (TF-IDF, TextRank, LexRank), de clustering ou de classifieurs (SVM, régression). Ces méthodes exploitent la structure et la fréquence des termes pour évaluer la pertinence des segments, garantissant la fidélité factuelle.

  Les modèles extractifs excellent à capturer des terminologies précises et nécessitent moins de données d'entraînement, étant précis, rentables et efficaces. Cependant, ils peuvent manquer de qualité expressive par rapport aux résumés humains, produisant souvent des résultats redondants, trop longs ou contextuellement incohérents.

- **Résumé Abstractif:** Les modèles de résumé abstractif génèrent de nouvelles phrases via des règles (templates, arbres syntaxiques, graphes) ou des modèles génératifs (RNN/LSTM/GRU, Transformers, BART, Pegasus). Cette approche permet une plus grande flexibilité stylistique et une condensation plus poussée, en combinant et reformulant plusieurs idées en une seule phrase. Cependant, ils sont plus complexes à développer, nécessitant des ensembles de données de haute qualité, des ressources computationnelles importantes et un temps d'entraînement prolongé.
- **Résumé Hybride:** Le résumé hybride combine les méthodes extractives et abstractives : d'abord extraction des segments clés, puis reformulation ou compression par un modèle génératif (E2SA, E2A). Cette stratégie vise à équilibrer la précision factuelle de l'extraction avec la fluidité du style génératif, en limitant la partie la plus coûteuse du résumé abstrait. En pratique, elle peut nécessiter une orchestration fine entre les deux étapes pour éviter les incohérences, et sa performance dépend souvent de la qualité du pré-traitement extractif.

La figure 5 suivante illustre les méthodes les plus importantes de résumé automatique découlant de ces trois familles.

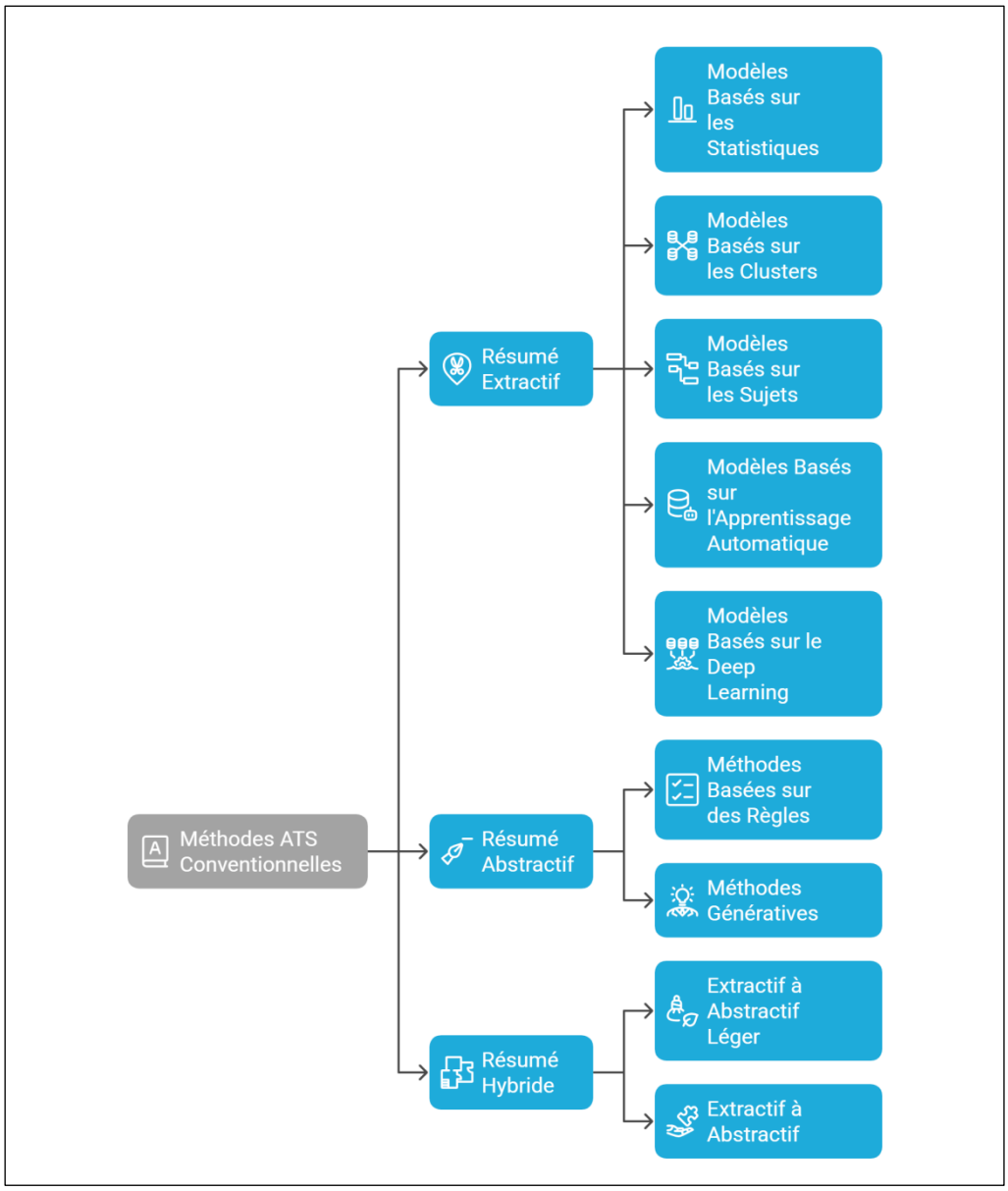

Figure 4: Méthode Conventionnelles de Résumé Automatique.

4. Méthodes de Résumé Basées sur les Grands Modèles Linguistiques

Les Grands Modèles Linguistiques (LLM), avec des milliards de paramètres, sont devenus centraux dans le traitement du langage naturel grâce à leur entraînement extensif sur diverses tâches et de vastes corpus textuels. Ils excellent dans des applications telles que le résumé, la réponse aux questions et le raisonnement logique. En ATS, les LLM ont atteint des performances comparables ou supérieures à celles

des humains, leurs résumés correspondant à la qualité et à la diversité de paraphrasage des textes humains [16].

4.1. Définition et Architecture des LLM

Un Grand Modèle Linguistique (LLM) est un système d'intelligence artificielle entraîné sur de vastes ensembles de données pour comprendre et générer du texte semblable à celui d'un humain. Les LLM sont généralement structurés comme des modèles autorégressifs, tels que GPT. La technologie clé que la plupart des LLM actuels utilisent est l'architecture Transformer. L'architecture Transformer se compose généralement de deux parties principales [17]:

- **L'Encodeur** : Traite la séquence d'entrée pour produire des représentations numériques (vecteurs d'intégration) qui encapsulent le sens du texte. BERT est une implémentation de transformeur qui n'utilise que l'encodeur.
- **Le Décodeur** : Génère le texte de sortie de manière séquentielle, en utilisant les sorties de l'encodeur et les mots déjà générés pour prédire le mot suivant. Les modèles à décodeur uniquement, comme les premiers modèles GPT, utilisent uniquement le composant décodeur.

L'innovation des Transformeurs repose sur trois concepts principaux :

- **Encodage Positionnel** : Permet au modèle de comprendre l'ordre des mots, car les transformeurs ne traitent pas le texte de manière séquentielle.
- **Mécanisme d'Auto-attention** : Permet au modèle de se concentrer sur les relations importantes entre les mots dans une séquence, aidant à lever l'ambiguïté et à comprendre les schémas sous-jacents de sens. L'attention multi-têtes (Multi-Head Attention) permet au modèle d'analyser plusieurs aspects du sens simultanément.
- **Couches Feed-Forward** : Transforme la représentation à toutes les positions de séquence.

La figure 6 ci-dessous résume cette architecture.

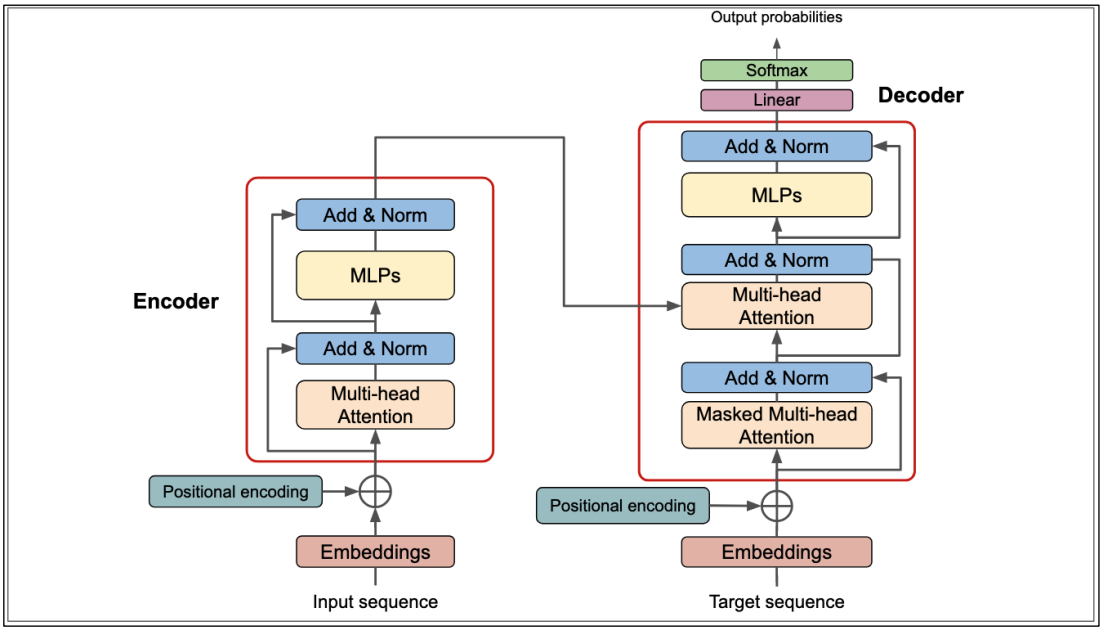

Figure 5: Architecture Encodeur-Décodeur d'un Transformer [18].

4.2. Types et Grandes Familles des Méthodes Basées sur les LLM

Les méthodes basées sur les LLM peuvent être classées en trois approches clés [16-18]:

1. **Méthodes basées sur l'ingénierie de prompts (***Prompt Engineering***)** : Ces méthodes impliquent la conception de prompts ou de templates efficaces pour guider les LLM à générer des résumés de haute qualité sans modifier les paramètres internes du modèle. L'avantage clé est son efficacité, minimisant le besoin d'un entraînement intensif et fonctionnant avec un petit ensemble d'exemples.
2. **Méthodes basées sur le réglage fin (***Fine-Tuning***)** : Dans cette approche, les LLM sont ajustés sur des ensembles de données de résumé dédiés. Le réglage des paramètres internes du modèle permet aux LLM de se spécialiser dans le résumé, améliorant ainsi leur précision et leur pertinence pour cette tâche.
3. **Méthodes basées sur la distillation des connaissances (***Knowledge Distillation***)** : Ces approches visent à extraire les connaissances des LLM pour créer des modèles plus petits et spécifiques à une tâche. Un modèle étudiant plus petit est supervisé par un modèle enseignant plus grand (le LLM). Cette approche permet au modèle étudiant d'hériter des capacités de résumé robustes du LLM, utile dans les

scénarios de ressources computationnelles limitées ou de strictes exigences de confidentialité des données.

5. Travaux Connexes

Dans cette section, nous présenterons les principaux domaines de recherche liés à notre projet de plateforme encyclopédique sur le patrimoine culturel algérien. Cette revue se concentrera sur quatres axes majeurs : (1) les plateformes de préservation du patrimoine culturel, (2) les techniques de web scraping pour la collecte de données culturelles, et (3) l'utilisation des modèles de langage (LLM) dans le traitement et la structuration de données patrimoniales et (4) les modèles de résumé automatique appliqués au contenu culturel.

5.1. Plateformes Encyclopédiques de Patrimoine Culturel

Europeana

- Description : Plateforme regroupant des millions d'objets culturels numérisés provenant de bibliothèques, musées et archives européennes.
- Forces : Interface multilingue, métadonnées standardisées, API ouverte pour les développeurs.
- Limites : Moins axée sur le patrimoine nord-africain, structure parfois trop rigide pour certains types de patrimoine immatériel.
- Pertinence : Modèle de référence pour l'organisation systématique de grandes collections patrimoniales.

Projet UNESCO Memory of the World

- Description : Initiative mondiale visant à préserver et diffuser le patrimoine documentaire de valeur universelle.
- Forces : Normes internationales de préservation, cadre méthodologique pour l'évaluation patrimoniale.
- Limites : Focus plus institutionnel que participatif, processus de catalogage centralisé.
- Pertinence : Offre des standards de classification que notre plateforme pourrait adapter.

CyArk

- Description : Organisation qui documente numériquement des sites patrimoniaux menacés à travers le monde.
- Forces : Modèles 3D haute résolution, données géospatiales précises.
- Limites : Concentré principalement sur le patrimoine bâti et moins sur les aspects immatériels.
- Pertinence : Approche technique pour la préservation numérique que nous pouvons partiellement intégrer.

5.2. Techniques de Web Scraping pour la Collecte de Données Culturelles

ArCo – Italian Cultural Heritage Knowledge Graph

- Description : Projet italien visant à créer un graphe de connaissances sur le patrimoine culturel, basé sur les catalogues officiels du ministère italien de la Culture.
- Forces : Architecture d'extraction multi-sources, enrichissement sémantique automatisé, utilisation de standards RDF et SPARQL.
- Limites : Dépendance à des structures de données relativement uniformes, complexité technique élevée.
- Pertinence : Framework méthodologique adaptable à notre contexte algérien.

Knowledge-based Relation Discovery in Cultural Heritage Knowledge Graphs

- Description : Approche basée sur la découverte de relations sémantiques dans les graphes de connaissances du patrimoine culturel, appliquée notamment aux biographies finlandaises.
- Forces : Extraction de relations sémantiques, enrichissement des données culturelles, utilisation de patrons ontologiques.
- Limites : Nécessite des données structurées et des ontologies bien définies.
- Pertinence : Méthodologie pour l'enrichissement sémantique des données culturelles.

5.3. Utilisation des LLM dans le traitement des données patrimoniales

AceGPT

- Description : Modèle de langage de grande taille localisé pour l'arabe, intégrant des instructions natives et un apprentissage par renforcement aligné sur les valeurs culturelles locales.
- Forces : Sensibilité aux nuances culturelles, capacité à traiter l'arabe dialectal, alignement sur les valeurs locales.
- Limites : Corpus d'entraînement encore limité pour certains aspects du patrimoine immatériel.
- Pertinence : Inspiration directe pour notre approche de traitement linguistique contextuel.

CAMeL – Cultural Adaptation of Multilingual Language Models

- Description : Ressource visant à mesurer les biais culturels dans les modèles de langage multilingues, en se concentrant sur les contextes arabes et occidentaux.
- Forces : Évaluation des biais culturels, adaptation des modèles aux contextes culturels spécifiques.
- Limites : Nécessite des ajustements spécifiques pour chaque culture ciblée.
- Pertinence : Méthodologie pour l'adaptation culturelle des modèles de langage.

5.4. Modèles de Résumé Automatique Appliqués au Contenu Culturel

BART – Denoising Sequence-to-Sequence Pre-training

- Description : Modèle de pré-entraînement pour la génération de texte, la traduction et la compréhension, efficace pour les tâches de résumé automatique.
- Forces : Préservation des informations critiques, adaptation au contenu patrimonial, efficacité multilingue.
- Limites : Risque de perte de nuances culturelles dans les résumés automatiques.

5.5. Synthèse comparative

Le tableau ci-dessous représente une synthèse comparative de ces différents travaux.

Tableau 1: Synthèse Comparative de Travaux et Projets Connexes.

