RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

ÉCOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département : Génie Industriel
Entreprise : Monad

Mémoire de projet de fin d'études

Pour l'obtention du diplôme d'ingénieur d'état en Génie Industriel
Option : Data Science & Intelligence Artificielle

Développement d'un Assistant Juridique Artificiel

Mehdi AMOR OUAHMED

Sous la direction de **Dr. Samia BELDJOUDI**, ENP
& **Dr. Adel AIT-HAMLAT**, Monad

Présenté et soutenu publiquement le (03/07/2024)

Composition du jury :

Président :	Dr. Iskander ZOUAGHI	MCA	ENP
Examinateur :	Dr. Oussama ARKI	MCA	ENP
Promotrice :	Dr. Samia BELDJOUDI	MCA	ENP

ENP 2024

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

ÉCOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département : Génie Industriel
Entreprise : Monad

Mémoire de projet de fin d'études

Pour l'obtention du diplôme d'ingénieur d'état en Génie Industriel
Option : Data Science & Intelligence Artificielle

Développement d'un Assistant Juridique Artificiel

Mehdi AMOR OUAHMED

Sous la direction de **Dr. Samia BELDJOUDI**, ENP
& **Dr. Adel AIT-HAMLAT**, Monad

Présenté et soutenu publiquement le (03/07/2024)

Composition du jury :

Président :	Dr. Iskander ZOUAGHI	MCA	ENP
Examinateur :	Dr. Oussama ARKI	MCA	ENP
Promotrice :	Dr. Samia BELDJOUDI	MCA	ENP

ENP 2024

Dédicaces

A la personne la plus importante dans ma vie, ma source de motivation, celle qui a toujours cru en moi et qui a toujours été à mes cotés, ***ma MAMAN****, la femme la plus courageuse au monde. Je t'aime Maman !*

En mémoire de mon cher père, bien que je n'aie pas eu la chance de profiter assez de ta présence, ce dont je suis le plus sûr, c'est que tu as toujours cru en moi. ***PAPA****, ton petit génie Mehdi va soutenir son ingéniorat, j'espère que je t'ai rendu fier !*

A mes deux sœurs : ***Mina****, ma deuxième mère, qui m'a toujours soutenu et* ***Majda*** *ma source d'inspiration.*

A mon frère ***Abderrahmene****, mon bras droit et la lumière qui guidait mon chemin.*

A mon beau-frère ***Amine****, ma chère nièce adorée* ***Yasmine****, et le tout petit nouveau membre de la famille* ***Mohammed Anis****. Je vous souhaite toute la réussite du monde.*

A mon grand-père ***Papa El Hadj****, qu'ALLAH le préserve. En mémoire de mes grands-mères* ***Mayma*** *et* ***Ma Hlima****, ainsi que de mon grand-père* ***Ali****.*

A tous mes amis et mes camarades de classe avec qui j'ai partagé des moments extraordinaires, merci pour ces 3 belles années. Je vous souhaite un avenir très prometteur, plein de succès.

Enfin, à tous mes professeurs du département qui nous ont transmis leurs savoir-faire et surtout à ceux qui ont cru en moi.

Mehdi.

Remerciment

Qu'il me soit permis de remercier et d'exprimer ma profonde gratitude en premier lieu à Dieu le tout-puissant de m'accorder sa bénédiction, de m'avoir donné de la force, de la volonté et de la patience pour mener à bien ce travail.

Ensuite, je voudrais remercier chaleureusement Dr Samia Beldjoudi, mon encadrante de l'école. Son aide précieuse, ses conseils éclairés et son soutien indéfectible ont été des atouts indispensables à la réalisation de ce projet.

Je tiens également à exprimer ma profonde gratitude à mon encadrant en entreprise chez Monad, Dr. Adel Ait-Hamlat. Sa guidance, son expertise et son soutien ont été inestimables tout au long de ce projet. Son engagement constant a été une source d'inspiration et a grandement contribué à la réussite de ce travail.

Un merci particulier à l'ensemble des personnes que j'ai rancontré à Monad, qui m'ont accueilli et m'ont permis de réaliser ce projet dans des conditions optimales. Je tiens à souligner l'apport de M. Malik Dahmani, Mme Nawel Sakhraoui, M. Amine Zidelmal dont la contribution a été d'une grande aide pour ce projet.

Je souhaite exprimer ma gratitude anticipée au jury qui évaluera ce travail. Je tiens à les remercier pour le temps et l'expertise qu'ils consacreront à la lecture et à l'évaluation de ce projet. Leur contribution à l'amélioration et à l'évaluation de la qualité de mon travail est très appréciée.

Je suis également reconnaissant envers le corps professionnel de l'École Nationale Polytechnique, département du génie industriel, pour la formation de qualité et l'encadrement professionnel qu'ils m'ont fournis.

ملخص

يهدف هذا المشروع إلى معالجة تحدي الوصول إلى النصوص التشريعية في الجزائر باستخدام تقنيات الذكاء الاصطناعي. الإطار التشريعي الجزائري ديناميكي، مع تحديثات وتغيرات متكررة للنصوص التشريعية، مما يعقد مهمة رجال القانون والمواطنين والشركات في الوصول إلى معلومات قانونية دقيقة ومحدثة. يركز هذا المشروع على تطوير مساعد قانوني ذكي قادر على الإجابة على استفسارات المستخدمين حول التشريعات الجزائرية.

تتضمن المنهجية المعتمدة عدة خطوات: استخراج النص الخام من الجرائد الرسمية الجزائرية، تحديد وتصنيف النصوص التشريعية، بناء رسم بياني للمعرفة، تحويل مقاطع النص إلى تمثيلات رقمية، واستخدام تقنيات البحث المتقدمة وتوليد النصوص لتقديم إجابات دقيقة وسياقية. يستخدم هذا المشروع تقنية الاسترجاع المدعوم بالتوليد لدمج تقنيتي استرجاع المعلومات مع توليد النصوص، مما يضمن إجابات مناسبة ومحدثة.

النتائج المتوقعة تشمل مساعد قانوني كفؤ قادر على فهم تعقيد النصوص التشريعية الجزائرية، وتوفير وصول سريع ومبسط إلى المعلومات القانونية. يهدف هذا المشروع إلى تحسين كفاءة رجال القانون والمواطنين من خلال تسهيل الوصول إلى المعلومات القانونية المعقدة وتقليل الوقت اللازم لفهم النصوص التشريعية. من خلال دمج تقنيات الذكاء الاصطناعي هذه، يساهم المشروع في تحديث وتنظيم العمليات القانونية في الجزائر.

الكلمات المفتاحية: الذكاء الاصطناعي - مساعد قانوني - التشريع الجزائري - استخراج النصوص - رسم بياني للمعرفة - استرجاع المعلومات - توليد النصوص - تقنية الاسترجاع المدعوم بالتوليد

Abstract

This project aims to address the challenge of accessing and understanding legislative texts in Algeria using artificial intelligence techniques. The Algerian legislative framework is dynamic, with frequent updates and modifications to legal texts, complicating the task for legal professionals, citizens, and businesses to access precise and up-to-date legal information. This project focuses on developing an intelligent legal assistant capable of responding to users' queries concerning Algerian legislation.

The methodology adopted involves several steps : extracting raw text from Algerian official journals, detecting and classifying legal texts, constructing a knowledge graph, vectorizing text segments, and using advanced search and text generation techniques to provide precise and contextual responses. This project employs the "Retrieval-Augmented Generation" (RAG) technique to combine information retrieval with text generation, ensuring relevant and up-to-date responses.

The expected results include a highly efficient legal assistant capable of navigating the complexity of Algerian legislative texts, providing simplified and quick access to legal information. This project aims to enhance the efficiency of legal professionals and citizens by facilitating access to complex legal information and reducing the time required to find and understand legal texts. By integrating these artificial intelligence technologies, the project contributes to the modernization and optimization of legal processes in Algeria.

Keywords : Artificial Intelligence - Legal Assistant - Algerian Legislation - Text Extraction - Knowledge Graph - Information Retrieval - Text Generation - RAG Technology

---

Résumé

Ce projet vise à relever le défi de l'accès et de la compréhension des textes législatifs en Algérie en utilisant des techniques d'intelligence artificielle. Le cadre législatif algérien est dynamique, avec de fréquentes mises à jour et modifications des textes juridiques, ce qui complique la tâche des professionnels du droit, des citoyens et des entreprises pour accéder à des informations juridiques précises et à jour. Ce projet se concentre sur le développement d'un assistant juridique intelligent capable de répondre aux requêtes des utilisateurs concernant la législation en Algérie.

La méthodologie adoptée repose sur plusieurs étapes : l'extraction de texte brut depuis les journaux officiels algériens, la détection et la classification des textes légaux, la construction d'un graphe de connaissance, la vectorisation des segments de texte, et l'utilisation de techniques avancées de recherche et de génération de texte pour fournir des réponses précises et contextuelles. Ce projet utilise la technique "Retrieval-Augmented Generation" (RAG) pour combiner la recherche d'information avec la génération de texte, garantissant ainsi des réponses pertinentes et à jour.

Les résultats attendus incluent un assistant juridique performant capable de naviguer dans la complexité des textes législatifs algériens, offrant un accès simplifié et rapide à l'information juridique. Ce projet vise à améliorer l'efficacité des professionnels du droit et des citoyens, en facilitant l'accès à des informations juridiques complexes et en réduisant le temps nécessaire pour trouver et comprendre les textes de loi. En intégrant ces technologies d'intelligence artificielle, le projet contribue à la modernisation et à l'optimisation des processus juridiques en Algérie.

Mots clés : Intelligence Artificielle - Assistant Légal - Législation Algérienne - Extraction de Texte - Graphe de Connaissance - Recherche d'Information - Generation de Texte - Technologie RAG

Table des matières

Liste des tableaux

Table des figures

Liste des acronymes

Introduction générale **14**

I Contexte et Cadre Général 16

1 Présentation de l'Organisme d'Accueil et du Contexte Juridique **17**

1.1 Introduction 17

1.2 L'Organisme d'Accueil : Monad 17

1.3 Le droit 18

1.4 Conclusion 19

2 Problématique et Description du Projet **20**

2.1 Introduction 20

2.2 Contexte et Enjeux 20

2.3 Justification du Projet 21

2.4 Objectifs du Projet 21

2.5 Méthodologie et Approche Adoptée 22

2.6 Résultats Attendus 23

2.7 Conclusion 23

II Etat De l'Art 24

3 Extraction et prétraitement de données textuelles **25**

3.1 Introdution . . . 25
3.2 Extraction de texte depuis les fichiers PDF . . . 25
3.3 Analyse de la mise en page des documents . . . 27
3.4 Expressions Régulières . . . 30
3.5 Conclusion . . . 31

4 Techniques d'apprentissage automatique et de traitement automatique du langage naturel 32

4.1 Introduction . . . 32
4.2 Apprentissage Automatique . . . 33
4.3 Apprentissage Profond . . . 33
4.4 Traitement automatique du langage naturel . . . 34
4.5 Réseaux de Neurones Récurrents . . . 37
4.6 Transformers . . . 39
4.6.1 Mécanisme d'attention . . . 39
4.6.2 Architecture . . . 39
4.6.3 Applications . . . 41
4.6.4 Évolution des Transformers . . . 41
4.7 Large Language Models . . . 41
4.7.1 Contexte . . . 42
4.7.2 Evolution et Impact . . . 42
4.7.3 Applications . . . 42
4.7.4 Limites . . . 43
4.8 Prompt Engineering . . . 44
4.9 Retrieval Augmented Generation . . . 46
4.10 Conclusion . . . 49

III Conception de la Solution 51

5 Extraction et Structuration des Données 53

5.1 Introduction . . . 53
5.2 Algorithme d'extraction . . . 53
5.3 L'utilisation des Expressions Régulières pour Structurer les Publications . . . 58

5.4 Construction du graphe 60
5.5 Conclusion 62

6 Développement du Chatbot Juridique : Workflow et Technologies 63
6.1 Introduction 63
6.2 Vue d'Ensemble du Workflow 63
6.3 Vectorisation des Données 65
6.4 Traitement des Requêtes Utilisateur 67
6.5 Recherche et Récupération d'Information 70
6.6 Génération de Réponses par le Modèle de Langage 71
6.7 Adaptabilité et Gestion des Connaissances 73
6.8 Conclusion 74

IV Mise en œuvre et Résultats 75

7 Mise en œuvre pratique du système 76
7.1 Introduction 76
7.2 Implémentation du système 76
7.3 Études de cas et résultats préliminaires 78
7.4 Conclusion 81

8 Évaluation et Validation 82
8.1 Introduction 82
8.2 Méthodes d'évaluation 82
8.3 Résultats 84
8.4 Conclusion 90

V Conclusion et Perspectives 91

Bibliographie 95

ANNEXE 99

Liste des tableaux

5.1 Publications extraites 59

8.1 Problèmes d'extraction identifiés 84
8.2 Densité du Graphe 85
8.3 Résumé des nœuds les plus importants 87
8.4 Tableau détaillé des communautés choisies 88
8.5 Répartition des rangs des réponses correctes 89

Table des figures

1.1 Logo de Monad 17

3.1 Exemple illustrant une analyse de la mise en page [1] 27

3.2 Exemple illustrant la binarisation 29

3.3 Exemple illustrant les résultats de la dilatation et de l'érosion 30

4.1 Représentation qui illustre la relation entre : Intelligence Artificielle, Machine Learning, et Deep Learning 33

4.2 Exemple de tokenization 35

4.3 Architecture des RNNs [2] 38

4.4 Différence entre RNN et LSTM [3] 38

4.5 Architecture des Transformers [4] 40

4.6 Diagramme illustrant l'approche suivie. 52

5.1 Exemple illustrant le fonctionnement de l'algorithme. 54

5.2 Exemple illustrant la gestion des cellules implicites. 57

5.3 Exemple illustrant le succès de l'approche adoptée pour gérer les implicit rows. . 57

5.4 Exemple de publication légale structurée par expressions régulières. 59

5.5 Un schéma qui illustre la structure du graphe 60

5.6 Visualisation graphique d'une partie du graphe des connaissances juridiques. Les nœuds correspondent aux publications légales présentées dans le tableau 5.1, aux articles et aux institutions selon les couleurs présentées dans le coin inféreur droit de la figure. Les arêtes représentent les différentes relations entre les nœuds. . . . 62

6.1 Diagramme illustrant le workflow du chatbot. 64

6.2 Diagramme illustrant le fonctionnement du reformulateur. 67

7.1 Architecture du système. 77

7.2 Capture d'écran illustrant la réponse de l'assistant 79

7.3 Capture d'écran illustrant la réponse de l'assistant 80

7.4 Capture d'écran illustrant la réponse de l'assistant 81

8.1 Histogramme des degrés 86

8.2 Représentations des communautés choisies 88

Liste des acronymes

- **PDF** : Portable Document Format
- **OCR** : Optical Character Recognition
- **DLA** : Document Layout Analysis
- **Regex** : Regular Expressions
- **ML** : Machine Learning
- **IA** : Intelligence Artificielle
- **NLP** : Natural Language Processing
- **ANN** : Artificial Neural Network
- **NER** : Named Entity Recognition
- **BOW** : Bag Of Words
- **TF-IDF** : Term Frequency-Inverse Document Frequency
- **RNN** : Recurrent Neural Network
- **LSTM** : Long Short Term Memory
- **BERT** : Bidirectional Encoder Representations from Transformers
- **GPT** : Generative Pretrained Transformer
- **LLM** : Large Language Models
- **SLM** : Statistic Language Models
- **NLM** : Neural Language Models
- **PLM** : Pre-trained Language Models
- **AGI** : Artificial General Intelligence
- **IR** : Information Retrieval
- **API** : Application Programming Interface
- **PaLM** : Pathways Language Model
- **COT** : Chain Of Thoughts
- **RAG** : Retrieval Augmented Genration
- **KG** : Knowledge Graph
- **MRR** : Mean Reciprocal Rank

Introduction générale

L'intelligence artificielle (IA) a révolutionné de nombreux domaines, offrant des opportunités significatives pour automatiser et optimiser des processus complexes. Dans le domaine juridique, en particulier, les avancées de l'IA ont commencé à remodeler la manière dont les services juridiques sont fournis, en offrant des solutions innovantes pour la recherche et l'analyse de la législation.