Projet / Plateforme	Objectif principal	Techniques utilisées	Langue / Localisation	Interaction utilisateur	Limites
Europeana [4]	Centraliser le patrimoine culturel européen numérisé	Métadonnées normalisées, API, moteurs de recherche	Multilingue, Europe	Navigation par thématique, recherche avancée	Faible représentation du Maghreb et du patrimoine immatériel
UNESCO Memory of the World[5]	Préserver le patrimoine documentaire mondial	Normalisation documentaire, registres collaboratifs	International, multilingue	Interface de consultation	Peu participatif, accès limité aux données en format brut
CyArk[6]	Documenter les sites menacés du patrimoine mondial en 3D	LiDAR, photogrammétrie, données géospatiales	Global (notamment zones à risque)	Exploration 3D interactive	Limité au patrimoine bâti, non interactif pour le contenu immatériel
ArCo[7]	Construire un graphe de connaissances du patrimoine culturel italien	Web scraping, RDF, SPARQL, ontologies	Italien, anglais	Accès SPARQL, navigation sémantique	Nécessite des sources structurées, courbe d'apprentissage technique élevée
AceGPT[8]	Modèle de langage aligné sur les valeurs et le contexte arabe	LLM (transformer), fine-tuning contextuel	Arabe standard et dialectal	Chatbot, génération de texte	Entraînement encore limité pour certains aspects culturels spécifiques

[4] https://www.europeana.eu/
[5] https://www.unesco.org/en/memory-world
[6] https://www.cyark.org/
[7] http://wit.istc.cnr.it/arco

[8] https://www.acegpt.org/

CAMeL (NYU Abu Dhabi)[9]	Adapter les LLM multilingues aux biais et contextes culturels arabes	LLM, tests de biais, tuning culturel	Arabe (Maghreb, Levant, Golfe), anglais	Benchmark, analyse de performance	Ne traite pas directement le contenu patrimonial
BART[10] (Meta)	Génération automatique de texte, résumé, traduction	Transformer seq2seq, fine-tuning	Multilingue	Résumés, génération de réponses	Rendu parfois trop générique ou perte d'information culturelle importante

L'analyse comparative met en lumière un ensemble de projets et plateformes traitant du patrimoine culturel sous différentes dimensions : documentation, visualisation 3D, modélisation sémantique, ou encore génération automatique de contenu. Toutefois, plusieurs lacunes subsistent, notamment en ce qui concerne le patrimoine culturel algérien, tant sur le plan de la représentation que sur celui de la technologie adaptée à son traitement :

- Faible représentation géographique et culturelle : Les projets comme *Europeana* ou *UNESCO Memory of the World* offrent une couverture internationale ou européenne, mais intègrent très peu le Maghreb, en particulier l'Algérie. Le patrimoine immatériel algérien est presque absent.
- Limites techniques ou participatives : Certains projets misent sur des technologies avancées (LiDAR, SPARQL, LLM), mais présentent une faible accessibilité aux non-experts ou ne permettent pas une interaction avec du contenu non-structuré ou immatériel.
- Non prise en compte des réalités linguistiques et culturelles locales : Malgré quelques initiatives en arabe comme *AceGPT* ou *CAMeL*, aucune ne traite directement du contenu patrimonial algérien, ni ne prend en compte la richesse des dialectes ou des pratiques culturelles régionales.
- Absence d'intégration de bout en bout : Aucun projet n'intègre l'ensemble de la chaîne de traitement allant de l'extraction de contenu non structuré (comme

[9] https://nyuad.nyu.edu/

[10] https://huggingface.co/docs/transformers/model_doc/bart

des pages web ou articles) à la synthèse sémantique et à la présentation finale des informations dans une logique de valorisation patrimoniale.

Ces constats révèlent un gap important que notre projet vise à combler, en proposant une approche holistique et adaptée au contexte algérien. Celle-ci repose sur une combinaison de web scraping intelligent, de traitement par modèles de langage (LLM) et de résumé automatique avec des modèles comme BART, afin de structurer et valoriser un contenu culturel jusque-là peu exploité par les technologies du TAL.

6. Conclusion

L'analyse des travaux connexes révèle que, bien que de nombreuses initiatives existent dans le domaine du patrimoine culturel numérique, mais en l'absence de consensus ou de Standard et vue la nature de nos données, nous préonisons d'utiliser une approche combinant web scraping intelligent, traitement par LLM et génération de résumés via BART spécifiquement pour le patrimoine algérien ce qui constituera une contribution originale en soit car la majeure partie du contenu culturel relatif au patrimoine algérien est sous forme de pages web statiques sans aucune possibilité de Traitement Automatique évolué visant à la conservation, l'expansion, et l'exploitation eficace de ce contenu. Cette revue de littérature nous permet maintenant d'aborder notre méthodologie en ayant clairement identifié les approches existantes, leurs forces et leurs limites, et notre positionnement distinctif qui intègre l'ensemble du pipeline de traitement, de l'extraction à la présentation synthétique des informations culturelles.

CHAPITRE II: CONCEPTION ET MODELISATION DE LA PLATEFORME

1. Introduction

La modélisation constitue une étape clé dans la conception d'un système informatique robuste et évolutif. Dans le cadre de ce projet, l'objectif est de développer une plate-forme intelligente dédiée à la conservation numérique du patrimoine culturel algérien, en exploitant les technologies avancées du web scraping, du résumé automatique, et de la gestion de données multilingues.

Ce chapitre présente de manière structurée l'architecture globale du système, les différentes couches fonctionnelles, ainsi que les modèles utilisés pour représenter les flux de données, les processus métiers et les interactions utilisateur. La modélisation adoptée repose sur une approche modulaire, intégrant des composants spécialisés pour l'acquisition de contenu, le résumé automatique via les modèles de langage, la structuration des objets patrimoniaux, et leur exposition à travers des interfaces programmatiques (API).

2. Architecture Générale de l'Application

Le schéma qui suit (figure 6) illustre les grandes étapes visant à la constitue du contenu encyclopédique de notre plate-forme.

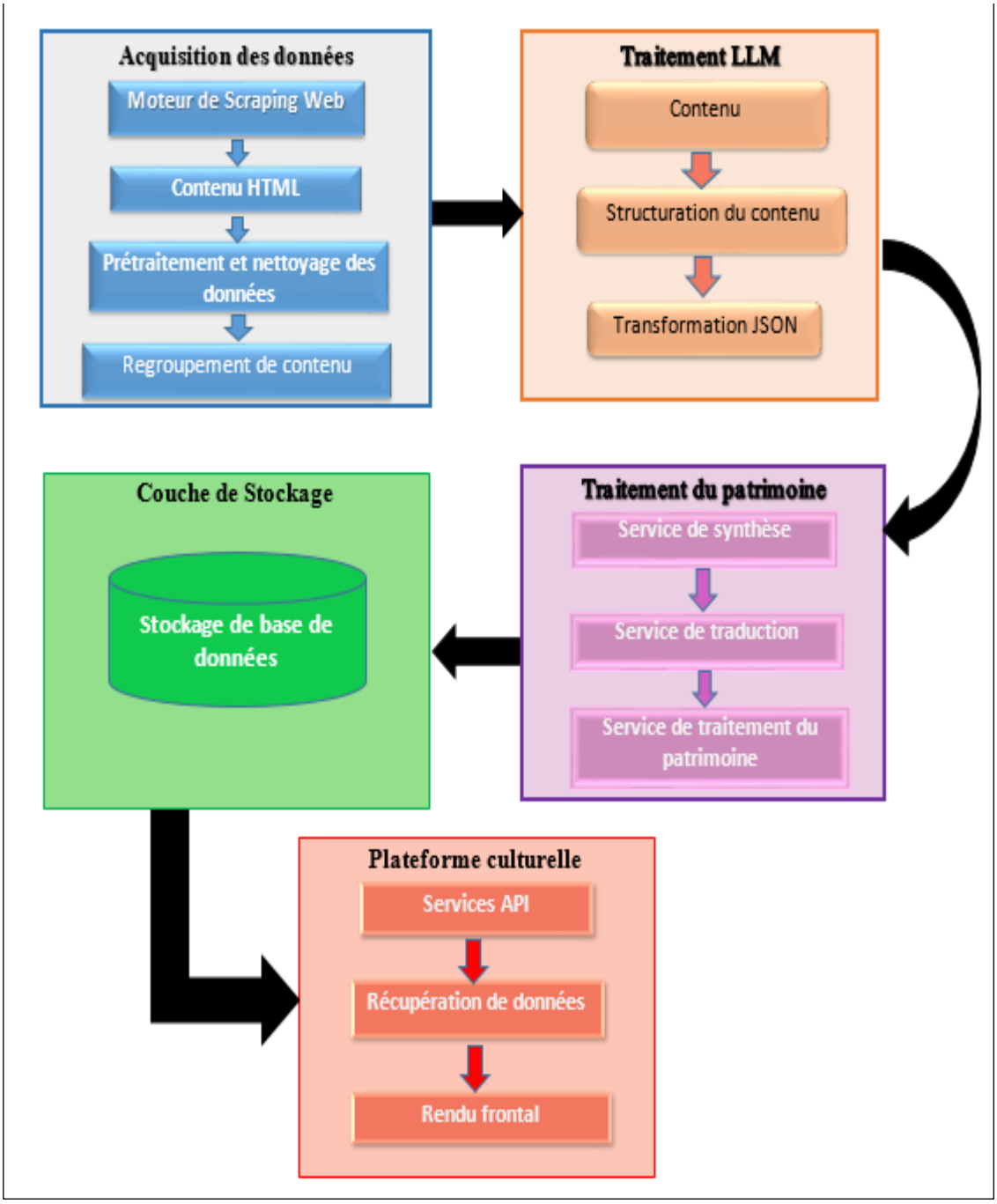

Figure 6: Architecture Générale de l'Application.

Dans ce qui suit, nous allons décrire cette architecture couche par couche.

2.1. Couche d'Acquisition de Données

La couche d'acquisition de données constitue le fondement du système d'extraction du patrimoine culturel algérien. Elle est chargée de récupérer et de préparer le contenu web brut en vue d'un traitement intelligent ultérieur. Cette couche implémente un

pipeline sophistiqué qui transforme les pages web non structurées en fragments de contenu propres et exploitables, prêts à être extraits grâce à la méthode LLM.

2.1.1. Moteur de Scraping Web

Ce composant joue un rôle central dans la collecte automatisée des données en parcourant les pages web ciblées afin d'en extraire le contenu pertinent.

Exemple :

Le moteur peut parcourir les pages de *www.tourismeblida.dz* pour récupérer les descriptions des lieux historiques de Blida.

- **Architecture d'implémentation**

Le composant de scraping web utilise Crawl4AI[11], un moteur d'exploration web asynchrone avancé, spécialement conçu pour l'extraction de contenu pilotée par l'IA. L'implémentation exploite les capacités asynchrones de Python pour permettre un traitement simultané et une utilisation efficace des ressources.

Les caractéristiques techniques principales de ce moteur de scraping sont résumées dans le tableau suivant :

Tableau 2: Spécification Techniques du Moteur de Web Scrapping.

Framework	Crawl4AI avec AsyncWebCrawler
Mode D'exécution	Traitement asynchrone avec asyncio[12] de Python
Gestion des Sessions	Identifiants de session uniques pour le suivi des requêtes
Configuration du navigateur	Paramètres de navigateur personnalisables via get_browser_config()

[11] https://docs.crawl4ai.com/

[12] https://docs.python.org/3/library/asyncio.html

- **Processus opérationnel:** Le moteur de scraping web lance des requêtes ciblées vers les sites web du patrimoine culturel, établissant des sessions d'exploration dédiées identifiées par des horodatages (heritage_crawl_session_{timestamp}). Le système utilise des paramètres de navigateur configurables pour optimiser la compatibilité avec diverses architectures web tout en préservant la discrétion.

2.1.2. Récupération de contenu HTML

Une fois le site identifié et l'exploration amorcée, le système procède à la récupération du contenu brut pour permettre une analyse plus fine.

Exemple :

Récupération d'un **`div`** contenant un paragraphe sur les origines du "Zellige algérien", accompagné d'images et de vidéos.

- **Processus d'acquisition de contenu:** Une fois la connexion établie, le moteur de scraping récupère les documents HTML complets grâce à la fonction fetch_page_body(). Ce processus capture l'intégralité de la structure DOM, y compris les scripts intégrés, les feuilles de style, les références multimédias et le contenu textuel.

- **Caractéristiques du volume de données**: Le contenu HTML récupéré varie généralement de quelques kilo-octets à plusieurs mégaoctets, selon la complexité et la richesse multimédia des sites web du patrimoine culturel ciblés. Le système enregistre les mesures de longueur du contenu à des fins de surveillance et d'optimisation.

2.1.3. Prétraitement et nettoyage des données

Avant toute tentative d'analyse ou d'extraction, les données brutes doivent être purifiées et structurées afin de garantir la pertinence et la qualité des résultats, elles doivent passer par plusieurs algorithmes de nettoyage :

- **Mise en œuvre de l'algorithme de nettoyage:** L'étape de prétraitement implémente un algorithme complet de nettoyage HTML via la fonction

clean_html_content(), en utilisant BeautifulSoup4 [13] pour l'analyse et la manipulation du *DOM*[14].

- **Suppression des éléments indésirables:** Le système supprime systématiquement les éléments non liés au contenu susceptibles d'interférer avec l'extraction des données patrimoniales comme par exemple les éléments: *script, style, noscript, meta, link, head, nav, header, footer, aside, form, input, button, select, textarea, iframe, embed, object.*

- **Filtrage basé sur les classes:** Des mécanismes de filtrage avancés ciblent les éléments dotés de classes CSS spécifiques, généralement associées au contenu non patrimonial : Classes filtrées : advertisement, ad, ads, sidebar, menu, navigation, cookie, popup, modal, social, share, comment, breadcrumb, pagination, footer, header.

- **Suppression des commentaires et des scripts:** L'algorithme de prétraitement identifie et supprime les commentaires HTML et les blocs de code JavaScript susceptibles d'introduire du bruit lors de la phase de traitement LLM ultérieure.

- **Amélioration de la qualité du contenu:** Au-delà des opérations de suppression, l'étape de prétraitement met en œuvre des procédures de normalisation du texte :

    - **Normalisation des espaces :** les espaces et les sauts de ligne excessifs sont consolidés.
    - **Extraction de contenu :** le contenu textuel pur est extrait tout en préservant la structure sémantique.
    - **Normalisation de l'encodage :** la cohérence de l'encodage des caractères est maintenue tout au long du processus.

2.1.4. Segmentation du Contenu

Après le nettoyage, le contenu doit être découpé en unités cohérentes et traitables pour les modèles de langage. Cette étape est cruciale pour optimiser les performances du système tout en préservant le sens des textes extraits.

[13] https://pypi.org/project/beautifulsoup4/

[14] Le DOM (Document Object Model) est une représentation d'un document HTML ou XML en tant qu'objet, permettant aux navigateurs web et aux langages de programmation comme JavaScript de manipuler le contenu, la structure et le style de la page web.

- **Stratégie de segmentation**: La dernière étape de la couche d'acquisition de données met en œuvre une segmentation intelligente du contenu grâce à la fonction chunk_content(). Ce processus répond aux limites pratiques des capacités de traitement LLM tout en préservant la cohérence du contenu.

- La segmentation repose sur des paramètres soigneusement définis afin d'équilibrer la taille des blocs et la lisibilité.Ils sont définis dans le tableau suivant:

Tableau 3: Spécification Techniques de la Segmentation de Contenu.

Taille maximale des blocs	2 000 caractères par bloc
Méthode de segmentation	Découpage tenant compte des limites des mots à l'aide de textwrap.wrap()
Préservation du contenu	Les mots longs et les termes composés restent intacts
Respect des limites	Les limites du langage naturel sont préservées afin de préserver le sens sémantique

- **Optimisation du traitemen**t: L'algorithme de segmentation utilise plusieurs stratégies d'optimisation :

  - **Contrôle des coupures** : break_long_words=False empêche la fragmentation artificielle des mots.
  - **Gestion des traits d'union** : break_on_hyphens=False préserve l'intégrité des termes composés.
  - **Traitement séquentiel** : Les blocs sont traités afin de préserver les relations contextuelles.