En Algérie, comme dans de nombreux autres pays, l'accès à une compréhension complète et à jour des textes législatifs est essentiel pour les professionnels du droit, les entreprises et les citoyens. Cependant, la gestion de cette vaste quantité de données législatives, souvent dispersées à travers différents journaux officiels (JOs), pose un défi majeur.

Ce mémoire se concentre sur le développement d'un assistant juridique artificiel pour l'Algérie, conçu pour répondre aux besoins spécifiques des utilisateurs en matière de législation. L'objectif est de créer un chatbot capable d'extraire, d'analyser et de fournir des informations précises et contextualisées sur les textes législatifs algériens (appelés souvent dans ce mémoire comme "publications légales"). Le système sera capable de détecter les relations entre les lois, décrets, arrêtés, etc., de construire un graphe de connaissances pour représenter ces relations, et d'analyser des articles spécifiques au sein de ces textes pour répondre efficacement aux requêtes des utilisateurs.

La réalisation de cet objectif nécessite l'intégration de techniques avancées en traitement automatique du langage naturel, en extraction d'informations, en traitement de graphe, et en génération de texte.

Ce mémoire est structuré en cinq parties de la manière suivante :

1. **Contexte et Cadre Général :** Dans cette première partie, nous introduisons l'organisme d'accueil et le contexte juridique algérien dans lequel ce projet se situe. Le premier chapitre offre une présentation détaillée de l'organisme d'accueil, ainsi que du contexte juridique algérien. Le deuxième chapitre expose la problématique spécifique que ce projet cherche à résoudre, en décrivant les défis liés à l'accès et à l'interprétation des textes juridiques en Algérie, ainsi que la description détaillée du projet.

2. **Etat de l'Art :** La deuxième partie examine l'état de l'art des techniques et des avancées dans le domaine du traitement automatique de language et de l'intelligence artificielle appliqués dans notre projet. Le premier chapitre discute des méthodes et des outils pour l'extraction et le prétraitement des données textuelles. Le deuxième chapitre explore les

techniques d'apprentissage automatique et de traitement du langage naturel les plus récentes qui seront utilisées pour le développement de notre assistant.

3. **Conception :** La troisième partie détaille la conception du système d'assistant juridique artificiel. Le premier chapitre présente les méthodes pour extraire et structurer les données textuelles extraites des journaux officiels, ainsi que la création d'un graphe de connaissance représentant les relations entre les publications. Le deuxième chapitre décrit le pipeline et les technologies utilisées pour le développement du chatbot juridique.

4. **Mise en œuvre et Résultats :** Cette quatrième partie se concentre sur la mise en œuvre et les résultats de notre assistant juridique artificiel. Le premier chapitre détaille l'architecture du système, l'intégration des composants tels que l'API backend, Neo4j, et les services d'OpenAI, ainsi que le déploiement sur AWS. Le deuxième chapitre est dédié à l'évaluation et à la validation de notre système. Il présente les critères d'évaluation utilisés pour mesurer la performance et détaille les résultats des tests effectués, en mettant en lumière les points forts du système ainsi que les axes d'amélioration identifiés.

5. **Conclusion et Perspectives :** La dernière partie propose une synthèse des contributions de ce travail et explore les perspectives d'amélioration et les extensions potentielles pour l'assistant juridique, ouvrant la voie à des recherches futures et à des applications plus larges.

Première partie

Contexte et Cadre Général

Chapitre 1

Présentation de l'Organisme d'Accueil et du Contexte Juridique

1.1 Introduction

Ce chapitre vise à introduire l'organisme d'accueil du projet ainsi que le contexte juridique dans lequel il s'inscrit. Il commence par une présentation détaillée de l'organisme. Ensuite, il explore les concepts et définitions juridiques essentiels, en se concentrant particulièrement sur le système juridique algérien et les différents types de publications légales. Cette mise en contexte est importante pour comprendre les enjeux et les objectifs de notre projet.

1.2 L'Organisme d'Accueil : Monad

Monad, dont le logo est représenté dans la figure 1.1, est une startup fondée en Mars 2023 par une équipe dévouée de passionnés d'IA et de linguistique. Leur objectif principal est de révolutionner la manière dont les entreprises Algériennes interagissent avec leurs données et leurs clients en utilisant des solutions de pointe en traitement du langage naturel.
Monad se positionne comme bien plus qu'une simple entreprise de données. Elle se présente comme un pionnier dans la création d'un web intelligent et interconnecté d'agents qui s'adaptent et évoluent en fonction des besoins de ses clients.

FIGURE 1.1 – Logo de Monad

La transformation de données textuelles brutes en informations exploitables est au cœur des activités de Monad. En tant qu'experts en IA, ils sont désireux de collaborer avec les équipes d'experts de leurs clients pour élaborer les solutions d'IA les plus personnalisées pour leurs entreprises. Leur expertise approfondie en NLP leur permet de fournir des informations

spécialisées et des solutions innovantes, adaptées de manière unique aux exigences spécifiques de leurs clients. Positionnée à l'avant-garde de l'analyse de données textuelles, Monad s'engage à exploiter les dernières avancées dans ce domaine pour fournir à ses clients les informations les plus pertinentes et influentes dérivées de leurs données.

1.3 Le droit

Le droit est un ensemble de règles et de normes établies par une autorité compétente, qui régissent les relations entre les individus et les institutions, et définissent les comportements à adopter dans une société donnée. Il émane d'un pouvoir législatif de l'Etat et est applicable dans un territoire déterminé sur lequel l'Etat exerce sa souveraineté. Le droit sert à réguler les interactions sociales et à établir les règles de conduite à suivre dans une communauté donnée.[5]

Le Droit Algérien : En ce qui concerne le droit algérien, il est composé d'une Constitution, des lois et des règlements adoptés par les autorités algériennes, ainsi que des conventions et traités internationaux auxquels l'Algérie adhère. Il est aussi influencé par des traditions juridiques françaises et islamiques, ainsi que par les coutumes et les pratiques locales.[6]

Le développement du droit algérien a été marqué par des événements clés de l'histoire de l'Algérie, notamment l'indépendance du pays en 1962. Depuis lors, le droit algérien a connu des évolutions significatives pour répondre aux besoins de la société et aux changements économiques, politiques et sociaux.

Publications Légales en Algérie

Les publications légales sont des textes juridiques officiels publiés dans les journaux officiels contenant les règles et normes juridiques applicables dans une société. Elles sont édictées par une autorité compétente comme un gouvernement ou une assemblée législative, établissant les règles de conduite et les sanctions en cas de non-respect. Elles peuvent prendre différentes formes et voici quelques exemples généraux des publications en Algérie :

- **La constitution :** C'est la loi fondamentale du pays. C'est le document qui contient l'ensemble des règles de droit constitutionnel qui déterminent la forme et l'organisation de l'état, les pouvoirs, leurs prérogatives et leurs rapports ainsi que les droits et les devoirs des citoyens.

- **Les lois :** Ce sont les textes juridiques qui émanent des deux chambres à savoir l'Assemblée Populaire Nationale (APN) et le Conseil de la Nation (CN).

- **Les ordonnances :** En cas de vacance de l'Assemblée Populaire Nationale (APN) ou durant les périodes d'intersession du parlement, le président de la République peut légiférer par ordonnance. Les textes ainsi pris, sont soumis à l'approbation des deux Chambres du parlement à leur prochaine session. Les ordonnances sont décidées en Conseil de ministres.

- **Les reglements :** Ce sont les matières autres que celles réservées à la loi relevant du pouvoir réglementaire du président de la République.

- **Les décrets :** Un décret qui est un acte exécutoire à portée générale ou individuelle est signé par le président de la République (Décret présidentiel) ou par le Premier ministre (Décret exécutif).

- **Les arrêtés :** C'est une décision exécutoire à portée générale ou individuelle émanant d'un ou plusieurs ministres (arrêté ministériel ou interministériel) ou d'autres autorités administratives (wilaya, commune, établissement public).

Liens entre les Publications Légales

Au sein du système juridique algérien, les publications légales ne sont pas des entités isolées, mais plutôt des composantes interconnectées formant un réseau complexe de relations. Ces liens peuvent prendre diverses formes, telles que des mises à jour, des citations mutuelles, ou des abrogations, contribuant ainsi à la dynamique et à l'évolution constante du paysage juridique. L'identification et la compréhension de ces liens revêtent une importance capitale pour une vision globale du système légal, permettant ainsi une modélisation efficace.

Journaux Officiels

Dans le contexte juridique algérien, les journaux officiels revêtent une importance particulière en tant que principale source de diffusion des textes législatifs et réglementaires émis par les autorités gouvernementales. Les lois, décrets, arrêtés, et autres textes publiés dans ces journaux définissent le cadre légal et réglementaire du pays. Les professionnels du droit, les citoyens et les entreprises s'appuient sur ces journaux pour connaître et comprendre les normes en vigueur, assurant ainsi la conformité aux règlements.

1.4 Conclusion

Ce chapitre a fourni un contexte essentiel pour comprendre notre projet en mettant en lumière l'organisme d'accueil, Monad, et son rôle dans le domaine de l'intelligence artificielle et du traitement du langage naturel. En outre, il a exploré le système juridique algérien, en mettant en évidence les différentes publications légales et leur interconnexion. Cette compréhension du contexte juridique est importante pour notre projet, car elle nous permet de naviguer avec confiance dans le cadre légal et réglementaire du pays. En résumé, ce chapitre a établi les bases nécessaires pour une exploration approfondie des enjeux et des objectifs de notre projet et du système juridique algérien.

Chapitre 2

Problématique et Description du Projet

2.1 Introduction

Dans un contexte juridique en constante évolution, l'accès rapide et précis aux informations légales constitue un enjeu majeur pour les professionnels du droit, les citoyens et les entreprises. En Algérie, la publication fréquente de nouvelles lois, décrets et arrêtés dans les journaux officiels rend la tâche de suivre et de comprendre les modifications législatives particulièrement complexe. Cette complexité, combinée à l'importance cruciale de rester informé des dernières évolutions légales, souligne la nécessité d'un outil technologique capable de simplifier l'accès à ces informations et d'en améliorer la compréhension.

Ce chapitre aborde les difficultés liées à la gestion de l'information juridique en Algérie et présente en détail le projet d'assistant juridique intelligent conçu pour y remédier. La première section contextualise la problématique et expose les enjeux, en soulignant les défis auxquels les utilisateurs sont confrontés face à l'abondance et à la complexité des textes juridiques, tout en justifiant la pertinence et l'importance de ce projet novateur. La deuxième section détaille les objectifs généraux et spécifiques du projet, en définissant clairement les buts à atteindre pour offrir une solution efficace aux problèmes identifiés. Enfin, la méthodologie et l'approche adoptées pour le développement de cet assistant juridique sont décrites, en présentant les techniques et les étapes clés mises en œuvre, notamment l'utilisation de la technique "Retrieval-Augmented Generation" (RAG), pour garantir que l'outil répondra aux attentes des utilisateurs en termes de précision et de rapidité.

En somme, ce chapitre offre une vue d'ensemble complète de la problématique abordée, des objectifs visés et de la méthodologie adoptée pour le développement de cet assistant juridique intelligent, posant ainsi les bases de la solution innovante proposée par ce projet.

2.2 Contexte et Enjeux

Le droit, en tant que système de règles régissant la vie en société, évolue constamment. En Algérie, le cadre législatif est particulièrement dynamique, avec de fréquentes mises à jour et

modifications des textes juridiques. Cette évolution rapide et continue pose un défi majeur : celui de l'accès et de la compréhension des textes légaux pour les professionnels du droit, les citoyens et les entreprises. Les journaux officiels (JOs), qui publient ces textes, sont souvent volumineux et complexes, rendant difficile la recherche d'informations précises et à jour.

Dans ce contexte, il est crucial de disposer d'outils efficaces pour naviguer dans cette masse d'informations juridiques. La problématique principale de ce projet repose sur la nécessité de faciliter l'accès à l'information juridique et de permettre une compréhension rapide et précise des textes de loi et de leurs interrelations. L'objectif est de créer un assistant juridique intelligent capable de répondre aux requêtes des utilisateurs concernant la législation en Algérie.

2.3 Justification du Projet

La création d'un assistant juridique basé sur l'intelligence artificielle se justifie par plusieurs facteurs :

- **Complexité et Volume de l'Information :** La grande quantité de textes juridiques publiés chaque année rend difficile le suivi et l'analyse manuelle.

- **Accessibilité :** Les professionnels du droit et les citoyens ont besoin d'un accès rapide et précis aux informations juridiques.

- **Évolution Continue :** Les lois et règlements étant fréquemment mis à jour, il est essentiel d'avoir un outil capable de suivre ces modifications en temps réel.

- **Efficacité :** Un assistant juridique intelligent permet de réduire le temps consacré à la recherche et à l'analyse des textes légaux, augmentant ainsi l'efficacité des professionnels du droit.

2.4 Objectifs du Projet

Le projet d'assistant juridique intelligent poursuit des objectifs à la fois généraux et spécifiques pour répondre aux besoins croissants d'accès et de compréhension des textes juridiques en Algérie.

Objectifs Généraux :

Le principal objectif général est de développer un assistant juridique basé sur l'intelligence artificielle, capable de fournir des réponses précises et rapides aux questions juridiques des utilisateurs. Cet assistant doit pouvoir naviguer efficacement dans la complexité et le volume des textes législatifs algériens, facilitant ainsi l'accès à l'information juridique pour les professionnels du droit, les citoyens et les entreprises.

En outre, l'assistant vise également à réduire le temps et les efforts nécessaires pour trouver des informations juridiques pertinentes, augmentant ainsi l'efficacité des utilisateurs dans leurs activités professionnelles ou personnelles.

Objectifs Spécifiques :

- Détecter les types de publications légales et capturer les relations entre eux.
- Construire une base de donnée initiale contenant les différentes publications légales et organiser ces informations dans un graphe de connaissance.
- Structurer les contenus des publication en extractant les articles de chaque publication, puis diviser leurs contenus en segments de texte.
- Vectoriser ces segments permettant une recherche efficace via une base de donnée vectorielle. Ensuite, utiliser un modèle de langage pour générer des réponses basées sur les contenus des segments pertinents appropriés à la requête de l'utilisateur.

2.5 Méthodologie et Approche Adoptée

La méthodologie adoptée pour ce projet repose sur une série d'étapes techniques et fonctionnelles visant à développer un assistant juridique intelligent. Une technique clé utilisée dans ce projet est "Retrieval-Augmented Generation" (RAG), qui combine la recherche d'information avec la génération de texte. Voici un aperçu des phases et des techniques employées :

1. **Extraction de Données :** Les textes bruts sont extraits des journaux officiels algériens.

2. **Détection et Classification :** Identification et classification des différents types de publications et des relations entre eux. Cela permet de constituer une base de données initiale de publications légales.

3. **Construction du Graphe de Connaissance :** Création d'un graphe où les nœuds représentent les publications légales et les arêtes représentent les relations entre ces textes.

4. **Division en Articles et Segments :** Chaque publication est divisée en articles, puis chaque article est segmenté en chunks (segments plus petits) pour une granularité fine dans l'analyse et la recherche.

5. **Vectorisation :** Les segments de texte sont vectorisés en utilisant des modèles de vectorisation avancés, comme les embeddings de phrases, pour convertir les textes en représentations numériques permettant une recherche efficace.

6. **Recherche et Génération de Réponses :** Lorsqu'une question est posée, elle est reformulée si nécessaire en fonction de l'historique de discussion, vectorisée, et utilisée pour rechercher les segments les plus pertinents pour les utiliser comme contexte pour le modèle de génération. Les réponses sont ensuite générées en utilisant un modèle de langage.

Cette approche hybride permet de bénéficier des avantages de la recherche d'information rapide et de la génération de texte contextuelle, assurant ainsi que les réponses fournies par l'assistant juridique sont à la fois précises et pertinentes. La technique RAG améliore significativement la capacité de l'assistant à comprendre et à répondre aux requêtes complexes en s'appuyant sur une base de connaissances riche et constamment mise à jour.

2.6 Résultats Attendus

Les résultats attendus de ce projet incluent :

- **Un assistant juridique performant :** Capable de fournir des réponses précises et rapides aux questions juridiques.

- **Un accès simplifié à l'information juridique :** Réduction du temps nécessaire pour trouver et comprendre les textes de loi.

- **Une mise à jour continue :** Capacité à suivre les évolutions législatives et à intégrer de nouveaux textes rapidement.

- **Amélioration de l'efficacité professionnelle :** Facilitation du travail des professionnels du droit grâce à un outil intelligent et performant.

2.7 Conclusion

Ce chapitre a détaillé la problématique liée à la gestion de l'information juridique en Algérie et a présenté notre projet d'assistant juridique intelligent, conçu pour y remédier. En mettant en lumière les défis auxquels les utilisateurs sont confrontés face à l'abondance et à la complexité des textes juridiques, nous avons justifié la pertinence et l'importance de ce projet novateur. Nous avons ensuite décrit les objectifs généraux et spécifiques du projet, qui visent à offrir une solution efficace aux problèmes identifiés. Enfin, nous avons expliqué la méthodologie et l'approche adoptées pour le développement de cet assistant juridique, en mettant en avant l'utilisation de la technique "Retrieval-Augmented Generation" (RAG). En somme, ce chapitre a fourni une vue d'ensemble complète de la problématique abordée, des objectifs visés et de la méthodologie adoptée pour le développement de notre assistant juridique intelligent, posant ainsi les bases de la solution innovante proposée par ce projet.

Deuxième partie

Etat De l'Art

Chapitre 3

Extraction et prétraitement de données textuelles

3.1 Introdution

Dans ce chapitre consacré à l'extraction et au prétraitement de données textuelles, nous aborderons les différentes techniques et méthodes permettant d'obtenir et de préparer des données textuelles à partir de diverses sources, notamment les fichiers PDF. L'objectif principal de ce processus est de rendre ces données exploitables pour les étapes ultérieures d'analyse et de traitement automatique du langage naturel.

Nous commencerons par examiner l'extraction de texte depuis les fichiers PDF, un format largement utilisé pour stocker et partager des documents. Nous décrirons les techniques d'extraction basées sur la structure interne du fichier, ainsi que la reconnaissance optique de caractères (OCR) pour traiter les documents scannés. Ensuite, nous aborderons l'analyse de la mise en page des documents, une étape cruciale pour comprendre et organiser le contenu des documents. Enfin, nous présenterons les expressions régulières, un outil puissant pour rechercher, manipuler et extraire des motifs spécifiques dans le texte.

Ce chapitre vise à fournir une base solide pour comprendre et mettre en œuvre les différentes techniques d'extraction et de prétraitement de données textuelles, essentielles pour mener à bien des projets liés au traitement automatique du langage naturel.

3.2 Extraction de texte depuis les fichiers PDF

L'extraction de texte à partir de fichiers PDF est une étape cruciale dans le traitement et l'analyse de données textuelles. Cette section présentera le format PDF, les techniques d'extraction de texte basées sur la structure interne du fichier, ainsi que la reconnaissance optique de caractères (OCR) pour traiter les documents scannés.

Format PDF

Le Portable Document Format, plus connu sous l'abréviation PDF, est un format de fichier polyvalent qui a été inventé pour faciliter la présentation et l'échange de documents en toute sécurité, quel que soit le matériel, l'application ou le système d'exploitation utilisé.[7]

Développé par "Adobe Systems" en 1993 et devenu une norme ouverte en 2008, le format PDF est aujourd'hui le principal langage de description de la page imprimée au monde, convenant aussi bien au papier qu'à l'utilisation en ligne.[8] À ce jour, il existe plus que 2.5 trillion de fichiers PDF,[9] témoignant de son adoption massive à l'échelle mondiale.

Chaque fichier PDF encapsule une description complète d'un document fixe et plat incluant le texte, les polices de caractères, les graphiques et d'autres informations nécessaires à son affichage. [10]

Extraction de texte basée sur la structure du PDF

Le format PDF stocke le texte sous forme de flux de caractères dans son code source, accompagné d'informations sur l'emplacement, la police, la taille et d'autres attributs. Pour extraire le texte d'un fichier PDF, il est nécessaire de comprendre comment le texte est stocké et organisé dans le code source, puis de récupérer les flux de caractères appropriés.

1. **Stockage du flux de texte dans le code source du fichier :** Chaque élément de texte dans un fichier PDF est représenté par un objet dans la structure du fichier, contenant des informations sur le contenu et la présentation. Cette organisation permet de préserver la mise en page et le formatage du document lors de son affichage ou de son impression.[8]

2. **Récupération du flux de texte à partir du code source :** Pour extraire le texte d'un fichier PDF, il est nécessaire de parcourir le code source et de récupérer les objets contenant les flux de caractères. Il existe plusieurs solutions open-source pour réaliser cette tâche, telles que PDFMiner, PyPDF2 et PDFBox. Ces outils permettent d'analyser la structure interne du fichier et d'en extraire le texte de manière efficace.

3. **Limites de cette technique :** Bien que l'extraction de texte basée sur la structure interne du PDF soit généralement efficace, elle présente certaines limites, notamment lorsqu'il s'agit de documents scannés. Ces documents sont souvent enregistrés sous forme d'images plutôt que de texte réel, et la technique d'extraction basée sur la structure du PDF ne peut pas récupérer le texte. Dans ce cas, il est nécessaire d'utiliser des méthodes de reconnaissance optique de caractères (OCR) pour extraire le contenu.

Reconnaissance Optique de Caractères

La reconnaissance optique de caractères (ou en anglais Optical Character Recognition - OCR) est une technologie informatique qui permet de convertir des documents physiques ou numériques en texte éditable et exploitable. Cette technologie repose sur l'analyse d'images pour identifier et extraire les caractères qu'elles contiennent.[11]

L'OCR est particulièrement utile pour extraire du texte à partir d'images, et peut être appliquée aux fichiers PDF contenant des documents scannés.

Historique

L'histoire de l'OCR remonte aux années 1920, avec les premiers brevets déposés par Emanuel Goldberg, un pionnier dans le domaine de la reconnaissance de caractères. Les premières machines OCR étaient conçues pour reconnaître des caractères spécifiques, tels que les chiffres ou les lettres majuscules. Ce n'est qu'à partir des années 1970 que les systèmes OCR ont commencé à se généraliser et à prendre en charge un plus large éventail de caractères et de polices.[12]

Application de l'OCR aux fichiers PDF

Pour appliquer l'OCR à un fichier PDF, il est nécessaire de convertir chaque page en une image, qui servira de base à l'analyse et à l'extraction des caractères.

Plusieurs logiciels et bibliothèques open-source sont disponibles pour réaliser cette tâche, tels que Tesseract[13], développé par Google, ou OCRopus[14]. Ces outils permettent d'analyser les images et d'extraire le texte qu'elles contiennent, en prenant en compte différentes polices, tailles et styles de caractères.

Limites de l'OCR

Bien que l'OCR soit une méthode efficace pour extraire du texte depuis des documents PDF scannés, elle présente certaines limites. La qualité de l'extraction dépend fortement de la qualité de l'image source : des images floues, déformées ou présentant des artefacts peuvent entraîner des erreurs de reconnaissance. De plus, l'OCR peut rencontrer des difficultés avec des polices de caractères peu communes ou des dispositions de texte complexes.

3.3 Analyse de la mise en page des documents

Pour surmonter les défis des techniques d'extraction, on a introduit une nouvelle technique qui sert à isoler les composants des documents avant de traiter chacun séparément. L'analyse de la mise en page des documents, également connue sous le nom de Document Layout Analysis (DLA), est une étape cruciale dans les systèmes de compréhension des documents. Son rôle est de détecter et d'annoter la structure physique des documents, facilitant ainsi les phases ultérieures d'analyse. Elle présente de nombreuses applications importantes, telles que la catégorisation du contenu et la reconnaissance de texte. L'objectif principal de la DLA est d'identifier les blocs homogènes du document et de déterminer leurs relations. [15]. La Figure 3.1 illustre un exemple d'analyse de la mise en page d'un document.

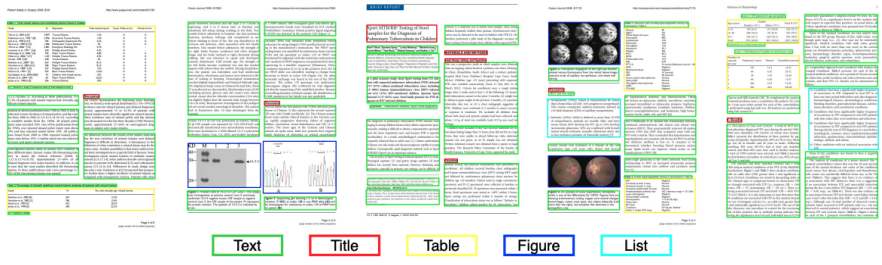


FIGURE 3.1 – Exemple illustrant une analyse de la mise en page [1]

Approches utilisées :

Bien que les fichiers PDF contiennent des représentations explicites et structurées de leur contenu, la structuration correcte de ce contenu en catégories interprétables par l'humain reste un défi et constitue le cœur de l'analyse de la mise en page des documents. [16]

Plusieurs méthodes ont été proposées dans la littérature pour relever ce défi, notamment la segmentation basée sur la géométrie, la segmentation basée sur le contenu, l'apprentissage automatique et l'intelligence artificielle, l'analyse de la mise en page basée sur les graphes, et l'utilisation de techniques de traitement d'image.

1. **Segmentation basée sur la géométrie :** Cette méthode implique la division du document en régions basées sur les caractéristiques géométriques, telles que la position, la taille et la forme des éléments du document. Elle est couramment utilisée pour identifier les en-têtes, les pieds de page, les colonnes de texte, etc.

2. **Segmentation basée sur le contenu :** Cette méthode implique la division du document en régions basées sur le contenu des éléments. Par exemple, les zones de texte peuvent être séparées des images, et les titres peuvent être séparés du texte du corps.

3. **Apprentissage automatique et intelligence artificielle :** Ces techniques peuvent être utilisées pour améliorer l'analyse et la segmentation de la mise en page. Par exemple, des algorithmes d'apprentissage automatique peuvent être entraînés pour reconnaître les différents éléments de la mise en page, tandis que des techniques d'intelligence artificielle peuvent être utilisées pour comprendre le contenu et le contexte des éléments. [17]

4. **Méthode basée sur les graphes :** Cette méthode implique la représentation de la mise en page du document sous forme de graphe, où les nœuds représentent les éléments du document et les arêtes représentent les relations spatiales entre eux. Cette représentation peut être utilisée pour analyser la structure du document et identifier les différentes régions.[16]

5. **Traitement d'image et règles bien définies :** Cette méthode implique l'utilisation de techniques de traitement d'image pour analyser la mise en page du document. Par exemple, des filtres peuvent être appliqués pour améliorer la qualité de l'image, et des algorithmes de détection de contours peuvent être utilisés pour identifier les différents éléments du document. De plus, des règles bien définies peuvent être utilisées pour guider l'analyse et la segmentation. Par exemple, une règle peut stipuler que les en-têtes sont toujours situés en haut de la page et utilisent une police de grande taille.