L'exemple illustré dans la figure 7 représente le résultats des traitements de la couche d'acquisition des données:

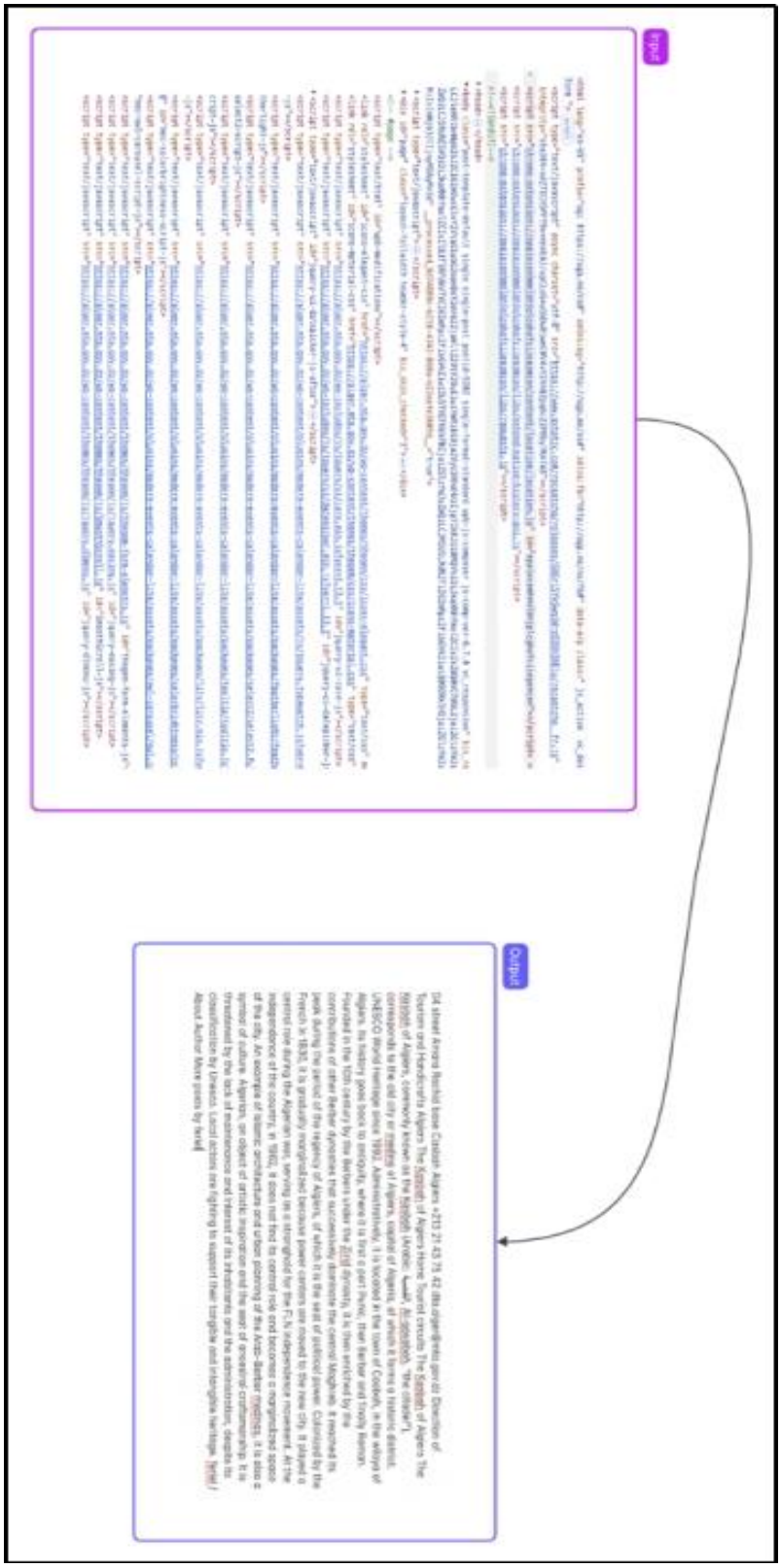

Figure 7: Illustration du flux de données à travers la couche "Acquisition de Données"

L'algorithme 1 ci-dessous, récapitule les différentes étapes de traitements effectués dans la couche d'acquisition des données.

Algorithm 1: Pseudocode de la couche d'acquisition de données pour l'extraction du patrimoine culturel

```
Fonction AcquisitionDonneesPatrimoine() :
    session_id ← "heritage_crawl_session_" + timestamp()
    navigateur ← get_browser_config()
    foreach site_cible dans liste_sites_patrimoine do
        crawler ← Crawl4AI.AsyncWebCrawler(config=navigateur)
        html_brut ← crawler.fetch_page_body(site_cible, session_id)
        contenu_nettoye ← NettoyerContenuHTML(html_brut)
        blocs_contenu ← SegmenterContenu(contenu_nettoye)
        foreach bloc dans blocs_contenu do
            envoyer bloc au module_LLM

Fonction NettoyerContenuHTML(html) :
    document ← BeautifulSoup(html)
    Supprimer balises : script, style, meta, link, head, nav, header, footer, etc.
    Supprimer éléments avec classes CSS : advertisement, sidebar, popup, etc.
    Supprimer commentaires HTML et scripts JS
    texte ← extraire contenu textuel
    Normaliser espaces et encodage
    return texte

Fonction SegmenterContenu(texte) :
    max_chars ← 2000
    segments ← textwrap.wrap(texte, width=max_chars,
        break_long_words=False, break_on_hyphens=False)
    return segments
```

Cette couche d'acquisition de données constitue la base essentielle de l'ensemble du pipeline d'extraction du patrimoine, garantissant que le traitement LLM ultérieur reçoive un contenu de haute qualité, correctement formaté et optimisé pour une extraction intelligente des informations du patrimoine.

2.2. Couche de traitement LLM

La couche de traitement LLM constitue le cœur intelligent du système d'extraction du patrimoine culturel. Elle transforme les fragments de contenu HTML nettoyés en objets de données patrimoniales structurés. Cette couche exploite les fonctionnalités avancées du modèle de langage étendu (MLM) via la plateforme Groq[15], mettant en

[15] https://groq.com/

œuvre un pipeline d'extraction sophistiqué combinant compréhension du langage naturel et génération de données structurées.

Exemple :

À partir du paragraphe “La danse Ahellil est pratiquée à Timimoun…”, le LLM extrait :

{

"title": "Danse Ahellil",

"category": "Danse",

"region": "Sud",

"period": "Antiquité",

"description": "Danse rituelle saharienne d'origine berbère, classée au patrimoine immatériel par l’UNESCO."

}

2.2.1. Extraction de contenu

Cette étape vise à identifier et extraire automatiquement les informations patrimoniales pertinentes à partir du texte nettoyé.

- **Architecture et configuration LLM** : Le processus d'extraction de contenu utilise DeepSeek-R1-Distill-Llama-70B, un modèle de langage de pointe accessible via la plate-forme d'inférence Groq. Ce choix de modèle offre un équilibre optimal entre vitesse de traitement, précision et rentabilité pour les tâches d'extraction de données patrimoniales.
- Les spécifications techniques de ce modèle sont résumées dans le tableau suivant:

Tableau 4: Spécification Techniques du Modèle de Langue pour l’Extraction de Contenu.

Modèle	groq/deepseek-r1-distill-llama-70b
Fournisseur	Plate-forme d'inférence Groq
Authentification	Sécurité basée sur des jetons API via des variables d'environnement
Mode de Traitement	Traitement par lots asynchrone avec gestion des erreurs

- **Mise en œuvre de la stratégie d'extraction**: Le système implémente la stratégie LLMExtractionStrategy du framework Crawl4AI, configurée avec des paramètres spécifiques pour l'extraction de données historiques :
- **Flux de travail de traitement intelligent** L'extraction de contenu s'effectue via la fonction process_chunk(), qui implémente des mécanismes de relance sophistiqués et des stratégies de limitation du débit :
- **Implémentation de la logique de relance:** Afin de garantir une résilience maximale, un système de relance intelligent est intégré, prenant en considération les spécificité décrite dans le tableau suivant:

Tableau 5: Spécification Techniques du Mécanisme de Relance.

Nombre maximal de tentatives	5 tentatives par bloc avec un délai de récupération exponentiel
Délai de base	5 secondes d'attente initiale
Stratégie de délai de récupération	Facteur de multiplication exponentiel (2 tentatives)
Gestion de la limite de débit	Détection Intelligente et attente adaptative

- **Mécanismes de traitement des blocs:** Chaque bloc HTML nettoyé subit un traitement LLM individuel, permettant :

  - **Traitement parallèle :** plusieurs blocs peuvent être traités simultanément
  - **Isolation des erreurs :** les blocs ayant échoué n'affectent pas la réussite globale de l'extraction
  - **Suivi de la progression :** Surveillance en temps réel de l'état du traitement
  - **Agrégation des résultats :** Combinaison transparente des résultats au niveau des blocs

2.2.2. Structuration du contenu

Cette phase convertit les éléments extraits en objets de données patrimoniales cohérents, conformes à un schéma prédéfini. Elle repose sur un modèle de données rigoureux pour assurer la standardisation et la validité des informations collectées.

- **Modélisation des données basée sur un schéma**: La phase de structuration du contenu met en œuvre un modèle de données via le framework BaseModel[16] de Pydantic[17], définissant la classe Heritage avec des spécifications de champs complètes comme décrites dans le tableau suivant:

[16] https://docs.pydantic.dev/latest/api/base_model/
[17] Pydantic est la bibliothèque de validation de données la plus utilisée pour Python.

Tableau 6: Spécification des Champs de la Classe Héritage.

Champ	Type	Description
Title	Str	Titre de l'objet patrimonial
Category	Str	Catégorie de l'objet patrimonial
Region	Str	Région associée
Period	Str	Période historique
Description	Str	Description détaillée
Image_url	Str	URL de l'image (optionnelle)

- **Extraction intelligente des champs**: Le LLM reçoit des instructions d'extraction détaillées qui guident le processus de structuration :
    - **Traitement multilingue** : Traduction automatique vers l'anglais lorsque le contenu source est dans d'autres langues
    - **Normalisation des noms** : Formatage et majuscules cohérents
    - **Désambiguïsation** : Résolution contextuelle des noms patrimoniaux ambigus
- **Système de classification des catégories**: Le système met en œuvre une taxonomie prédéfinie pour la catégorisation du patrimoine :
    - **Catégories :** Alimentation, Musique, Vêtements, Artisanat, Architecture, Danse, Langue, Coutumes, Mythologie, Site naturel, Site religieux.
    - **Cadre d'attribution régionale**: La classification géographique suit la structure administrative de l'Algérie: régions Nord, Sud, Est, Ouest, et Centre de l'Algérie.
    - **Contextualisation temporelle:** L'extraction avancée des périodes met en œuvre un traitement intelligent des dates :
      - *Calcul du siècle :* Conversion automatique des années en siècles
      - *Gestion des périodes* av. J.-C./a.c. : Valeurs négatives pour les périodes av. J.-C., positives pour les périodes après. J.-C.
      - *Datation contextuelle* : Inférence à partir de références historiques et de contextes culturels
      - *Patrimoine intemporel* : Gestion des champs vides pour le patrimoine naturel ou non daté
    - **Préservation des descriptions:**
      - *Extraction du contenu intégral* : Paragraphes descriptifs complets sans résumé.
      - *Services de traduction* : Traduction automatique en anglais tout en préservant le sens.
      - *Maintenance du contexte* : Préservation des nuances culturelles et historiques

Exemple :

L'objet "ksar" est classé dans la catégorie "Architecture", avec région "Sud", période "Moyen Âge".

2.2.2. Transformation JSON

Cette étape inclut:

- **Génération de données structurées**: La phase de transformation JSON convertit les réponses structurées du LLM en objets JSON standardisés, conformes au schéma Heritage. Ce processus garantit la cohérence des données et la compatibilité des API.
- **Gestion des formats de réponse:** Le système prend en charge plusieurs formats de réponse LLM :
    - *Réponses de liste* : plusieurs objets patrimoniaux extraits de blocs uniques
    - *Réponses d'objet unique* : éléments patrimoniaux individuels regroupés sous forme de liste.
    - *Réponses vides* : gestion fluide des blocs sans données patrimoniales extractibles.
- **Validation des données et assurance qualité**: Le processus de transformation met en œuvre des mécanismes de validation basé sur le filtrage de la qualité du contenu**:** Grâce à la fonction validate_heritage_items(), le système applique :
    - *Longueur minimale de la description* **:** seuil de 100 mots pour des descriptions pertinentes
    - *Validation des champs obligatoires* **:** champs de titre et de description obligatoires
    - *Détection de contenu générique* **:** filtrage des espaces réservés ou du texte du modèle
    - *Analyse de la structure des phrases* **:** exigence d'au moins deux phrases pour les descriptions
- **Indicateurs de qualité et rapports**
    - *Statistiques de validation* : rapports en temps réel sur les éléments validés/filtrés
    - *Scores de qualité* **:** évaluation quantitative de la précision de l'extraction
    - *Catégorisation des erreurs* : classification détaillée des raisons de contenu filtré
- **Normalisation des sorties:** La transformation JSON finale produit des objets patrimoniaux au format cohérent avec :
    - *Conformité du schéma* **:** Respect total des spécifications du modèle patrimonial.
    - *Enrichissement des métadonnées* **:** Horodatages de traitement supplémentaires et attribution de la source.
    - *Gestion des erreurs* **:** Dégradation progressive en cas d'échec d'extraction partielle.
    - *Logique d'agrégation* **:** Combinaison intelligente de résultats multi-blocs.

- **Intégration et performances des couches:** Cette dernière section évalue les performances globales de la couche LLM, tant du point de vue de l'efficacité du traitement que de la qualité des résultats produits.La couche de traitement LLM implémente plusieurs stratégies d'optimisation :

    - *Traitement par lots* : Gestion efficace de plusieurs blocs de contenu
    - *Opérations asynchrones* : Traitement non bloquant pour un débit amélioré
    - *Gestion des ressources* : Limitation intelligente du débit des API et optimisation des jetons
    - *Reprise après erreur* : Mécanismes de nouvelle tentative robustes pour les échecs temporaires

L’exemple illustré dans la figure 7 représente le résultats des traitements de la couche de Traitement par LLM:

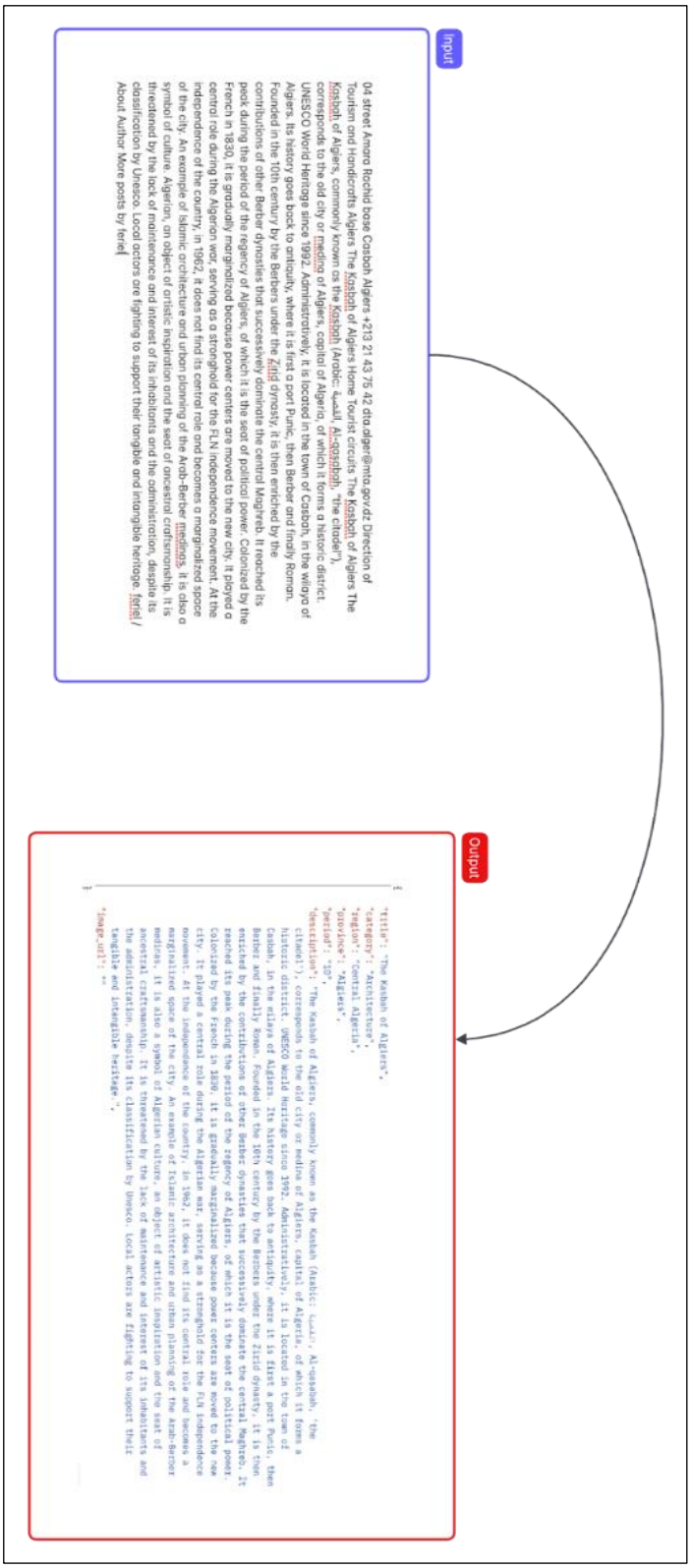

Figure 7: Illustration du flux de données à travers la couche "Traitement de LLM"

L'algorithme 2 ci-dessous, récapitule les différentes étapes de traitements effectués dans la couche 2.

Algorithm 2: Pseudocode de la couche LLM pour l'extraction et la structuration du contenu patrimonial

```
Fonction TraitementLLM(blocs_nettoyes) :
    foreach bloc dans blocs_nettoyes do
        tentative ← 0
        while tentative < 5 et pas de réponse valide do
            réponse ← process_chunk(bloc)
            if réponse valide then
                résultat ← réponse
                break
            attente ← 5 * 2^tentative secondes
            attendre(attente)
            tentative ← tentative + 1
        Ajouter résultat à la liste des extraits

Fonction StructurerContenuréponses :
    foreach réponse dans réponses do
        objets ← transformer_en_JSON(réponse)
        foreach objet dans objets do
            if validate_heritage_items(objet) then
                Ajouter objet à la base de données

Fonction validate_heritage_items(objet) :
    if longueur(description) < 100 ou titre est vide ou description est vide then
        return Faux
    if contenu générique détecté ou nombre_phrases(description) < 2 then
        return Faux
    return Vrai
```

Cette couche de traitement LLM transforme le contenu HTML brut et nettoyé en objets de données patrimoniales structurés et de haute qualité grâce à un traitement sophistiqué du langage naturel, garantissant ainsi que les informations extraites préservent leur authenticité culturelle tout en respectant des formats de données standardisés adaptés au stockage en base de données et à la présentation sur plate-forme.