Techniques de traitement d'images utilisées

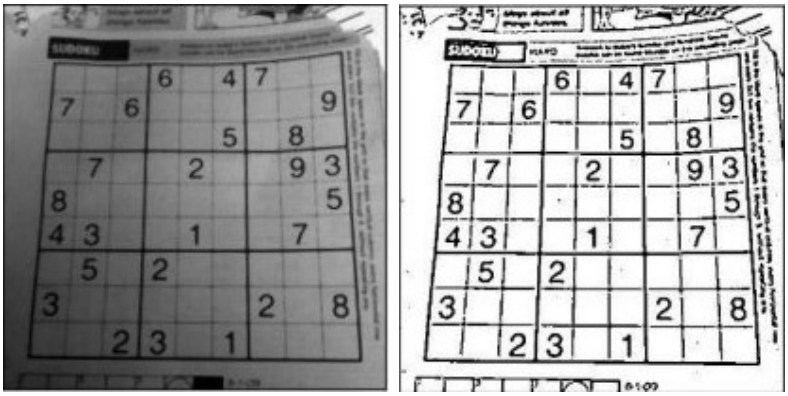
Dans la continuité de notre discussion sur l'analyse de la mise en page des documents, il est important d'aborder les techniques de traitement d'image qui sont largement utilisées pour segmenter la mise en page dans les documents PDF. Ces techniques jouent un rôle important dans l'amélioration de la précision et de l'efficacité de l'analyse et de l'extraction de données à partir de documents PDF.

Dans cette sous-section, nous nous concentrerons sur des techniques comme la binarisation, la dilatation, l'érosion, et la détection de lignes. Nous verrons également comment ces techniques

peuvent être combinées pour améliorer la précision de la segmentation.

Binarisation des images

La binarisation, comme illustré dans la Figure 3.2, est une technique de traitement d'images qui consiste à convertir une image en niveaux de gris ou en couleur en une image binaire, où chaque pixel est soit noir soit blanc. Cette transformation simplifie considérablement l'image originale, facilitant ainsi son analyse ultérieure. L'importance de la binarisation réside particulièrement dans la détection des lignes, une tâche fondamentale dans de nombreuses applications de vision par ordinateur. En réduisant l'image à ses éléments les plus basiques, la binarisation permet d'identifier et de localiser les lignes de manière plus précise et efficace.



(a) Image Originale (b) Réultat

FIGURE 3.2 – Exemple illustrant la binarisation

Dilatation et Erosion

La transformation par dilatation et érosion est une paire d'opérations morphologiques couramment utilisées dans le traitement d'image pour modifier la forme des objets présents dans une image. Ces opérations agissent sur la structure des pixels de l'image, influençant la taille, la forme et la position des objets. La Figure 3.3 illustre l'effet de chacune d'eux.

La dilatation est une opération qui consiste à convoluer une image A avec un filtre (élément structurant) (B) qui peut avoir n'importe quelle forme ou taille, dans notre cas une ligne horizontale ou verticale. Le filtre B a un point d'ancrage défini, généralement le centre du filtre. Lorsque le filtre B est balayé sur l'image, nous calculons la valeur maximale des pixels dans sa fenêtre B_p (où on place le point d'ancrage sur le pixel) et remplaçons le pixel de l'image par cette valeur maximale. L'opération de dilatation est définie comme dans l'équation 3.1 :

$$Dil_B(A) = \{max(B_p) \mid p \in X\} \tag{3.1}$$

L'érosion, quant à elle, est l'opération inverse de la dilatation. Elle vise à réduire la taille des zones d'intérêt dans une image. De manière similaire à la dilatation, l'érosion attribue à chaque pixel de l'image la valeur minimale dans sa fenetre. L'opération de est définie donc comme dans l'équation 3.2 :

$$Eros_B(A) = \{min(B_p) \mid p \in X\} \tag{3.2}$$

On applique l'érosion à l'image dilatée avec les mêmes filtres. L'érosion a pour effet de rétrécir les zones noires.
Ces opérations sont souvent utilisées en duo pour atteindre des objectifs spécifiques comme par exemple mieux détecter les lignes présentes dans l'image.

(a) Image Originale (b) Dilatation (c) Erosion

FIGURE 3.3 – Exemple illustrant les résultats de la dilatation et de l'érosion

Détection des Lignes en utilisant la Transformée de Hough

Cette méthode utilise la transformée de Hough pour identifier les lignes droites dans l'image. La méthode repose sur le principe de transformer les coordonnées des pixels de l'image d'origine en un espace de paramètres appelé espace de Hough. Dans cet espace, les lignes droites sont représentées par des points correspondant à leurs paramètres, tels que l'angle et la distance par rapport à un point de référence.

L'étape suivante consiste à appliquer une stratégie de seuillage pour identifier les paramètres de ligne les plus significatifs dans l'espace de Hough. Ces paramètres représentent les lignes détectées dans l'image. Une fois les paramètres de ligne identifiés, la méthode de HoughLinesP effectue un tracé précis des lignes sur l'image d'origine.

L'identification de ces lignes peut ensuite être utilisée pour extraire des caractéristiques importantes de l'image, telles que les bords, les contours et les formes, ce qui est essentiel pour des tâches telles que la reconnaissance d'objets.

3.4 Expressions Régulières

Une expression régulière se définit librement comme une chaîne de lettres, de chiffres et de symboles spéciaux décrivant une ou plusieurs chaînes de recherche. La chaîne peut contenir des informations fixes ou variables. [18]

Par exemple, nous souhaitons rechercher la chaîne de caractères "mot clé" dans un texte, mais nous ne savons pas si l'auteur a orthographié la chaîne de caractères "mot clé" ou "mot cle". Une expression régulière nous permet de spécifier les informations variables dans les chaînes de recherche, tout en limitant la portée de la recherche. Ainsi, "mot clé" et "mot cle" sont tous deux valables pour la recherche, mais "mot clè" ne l'est pas.

Utilisation des expressions régulières dans l'extraction de texte

Les expressions régulières peuvent être utilisées pour extraire des informations spécifiques d'un document texte, telles que des adresses e-mail, des numéros de téléphone ou des dates. Elles peuvent également être utilisées pour nettoyer et formater le texte, par exemple en supprimant les caractères indésirables ou en remplaçant des chaînes de caractères par d'autres.

Outils et bibliothèques pour les expressions régulières

Il existe de nombreux outils et bibliothèques pour travailler avec les expressions régulières dans différents langages de programmation. Par exemple, le module 're' en Python, la classe 'Pattern' en Java, ou la fonction 'grepl' en R. Ces outils fournissent des méthodes pour rechercher, remplacer et extraire des motifs dans le texte à l'aide d'expressions régulières.

3.5 Conclusion

Dans ce chapitre, nous avons exploré les différentes techniques et méthodes d'extraction et de prétraitement de données textuelles à partir de fichiers PDF. Nous avons étudié le format PDF, l'extraction de texte basée sur la structure interne du fichier, la reconnaissance optique de caractères (OCR) pour traiter les documents scannés, l'analyse de la mise en page des documents et l'utilisation des expressions régulières pour rechercher et manipuler des motifs spécifiques dans le texte.

Ces compétences sont essentielles pour préparer les données textuelles en vue d'une analyse plus approfondie, telles que les tâches liées au traitement automatique du langage naturel (NLP). En maîtrisant ces techniques, il est possible d'extraire et de préparer des données textuelles de manière efficace et précise.

Dans le chapitre suivant, nous nous appuierons sur ces connaissances pour aborder d'autres aspects du traitement automatique du langage naturel, l'analyse sémantique et les modèles de langage.

Chapitre 4

Techniques d'apprentissage automatique et de traitement automatique du langage naturel

4.1 Introduction

L'apprentissage automatique et le traitement automatique du langage naturel sont deux domaines de recherche en pleine expansion qui ont un impact significatif sur diverses industries, allant de la technologie à la santé en passant par l'éducation. Ces domaines visent à permettre aux machines de comprendre et d'interagir avec les humains de manière naturelle et intuitive, en utilisant des algorithmes et des modèles pour traiter et analyser de grandes quantités de données textuelles.

Ce chapitre présentera les concepts clés et les techniques avancées dans les domaines de l'apprentissage automatique et du traitement automatique du langage naturel. Nous commencerons par une introduction à l'apprentissage automatique, en expliquant ses différents types et en présentant ses applications pratiques. Nous aborderons ensuite l'apprentissage profond, une branche de l'apprentissage automatique qui a révolutionné la façon dont les machines traitent et interprètent les données.

Nous nous concentrerons ensuite sur le traitement automatique du langage naturel, en expliquant ses tâches principales et en présentant les techniques utilisées pour les accomplir. Nous explorerons également la tokenisation, la vectorisation du texte et les embeddings, qui sont des étapes cruciales dans le traitement du langage naturel.

Nous discuterons ensuite des réseaux de neurones récurrents et des transformers, deux architectures de réseaux de neurones largement utilisées dans le traitement automatique du langage naturel.

Nous discuterons aussi des Large Language Models (LLMs), qui sont des modèles de langage de grande taille pré-entraînés sur de vastes quantités de données textuelles. Nous verrons comment ces modèles ont amélioré les performances des tâches de traitement automatique du langage naturel et ont ouvert la voie à de nouvelles applications.

Enfin, nous aborderons les approches récentes qui permettent à ces modèles de fournir des réponses plus précises et plus informatives, même pour les tâches nécessitant des connaissances externes.

4.2 Apprentissage Automatique

L'apprentissage automatique, couramment appelé machine learning (ML), constitue un sous-domaine de l'Intelligence Artificielle (IA), qui se concentre sur la programmation d'ordinateurs pour qu'ils puissent apprendre à partir de données.

Selon *Arthur Samuel*, l'apprentissage automatique est "le domaine d'étude qui donne aux ordinateurs la capacité d'apprendre sans être explicitement programmés"[19]

En 1997, *Tom Mitchell* a proposé une définition plus orientée vers l'ingénierie, en décrivant l'apprentissage automatique comme "un programme informatique qui apprend de l'expérience E en ce qui concerne une tâche T et une mesure de performance P, si sa performance sur T, mesurée par P, s'améliore avec l'expérience E."[20]

L'objectif majeur de l'apprentissage automatique est de reproduire le mécanisme d'apprentissage humain en utilisant des algorithmes d'optimisation mathématique. Ces algorithmes évoluent progressivement en assimilant davantage de données, acquérant ainsi la capacité de formuler des prédictions ou de prendre des décisions, tout cela sans nécessiter une programmation explicite pour accomplir la tâche.

4.3 Apprentissage Profond

L'apprentissage profond, également connu sous le nom de Deep Learning en anglais, constitue une branche du Machine Learning qui se distingue par la capacité de ses modèles à avoir des prédictions sans nécessiter l'extraction manuelle de caractéristiques.

Bien que souvent perçu comme un domaine de recherche récent, le deep learning remonte aux années 1940. Il ne semble nouveau que parce qu'il a été relativement impopulaire pendant plusieurs années avant sa popularité actuelle, et parce qu'il a été rebaptisé de nombreuses fois, reflectant l'influence de différents chercheurs et de différentes perspectives. [21]

Cette méthode d'apprentissage s'inspire directement du cerveau humain et repose sur l'utilisation de réseaux de neurones artificiels (Artificial Neural Networks : ANN). La Figure 4.1 illustre la relation entre l'Intelligence Artificielle, le Machine Learning et le Deep Learning.

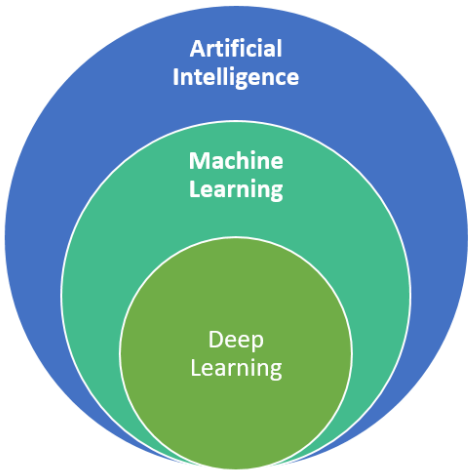


FIGURE 4.1 – Représentation qui illustre la relation entre : Intelligence Artificielle, Machine Learning, et Deep Learning

4.4 Traitement automatique du langage naturel

Depuis l'invention des ordinateurs, la communication fluide à travers le langage naturel a été une technologie de rêve. Le fait que le langage naturel est extrêmement difficile à formuler mathématiquement rend cette tâche encore plus complexe.

Cependant, l'évolution actuelle nous rapproche progressivement de cette ambition, marquée notamment par l'émergence du domaine du "Traitement du Langage Naturel", également connu sous le nom de "Natural Language Processing" (NLP) qui est défini de manière générale comme la manipulation automatique du langage naturel par des logiciels.

Définition

Le NLP est un terme collectif désignant le traitement informatique automatique des langues humaines. Il s'agit à la fois d'algorithmes qui prennent en entrée un texte produit par l'homme et d'algorithmes qui produisent en sortie un texte d'apparence naturelle. [22] L'importance du NLP réside dans sa capacité à combler le fossé entre la communication humaine et les systèmes informatiques, en permettant à des applications telles que les assistants virtuels, les chatbots et les services de traduction linguistique d'interagir avec les utilisateurs d'une manière plus naturelle et intuitive. [23]

Tâches de NLP

Plusieurs tâches NLP décomposent le texte humain de manière à aider l'ordinateur à comprendre ce qu'il traite. Voici quelques-unes de ces tâches :

1. **Classification de texte :** Catégorisation d'un texte d'entrée en groupes prédéfinis. Cela comprend, par exemple, l'analyse des sentiments et la catégorisation des sujets. Les entreprises peuvent utiliser l'analyse des sentiments pour comprendre l'opinion des clients sur leurs services. Le filtrage des courriels est un autre exemple de catégorisation thématique dans lequel les courriels peuvent être classés dans des catégories telles que "Personnel", "Social", "Promotions" et "Spam".

2. **Traduction automatique :** Traduction automatique d'un texte d'une langue à une autre. Il convient de noter que cela peut inclure aussi des domaines tels que la traduction de code d'un langage de programmation à un autre, par exemple de Python à C++.

3. **Reconnaissance d'entités nommées :** La reconnaissance d'entités nommées (NER) est une tâche qui consiste à identifier et à classer les entités nommées dans un texte, telles que les noms de personnes, de lieux, d'organisations, etc.

4. **Génération de texte :** Génération d'un texte de sortie cohérent et pertinent sur la base d'un texte d'entrée donné, appelé une requête (prompt).

Tokenization

La tokenization est une étape cruciale dans le traitement automatique du langage naturel. Elle consiste à découper un texte en unités plus petites, appelées tokens, qui peuvent être des mots, des phrases ou des caractères. Cette étape permet de transformer un texte brut en une séquence de tokens qui pourront être traités par les algorithmes de NLP.