2.3. Couche de traitement du patrimoine

La couche de traitement du patrimoine sert à l'enrichissement du contenu et à la préparation multilingue du système d'extraction du patrimoine culturel algérien. Elle relie les sorties de la couche de traitement LLM à la couche de stockage, en mettant en œuvre des services sophistiqués de synthèse et de traduction pour enrichir les données patrimoniales avec des résumés condensés et une prise en charge multilingue de l'arabe et du français.

2.3.1. Service de résumé

Le service de résumé permet de condenser efficacement les descriptions patrimoniales longues tout en préservant leur sens et leur richesse culturelle. Il s'appuie sur un modèle de pointe en traitement du langage naturel.

- **Implémentation du modèle BART**: Le composant de résumé utilise BART (Transformateurs bidirectionnels et autorégressifs), un modèle séquence à séquence de pointe spécialement conçu pour les tâches de résumé de texte. Cette implémentation permet une condensation intelligente du contenu tout en préservant le contexte du patrimoine culturel et les informations essentielles.
- **Spécifications techniques**: Les points ci-dessous présentent les détails techniques qui régissent le fonctionnement du module de résumé.
    - *Architecture du modèle* : Résumé basé sur un transformateur BART
    - *Mode de traitement* : Traitement par lots pour plusieurs descriptions du patrimoine
    - *Préservation du contenu* : Préservation du contexte culturel et de l'importance du patrimoine
    - *Optimisation de la longueur* : Résumés condensés permettant une prévisualisation rapide
- **Processus de résumé:** Le service SummarizerService implémente la méthode summary_paragraphs() qui traite les descriptions patrimoniales par lots :

```
descriptions = [h.get("description", "") for h in heritages]
summaries = HeritageProcessingService.summarizer.summarize_paragraphs(descriptions)
```

Figure 8 Méthode summary_paragraph()

- **Caractéristiques du traitement :** Ce service présente plusieurs caractéristiques clés qui garantissent un traitement performant, pertinent et respectueux du contexte culturel des données patrimoniales :
    - *Efficacité du traitement par lots* : Plusieurs descriptions traitées simultanément

- *Préservation du contexte* : Importance culturelle préservée sous une forme condensée
- *Assurance qualité* : Résumés pertinents qui capturent l'essence du patrimoine
- *Optimisation des performances* : Traitement efficace des grands ensembles de données patrimoniales

Exemple :

Texte original :

“Le burnous est un long manteau traditionnel en laine, porté lors des cérémonies, en particulier dans les régions montagneuses…”

Résumé automatique :

“Le burnous est un manteau traditionnel algérien utilisé lors de cérémonies.”

2.3.2. Service de traduction

Afin de garantir l’accessibilité du patrimoine culturel à un public diversifié, un service de traduction multilingue est intégré à la couche. Il permet de traduire les données en arabe et en français, tout en respectant le contexte culturel.

- **Intégration du modèle Helsinki-NLP:** Le composant de traduction exploite les modèles Helsinki-NLP[18], fournissant une traduction automatique neuronale de haute qualité pour les paires de langues arabe et française. Cette implémentation garantit l'accessibilité du contenu du patrimoine culturel à travers de multiples communautés linguistiques.
- **Spécifications techniques**: Les paramètres techniques suivants encadrent le fonctionnement du service de traduction.
  - *Cadre de traduction* : Modèles de transformation Helsinki-NLP.
  - *Langues prises en charge* : Anglais vers arabe (ar), anglais vers français (fr)
  - *Méthode de traitement* : Traduction au niveau du champ pour le titre, la description et le résumé.
  - *Sensibilité culturelle* : Traduction contextuelle préservant la terminologie du patrimoine.
- **Structure de données multilingue:** Le service de traduction produit du contenu multilingue structuré selon le format suivant :

[18] https://huggingface.co/Helsinki-NLP

```
"translations": {
    "ar": {
        "title": ar_title,
        "description": ar_description,
        "summary": ar_summary
    },
    "fr": {
        "title": fr_title,
        "description": fr_description,
        "summary": fr_summary
    }
}
```

Figure 9 : Illustration de la Structure Traduction.

2.3.3. Service de traitement du patrimoine

Ce service agit comme une passerelle entre les couches d'extraction et de stockage. Il orchestre l'ensemble du processus d'enrichissement via une approche modulaire combinant synthèse et traduction.

- **Pipeline de traitement intégré**: Le service de traitement du patrimoine orchestre l'intégralité du flux de travail d'amélioration du patrimoine via la méthode process_scraped_heritages(), combinant les opérations de synthèse et de traduction dans un pipeline de traitement unifié.

- **Flux de traitement**: Les différentes étapes traversées par un objet patrimonial dans le pipeline, depuis sa réception jusqu'à sa version enrichie prête à être stockée sont illustrées dans le tableau ci-dessous.

Tableau 7: Étapes du Processus de Gestion d'un Element du Patrimoine.

Étape 1 : Synthèse du contenu
- Traitement par lots des descriptions patrimoniales à l'aide du modèle BART - Génération de résumés concis préservant l'importance culturelle - Conservation du contexte patrimonial et des informations essentielles
Étape 2 : Traduction multilingue
- Traduction du titre, de la description et du résumé en arabe - Traduction du titre, de la description et du résumé en français - Préservation de la terminologie culturelle et du vocabulaire patrimonial
Étape 3 : Amélioration de la structure des données
- Association du contenu original aux résumés générés - Intégration des traductions multilingues dans un format structuré - Préparation des objets patrimoniaux enrichis pour le stockage en base de données

- **Structure améliorée des objets patrimoniaux**: Le service de traitement produit des objets patrimoniaux complets.


```
{
  "title": "Titre en anglais",
  "description": "Description complète en anglais",
  "summary": "Résumé généré par l'IA en anglais",
  "category": "Catégorie patrimoniale",
  "region": "Région géographique",
  "period": "Période historique",
  "image_url": "URL de l'image",
  "source_url": "URL source",
  "translations": {
    "ar": {
      "title": "العنوان بالعربية",
      "description": "الوصف الكامل بالعربية",
      "summary": "الملخص بالعربية"
    },
    "fr": {
      "title": "Titre en français",
      "description": "Description complète en français",
      "summary": "Résumé en français"
    }
  }
}
```


Figure 10: Illustration de la Structure de l'Objet Patrimonial après le Service de Traitement.

2.4. Couche de stockage

La couche de stockage constitue le fondement de la gestion persistante des données du système d'extraction du patrimoine culturel algérien. Elle est responsable du stockage, de l'organisation et de la récupération fiables des objets de données patrimoniales structurées avec un support multilingue. Cette couche met en œuvre une architecture de base de données sophistiquée qui relie les résultats de traitement intelligents (le résumé et la traduction) aux exigences d'accès aux données de la plate-forme culturelle, garantissant l'intégrité des données, l'évolutivité et des mécanismes de récupération efficaces pour le contenu du patrimoine multilingue.

2.4.1. Architecture et configuration de la base de données

Ce volet présente les choix technologiques et les stratégies mises en œuvre pour assurer une base de données robuste, évolutive et adaptée à la gestion multilingue des données patrimoniales :

- **Intégration de MongoDB Atlas**: La couche de stockage utilise MongoDB Atlas, un service de base de données NoSQL cloud-native, spécialement choisi pour son modèle de stockage orienté document qui s'adapte naturellement à la structure hiérarchique des données du patrimoine culturel avec des traductions multilingues imbriquées. L'implémentation exploite la conception de schéma flexible de MongoDB pour gérer différentes structures d'objets patrimoniaux tout en préservant la cohérence entre les différentes versions linguistiques.

- **Spécifications techniques**: Les spécifications techniques suivantes définissent les composants essentiels utilisés pour sécuriser et connecter le système à la base de données cloud :

    - *Fournisseur de base de données* : MongoDB Atlas (hébergé dans le cloud)
    - *Infrastructure de connexion* : MongoEngine ODM (Object Document Mapper)
    - *Authentification* : Gestion des identifiants basée sur des variables d'environnement
    - *Protocole de sécurité* : Chiffrement SSL/TLS avec validation des certificats
    - *Prise en charge multilingue* : Structure de document imbriquée pour le stockage des traductions
- **Gestion des connexions améliorée**: La configuration de la base de données met en œuvre une stratégie de connexion robuste via la fonction initialize_db(), conçue pour gérer les complexités de la connectivité des bases de données cloud et les exigences de compatibilité Python 3.12.

- **Stratégie de connexion principale**: Le mécanisme de connexion principal implémente une configuration SSL/TLS complète :

    - *Validation de l'autorité de certification* **:** Utilise certifi.where() pour l'emplacement du paquet de certificats de confiance.
    - *Sécurité TLS* **:** Applique la sécurité de la couche transport avec une validation stricte des certificats.
    - *Résilience en écriture* **:** Active les mécanismes de nouvelle tentative automatique pour les opérations d'écriture.
    - *Gestion du délai d'expiration* **:** Délai d'expiration de la sélection du serveur étendu (30 secondes) pour une connectivité cloud fiable.
- **Mécanisme de connexion de secours**: Le système intègre une logique de secours intelligente pour garantir la résilience de la connectivité :

    - *Dégradation progressive* **:** Retour automatique à des paramètres de connexion simplifiés.
    - *Simplification des paramètres* **:** Supprime les paramètres de requête complexes pour les connexions SSL de base.
    - *Isolation des erreurs* **:** Gestion des exceptions distincte pour les tentatives principales et de secours.
    - *Vérification de la connexion* **:** Rapports d'état en temps réel pour les deux stratégies de connexion.
- **Cadre de sécurité de l'environnement**: La configuration implémente une gestion sécurisée des identifiants via :

- *Isolation des variables d'environnement* **:** URI de base de données stockés dans des fichiers .env
- *Sécurité de production* **:** Identifiants séparés Base de code
- *Protection des chaînes de connexion :* composants URI protégés contre toute exposition

2.4.2. Modélisation des données et conception de schémas

Cette section décrit la structuration logique des données patrimoniales dans la base de données, avec un accent particulier sur la prise en charge du multilinguisme et la flexibilité du schéma.

- **Structure de documents patrimoniaux avec prise en charge multilingue**: La couche de stockage implémente le modèle de document patrimonial amélioré à l'aide de la classe Document de MongoEngine[19], créant ainsi une structure de données complète qui capture tous les aspects essentiels des objets du patrimoine culturel algérien avec une prise en charge complète de la traduction multilingue.
- **Architecture de champs de base améliorée**:

```python
class Heritage(Document):
    """
    Document for heritage sites with multilingual support.
    """
    title = StringField(required=True)
    description = StringField(required=True)
    summary = StringField(required=True)
    category = StringField(required=True)
    region = StringField(required=True)
    period = StringField()
    image_url = StringField()
    source_url = StringField()
    translations = DictField()  # Holds translations like {"ar": {"title": "...", "description": "..."}}
    created_at = DateTimeField(default=datetime.now)
    updated_at = DateTimeField(default=datetime.now)
```

Figure 11 : Illustration du Document Heritage.

Le tableau ci-dessous récapitule les champs du document héritage.

Tableau 8: Les Champs du Document Heritage.

Champs	Description
Champ Titre	Identifiant principal avec validation obligatoire (original anglais)
Champ Description	Contenu narratif patrimonial complet (original anglais)
Champ Résumé	Informations patrimoniales condensées générées par l'IA à l'aide du modèle BART
Champ Catégorie	Taxonomie de classification du patrimoine
Champ Région	Système d'attribution géographique

[19] http://mongoengine.org/

Champ Période	Contexte temporel historique (valeur nulle pour un patrimoine intemporel)
Champ URL Image	Stockage de références de médias visuels
Champ URL Source	Suivi de l'attribution du contenu original
Champ Traductions	DictField contenant des objets de traduction imbriqués

- **Traductions arabes** :

  {"ar": {"title": "...", "description": "...", "summary": "..."}}

- **Traductions française :**

  {"fr": {"title": "...", "description": "...", "summary": "..."}}

- **Structure extensible :** Prise en charge de langues supplémentaires selon les besoins

  - Date de création : Génération automatique d'un horodatage à la création du document
  - Date de mise à jour : Mise à jour dynamique de l'horodatage lors de la modification du document

- Spécification de la structure de traduction: Le champ de traduction implémente une structure de dictionnaire imbriquée :

```
translations = {
    "ar": {
        "title": "العنوان بالعربية",
        "description": "الوصف الكامل بالعربية",
        "summary": "الملخص بالعربية"
    },
    "fr": {
        "title": "Titre en français",
        "description": "Description complète en français",
        "summary": "Résumé en français"
    }
}
```

Figure 12 : Illustration de la structure du champs « translations »

- **Stratégies d'optimisation de la base de données:** Le modèle de document implémente des optimisations stratégiques de base de données :

  - *Nommage des collections* : spécification explicite des collections « héritages »
  - *Stratégie d'index primaire* : indexation des champs de titre pour des performances de recherche améliorées

- *Indexation multilingue :* indexation des titres en arabe et en français pour une recherche multilingue
- *Indexation par catégories et régions :* capacités de filtrage améliorées
- *Optimisation des requêtes* : structures d'index préparées pour les modèles d'accès courants

- **Gestion automatique des horodatages**: Le modèle implémente une gestion intelligente des horodatages grâce à la substitution de méthodes, cette implémentation assure :

  - *Maintenance de la piste d'audit* : suivi automatique des modifications de données
  - *Cohérence temporelle* : mises à jour fiables de l'horodatage à chaque sauvegarde
  - *Intégrité des données* : chaîne d'héritage préservée tout en ajoutant un comportement personnalisé

2.5. Couche Contrôleur de Base de Données et Intégration Multilingue

La couche de stockage constitue un élément central du système d'extraction du patrimoine culturel, en assurant à la fois la persistance fiable des données et leur préparation à l'exposition via une API multilingue. Elle repose sur une architecture bien structurée séparant la logique métier des opérations de persistance, via un contrôleur dédié et des services spécialisés.

2.5.1. Architecture orientée service

Le module DatabaseController agit comme interface principale entre la couche application et les opérations de stockage. Il délègue les traitements complexes à un service HeritageService, qui encapsule :

- La gestion fine des objets patrimoniaux traduits ;
- Le traitement spécifique des données multilingues ;
- La centralisation des erreurs pour une robustesse accrue.

Ce découplage favorise la maintenabilité, la réutilisabilité du code, et une meilleure gestion des évolutions fonctionnelles.

2.5..2 Intégration dans le pipeline LLM

La couche de stockage reçoit les objets multilingues au format JSON, générés en amont par les modèles de langage. Elle assure leur validation structurelle (format, complétude, champs obligatoires), leur normalisation (codes de langue, dates ISO,

identifiants compatibles API), puis leur structuration via un schéma de données formel (ex. : Pydantic ou MongoEngine). Chaque objet est ensuite validé sur les points suivants :

- Présence des langues de traduction requises (arabe, français, anglais) ;
- Structure correcte des traductions et résumés ;
- Intégrité des métadonnées (source, horodatage, contexte).

Un mécanisme de traitement par lots est également intégré, permettant la gestion simultanée de plusieurs objets patrimoniaux issus d'une même session.

2.5..3 Garanties de persistance et cohérence

L'intégration de transactions atomiques garantit la synchronisation des données traduites, ainsi que la mise à jour automatique des horodatages et des index. La couche assure également :

- La durabilité des enregistrements via MongoDB Atlas (sauvegardes et réplication) ;
- La cohérence des objets multilingues selon les contraintes ACID ;
- L'interopérabilité avec les systèmes externes grâce à une exposition API en JSON normalisé.

2.5.4. Préparation à l'exposition API

Une fois validées et stockées, les données sont prêtes à être exposées via une interface API. Plusieurs formats de réponse sont pris en charge :

- Réponses par langue ;
- Objets complets avec toutes les traductions ;
- Réponses d'erreur standardisées.

La prise en charge des opérations par lots et la compatibilité avec des filtrages linguistiques assurent une intégration fluide avec les plates-formes culturelles exploitant ces données.

2.5.5. Suivi des performances et de la qualité

La couche intègre un système de journalisation et d'analyse des indicateurs :

- Taux de réussite des opérations ;
- Qualité et exhaustivité des traductions ;
- Temps de réponse et erreurs typées (structure, langue, contenu) ;
- Suivi de la croissance du volume de données et de la distribution linguistique.

```
{
    "category": "Natural Site",
    "created_at": "2025-06-03T16:17:27.546000",
    "description": "The Tassili n'Ajjer is a vast plateau located in the southeast of Algeria, bordering
        Libya, Niger, and Mali, covering an area of 72,000 km². The exceptional density of paintings and
        engravings, along with numerous prehistoric remains, are extraordinary testimonies to
        prehistory. From 10,000 years before our era to the first centuries AD, successive populations
        left many archaeological traces, including habitats, tumuli, and enclosures, which have yielded
        abundant lithic and ceramic material. However, it is the rock art (engravings and paintings)
        that brought Tassili worldwide fame from 1933 onwards, the year of its discovery. To date, 15,
        000 engravings have been cataloged. The site is also of great geological and aesthetic interest:
        the panorama of geological formations with their 'rock forests' of eroded sandstone offers the
        image of a strange lunar landscape.",
    "id": "683f12070b1d6c70779bd37c",
    "image_url": "https://whc.unesco.org/uploads/thumbs/site_0179_0001-750-750-20090910111123.jpg",
    "period": "-6",
    "region": "Southern Algeria",
    "source_url": "",
    "summary": "The Tassili n'Ajjer is a vast plateau located in the southeast of Algeria, bordering
        Libya, Niger, and Mali. The exceptional density of paintings and engravings, along with numerous
        prehistoric remains, are extraordinary testimonies to prehistory. To date, 15,000 engravings
        have been cataloged.",
    "title": "Tassili n'Ajjer",
    "translations": {…
    },
    "updated_at": "2025-06-03T16:17:27.546000"
},
```


Figure 13: Exemple de la structure final de l'objet patrimonial stockée dans la base de donnée

2.5. Couche Plate-forme Culturelle

La couche plate-forme culturelle constitue l'interface utilisateur finale du système d'extraction du patrimoine culturel algérien. Elle transforme les données patrimoniales structurées et multilingues stockées dans la base de données en une expérience utilisateur interactive et accessible. Cette couche implémente une architecture moderne basée sur une API REST Flask[20] et une interface utilisateur Vue.js[21], offrant une plate-forme complète pour la découverte, l'exploration et la consultation du patrimoine culturel algérien avec une prise en charge multilingue complète (anglais, arabe, français).

2.5.1. Service API

Le service API constitue la couche d'interface entre les données patrimoniales stockées et l'application frondent, fournissant des endpoints RESTful pour l'accès aux données avec une prise en charge multilingue intégrée.

[20] https://flask-restful.readthedocs.io/en/latest/

[21] https://vuejs.org/

- **Architecture d'implémentation**: Le service API utilise Flask, un framework web Python léger et flexible, particulièrement adapté à la création d'APIs REST rapides et efficaces. L'implémentation exploite la simplicité de Flask pour créer une interface de programmation claire et bien structurée qui expose les données patrimoniales multilingues de manière cohérente et accessible.
- **Spécifications techniques***:* Les caractéristiques techniques principales du service API sont les suivantes :
    - *Framework* **:** Flask (Python)
    - *Architecture* **:** API REST
    - *Format de données :* JSON avec support multilingue
    - *Communication base de données* **:** Intégration directe avec la couche de stockage MongoDB
    - *Gestion des erreurs* : Codes de statut HTTP standardisés
    - *Performance* **:** Réponses optimisées pour les données patrimoniales volumineuses
- **Endpoints principaux**: Le service API expose deux endpoints fondamentaux pour l'accès aux données patrimoniales :

    ✧ **Endpoint de récupération complète des patrimoines**

    @app.route('/api/heritages', methods=['GET'])

    Cet endpoint permet la récupération de l'ensemble de la collection patrimoniale stockée dans la base de données MongoDB. Il retourne un tableau JSON contenant tous les objets patrimoniaux avec leurs traductions complètes en anglais, arabe et français, leurs résumés générés par l'IA, et toutes les métadonnées associées.

    Caractéristiques de l'endpoint :

    Méthode HTTP **:** GET
    Réponse : Tableau JSON d'objets patrimoniaux multilingues
    Structure de données **:** Objets Heritage complets avec champs de traduction imbriqués
    Performance **:** Optimisé pour gérer de grandes collections patrimoniales
    Gestion des erreurs **:** Codes d'erreur HTTP appropriés en cas d'échec de récupération

    ✧ **Endpoint de récupération patrimoniale individuelle**

    @app.route('/api/heritages/<heritage_id>', methods=['GET'])

Cet endpoint permet l'accès à un objet patrimonial spécifique via son identifiant unique MongoDB ObjectId. Il retourne l'objet patrimonial complet avec toutes ses traductions et métadonnées associées.

Caractéristiques de l'endpoint :

Méthode HTTP **:** GET
Paramètre **:** heritage_id (ObjectId MongoDB)
Réponse **:** Objet JSON patrimonial individuel avec traductions complètes
Validation **:** Vérification de l'existence de l'identifiant patrimonial
Gestion des erreurs **:** Retour 404 si l'objet patrimonial n'existe pas

- **Intégration avec la couche de stockage**: Le service API communique directement avec la couche de stockage MongoDB via le contrôleur DatabaseController et le service HeritageService. Cette intégration garantit :

  - **Cohérence des données :** Accès direct aux données patrimoniales validées et structurées
  - **Performance optimisée :** Récupération efficace des données multilingues sans transformation supplémentaire
  - **Intégrité transactionnelle :** Garantie de cohérence des données lors des opérations de lecture
  - **Gestion des traductions :** Récupération automatique de toutes les versions linguistiques disponibles

2.5.2. Récupération de données

La récupération de données implémente les mécanismes de communication entre l'interface utilisateur Vue.js et le service API Flask, utilisant des requêtes HTTP asynchrones pour une expérience utilisateur fluide et réactive.

- **Architecture de communication**: La communication entre le frontend Vue.js et l'API Flask s'effectue via la bibliothèque Axios, un client HTTP basé sur les promesses JavaScript qui permet des requêtes asynchrones optimisées pour les applications web modernes.
- **Spécifications techniques de récupération:** Les caractéristiques techniques de la récupération de données sont les suivantes :
  - *Bibliothèque HTTP* **:** Axios pour les requêtes asynchrones
  - *Format d'échange :* JSON avec support multilingue complet
  - *Méthode de communication* **:** Requêtes HTTP GET vers les endpoints Flask
  - *Gestion des erreurs* **:** Interception et traitement des erreurs de communication
  - *Performance* **:** Requêtes optimisées avec gestion du cache côté client
- **Fonctionnalités de recherche et filtrage avancées:** La couche de récupération implémente des mécanismes de recherche et de filtrage sophistiqués pour

permettre aux utilisateurs de naviguer efficacement dans la collection patrimoniale.

- **Recherche multilingue:** Le système offre des capacités de recherche avancées qui fonctionnent dans les trois langues supportées :

  Recherche dans les titres : Recherche simultanée dans les titres anglais, arabes et français
  Recherche dans les descriptions : Recherche textuelle complète dans toutes les descriptions traduites
  Recherche dans les résumés : Recherche dans les résumés générés par l'IA dans toutes les langues
  Logique de pertinence : Algorithme de classement des résultats par pertinence multilingue

- **Filtrage par critères patrimoniaux:** Le système propose des options de filtrage avancées basées sur les métadonnées patrimoniales :

  Filtrage par région géographique : Nord de l'Algérie, Sud de l'Algérie, Est de l'Algérie, Ouest de l'Algérie, et Centre de l'Algérie
  Filtrage par catégorie patrimoniale : Alimentation, Musique, Vêtements, Artisanat, Architecture, Danse, Langue, Coutumes, Mythologie, Site naturel, Site religieux
  Filtrage par période historique : Filtrage par siècles spécifiques, Gestion des périodes avant J.-C. et après J.-C., Options de plage temporelle pour les recherches avancées.

- **Gestion des données multilingues côté client:** La récupération de données implémente une logique intelligente pour la gestion des contenus multilingues côté client :

  - **Stratégie de langue par défaut**

    Langue primaire : L'anglais est affiché par défaut lors du chargement initial
    Contenu de fallback : Si une traduction spécifique n'est pas disponible, le système revient au contenu anglais
    Cohérence d'affichage : Maintien de la cohérence linguistique à travers tous les éléments de l'interface

  - **Commutation de langue dynamique**

    Changement en temps réel : Mise à jour instantanée du contenu lors du changement de langue
    Préservation de l'état : Maintien de l'état de navigation et des filtres lors du changement de langue
    Synchronisation des données : Récupération des traductions appropriées depuis l'objet patrimonial complet

2.5.3. Rendu frontal

Le rendu frontal implémente une interface utilisateur moderne et responsive utilisant Vue.js et Tailwind CSS, offrant une expérience utilisateur optimale pour l'exploration du patrimoine culturel algérien avec une prise en charge multilingue complète.

- **Architecture de l'interface utilisateur**: L'application frontend utilise Vue.js, un framework JavaScript progressif qui permet la création d'interfaces utilisateur réactives et modulaires. L'architecture Single Page Application (SPA) offre une navigation fluide et une expérience utilisateur moderne sans rechargement de page.
- **Spécifications techniques du frontend:** Les caractéristiques techniques principales de l'interface utilisateur sont les suivantes :
  - *Framework JavaScript* : Vue.js pour la réactivité et la gestion des composants
  - *Framework CSS* : Tailwind CSS pour un design moderne et responsive
  - Architecture : Single Page Application (SPA)
  - *Gestion d'état* : Vuex ou Composition API pour la gestion des données globales
  - *Internationalisation* : Vue I18n pour la gestion des contenus statiques multilingues
  - *Routage* : Vue Router pour la navigation entre les vues
- **Système de gestion multilingue**: L'interface utilisateur implémente un système de gestion multilingue sophistiqué qui combine le contenu dynamique des données patrimoniales avec l'internationalisation des éléments statiques de l'interface.
  - Gestion du contenu dynamique multilingue
  - Bouton de sélection de langue intégré dans l'interface utilisateur
  - Options disponibles : Anglais, Arabe, Français
  - Changement instantané du contenu lors de la sélection d'une nouvelle langue
  - Internationalisation des éléments statiques
  - Le système utilise Vue I18n pour la gestion des éléments statiques de l'interface :
    Labels et boutons : Traduction automatique des éléments de navigation
    Messages système : Notifications et messages d'erreur multilingues
    Textes d'interface : Tous les textes fixes traduits selon la langue sélectionnée
    Formats régionaux : Adaptation des formats de date et de nombre selon la langue
- **Fonctionnalités de recherche avancée:** L'interface utilisateur implémente des fonctionnalités de recherche sophistiquées qui exploitent les capacités multilingues du système.
  - ✧ Barre de recherche intelligente :

Recherche simultanée dans toutes les langues disponibles
Auto-complétion basée sur les contenus patrimoniaux
Suggestions de recherche contextuelles
Mise en évidence des termes recherchés dans les résultats

- ✧ Filtres de recherche avancés :

Sélecteur de région avec interface graphique
Filtrage par catégorie
Sélecteur de période historique avec timeline interactive
Combinaison de filtres multiples pour des recherches précises

- ✧ Présentation des résultats :

Grille responsive adaptée aux différentes tailles d'écran
Cartes patrimoniaux avec aperçu du contenu multilingue
Pagination intelligente pour la gestion de grandes collections
Tri des résultats par pertinence, date, ou ordre alphabétique

- ✧ Détails patrimoniaux :

Vue détaillée avec toutes les informations multilingues
Affichage des images patrimoniales avec galerie intégrée
Liens vers les sources originales lorsque disponibles
Partage social et options d'export

2.5.4. Administration de la Plate-forme

- Le diagramme de séquences ci-dessous illustre les différentes fonctionnalités d'administration de la plate-forme encyclopédique pour la conservation du patrimoine culturel Algérien.

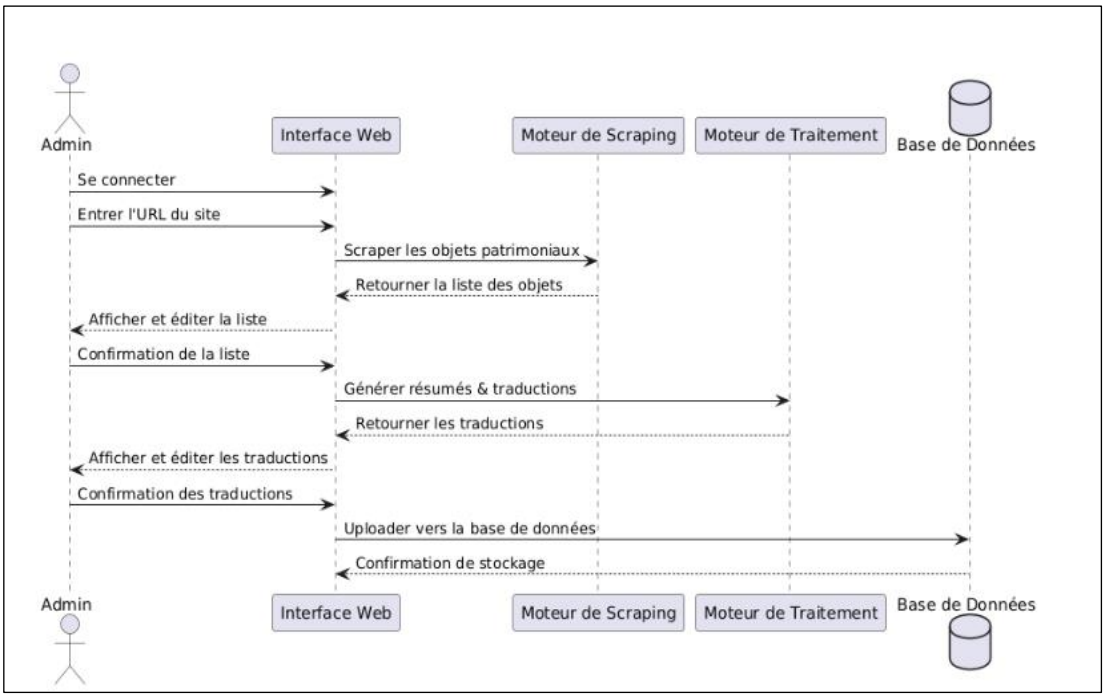

Figure 14: Diagramme de séquences de l'Administration de l Plate-forme.

3. Conclusion

Ce chapitre a permis de formaliser l'ensemble des composants qui constituent l'ossature technique de la plate-forme de conservation du patrimoine culturel algérien. L'approche de modélisation adoptée a mis en évidence une architecture orientée services, articulée autour de couches indépendantes mais interconnectées, chacune assumant une responsabilité précise : collecte, traitement, structuration, validation, et exposition des objets patrimoniaux. Le chapitre suivant est consacré au développement et à l'évaluation de la plateforme, où les composants modélisés précédemment sont implémentés, testés et validés à travers divers scénarios fonctionnels et techniques.

CHAPITRE III: TESTS ET DEVELOPPEMENT DE LA PLATEFORME

1. Introduction

Après avoir posé les fondements théoriques et techniques liés à l'extraction, au traitement sémantique et au stockage des données patrimoniales, ce chapitre aborde la phase pratique du projet. Il s'agit ici de détailler le processus de mise en œuvre de l'application développée, depuis le choix de l'environnement technologique jusqu'à l'évaluation de ses performances à travers un scénario d'usage réaliste.

Dans un premier temps, nous présentons l'environnement de développement utilis. Nous décrivons ensuite un scénario d'évaluation fonctionnel, conçu pour simuler une situation réelle d'extraction et d'organisation d'informations issues de sources en ligne. Enfin, une présentation détaillée de l'application développée est proposée.

2. Environnement de développement

Cet environnement repose sur deux composantes essentielles : l'infrastructure matérielle utilisée pour le développement et les tests, ainsi que les outils logiciels mobilisés pour assurer l'efficacité, la scalabilité et la robustesse du système.

2.1 Environnement matériel

Les caractéristiques techniques de la machine sont résumés dans le tableau suivantes :

Tableau 9: Spécifications Techniques du Matériel Utilisé.

Outil	Description
Modèle	MacBook Air (M1, 2020)
Processeur	Puce Apple M1 (8 cœurs, architecture ARM)
Mémoire vive (RAM)	8 Go
Système d'exploitation	macOS Ventura, version 13.2.1
Affichage	Écran Retina intégré de 13,3 pouces avec une résolution de 2560 × 1600
Stockage interne	245,11 Go SSD (avec environ 1,55 Go d'espace libre au moment de l'utilisation)

2.2 Environnement logiciel

L'environnement logiciel repose principalement sur des outils open source, organisés en plusieurs couches fonctionnelles :

2.2.1. Backend

Les outils utlisés pour leback-end sont:

- **Flask (Python) :** Flask est un micro-framework web écrit en Python, conçu pour être léger, flexible et simple à utiliser. Il permet de créer rapidement des applications web, des API REST ou des backends complets.Il repose sur Werkzeug (pour la gestion des requêtes HTTP) et Jinja2 (pour le rendu des templates HTML), et ne vient qu'avec les composants essentiels.