Le but principal de la tokenisation est de représenter le texte d'une manière qui soit significative pour les machines sans perdre son contexte. En convertissant le texte en tokens, les algorithmes peuvent identifier plus facilement les motifs. Cette reconnaissance de motifs est cruciale car elle permet aux machines de comprendre et de répondre à l'entrée humaine [24]. Un exemple illustrant le processus de tokenisation est présenté dans la Figure 4.2

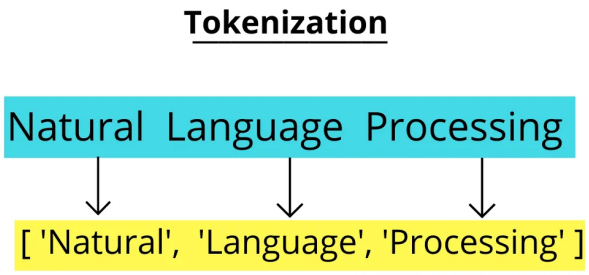


FIGURE 4.2 – Exemple de tokenization

Les manières de découper le texte en tokens varient selon le niveau de détail nécessaire et les besoins spécifiques de la tâche. Voici une exploration des différents types :

- **Tokenisation de mots :** Cette méthode décompose le texte en mots individuels. C'est l'approche la plus courante et elle est particulièrement efficace pour les langues ayant des limites de mots claires, comme l'anglais.

- **Tokenisation de caractères :** Ici, le texte est segmenté en caractères individuels. Cette méthode est bénéfique pour les langues qui manquent de limites de mots claires ou pour les tâches qui nécessitent une analyse granulaire, comme la correction orthographique.

- **Tokenisation de sous-mots :** En trouvant un équilibre entre la tokenisation de mots et de caractères, cette méthode décompose le texte en unités qui peuvent être plus grandes qu'un seul caractère mais plus petites qu'un mot complet. Par exemple, "Chatbots" pourrait être tokenisé en "Chat" et "bots". Cette approche est particulièrement utile pour les langues qui forment un sens en combinant de plus petites unités ou lorsqu'on traite avec des mots hors vocabulaire dans les tâches NLP.

La tokenisation joue un rôle crucial dans la préparation du texte pour les tâches de NLP. Une fois le texte découpé en tokens, la prochaine étape consiste à convertir ces tokens en représentations numériques pour permettre aux modèles d'apprentissage automatique de les traiter. Cette étape s'appelle la vectorisation du texte.

Vectorisation du Texte et Embeddings

Pour surmonter le fait que les ordinateurs ne comprennent pas le langage naturel, il est nécessaire de représenter les mots et les phrases sous une forme que les algorithmes peuvent traiter. Cette représentation est généralement sous forme de vecteurs, d'où le terme de vectorisation du texte.

Embeddings sont fondamentalement une forme de représentation des mots qui relie de manière significative la compréhension humaine de la connaissance à la compréhension d'une machine. Ils sont des représentations éparses d'un texte dans un espace à n dimensions, qui tentent de capturer la signification des mots. [25]

Techniques de vectorisation

La vectorisation du texte est une étape cruciale dans le traitement du langage naturel. Il existe plusieurs techniques pour convertir les mots et les phrases en vecteurs. Voici les plus couramment utilisées :

- **One-Hot Encoding :** La méthode One-Hot est la plus simple des techniques de vectorisation. Elle consiste à représenter chaque mot du vocabulaire par un vecteur binaire de la taille du vocabulaire. Chaque mot est représenté par un vecteur où tous les éléments sont à zéro, sauf un qui est à un, correspondant à la position du mot dans le vocabulaire. Cette méthode est simple mais elle a un inconvénient majeur : elle ne prend pas en compte la sémantique des mots, c'est-à-dire que deux mots proches en termes de sens pourraient être très éloignés en termes de vecteurs.

- **Bag Of Words (BOW) :** Le modèle Bag of Words est une autre méthode de vectorisation. Elle consiste à représenter un texte sous forme d'un sac de mots, en ignorant l'ordre des mots et leur contexte. Chaque mot unique est considéré comme une caractéristique, et un vecteur de fréquence de chaque mot est créé pour représenter le texte.

- **TF-IDF (Term Frequency-Inverse Document Frequency) :** Le TF-IDF est une variante du modèle Bag of Words. Elle consiste à pondérer la fréquence d'apparition d'un mot dans un texte par sa fréquence inverse dans l'ensemble des textes.
  L'idée derrière la méthode TF-IDF est de donner une pondération élevée aux termes qui apparaissent fréquemment dans un document, mais pas dans l'ensemble de la collection de documents. Cela permet de mettre en avant les termes qui sont caractéristiques d'un document par rapport aux autres documents.

  La pondération TF-IDF d'un terme t dans un document d est calculée comme dans l'équation 4.1 :

$$\text{TFIDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t) \tag{4.1}$$

  où :

  - $\text{TF}(t, d)$: la fréquence du terme t dans le document d (nombre de fois où le terme apparaît dans le document).
  - $\text{IDF}(t)$: la mesure d'inverse de la fréquence de document du terme t, calculée comme dans l'équation 4.2 :

$$\text{IDF}(t) = \log\left(\frac{N}{\text{df}(t)}\right) \tag{4.2}$$

où :

* N : le nombre total de documents dans la collection.
* $\text{df}(t)$: le nombre de documents dans la collection qui contiennent le terme t.

- **Word2Vec :** Word2Vec est une méthode plus avancée de vectorisation du texte. Elle consiste à apprendre des représentations de mots à partir de leur contexte d'utilisation dans le texte.
  Elle permet de créer des représentations vectorielles d'un vocabulaire à partir d'un corpus de textes, où chaque mot est représenté par un vecteur de nombres réels. Ces vecteurs sont construits de telle manière que les mots qui ont un contexte similaire dans le corpus ont des représentations vectorielles similaires.

- **GloVe (Global Vectors for Word Representation) :** GloVe [26] est une autre méthode de vectorisation avancée. Elle consiste à apprendre des représentations de mots à partir de leur co-occurrence dans le texte.
  Le principe fondamental de GloVe repose sur l'idée que les mots qui apparaissent fréquemment dans des contextes similaires partagent une signification sémantique similaire. Le modèle GloVe utilise une matrice de co-occurrence qui enregistre la fréquence à laquelle chaque paire de mots apparaît ensemble dans un corpus de texte. Cette matrice est ensuite utilisée pour estimer les représentations vectorielles des mots.

- **Méthodes basées sur les grands modèles de langage :** Il convient de noter qu'il existe également des méthodes de vectorisation plus récentes et plus avancées qui sont basées sur les grands modèles de langage. Elles sont venues pour surpasser les limites des anciennes techniques. Ces méthodes, telles que BERT [27], GPT [28], et ELMo [29], utilisent des réseaux de neurones profonds pour apprendre des représentations de mots à partir de vastes quantités de données textuelles. Elles sont capables de capturer des aspects plus subtils de la sémantique des mots et de leur contexte d'utilisation.
  Cependant, la discussion détaillée de ces méthodes dépasse le cadre de cette section sur la vectorisation du texte. Nous les aborderons en détail dans les sections suivantes consacrées aux grands modèles de langage.

Une fois que le texte a été tokenisé et chaque token a être représenté sous forme de vecteur, ces vecteurs peuvent ensuite être utilisés comme entrée pour les modèles de deep learning comme les réseaux de neurones récurrents qui peuvent alors traiter ces séquences pour effectuer des tâches de NLP telles que la classification de texte, la génération de texte ou la traduction automatique.

4.5 Réseaux de Neurones Récurrents

Les réseaux de neurones récurrents (RNN) ont joué un rôle crucial dans la révolution du NLP en permettant aux modèles d'apprentissage profond de prendre en compte le contexte et la temporalité des données textuelles. Les RNN sont des architectures de réseaux de neurones qui utilisent des boucles pour traiter des séquences de données, ce qui les rend particulièrement adaptés aux tâches NLP telles que la classification de texte, la traduction automatique et la génération de texte.

Un réseau neuronal récurrent (RNN) est un type de réseau neuronal artificiel dans lequel les connexions entre les unités forment un cycle dirigé. Cela crée un état interne du réseau qui lui permet d'afficher un comportement dynamique. Contrairement aux réseaux neuronaux de type feed forward, les RNN peuvent utiliser leur mémoire interne pour traiter et travailler sur des séquences arbitraires d'entrées. [30]

Les réseaux neuronaux récurrents ont été créés dans les années 1980, mais ils ont récemment gagné en popularité grâce à l'augmentation de la puissance de calcul. Ils sont particulièrement utiles pour les données séquentielles comme les données textuelles, car chaque neurone peut utiliser sa mémoire interne pour conserver les informations relatives à l'entrée précédente. La Figure 4.3 illustre l'architecture des RNNs.

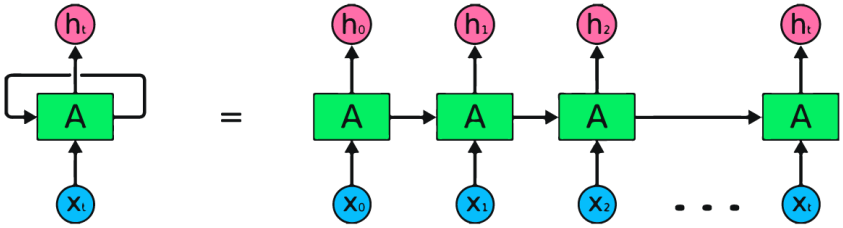


FIGURE 4.3 – Architecture des RNNs [2]

Autres types des RNN

Les RNN peuvent traiter le contexte dès le début de l'énoncé, ce qui permet des prédictions plus précises d'un mot à la fin de l'énoncé. Dans la pratique, cela n'est pas nécessairement vrai pour tous les types de RNN, car les RNN sont en fait limités à quelques pas en arrière. C'est l'une des principales raisons pour lesquelles les RNN doivent être utilisés avec une mémoire à long terme (LSTM) pour obtenir d'excellents résultats. L'ajout d'une LSTM au réseau revient à ajouter une unité de mémoire à l'intérieur du réseau, capable de se souvenir du contexte depuis le tout début de l'entrée. Ainsi, si la phrase comporte 10 mots et que nous voulons prédire le 11e mot, les 10 mots sont traités par les RNN et leurs poids à chaque étape sont sauvegardés à l'aide de la LSTM, ce qui permet de prédire la probabilité du 11e mot en conséquence [30]. La Figure 4.4 illustre la différence entre les deux architectures.

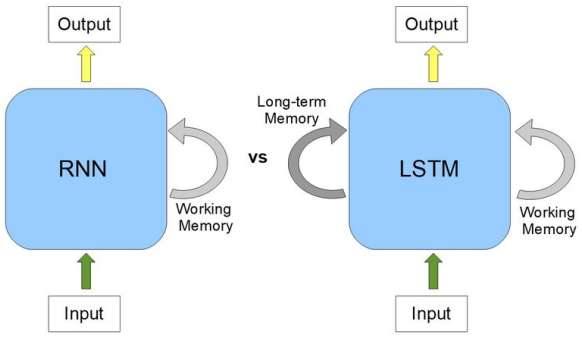


FIGURE 4.4 – Différence entre RNN et LSTM [3]

Limites

Malgré les améliorations apportées par les LSTM, les RNN présentent encore certaines limites. L'une des principales limites est le problème de dépendance à long terme. Bien que les LSTM soient conçus pour gérer de longues séquences, leur performance peut se dégrader lorsque la distance entre les mots dépendants dans une phrase est très grande. Cette limitation est due à la nature séquentielle des RNN, qui traite les données étape par étape, ce qui rend difficile la capture des dépendances à long terme.

Ces limites ont ouvert la voie à l'émergence d'une nouvelle architecture : les Transformers. Introduits par *Vaswani et al.* dans leur article *"Attention is All You Need"* [4], ont révolutionné le domaine du NLP en proposant une nouvelle méthode pour traiter les données séquentielles, basée sur l'attention.

4.6 Transformers

Dans la continuité de notre exploration des techniques d'apprentissage automatique et de traitement automatique du langage naturel, nous allons maintenant nous pencher sur une avancée majeure dans le domaine du NLP : les Transformers.

Cette section présentera les concepts clés des Transformers, leur fonctionnement et leurs applications dans diverses tâches de NLP. Nous verrons également comment cette innovation a contribué à l'amélioration des performances des modèles de deep learning dans le traitement du langage naturel et a ouvert la voie à de nouvelles possibilités dans ce domaine en constante évolution.

4.6.1 Mécanisme d'attention

Pour comprendre les transformers, nous devons d'abord comprendre le mécanisme d'attention. Le mécanisme d'attention permet aux transformers d'avoir une mémoire à long terme extrêmement longue. Un modèle de transformer peut "se concentrer" sur tous les tokens précédents qui ont été générés.

Étant donnée une séquence d'entrée $X = \{x_1, x_2, \ldots, x_n\}$, le mécanisme d'attention calcule une somme pondérée de tous les éléments de X, où les poids sont déterminés par la pertinence de chaque élément par rapport à l'élément courant. Cela peut être représenté mathématiquement comme dans l'équation 4.3 :

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{4.3}$$

où Q, K et V sont les matrices de requête, de clé et de valeur respectivement, et d_k représente la dimension de la clé.

4.6.2 Architecture

Conçus à l'origine pour la transduction de séquences ou la traduction automatique neuronale, les transformers excellent dans la conversion de séquences d'entrée en séquences de sortie. Il

s'agit du premier modèle de transduction reposant entièrement sur le mécanisme d'attention pour calculer les représentations de son entrée et de sa sortie sans utiliser de RNN ou de convolution. La principale caractéristique de l'architecture des transformateurs est qu'ils conservent le modèle encodeur-décodeur [31]. La Figure 4.5 illustre l'architecture des Transformers.

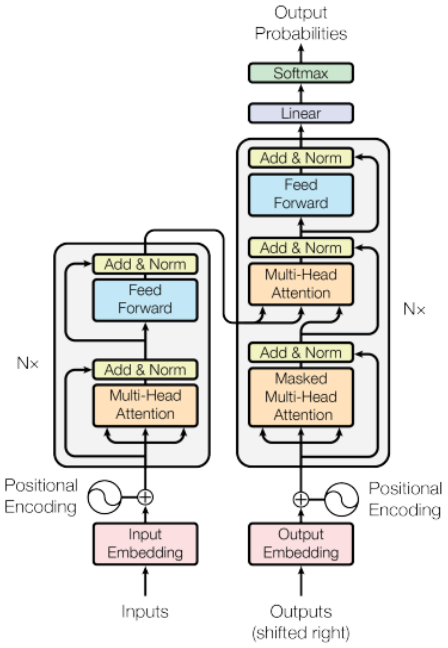


FIGURE 4.5 – Architecture des Transformers [4]

Encodeur

Dans un modèle transformer, l'encodeur est chargé d'encoder la séquence d'entrée. Il se compose d'une pile de couches identiques, comprenant chacune un mécanisme d'auto-attention multi-têtes et un réseau de neurones à propagation avant. L'encodeur traite la séquence d'entrée en parallèle et capture les dépendances entre différents éléments.