- **Base de données - MongoDB (NoSQL):** MongoDB est une base de données NoSQL orientée documents, conçue pour être flexible, scalable et performante. Elle permet de stocker les données sous forme de documents JSON-like (au format BSON), ce qui facilite la manipulation de structures complexes et évolutives. Elle repose sur un modèle sans schéma fixe, ce qui nous a offre une grande liberté dans la conception des données — et s'adapte facilement aux applications modernes nécessitant une forte évolutivité ou des traitements massifs.

- **Test des API - Postman[22] :** Postman est un outil graphique conçu pour tester, documenter et automatiser les API, en particulier les routes HTTP. Il permet d'envoyer facilement des requêtes GET, POST, PUT, DELETE vers un serveur, d'analyser les réponses et de vérifier le bon fonctionnement des points de terminaison d'une API. Grâce à son interface intuitive, Postman nous a facilité la phase de développement, de débogage et de validation des services web, tout en offrant des fonctionnalités avancées comme la gestion des environnements, les jeux de tests et la collaboration en équipe.

2.2.2 Frontend

Le frontend correspond à la partie visible de l'application, celle avec laquelle l'utilisateur interagit directement. Il englobe la conception des interfaces, l'affichage

[22] https://www.postman.com/

dynamique des données et la gestion des actions utilisateur. Pour offrir une expérience utilisateur fluide, responsive et esthétique, notre projet s'appuie sur des technologies web modernes et performantes, détaillées ci-dessous :

- **Langage HTML:**HTML (HyperText Markup Language) est le langage de balisage standard utilisé pour structurer le contenu des pages web. Il permet de définir les éléments de base d'une interface tels que les titres, paragraphes, boutons, images, liens ou formulaires. Conçu pour être simple et lisible, HTML fournit l'ossature du frontend, sur laquelle viennent se greffer les styles (via CSS) et les interactions dynamiques (via JavaScript). Dans notre projet, HTML nous a permis de construire la structure de chaque page affichée à l'utilisateur.

- **Framework CSS :** Tailwind CSS un framework CSS utilitaire qui permet de styliser rapidement les interfaces web à l'aide de classes prédéfinies directement dans le code HTML. Plutôt que d'écrire des feuilles de style personnalisées, on applique des classes pour gérer les couleurs, les espacements, les tailles, les polices, les bordures, etc.

  Ce système rend le développement plus rapide, cohérent et flexible. Dans notre projet, Tailwind CSS nous a permis de concevoir une interface responsive, moderne et harmonieuse, tout en gardant le code clair et structuré.

- **Framework JS :** Vue.js est un framework JavaScript progressif utilisé pour construire des interfaces utilisateur interactives. Léger, rapide et modulaire, il permet de créer des composants réutilisables, de gérer le DOM de manière réactive et de dynamiser l'affichage des données sans recharger la page. Grâce à sa simplicité et sa courbe d'apprentissage douce, Vue.js nous a permis d'intégrer des fonctionnalités dynamiques (comme les mises à jour en temps réel ou l'affichage conditionnel) tout en gardant une structure de code propre et évolutive.

- **Communication avec le backend - Axios:** Axios est une bibliothèque JavaScript permettant d'envoyer des requêtes HTTP depuis le frontend vers un serveur backend. Elle est utilisée pour interagir avec des API (REST), en effectuant des appels GET, POST, PUT, DELETE, etc., de manière simple et efficace. Axios facilite la récupération ou l'envoi de données sans recharger la page, ce qui améliore l'expérience utilisateur. Dans notre projet, Axios a joué un rôle essentiel

pour connecter le frontend réalisé avec Vue.js au backend développé avec Flask, en assurant la communication entre les deux couches.

2.2.3. Divers Outils

- **Système de contrôle de version - GitHub**: GitHub est une plateforme web basée sur Git [23] , un système de contrôle de version distribué. Elle permet aux développeurs de suivre l'historique des modifications de leur code, de travailler en collaboration et de gérer les différentes versions d'un projet de manière organisée. Dans notre projet, GitHub nous a permis de centraliser le code source, de sauvegarder chaque évolution, de gérer les branches pour les nouvelles fonctionnalités, et de faciliter la collaboration à distance. Grâce à ses fonctionnalités comme les commits, les pull requests et les issues, nous avons pu assurer un développement structuré, traçable et sécurisé.

- **Interface en ligne de commande - Terminal Bash:** Le Terminal Bash (Bourne Again SHell) est une interface en ligne de commande utilisée pour interagir directement avec le système d'exploitation à l'aide de commandes textuelles. Il permet d'exécuter rapidement des opérations comme la navigation dans les fichiers, l'exécution de scripts, l'installation de dépendances ou encore le lancement de serveurs. Dans notre projet, l'utilisation du terminal Bash a été essentielle pour lancer le backend, initialiser les dépôts Git, gérer les environnements virtuels Python, ou encore exécuter les commandes nécessaires au bon fonctionnement des différentes technologies. Il nous a offert un contrôle rapide, précis et automatisable sur l'ensemble du cycle de développement.

3. Scénarios d'évaluation

Un scénario d'évaluation est une mise en situation planifiée qui décrit les conditions, les objectifs, les outils, les critères et les modalités selon lesquels une solution, un système, un projet ou une compétence sera testé ou évalué. Il simule un

[23] Git est un système de contrôle de version distribué, utilisé pour suivre les modifications apportées aux fichiers, particulièrement dans le développement logiciel.

contexte réaliste permettant de mesurer la performance, la pertinence ou la conformité des résultats attendus.

3.1 Évaluation de la qualité des Résumés Automatique

Dans ce scénario, nous avons comparé les performances de trois modèles de résumé de texte préentraînés et fine-tunés sur des tâches de summarization :

- **T5-large** (Transformer by Google)
- **BART-large** (fine-tuné sur CNN/DailyMail)
- **Pegasus** (fine-tuné sur XSum)

Le processus d'évaluation se déroule comme suit :

a. Un jeu de données personnalisé a été utilisé, contenant des articles (paragraphes) et leurs résumés de référence.
b. Les trois modèles ont généré des résumés automatiquement pour 20 exemples extraits de ce jeu de données.
c. Les résumés générés ont été évalués à l'aide de la métrique ROUGE (ROUGE-1, ROUGE-2 et ROUGE-L), qui mesure les chevauchements de n-grammes et la structure linguistique avec le résumé de référence.
d. Les scores moyens de F-mesure pour chaque modèle ont été calculés et visualisés sous forme de graphique.

3.1.1. Résultats de l'Evaluation des Résumés Automatiques

Le tableau ci-dessous, décrit les résultats obtenus lors de l'évaluation des modèles de Résumé Automatique.

Tableau 10: Résultats d'Évaluation des Modèles de Résumé Automatique.

Modèle	**ROUGE-1 (%)**	**ROUGE-2 (%)**	**ROUGE-L (%)**
T5	32.14	8.36	22.03
BART	**33.27**	**10.45**	**23.25**
Pegasus	25.45	6.35	18.65

3.1.2. Discussion des Résultats

Nous pouvons clairement observé que le modèle *BART* a surpassé les deux autres modèles sur les trois sous-métriques ROUGE, avec un léger avantage sur *T5*. Cela

s'explique notamment par son fine-tuning sur des données similaires (CNN/DailyMail), ce qui le rend plus apte à générer des résumés cohérents et pertinents dans des styles journalistiques.

Par ailleurs, le modèle *T5-large* obtient des scores proches de ceux de *BART*, surtout en *ROUGE-1* et *ROUGE-L* (qualité de production des séquences de 1 et n-gram respectivement), montrant sa capacité à extraire les points clés, bien qu'il ait tendance à être un peu plus verbeux.

En revanche, Pegasus, bien que conçu spécifiquement pour la summarization, montre une performance plus faible ici, probablement en raison de la nature du dataset utilisé (le modèle étant fine-tuné sur XSum[24], un dataset très court et précis).

Il est ànoté que la métrique ROUGE-2 est plus exigeante, car elle évalue les bi-grammes. BART excelle ici avec 10.45%, indiquant une meilleure continuité sémantique entre les mots.

Bien que ce scénario nous a permis d'avoir une appréciation sur différents modèles de Résumé Automatique afin d'en choisir un, il a présenté les limites suivantes :

- L'évaluation a été réalisée sur 20 échantillons seulement, ce qui limite la généralisation.
- Les résumés ont été générés avec un nombre de beams faible (2) pour des raisons de performance, ce qui peut avoir un impact sur la qualité maximale potentielle.

3.1.3. Visualisation des Résultats d'Evaluation du Résumé Automatique

Le graphe généré permet une comparaison visuelle claire, où BART domine les trois catégories, suivi de T5, puis Pegasus :

[24] https://paperswithcode.com/sota/text-summarization-on-x-sum

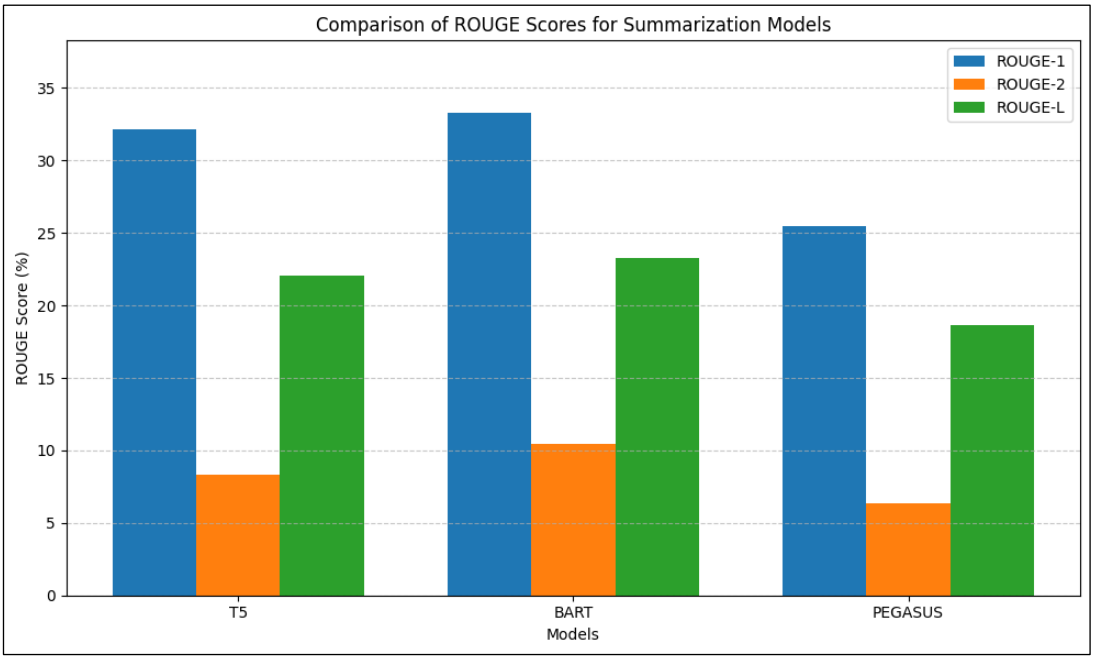

Figure 15: Comparaison des catégories ROUGE pour les Modèles de Résumé Automatique.

3.2. Evaluation de la qualité de Traduction

Les modèles de traduction (ou translation models) sont des systèmes informatiques conçus pour traduire automatiquement un texte d'une langue source vers une langue cible. Leur objectif principal est de produire une traduction qui respecte à la fois le sens original du texte et les règles linguistiques de la langue cible.

Dans cette évaluation, deux modèles de traduction automatique de l'anglais vers l'arabe ont été testés :

- **Helsinki-NLP/opus-mt-en-ar :** un modèle open-source basé sur MarianMT[25].
- **Google Translate :** le service de traduction de Google.

Les données utilisées proviennent d'un corpus contenant des paragraphes en anglais avec leurs traductions humaines correspondantes en arabe. L'évaluation s'est faite sur 50 échantillons, à l'aide de la métrique ROUGE (ROUGE-1, ROUGE-2 et ROUGE-L), qui mesure la similarité en termes de n-grammes entre les traductions automatiques et les traductions de référence.

[25] Un Framework de Traduciton siimilaire à BART (https://huggingface.co/docs/transformers/model_doc/marian)

Le pipeline d'évaluation est le suivant:

a. Traduction de chaque paragraphe par les deux systèmes.
b. Calcul des scores ROUGE pour chaque paire (traduction vs référence).
c. Agrégation des scores (moyennes, écarts-types).
d. Visualisation comparative.

3.2.1. Résultats de l'Evaluation de la Traduction Automatique

Les scores ROUGE moyens sont présentés dans le tableau suivant :

Tableau 11: Résultats d'Évaluation des Modèles de la Traduction Automatique.

Modèle	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
Helsinki-NLP	**0.3958**	**0.1848**	**0.3718**
Google Translate	0.2493	0.1013	0.2493

Il est à noté que les écarts-types ont également été calculés, les valeurs exactes sont visibles dans les barres d'erreur du graphique.

3.2.2. Discussion des Résultats

Le modèle open source Helsinki-NLP surpasse Google Translate dans ce test, sur l'ensemble des métriques :

- **ROUGE-1 :** reflétant le recouvrement de mots simples, Helsinki atteint ~0.40 contre ~0.25 pour Google.
- **ROUGE-2 :** mesurant les bigrammes, Helsinki dépasse 0.18 tandis que Google reste autour de 0.10.
- **ROUGE-L :** évaluant la similarité en structure de phrases (longest common subsequence), Helsinki atteint ~0.37.

Interprétations possibles :

- **Qualité du corpus de test :** le contenu du corpus et la nature des phrases peuvent avantager un modèle plus formel et aligné sur des données structurées comme Helsinki.

- **Variabilité de Google Translate :** étant un service généraliste, il peut générer des paraphrases ou des traductions plus libres, ce qui baisse les scores ROUGE malgré une qualité perçue souvent correcte.
- **Spécialisation du modèle Helsinki :** entraîné spécifiquement sur le domaine de la traduction anglais-arabe avec un vocabulaire aligné, Helsinki semble ici mieux adapté au test.

3.2.3. Visualisation des Résultats d'Evaluation de la Traduction Automatique

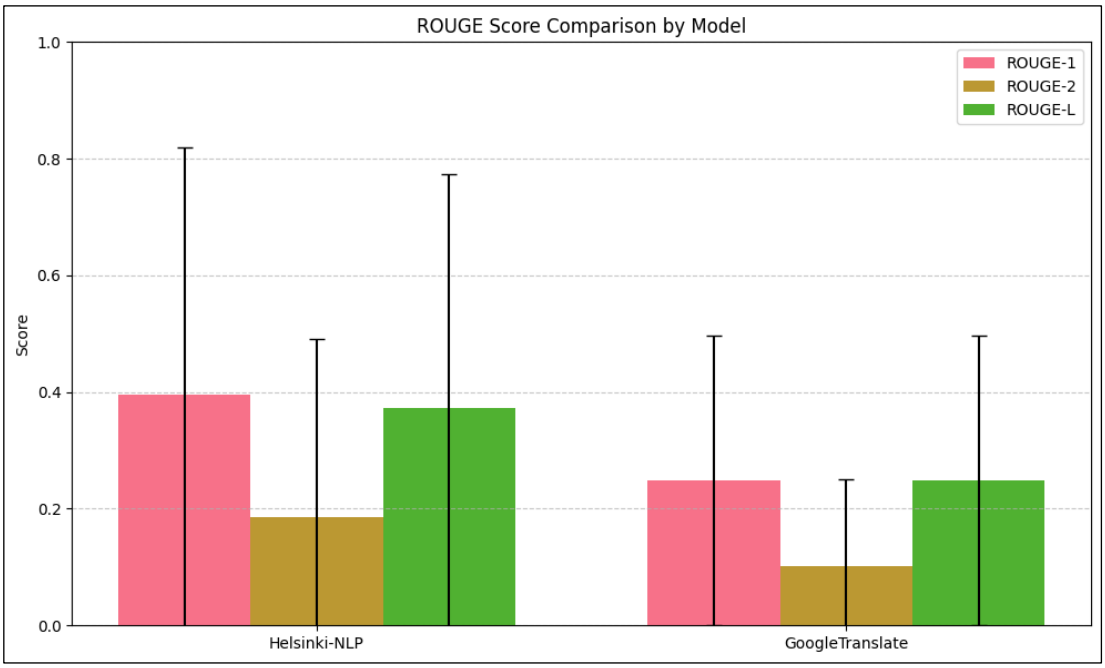

Figure 16: Comparaison des Métriques ROUGE Score Obtenues par les Modèles de Traduction Automatique.

3.3. Comparaison avec l'état del'art

Au vue des Travaux existants, notre plate-forme encyclopédique sur le patrimoine culturel algérien se distingue des travaux existants par:

- **Spécificité culturelle :** La plate-forme s'intéresse uniquement au patrimoine algérien dans sa diversité matérielle et immatérielle, contrairement aux approches plus génériques.

- **Pipeline intégré complet** : Chaîne de traitement cohérente allant de l'extraction par web scraping guidé par LLM jusqu'à la génération de résumés concis avec BART, offrant une solution de bout en bout.

- **Traitement bilingue :** Prise en compte simultanée de l'arabe et du français reflétant la richesse linguistique du contexte algérien.

- **Résumés contextuels :** Utilisation de BART pour générer des descriptions synthétiques qui préservent les nuances culturelles essentielles du patrimoine algérien.

- **Structure encyclopédique participative :** Contrairement aux approches purement institutionnelles, notre plate-forme intègre une dimension collaborative tout en maintenant une rigueur scientifique.

Tableau 12: Synthèse des points importants de notre approches.

Projet / Plateforme	Objectif principal	Techniques utilisées	Langue / Localisation	Interaction utilisateur	Limites
Notre Projet	Centraliser, structurer et valoriser le patrimoine culturel algérien (matériel et immatériel)	Web scraping avec LLM et structuration, résumé automatique avec BART, moteur de recherche intelligent	Anglais, Arabe, Français	Interface web interactive, moteur de recherche, affichage visuel et résumé généré	Manque de sources uniformes, défis liés à la collecte de données fiables

4. Présentation de la Plate-forme Turathna

Dans cette section,nous allons présenter la plate-forme développée dans le cadre de la conservation du patrimoine culturel Algérien.

4.1. Sitemap schéma

La figure ci-dessous représente l'arborescence des pages de notre plate-forme.

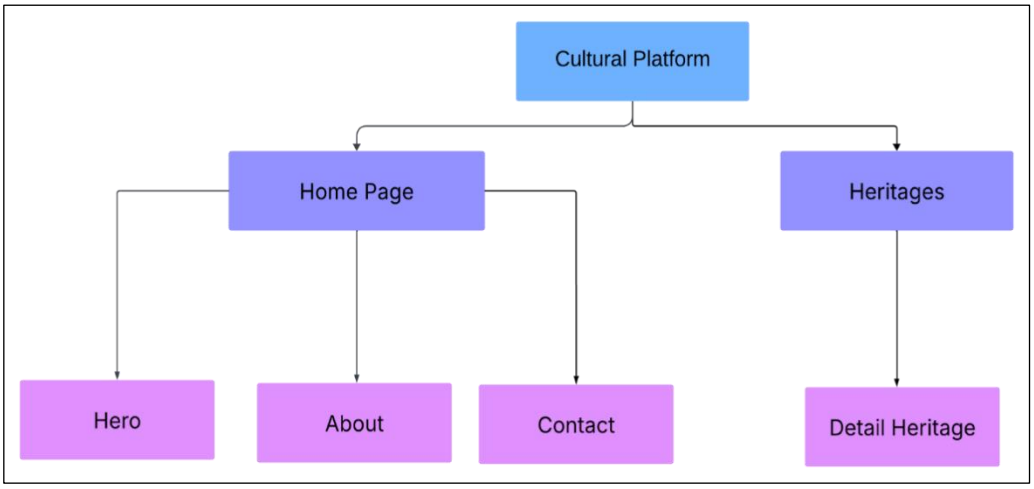

Figure 17: Schéma d'accessibilité de la Plate-forme.

Cette Arborescence contient plusieures pages:

- **Cultural Platform:** C'est la structure principale de la plateforme Turathna, qui représente l'écosystème global du site web. Elle regroupe l'ensemble des pages et fonctionnalités proposées aux utilisateurs, et sert de base à l'architecture de navigation.
- **Home Page:** Il s'agit de la page d'accueil de la plate-forme, accessible dès l'arrivée de l'utilisateur. Elle joue un rôle crucial d'introduction en présentant la mission de Turathna, une vue globale de son contenu, et des liens directs vers les sections clés. Elle est conçue pour être attrayante, informative et responsive.
- **Hero:** La section « Hero » est la partie supérieure de la Home Page, qui capte immédiatement l'attention de l'utilisateur. Elle contient généralement une image forte liée au patrimoine culturel algérien, un slogan accrocheur, et un bouton d'appel à l'action. Elle a un rôle marketing et informatif à la fois.

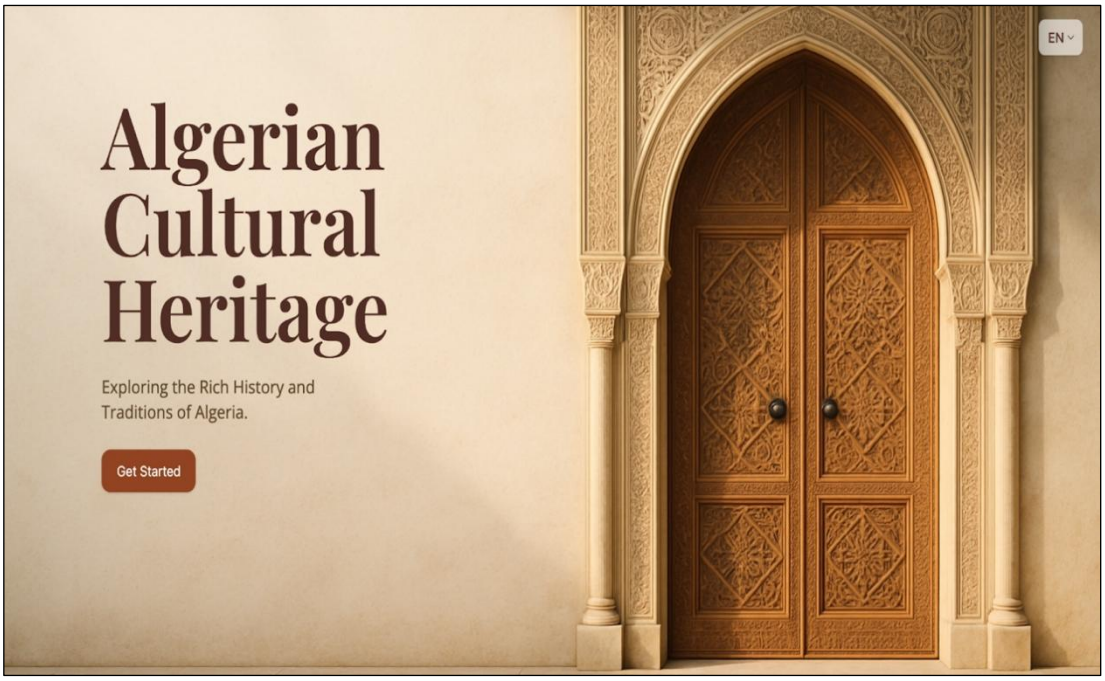

Figure 18: Interface "Hero" de la plate-forme Turathna.

- **About:** La section « About » présente les objectifs, la vision et la motivation derrière Turathna. Elle explique le contexte de création de la plateforme, ses sources, ses données patrimoniales, et l'importance de la valorisation du patrimoine culturel.

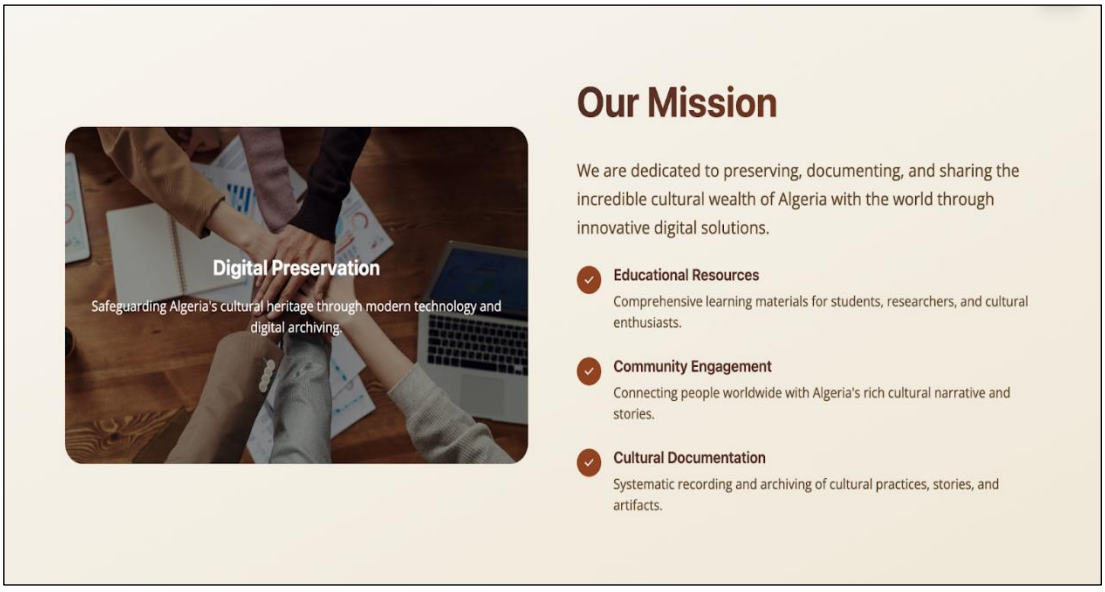

Figure 19: Interface "About" de la plate-forme Turathna.

- **Contact :** La section « Contact » permet aux utilisateurs de communiquer avec les administrateurs de la plateforme. Elle inclut un formulaire de contact et des liens vers les réseaux sociaux. Cette section est essentielle pour assurer un lien interactif avec la communauté.

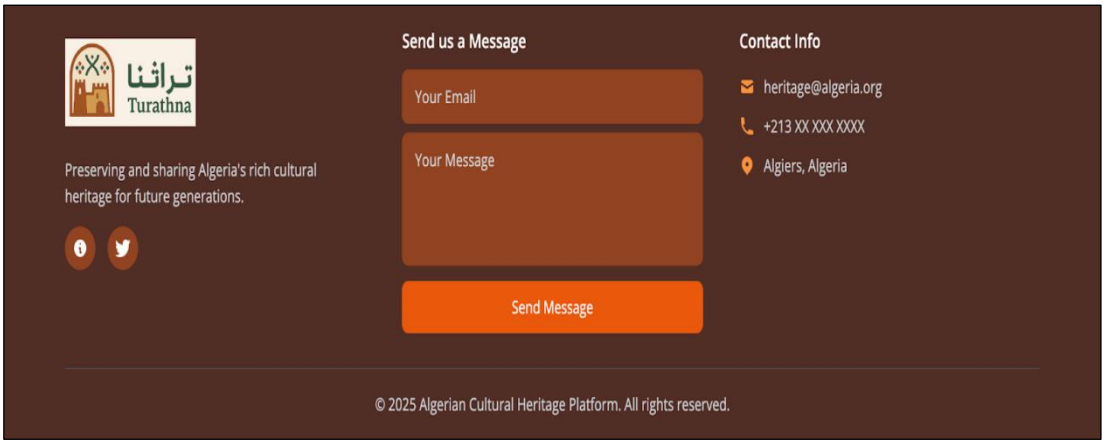

Figure 20: Interface de "Contact" de laplate-formeTurathna.

- **Heritages:** La page « Heritages » regroupe tous les éléments patrimoniaux recensés dans la plateforme. Elle affiche une grille des contenus patrimoniaux, accompagnés de filtres et d'un moteur de recherche. Cette page constitue le cœur du contenu de Turathna.

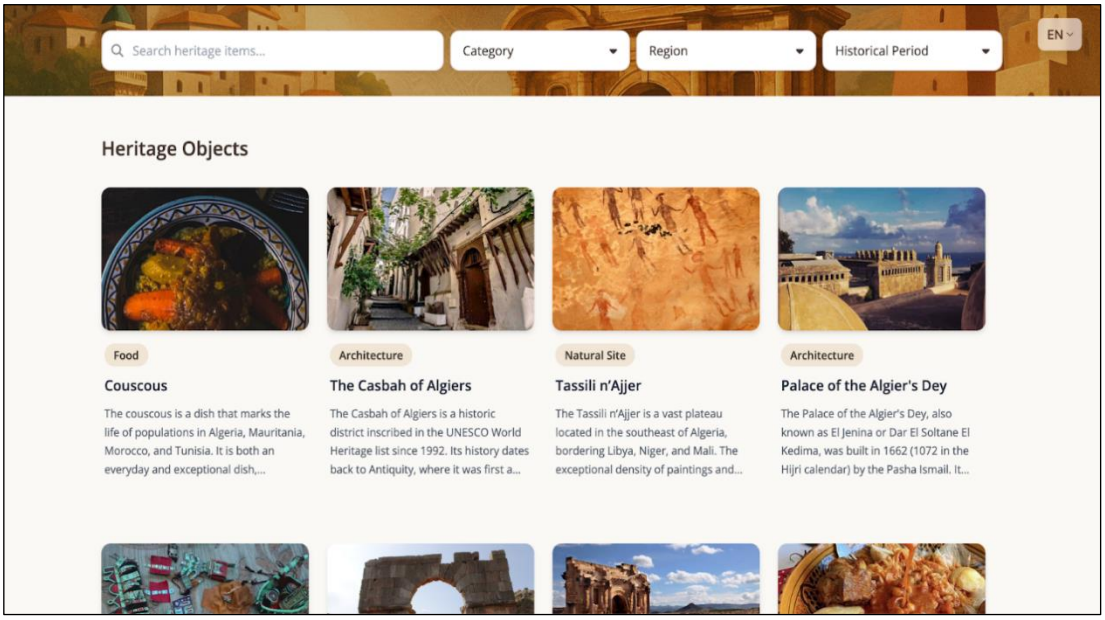

Figure 21: Interface "Heritages" de laplate-forme Turathna.

- **Detail Heritage:** Il s'agit de la page de détail dédiée à chaque élément patrimonial. Elle fournit une description complète (texte, image, géolocalisation, etc.), avec des métadonnées, des catégories, et éventuellement des suggestions d'éléments similaires. Elle joue un rôle de fiche descriptive approfondie pour chaque patrimoine.

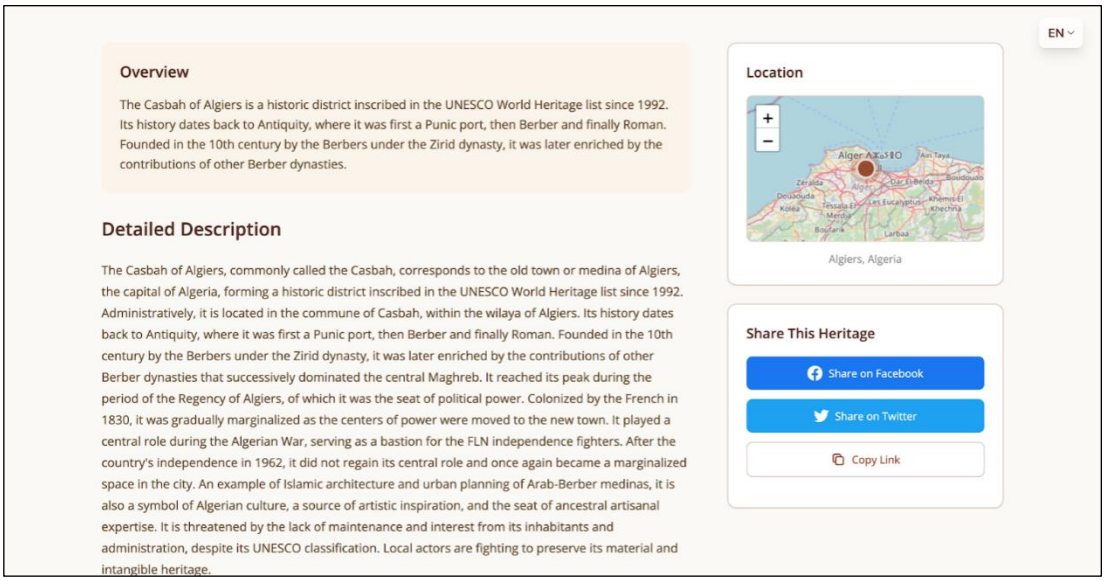

Figure 22: Interface "Heritages Detail" de la plate-forme Turathna.

4.2. Volé Administration de la Plate-forme Turathna

Dans cette section, nous allons présenter les différentes pages constituant le volé administration de la plateforme Turathna.