Soit $X = \{x_1, x_2, \ldots, x_n\}$ la séquence d'entrée. L'encodeur prend X et produit une séquence de représentations encodées $Z = \{z_1, z_2, \ldots, z_n\}$. Le processus d'encodage peut être exprimé comme comme dans l'équation 4.4 :

$$Z = \text{Encoder}(X) = \text{EncoderLayer}(\ldots(\text{EncoderLayer}(X))) \tag{4.4}$$

où *EncoderLayer* représente une seule couche dans l'encodeur, et les points de suspension indiquent l'empilement de plusieurs couches.

Décodeur

Les couches de décodeurs prennent la représentation vectorielle générée par les encodeurs et génèrent une séquence de sortie, souvent mot par mot. Chaque couche de décodeur comprend également un mécanisme d'attention, mais il est légèrement modifié pour se concentrer sur les parties pertinentes de la représentation vectorielle générée par les encodeurs.

Le décodeur est autorégressif : il débute avec un token de démarrage et prend en entrée une liste des sorties précédentes, ainsi que les sorties de l'encodeur qui contiennent les informations

d'attention provenant de l'entrée. Le décodeur cesse de décoder lorsqu'il génère un token en sortie.[32]

4.6.3 Applications

Les Transformers ont révolutionné le domaine du NLP et ont été appliqués à diverses tâches, améliorant considérablement les performances des modèles. [33]
Voici quelques exemples d'applications des Transformers :

- **Traduction Automatique :** La traduction automatique est l'une des premières applications des Transformers. Les modèles basés sur les Transformers, tels que BERT (Bidirectional Encoder Representations from Transformers) [27], ont considérablement amélioré la qualité de la traduction en prenant en compte le contexte global de la phrase, ce qui a permis de mieux comprendre les nuances et les subtilités linguistiques.
- **Génération de Texte :** Les Transformers sont également utilisés pour la génération de texte, comme la rédaction d'articles, la création de dialogues pour les chatbots, et même la génération de poèmes ou de paroles de chansons. GPT (Generative Pretrained Transformer) [28], développé par OpenAI, est un exemple de modèle de Transformer capable de générer un texte cohérent et pertinent en fonction d'une requête donnée.
- **Résumé Automatique :** Les Transformers ont montré leur efficacité dans cette tâche en étant capables de comprendre et de synthétiser des informations à partir de textes longs.
- **Reconnaissance d'Entités Nommées :** Les Transformers ont montré leur efficacité dans cette tâche en étant capables de comprendre le contexte dans lequel les entités apparaissent et en améliorant ainsi la précision des modèles de NER.

4.6.4 Évolution des Transformers

Les transformers, introduits initialement pour des tâches comme la traduction automatique neuronale, ont démontré leur capacité à encoder et à décoder des séquences en utilisant des mécanismes d'attention. Cette architecture a permis de surmonter les limitations des RNN et des convolutions en offrant une approche plus parallèle et efficace pour traiter les séquences.

Les encodeurs-décodeurs constituant les transformers ont fourni une base solide pour le développement de modèles plus vastes et plus puissants.
Une avancée très populaire reposant sur les transformers est ce que l'on appelle les Grands Modèles de Langage (LLMs), qui vont au-delà de cette architecture en utilisant ces principes de manière encore plus étendue pour des tâches de compréhension du langage et de génération de texte.

4.7 Large Language Models

Les avancées significatives récentes dans le domaine des modèles de langage sont largement attribuables aux transformers, à l'augmentation de la puissance de calcul et à la disponibilité de vastes ensembles de données d'entraînement. Ces développements ont ouvert la voie à la création de grands modèles de langage (LLM, pour Large Language Models), capables d'approcher les performances humaines dans une variété de tâches. [34]

Ces modèles ont attiré beaucoup d'attention en raison de leurs performances remarquables dans un large éventail de tâches de traitement du langage naturel, notamment depuis la sortie de ChatGPT en novembre 2022. La capacité des LLM à comprendre et à générer du langage de manière générale est acquise grâce à l'entraînement de milliards de paramètres de modèle sur d'énormes quantités de données textuelles.[35]

4.7.1 Contexte

La modélisation linguistique, une approche clé pour le développement de l'intelligence linguistique des machines, vise à prédire la probabilité de séquences de mots, afin de prédire les probabilités de futurs tokens (ou tokens manquants). Ce domaine a connu plusieurs étapes majeures, chacune apportant des avancées significatives. [35]

Tout d'abord, les modèles linguistiques statistiques (SLM) ont été développés pour prédire la probabilité de séquences de mots. Ensuite, les modèles linguistiques neuronaux (NLM) ont été introduits, suivis par les modèles linguistiques pré-entraînés (PLM). Les grands modèles de langage (LLM) représentent la dernière étape de cette évolution et ont un impact significatif sur la communauté de l'IA.

L'émergence de ChatGPT et de GPT-4 [36] a même conduit à réévaluer les possibilités de l'intelligence artificielle générale (AGI pour Artificial General Intelligence).

4.7.2 Evolution et Impact

Dans la lignée de cette évolution, OpenAI, la société à l'origine de ChatGPT, a publié en février 2023 un article technique intitulé *"Planification pour l'AGI et au-delà"*. Cet article discute des plans à court et à long terme pour approcher l'AGI en utilisant les LLM [37]. De plus, un article plus récent a avancé que GPT-4 pourrait être considéré comme une version précoce d'un système AGI [38].

Les progrès rapides des LLM révolutionnent les domaines de recherche en IA. Dans le traitement du langage naturel, les LLM servent de solveurs de tâches linguistiques générales, transformant ainsi le paradigme de recherche [39]. Dans le domaine de la recherche d'information (IR), les moteurs de recherche traditionnels sont confrontés à une nouvelle méthode de recherche d'information via des chatbots d'IA (comme ChatGPT) [40]. Dans le domaine de la vision par ordinateur, les chercheurs s'efforcent de développer des modèles vision-langage similaires à ChatGPT pour mieux servir les dialogues multimodaux [41]. Notamment, GPT-4 a intégré l'information visuelle en supportant une entrée multimodale.

4.7.3 Applications

Les grands modèles de langage (LLM) ont révolutionné le domaine du NLP, en excellant dans diverses tâches classiques telles que l'analyse de texte au niveau des mots et des phrases, le marquage de séquences, l'extraction de relations et la génération de texte. [39]

Au-delà du NLP général, les LLM trouvent des applications croissantes dans des domaines spécialisés tels que la santé, l'éducation, le droit, la finance et la recherche scientifique. Par exemple, en santé, les LLM sont utilisés pour des tâches allant de l'analyse de la santé mentale

à l'assistance médicale, tandis que dans l'éducation, ils servent d'assistants à l'écriture et à la lecture. Malgré que les LLM continuent de démontrer leur potentiel transformationnel en augmentant l'efficacité et l'automatisation des tâches dans des domaines spécifiques, l'utilisation généralisée de ces modèles soulève des défis tels que le risque de biais, de plagiat et de fiabilité, notamment dans des secteurs sensibles comme la santé et l'éducation. [39]

4.7.4 Limites

Il est important de se rappeler que les LLM sont formés pour prédire un token. Bien que le fine-tuning et l'alignement améliorent leurs performances et ajoutent différentes dimensions à leurs capacités, il existe encore quelques limitations importantes qui se présentent, particulièrement si elles sont utilisées de manière naïve. Certaines d'entre elles incluent les points suivants :

- **Ils n'ont pas mémoire :** Les LLM par eux-mêmes ne peuvent même pas se souvenir de ce qui leur a été envoyé dans la requête précédente. C'est une limitation importante pour de nombreux cas d'utilisation qui nécessitent une certaine forme d'état.

- **Ils sont probabilistes :** Si vous envoyez plusieurs fois la même requête à un LLM, vous êtes susceptible d'obtenir des réponses différentes. Bien qu'il existe des paramètres, et en particulier la température [1], pour limiter la variabilité de la réponse, c'est une propriété inhérente à leur entraînement qui peut créer des problèmes.

- **Ils ont des informations obsolètes :** Un LLM par lui-même ne connaît même pas l'heure ou le jour actuel et n'a pas accès à toute information qui n'était pas présente dans son ensemble d'entraînement.

- **Ils hallucinent :** Les LLM n'ont pas de notion de "vérité" et ils ont généralement été entraîné sur un mélange de contenu bon et mauvais. Ils peuvent produire des réponses très plausibles mais fausses.

Bien que toutes les limites mentionnées précédemment puissent avoir un impact significatif sur certaines utilisations, il est utile d'examiner de plus près le problème des hallucinations. Ce dernier a attiré beaucoup d'attention récemment et a donné lieu à de nombreuses approches de prompts et des méthodes d'augmentation des LLM que nous décrirons plus tard.

Hallucinations

Dans le domaine des LLM, le phénomène des hallucinations a suscité une attention considérable ces derniers moments surtout avec l'utilisation excessive du chat GPT ces derniers mois.

Défini dans la littérature, notamment dans l'article [42], l'hallucination dans un LLM est caractérisée comme "la génération de contenu qui est dépourvu de sens ou qui n'est pas fidèle à la source fournie". Ce terme, bien qu'ancré dans le langage psychologique, a été approprié dans le domaine de l'intelligence artificielle.

1. La 'température' est un paramètre utilisé lors de la génération de texte qui contrôle le degré d'incertitude dans les prédictions du modèle.

Selon la catégorisation de [42], il existe deux principaux types d'hallucinations, à savoir l'hallucination intrinsèque et l'hallucination extrinsèque.

1. **Hallucinations intrinsèques :** Elles entrent en conflit directement avec le matériel source, introduisant des inexactitudes factuelles ou des incohérences logiques.
2. **Hallucinations extrinsèques :** Ces dernières, bien qu'elles ne contredisent pas, ne peuvent pas être vérifiées par rapport à la source, englobant des éléments non confirmables.

La définition de "source" dans le contexte des LLM varie selon la tâche. Dans les tâches basées sur le dialogue, elle fait référence à la "connaissance du monde", tandis que dans la résumé de texte, elle se rapporte au texte d'entrée lui-même.

4.8 Prompt Engineering

L'apprentissage automatique, en particulier dans le traitement du langage naturel, a évolué de l'utilisation de méthodes entièrement supervisées, qui dépendaient fortement de l'ingénierie des caractéristiques et de l'entraînement sur des ensembles de données spécifiques à la tâche, vers une ère où l'accent est mis sur l'ingénierie de l'architecture permettant aux modèles d'apprendre directement des caractéristiques pertinentes. Le passage à l'approche de «pré-entraînement et fine-tuning» entre 2017 et 2019 a représenté une avancée majeure, permettant aux modèles d'être d'abord entraînés sur de vastes quantités de données textuelles non spécifiques avant d'être ajustés pour des tâches spécifiques, soulignant l'importance de l'ingénierie des objectifs d'entrainement. Plus récemment, le paradigme de «pré-entraînement, prompt et prédiction» a marqué une évolution, s'éloignant du fine-tuning au profit de l'utilisation de prompts textuelles pour orienter les modèles pré-entraînés vers des solutions tâche-spécifiques sans entraînement supplémentaire. Cette méthode soulève cependant le challenge du prompt engineering, c'est-à-dire la sélection de prompts optimales pour permettre au modèle de résoudre efficacement différentes tâches. [43]

Définition

L'ingénierie de prompt consiste à concevoir des instructions spécifiques pour guider les LLM dans la génération de réponses plus précises et plus pertinentes. Cela implique le développement d'une fonction de prompt $f_{\text{prompt}}(x)$ visant à optimiser les performances dans les tâches en aval [43]. Traditionnellement, cela implique l'ingénierie de templates, où soit des algorithmes, soit des experts humains recherchent le template le plus efficace pour chaque tâche attendue du modèle. Cette méthode présente des similitudes avec la technique pédagogique d'enseigner par questionnement, rappelant le guidage du processus de réflexion d'un enfant. Tout comme une question bien formulée oriente la pensée d'un enfant, un prompt soigneusement conçu peut influencer la sortie d'un modèle d'IA, en particulier un grand modèle de langage (LLM). L'ingénierie de prompt constitue ainsi le lien critique entre l'intention humaine et la réponse de la machine, façonnant l'interaction entre les utilisateurs et les systèmes IA [44].

Techniques utilisées

L'ingénierie de prompt emploie diverses techniques pour optimiser l'efficacité des prompts. Chaque méthode a ses propres avantages et défis, et leur sélection dépend du contexte de la tâche, des ressources disponibles, et des caractéristiques du modèle de langage.

1. **Zero-Shot Prompting :** Le Zero-Shot Prompting [45] offre un changement de paradigme dans l'exploitation des grands modèles de langage (LLM). Cette technique élimine le besoin de données d'entraînement extensives, en se basant plutôt sur des prompts soigneusement conçues qui guident le modèle vers de nouvelles tâches. Plus précisément, le modèle reçoit une description de la tâche dans l'invite, mais manque de données étiquetées pour s'entraîner sur des mappages spécifiques d'entrée-sortie. Le modèle exploite ensuite ses connaissances préexistantes pour générer des prédictions basées sur le prompt donné pour la nouvelle tâche.

2. **Few-Shot Prompting :** Le prompt en few-shot [46] fournit aux modèles quelques exemples d'entrée-sortie pour induire une compréhension d'une tâche donnée, contrairement au prompt en zero-shot, où aucun exemple n'est fourni. Fournir même quelques exemples de haute qualité peut améliorer les performances du modèle sur des tâches complexes par rapport à l'absence de démonstration. Cependant, le prompt en few-shot nécessite des tokens supplémentaires pour inclure les exemples, ce qui peut devenir prohibitif pour des entrées de texte plus longues. De plus, la sélection et la composition des exemples peuvent influencer le comportement du modèle, et des biais comme la préférence pour les mots fréquents peuvent encore affecter les résultats en few-shot [47]. Bien que le prompt en few-shot améliore les capacités pour des tâches complexes, en particulier parmi les grands modèles pré-entraînés, un ingénierie de prompts soigneuse est cruciale pour atteindre des performances optimales et atténuer les biais involontaires du modèle.

3. **Chain-of-Thought (CoT) :** Les LLMs peinent souvent face au raisonnement complexe, limitant leur potentiel. Dans le but de combler cette lacune, [48] ont introduit le prompt en Chaîne de Pensée (Chain Of Thought - CoT) comme une technique pour solliciter les LLM de manière à faciliter des processus de raisonnement cohérents et étape par étape. La contribution principale réside dans la proposition et l'exploration du prompt en CoT, démontrant son efficacité à susciter des réponses plus structurées et réfléchies de la part des LLM par rapport aux prompts traditionnelles. À travers une série d'expériences, les auteurs mettent en avant les qualités distinctives du prompt en CoT, soulignant sa capacité à guider les LLM à travers une chaîne de raisonnement logique. Par exemple, le prompt montrerait le processus de raisonnement et la réponse finale pour un problème de mathématiques à étapes multiples et imiterait comment les humains décomposent les problèmes en étapes intermédiaires logiques.