4.2.1. Schéma d'accessibilité des Interfaces d'Administrtion

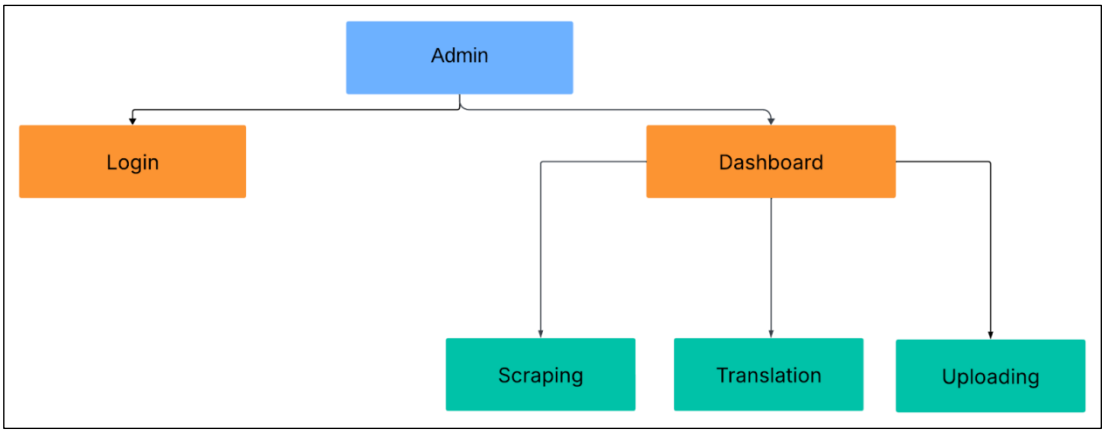

Figure 23:Schéma d'Accessibilité des Interfaces d'Administration de Turathna.

Le volé administration est constitué des pages suivantes:

- **Admin:** C'est la structure principale du back-office réservé aux administrateurs de la plateforme Turathna. Elle regroupe les interfaces nécessaires à la gestion des contenus patrimoniaux, l'automatisation des tâches (scraping, traduction), ainsi que la supervision des données. L'accès y est restreint via authentification.
- **Login:** Page d'authentification sécurisée permettant aux administrateurs d'accéder à l'interface d'administration. Elle vérifie les identifiants et génère une session pour l'utilisateur. Elle est connectée à un système de gestion de sessions (Flask session).

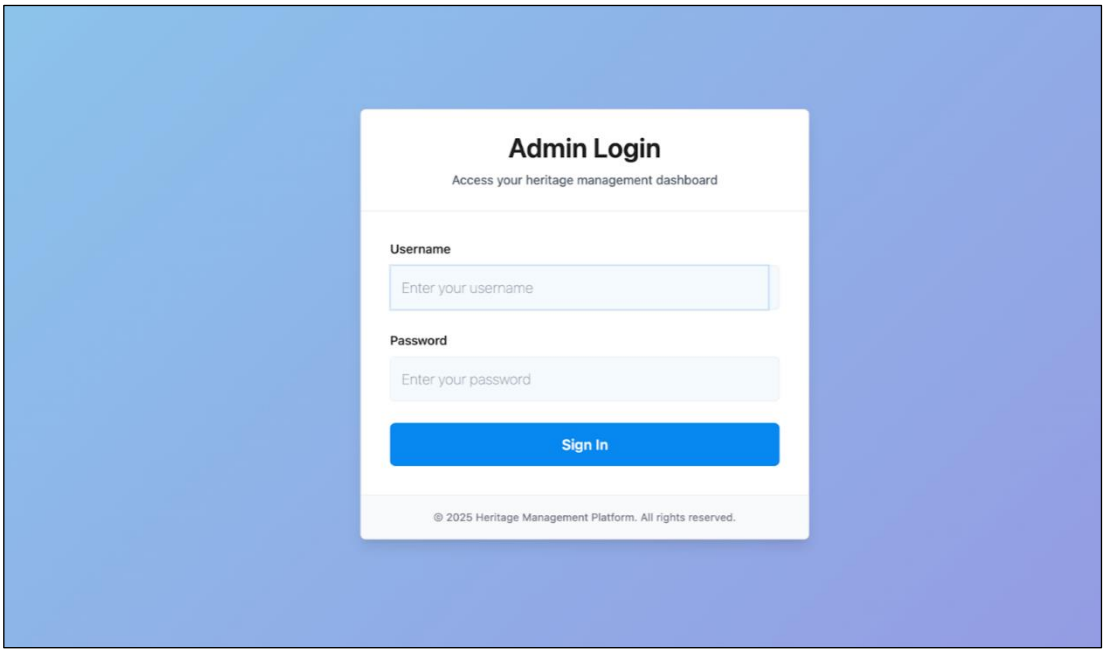

Figure 24: Interface de "Login" de la plate-forme Turathna.

- **Dashboard:** Il s'agit de l'espace central après connexion. Il fournit une vue d'ensemble des activités de la plateforme, des statistiques clés, et des raccourcis vers les modules fonctionnels (scraping, traduction, upload). Il constitue le point d'entrée vers les outils de gestion.
- **Scraping:** Ce module permet d'extraire automatiquement des données culturelles depuis des sources en ligne (sites web, bases de données publiques, etc.). Il utilise des scripts (ex. en Python) qui collectent, nettoient et structurent les données avant leur intégration dans la base. L'interface permet de lancer, surveiller et consulter les résultats des tâches de scraping.

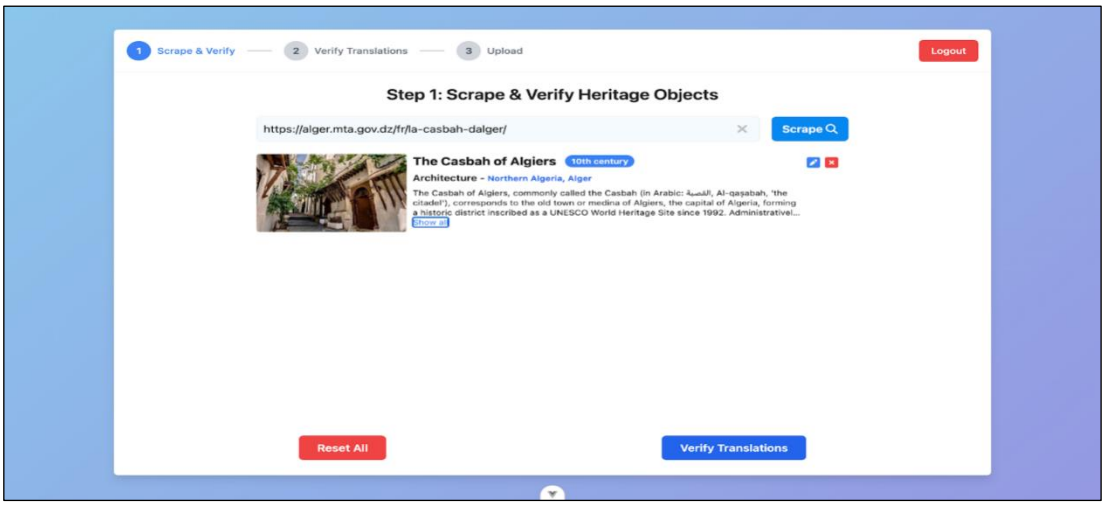

Figure 25: Interface "Scraping" de la plate-forme Turathna.

- **Translation:** Module permettant la traduction automatique des contenus patrimoniaux. Il peut être basé sur des modèles pré-entraînés (comme MarianMT ou mBART) pour générer les versions multilingues des fiches. Ce module peut inclure un outil de validation manuelle pour réviser les traductions automatiques.

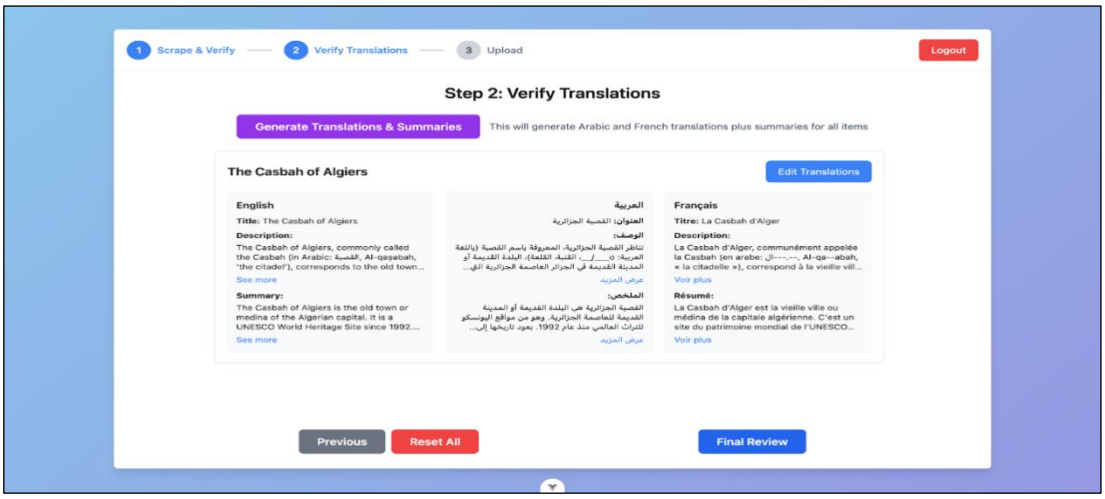

Figure 26: Interface "Translations" de la plate-forme Turathna.

- **Uploading:** Module permettant aux administrateurs de téléverser manuellement de nouveaux contenus : fiches patrimoniales, images, vidéos ou documents. Il comprend des champs de métadonnées, un gestionnaire de fichiers, et des validations côté client et serveur.

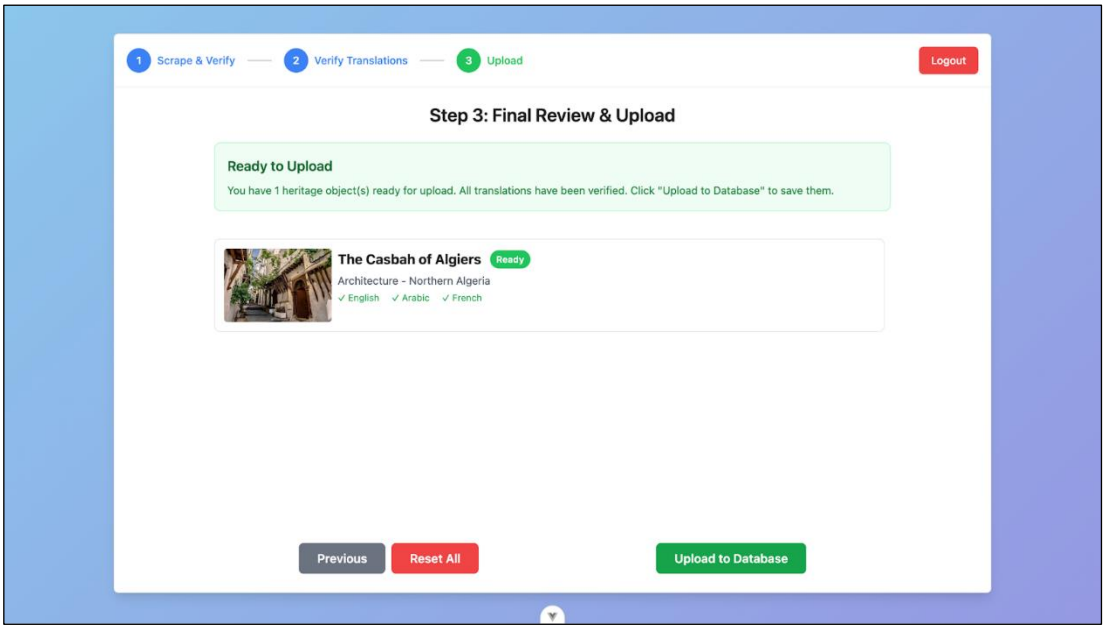

Figure 27: Interface "Upload" de la plate-forme Turathna.

5. Conclusion

Ce chapitre a concrétisé les aspects théoriques et techniques du projet en détaillant le processus de développement de la plate-forme de conservation du patrimoine culturel algérien. Il a présenté l'environnement de développement adopté, la structure générale de l'application, ainsi que les outils et technologies mobilisés pour assurer le traitement sémantique, la gestion multilingue et la structuration des données patrimoniales.

À travers un scénario d'usage réaliste, le système a été évalué sur sa capacité à extraire, organiser et restituer des objets patrimoniaux issus de sources en ligne. Les résultats obtenus démontrent la viabilité de la solution proposée, tout en soulignant son potentiel pour une exploitation future à grande échelle. Ces premiers tests constituent une base solide en vue des améliorations fonctionnelles et des intégrations envisagées dans les perspectives du projet.

CONCLUSION GENERALE

Conclusion Générale

Ce mémoire a présenté la conception, le développement et l'évaluation d'une plateforme intelligente dédiée à la valorisation du patrimoine culturel algérien à partir de sources web. En l'absence d'une solution existante équivalente dans le contexte national, ce travail propose un pipeline complet et original intégrant des techniques de web scraping, de traitement automatique du langage, de structuration de données multilingues, de traduction et de résumé automatique. L'architecture modulaire mise en place repose sur des outils avancés tels que Crawl4AI pour l'exploration web, DeepSeek-R1 pour l'extraction structurée, Helsinki-NLP pour la traduction, et BART pour la génération de résumés. Le système a démontré sa capacité à transformer des contenus non structurés en objets patrimoniaux cohérents, enrichis et exploitables dans plusieurs langues, répondant ainsi à un besoin urgent de numérisation et d'accessibilité culturelle.

L'évaluation fonctionnelle menée dans un scénario réaliste a confirmé la pertinence de l'approche adoptée et la qualité des données produites. En plus de son apport méthodologique et technique, ce projet pose les bases d'une plateforme évolutive, adaptée à la diversité du patrimoine algérien, tant matériel qu'immatériel. Il ouvre la voie à plusieurs perspectives de recherche et d'amélioration, telles que:

- L'intégration de mécanismes d'interaction utilisateur,
- La mise en place d'un moteur de recommandation basé sur les préférences culturelles, ou encore
- L'exploitation de modèles plus spécialisés pour des thématiques patrimoniales ciblées.

À terme, cette plate-forme pourrait s'inscrire dans une stratégie nationale de conservation numérique, en facilitant l'accès, la transmission et la valorisation du patrimoine auprès d'un large public.

Références Bibliographiques

[1] Commission économique des Nations unies pour l'Afrique (2023, juin). La révolution des données en Afrique du Nord : Mettre les données au service de la transformation structurelle (Rapport).

[2] Dang Tuan Nguyen. Université de Caen, Année : 2006. Extraction d'information à partir de documents Web multilingues p

[3] G. Druey, "Étude, conception, collecte, curation et évaluation d'un scraping de sites web liés au transport maritime pour améliorer la prédiction du fret de matières premières," Master en Sciences de l'information, Haute école de gestion de Genève (HEG-GE), Genève, 2020. doi: 10.5281/zenodo.3980515.

[4] H. Rezzag et S. Bouaroura, "Extracteur de données web sur la comparaison de prix," Mémoire de MASTER II RECHERCHE en Informatiques, Département de Mathématiques et informatique, Université de Ghardaïa, Ghardaïa, 2020

[5] BADRACHIR, "EXTRACTION DE DONNÉES À PARTIR DU WEB," Mémoire de Maîtrise en Informatique, Université du Québec à Montréal, Montréal, 2013.

[6] Mitchell, Rayen. Web Scraping with Python: Data Extraction from the Modern Web . O'Reilly Media.

[7] M. I. Bourhadoun, "Impact des méthodes analytiques dans le contexte des données massives," Mémoire de Fin d'études Master, Département d'Informatique, Université de 8 Mai 1945 – Guelma, Guelma, 2020.

[8] Le Web Crawling : Qu'est-ce que l'indexation des pages web ? Site web : https://datascientest.com/web-crawling-tout-savoir

[9] Semji. (n.d.). Crawler un site web : méthodes et techniques. Semji. Récupéré de https://semji.com/fr/guide/crawler-un-site-web-methodes-et-techniques/

[10] O. S. Benmahammed et C. Barouda, « Un résumé automatique du texte basé Sur une méthode abstractive », Mémoire de Master, Département de Mathématiques et informatique, Université Abdelhamid Ibn Badis - Mostaganem, Mostaganem, Algérie, 2022.

[11] H. P. Luhn, "The Automatic Creation of Literature Abstracts," IBM Journal of Research and Development, vol. 2, no. 2, pp. 159–165, 1958.

[12] I. Mani, Automatic Summarization, Amsterdam: John Benjamins Publishing Company, 2001.

[13] A. Vaswani et al., "Attention Is All You Need," in Advances in Neural Information Processing Systems (NeurIPS), 2017, pp. 5998–6008.

[14] T. Brown et al., "Language Models are Few-Shot Learners," in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, 2020, pp. 1877–1901.

[15] J. Zhang et al., "PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization," in Proc. Int. Conf. on Machine Learning (ICML), 2020, pp. 11328–11339.

[16] Y. Zhang, H. Jin, D. Meng, J. Wang, et J. Tan, « A Comprehensive Survey on Automatic Text Summarization with Exploration of LLM-Based Methods », ArXiv Prepr., arXiv:2407.03960, 2024.

[17] M. H. T. de Boer, Q. Smit, M. van Bekkum, A. Meyer-Vitali, et T. Schmid, « Design Patterns for LLM-based Neuro-Symbolic Systems », Neurosymbolic Artif. Intell., vol. 0, no. 0, pp. 1–19, 2024

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," arXiv Preprint arXiv:1706.03762, Jun. 2017.