4. **Self-Consistency :** les auteurs de [49] ont introduit la self-consistence, une stratégie de décodage améliorant la performance du raisonnement par rapport au décodage gourmand dans les prompts en Chaîne de Pensée (CoT). Pour les tâches de raisonnement complexe comportant plusieurs chemins valides, la self-consistence génère des chaînes de raisonnement diverses en échantillonnant à partir du décodeur du modèle de langage. Elle identifie ensuite la réponse finale la plus cohérente en marginalisant ces chaînes échantillonnées. Cette approche tire parti de l'observation selon laquelle les problèmes nécessitant une analyse réfléchie impliquent souvent une plus grande diversité de raisonnement.

En conclusion, le prompt engineering offre une approche innovante pour optimiser l'utilisation des grands modèles de langage en guidant leur génération de réponses. Cependant, cette méthode dépend fortement de la qualité et de la pertinence des prompts utilisés, ce qui souligne la nécessité de techniques d'optimisation plus avancées. La section suivante explorera une approche complémentaire qui vise à améliorer la génération de réponses en enrichissant les modèles de langage avec des informations contextuelles pertinentes, tirées des sources externes.

4.9 Retrieval Augmented Generation

Les LLM ont révolutionné la génération de texte, cependant leur dépendance à des données d'entraînement limitées et statiques entrave leur capacité à fournir des réponses précises, en particulier pour les tâches nécessitant des connaissances externes. La méthode de sollicitation traditionnelle est insuffisante, nécessitant un réentraînement coûteux. La Génération Augmentée par Récupération (Retrieval Augmented Generation - RAG) émerge comme une solution innovante, intégrant la recherche d'informations dans le processus de prompting. [47]

Principe

Le RAG [50] analyse l'entrée de l'utilisateur, élabore une requête ciblée et fouille une base de connaissances préétablie à la recherche de ressources pertinentes. Les extraits récupérés sont intégrés dans le prompt original, l'enrichissant d'un contexte de fond. Le prompt augmenté permet au LLM de générer des réponses créatives et factuellement précises. L'agilité du RAG surmonte les limitations statiques, la rendant changeante pour les tâches exigeant des connaissances à jour.

Processus

Le paradigme de recherche RAG représente une méthodologie qui a gagné en importance peu de temps après l'adoption généralisée de ChatGPT. Il suit un processus comprenant l'indexation, la récupération et la génération.

1. **L'indexation :** commence par le nettoyage et l'extraction des données brutes dans différents formats tels que PDF, HTML, Word et Markdown, qui sont ensuite convertis en un format texte uniforme. Pour répondre aux limitations contextuelles des modèles linguistiques, le texte est segmenté en morceaux plus petits et plus digestes. Ces morceaux sont ensuite encodés en représentations vectorielles et stockés dans une base de données vectorielle. Cette étape est cruciale pour permettre des recherches de similarité efficaces lors de la phase de récupération ultérieure.

2. **Récupération :** Lors de la réception d'une requête utilisateur, le système RAG utilise le même modèle d'encodage utilisé lors de la phase d'indexation pour transformer la requête en une représentation vectorielle. Il calcule ensuite les scores de similarité entre le vecteur de la requête et le vecteur des morceaux dans le corpus indexé.

   Le système priorise et récupère les k meilleurs morceaux qui démontrent la plus grande similarité avec la requête. Ces morceaux sont ensuite utilisés comme contexte élargi dans le prompt.

3. **Génération :** La requête posée et les documents sélectionnés sont synthétisés en un prompt cohérent auquel un grand modèle linguistique est chargé de formuler une réponse. L'approche du modèle pour répondre peut varier en fonction de critères spécifiques à la tâche, lui permettant soit de s'appuyer sur ses connaissances paramétriques inhérentes, soit de restreindre ses réponses aux informations contenues dans les documents fournis. Dans le cas de dialogues en cours, tout historique conversationnel existant peut être intégré dans le prompt, permettant au modèle de s'engager efficacement dans des interactions de dialogue à plusieurs tours.

Optimoisation de l'Indexation

Dans la phase d'indexation, les documents seront traités, segmentés et transformés en embeddings pour être stockés dans une base de données vectorielle. La qualité de la construction de l'index détermine si le bon contexte peut être obtenu dans la phase de récupération.

Stratégie de découpage

La méthode la plus courante consiste à diviser le document en chunks [2] selon un nombre fixe de tokens (par exemple, 100, 256, 512) [51]. Les chunks plus grands peuvent capturer plus de contexte, mais ils génèrent également plus de bruit, nécessitant plus de temps de traitement et des coûts plus élevés. Alors que les chunks plus petits peuvent ne pas transmettre entièrement le contexte nécessaire, ils ont moins de bruit. Cependant, les chunks entraînent une troncature au sein des phrases, ce qui incite à optimiser les méthodes de divisions récursives et de fenêtres glissantes [52]. Néanmoins, ces approches ne parviennent toujours pas à trouver un équilibre entre la complétude sémantique et la longueur du contexte. Par conséquent, des méthodes comme Small2Big [53] ont été proposées.

Base de Données Vectorielle

Une base de données vectorielle est un type spécifique de base de données qui enregistre des informations sous forme de vecteurs multidimensionnels représentant certaines caractéristiques ou qualités. [54]

Le nombre de dimensions dans chaque vecteur peut varier largement, allant de quelques-unes à plusieurs milliers, en fonction de la complexité et du niveau de détail des données. Ces données, qui peuvent inclure du texte, des images, de l'audio et de la vidéo.

Les bases de données traditionnelles stockent des chaînes des données scalaires sous forme de lignes et de colonnes. En revanche, une base de données vectorielle fonctionne sur des vecteurs, donc la manière dont elle est optimisée et interrogée est assez différente. [55]

Dans les bases de données traditionnelles, nous interrogeons généralement les lignes de la base de données où la valeur correspond exactement à notre requête. Dans les bases de données vectorielles, nous appliquons une métrique de similarité pour trouver un vecteur qui est le plus similaire à notre requête.

2. Les chunks sont des portions ou des groupes de texte dans un langage naturel qui représentent des unités sémantiques cohérentes

Une base de données vectorielle utilise une combinaison d'algorithmes qui participent tous à la recherche du voisin le plus proche approximatif. Ces algorithmes sont assemblés dans un pipeline qui fournit une récupération rapide et précise des voisins d'un vecteur interrogé. [55]

Métriques de Similarité

Voici quelques exemples de mesures de similarité couramment utilisées dans la littérature. Cependant, le choix d'une mesure de similarité appropriée dépend des caractéristiques des données et des objectifs particuliers de la tâche à réaliser.

- **Similarité Cosinus (Cosine Similarity) :** La similarité cosinus est une mesure de similarité couramment utilisée qui calcule le cosinus de l'angle entre deux vecteurs. Elle est particulièrement utile pour comparer la similarité entre des documents, des plongements de texte ou des données de grande dimension. La similarité cosinus entre deux vecteurs A et B peut être calculée comme dans l'équation 4.5 :

$$\text{cosine similarity}(A, B) = \frac{A \cdot B}{\|A\|\|B\|} \tag{4.5}$$

  où $\cdot$ représente le produit scalaire et $\|\cdot\|$ représente la norme euclidienne.

- **Distance Euclidienne (Euclidean Distance) :** La distance euclidienne est une mesure de dissimilarité couramment utilisée qui calcule la distance en ligne droite entre deux points dans l'espace euclidien. Elle est largement utilisée dans les algorithmes de regroupement, tels que le k-means. La distance euclidienne entre deux vecteurs A et B de même dimension peut être calculée comme dans l'équation 4.6 :

$$\text{euclidean distance}(A, B) = \sqrt{\sum_{i=1}^{n}(A_i - B_i)^2} \tag{4.6}$$

  où A_i et B_i représentent les i-èmes éléments des vecteurs A et B, respectivement.

- **Similarité de Jaccard (Jaccard Similarity) :** La similarité de Jaccard est une mesure couramment utilisée pour comparer la similarité entre ensembles. Elle est particulièrement utile dans le traitement de texte et les systèmes de recommandation. La similarité de Jaccard entre deux ensembles A et B est calculée comme le rapport entre la taille de leur intersection et la taille de leur union comme montré dans l'équation 4.7 :

$$\text{jaccard similarity}(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{4.7}$$

  où $|A|$ et $|B|$ désignent les cardinalités des ensembles A et B, respectivement.

- **Distance de Hamming (Hamming Distance) :** La distance de Hamming est une mesure de similarité utilisée pour comparer des vecteurs binaires de même longueur. Elle calcule le nombre de positions où les éléments correspondants de deux vecteurs sont différents. La distance de Hamming entre deux vecteurs binaires A et B peut être calculée comme dans l'équation 4.8 :

$$\text{hamming distance}(A, B) = \sum_{i=1}^{n}(A_i \oplus B_i) \tag{4.8}$$

  où A_i et B_i représentent les i-èmes éléments des vecteurs A et B, respectivement.

Index Structurel

Une méthode efficace pour améliorer la recherche d'informations consiste à établir une structure hiérarchique pour les documents. En construisant une telle structure, le système RAG peut accélérer la récupération et le traitement des données pertinentes.

- **Index de Graphe de Connaissances :** L'utilisation du graphe de connaissances (Knowledge Graphs - KG) dans la construction de la structure hiérarchique des documents contribue à maintenir la cohérence. Il délimite les connexions entre différents concepts et entités, réduisant ainsi considérablement le risque d'illusions. Un autre avantage est la transformation du processus de recherche d'informations en instructions que les LLM peuvent comprendre, améliorant ainsi la précision de la récupération des connaissances et permettant aux LLM de générer des réponses contextuellement cohérentes, améliorant ainsi l'efficacité globale du système RAG.

En conclusion, la Génération Augmentée par Récupération offre une approche innovante pour améliorer les performances des grands modèles de langage en intégrant la recherche d'informations dans le processus de génération de réponses. En combinant l'indexation, la récupération et la génération, le RAG permet aux modèles de fournir des réponses plus précises et plus informatives, même pour les tâches nécessitant des connaissances externes. L'optimisation de l'indexation, le choix des métriques de similarité appropriées et la structure hiérarchique des données sont des aspects cruciaux pour garantir l'efficacité et la pertinence du processus de récupération. En somme, le RAG représente une avancée significative dans le domaine des modèles de langage et ouvre la voie à des applications plus performantes et mieux informées.

4.10 Conclusion

Dans ce chapitre, nous avons exploré les concepts clés et les techniques avancées dans les domaines de l'apprentissage automatique et du traitement automatique du langage naturel. Nous avons vu comment l'apprentissage automatique, en particulier l'apprentissage profond, a révolutionné la façon dont les machines traitent et interprètent les données. Nous avons également examiné les différentes tâches du traitement automatique du langage naturel, telles que la classification de texte, la traduction automatique et la génération de texte, ainsi que les techniques utilisées pour les accomplir.

Nous avons discuté de l'importance de la tokenisation, de la vectorisation du texte et des embeddings dans le traitement du langage naturel. Nous avons également présenté les Large Language Models (LLMs), qui ont amélioré les performances des tâches de traitement automatique du langage naturel et ont ouvert la voie à de nouvelles applications. Enfin, nous avons abordé la Retrieval-Augmented Generation (RAG), une approche récente qui combine la génération de texte avec la récupération d'informations pertinentes à partir de sources externes, permettant aux modèles de fournir des réponses plus précises et plus informatives.

En somme, l'apprentissage automatique et le traitement automatique du langage naturel sont des domaines en constante évolution qui offrent de nombreuses opportunités pour améliorer l'interaction homme-machine. Les avancées récentes dans ces domaines ont le potentiel de transformer diverses industries en permettant aux machines de comprendre et d'interagir avec les humains de manière plus naturelle et intuitive. Cependant, il reste encore de nombreux défis à relever, tels que la nécessité de disposer de grandes quantités de données annotées, la gestion

de la complexité des modèles et la prise en compte des biais dans les données. Les futures recherches dans ces domaines devront donc se concentrer sur la résolution de ces défis pour permettre une adoption plus large et plus efficace de ces technologies.

Troisième partie

Conception de la Solution

Introduction

Dans le cadre de ce projet de fin d'étude, nous avons entrepris le développement d'un assistant juridique artificiel visant à faciliter l'accès et la compréhension de la législation en Algérie. Ce projet se décline en plusieurs étapes clés, allant de l'extraction des publications légales à partir des Journaux Officiels, à la création d'un chatbot capable de répondre de manière contextuelle et précise aux requêtes des utilisateurs. Cette section présente une vue d'ensemble de l'approche méthodologique adoptée et introduit les chapitres détaillant les différentes phases du projet.

Chapitre 3 : Extraction et Structuration des Données

La première phase du projet consiste en l'extraction et la structuration des publications légales. Ce chapitre explore les algorithmes et techniques utilisés pour extraire les informations des Journaux Officiels algériens. Nous décrivons en détail les méthodes d'extraction de texte brut, et l'utilisation des expressions régulières pour structurer les données extraites. Une attention particulière est accordée à la construction d'un graphe de connaissances, qui permet de représenter les relations complexes entre les différentes publications légales.

Chapitre 4 : Développement du Chatbot Juridique - Workflow et Technologies

La seconde phase se concentre sur le développement du chatbot juridique. Ce chapitre décrit le workflow général et les technologies employées pour transformer les données structurées en un assistant interactif capable de répondre aux requêtes des utilisateurs. Nous détaillons chaque étape du processus, depuis la reformulation des questions, la vectorisation des textes, l'utilisation d'une base de données vectorielle, jusqu'à la génération de réponses contextuelles à l'aide de modèles de langage avancés.

Le diagramme illustrant l'approche suivie est présenté dans la Figure 4.6.

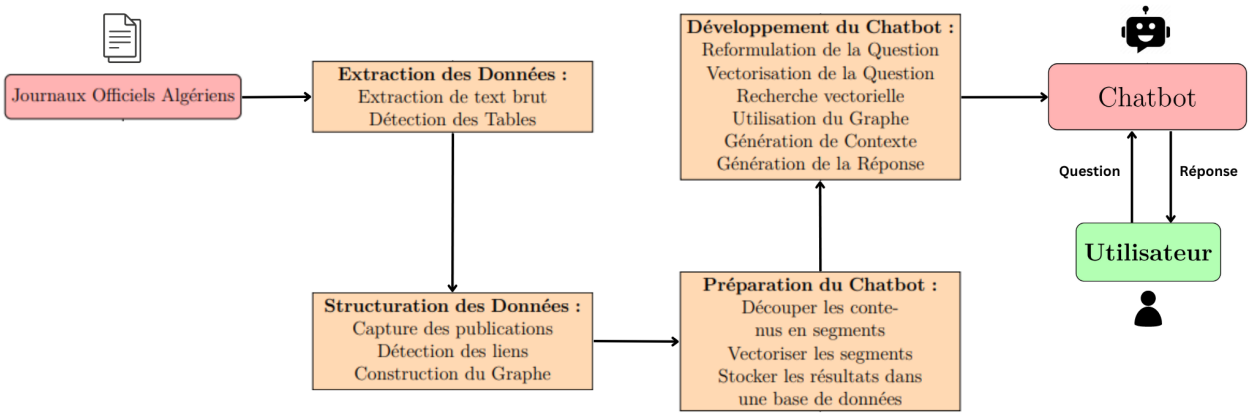


FIGURE 4.6 – Diagramme illustrant l'approche suivie.

Chapitre 5

Extraction et Structuration des Données

5.1 Introduction

L'extraction et la structuration des données sont des étapes cruciales pour la réussite du projet d'assistant juridique artificiel. Ce chapitre se concentre sur les techniques et les algorithmes utilisés pour extraire les publications légales des Journaux Officiels algériens et les structurer de manière à permettre leur intégration dans un système de graphe de connaissances.

Dans un premier temps, nous décrirons l'algorithme général d'extraction et les défis associés à ce processus. Une section dédiée à l'utilisation des expressions régulières expliquera comment ces outils permettent de structurer efficacement les publications, en soulignant la flexibilité et l'importance de ce modèle.

Enfin, nous aborderons la construction d'un graphe de connaissance, en mettant en lumière l'importance de la modélisation en graphe pour capturer les relations complexes entre les différentes publications légales. Les composantes essentielles du graphe et les types de liens entre les publications seront également discutés en détail.

5.2 Algorithme d'extraction

L'étape d'extraction est fondamentale dans notre méthode. Elle vise à convertir les documents PDF des journaux officiels en texte et tables structurés. Les journaux officiels constituent la seule source disponible pour extraire les publications légales, et ces documents sont publiés sur le site du secrétariat général du gouvernement.

L'algorithme proposé traverse diverses étapes pour surmonter les défis liés à la complexité des documents. Ces défis incluent la présence de textes disposés en une colonne, parfois en deux colonnes, l'existence des bordures, des lignes séparatrices et même la complexité des tables avec des cases implicites.

De plus, nous avons choisi d'utiliser les versions françaises de ces documents plutôt que celles en arabe en raison des limitations actuelles des technologies traitant efficacement la langue arabe.

Notre approche combine traitement d'image et analyse structurelle pour une extraction précise du contenu. Nous allons détailler chaque étape de l'algorithme, en mettant en lumière nos choix et considérations.

Algorithme général

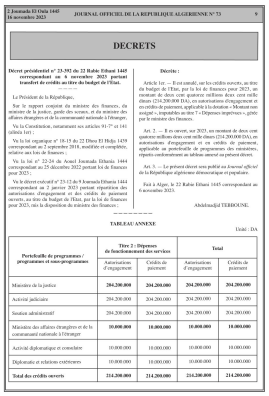
Le processus d'extraction consiste en différentes étapes conçues pour analyser les documents PDF des journaux officiels. Le processus global peut être résumé comme suit :

Algorithme 1 : Algorithme d'extraction

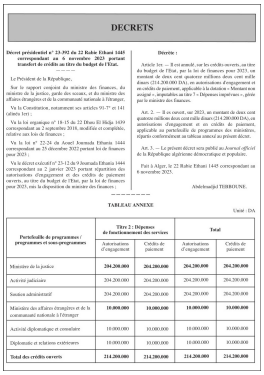
Data : Document PDF représentant un journal officiel
Result : Texte et tables contenus dans le document

```
Extraire les pages;
for chaque page do
    Détecter toutes les lignes horizontales et verticales;
    Enelever les bordures;
    Détecter les lignes verticales séparatrices de texte;
    Détecter les tables (Intersection de lignes horizontales et des lignes verticales);
    Extraire les rectangles représentants les zones de textes et les rectangles
     représentants les tables;
    for chaque rectangle do
        if le rectangle est une zone de texte then
            Extraire le texte contenant dans le rectangle;
        else
            // Le rectangle contient une table
            Détecter les cellules de la tables;
            for chaque cellule do
                Extraire le texte de la cellule;
            Aggréger les textes extraits dans une table pour former la table originale;
return Texte, Tables;
```

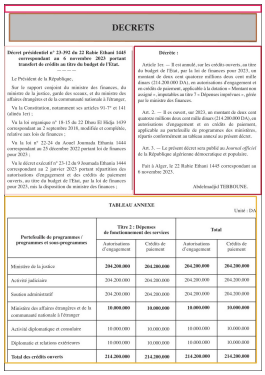
La Figure 5.1 illustre le fonctionnement de l'algorithme.



La page originale



La page après enlèvement des bordures.



La même page avec les différents rectangles.

FIGURE 5.1 – Exemple illustrant le fonctionnement de l'algorithme.

Identification des Lignes Verticales et Horizontales

La phase d'identification des lignes verticales et horizontales constitue une étape fondamentale de notre algorithme d'extraction car elle permet de définir la mise en page, d'isoler les différentes zones de texte et de tables, et de créer une base solide pour les étapes ultérieures du traitement.Le processus s'effectue en suivant les etapes suivantes :

- **Conversion de La Page en Image :** Chaque page du document a été converti en une image. Cette transformation en image a facilité la manipulation des éléments visuels du document.

- **Dilatation et Érosion :** Ces techniques sont utilisées pour améliorer la détection des lignes. La dilatation permet d'élargir les lignes noires sur un fond blanc, facilitant ainsi leur identification. Ensuite, l'érosion réduit les zones élargies pour retrouver les lignes à leur taille d'origine, tout en éliminant les imperfections.

  Ces opérations sont souvent utilisées en duo pour atteindre des objectifs spécifiques et dans notre cas le but était de mieux détecter les lignes présentes dans l'image.

- **Détection des Lignes avec HoughLinesP :** La fonction HoughLinesP de la bibliothèque OpenCV a été ensuite utilisée pour détecter les lignes présentes dans l'image après les transformations. Cette méthode utilise la transformée de Hough pour identifier les lignes droites dans l'image.

Elimination des bordures

L'élimination des bordures vise à préparer les pages document PDF en éliminant les éléments visuels non essentiels, facilitant ainsi les étapes ultérieures du processus d'extraction.

- **Analyse Visuelle des Documents :** Avant de procéder à l'enlèvement des bordures, une analyse visuelle des documents a été réalisée pour identifier les caractéristiques spécifiques des bordures présentes. Cette étape a permis de déterminer que les bordures étaient représentées par trois lignes horizontales en haut, deux lignes en bas, et deux lignes verticales de chaque côté.

- **Délimitation de l'Image :** Après la détection des lignes horizontales et verticales sur l'image de chaque page, l'identification des lignes carréctirisant les bordures a été faite pour déterminer les limites des bordures. Enfin, les coordonnées des lignes détectées ont été utilisées pour délimiter l'image, éliminant ainsi les zones correspondantes aux bordures.

En résumé, l'enlèvement des bordures constitue une étape préliminaire essentielle visant à préparer les page du document en éliminant les éléments visuels indésirables. Cette opération facilite la détection précise des structures textuelles et tabulaires.

Détection des Zones de Texte

Un élément important pour bien détecter le texte est la détection des lignes verticales séparatrices. Elles jouent un rôle important dans la structuration du contenu, en particulier lorsque le texte est organisé en deux colonnes. La détection précise de ces lignes est essentielle pour garantir la cohérence de la lecture du texte.

Les lignes verticales séparatrices de texte sont définies comme des éléments visuels destinés à diviser le texte en deux colonnes distinctes. Ces lignes peuvent être interrompues par des tables, ce qui peut complexifier la détection, car une table peut couvrir toute la largeur de la page, chevauchant ainsi les lignes verticales.

La détection des lignes verticales séparatrices de texte s'effectue en plusieurs étapes :

- La première étape consiste à détecter toutes les lignes verticales présentes sur la page. Cette détection brute sert de base pour les étapes de filtrage ultérieures.

- Ensuite, un filtrage est appliqué pour exclure les lignes verticales qui croisent des lignes horizontales. Ces intersections sont typiquement associées aux structures tabulaires plutôt qu'au texte courant. En éliminant ces lignes, l'algorithme se concentre sur les lignes verticales pertinentes pour la structure du texte, minimisant ainsi les erreurs de classification entre texte et tableaux.

- Pour affiner davantage la détection, seules les lignes verticales situées près du centre de la page sont conservées. Une marge d'erreur ϵ est utilisée pour compenser les petites imperfections de détection. Cette étape vise à exclure les lignes verticales situées aux bords de la page, qui sont souvent des artefacts ou des éléments non structurants.

La détection précise de ces lignes verticales séparatrices est importante pour garantir une lecture cohérente du texte, en suivant l'ordre des colonnes. Cette approche contribue à une extraction de texte plus structurée et facilite la suite du processus d'analyse.

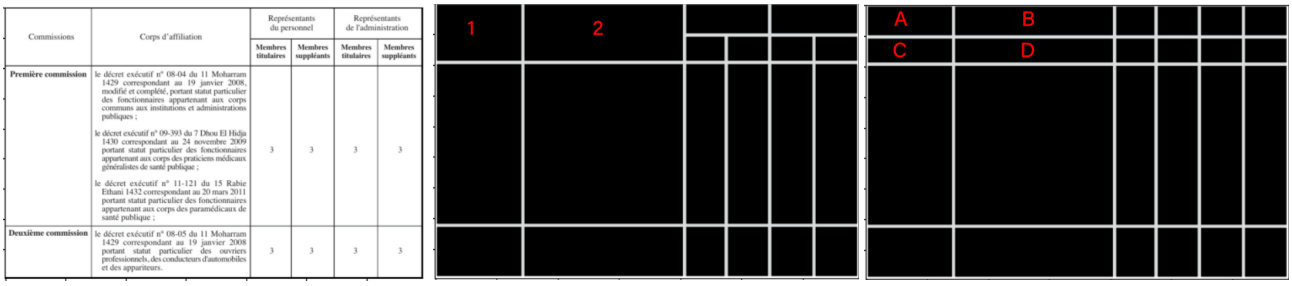
Détection des Tables

La détection des tables constitue une des étapes les plus impoprtantes dans le processus d'extraction, permettant de structurer et d'analyser les données tabulaires présentes dans les journaux officiels algériens. Le processus de détection des tables s'effectue selon la séquence d'étapes détaillée ci-dessous :

- **Identification des Rectangles Représentant les Tables :** À partir des lignes verticales restantes, on détermine les rectangles qui sont l'intersection de lignes verticales et horizontales, formant les contours des tables. Puis, les rectangles les plus grands sont sélectionnés, représentant ainsi les tables potentielles sur la page.

- **Gestion des Lignes Implicites :** Un défi majeur dans la détection des tables réside dans la gestion des ligne implicites ou souvent appelées 'implicit rows'. Pour surmonter ce problème, les lignes verticales et horizontales à l'intérieur des rectangles sont redessinées,

les étendant sur toute la longueur et la largeur du rectangle. Cela garantit une représentation complète des cellules, même si certaines lignes n'étaient pas explicitement présentes dans la table originale.



La table originale. La grille existante. La grille dessinée.

FIGURE 5.2 – Exemple illustrant la gestion des cellules implicites.

- **Extraction du Texte des Cellules :** Les cellules sont extraites en utilisant la grille existante dans la table originale, créant ainsi des rectangles représentant chaque cellule de la table. Le texte est ensuite extrait à partir de ces cellules.

- **Correspondance et Fusion des Cellules :** En comparant les cellules existantes dans la table initiale avec celles dessinées, une correspondance est établie. Si une cellule existante englobe une cellule dessinée, le texte de la cellule existante est transféré à la cellule dessinée. Par exemple dans la Figure 5.2, la cellule '1' englobe les cellules 'A' et 'C' et la cellule '2' englobe les cellules 'B' et 'D'. Ce processus permet de garantir la précision de l'extraction, même en présence de cellules implicites. Un exemple de réussite de notre approche est illustré dans la Figure 5.3.

Filière	Postes supérieurs	Nombre
Administration générale	Chargé d'études et de projet	6
	Attaché de cabinet	4
	Assistant de cabinet	1
	Chargé d'accueil et d'orientation	1
Documentaliste-archiviste	Chargé de programmes documentaires	1
Traduction-interprétariat	Chargé de programmes traduction-interprétariat	1
Informatique	Responsable de bases de données	1
	Responsable de réseaux	1
Statistiques	Chargé de programmes statistiques	1

La table originale.

Filière	Postes supérieurs	Nombre
Administration générale	Chargé d'études et de projet	6
Administration générale	Attaché de cabinet	4
Administration générale	Assistant de cabinet	1
Administration générale	Chargé d'accueil et d'orientation	1
Documentaliste-archiviste	Chargé de programmes documentaires	1
Traduction-interprétariat	Chargé de programmes traduction-interprétariat	1
Informatique	Responsable de bases de données	1
Informatique	Responsable de réseaux	1
Statistiques	Chargé de programmes statistiques	1

La table extraite.

FIGURE 5.3 – Exemple illustrant le succès de l'approche adoptée pour gérer les implicit rows.

Limites et Perspectives de l'Extraction

Notre approche se concentre actuellement sur les journaux officiels publiés depuis 2002. Les numéros antérieurs, souvent numérisés sous forme d'images scannées, présentent des difficultés pour l'extraction automatique du texte. Cependant, nous avons veillé à inclure les lois les plus importantes d'avant 2002. Ces lois, en particulier les codes ou les lois qui entretiennent plusieurs relations avec d'autres lois plus récentes, ont été intégrées manuellement pour assurer la cohérence et la continuité de notre analyse. Nous explorons également activement des solutions pour surmonter les limitations techniques liées aux publications antérieures à 2002 afin d'élargir notre champ d'analyse.

5.3 L'utilisation des Expressions Régulières pour Structurer les Publications

Au-delà de l'extraction du texte brut depuis les journaux officiels, l'étape cruciale d'identification et de capture des publications légales a été réalisée grâce à l'utilisation d'expressions régulières. L'utilisation d'expressions régulières (regex) s'avère être une technique puissante et flexible pour cette tâche, permettant de capturer des éléments clés tels que le type de publication, le pouvoir émetteur, les numéros, les dates selon différents calendriers, les références à d'autres publications, ainsi que les institutions associées.

Méthodologie

L'approche repose sur l'utilisation ciblée de regex pour segmenter les textes, détecter les références inter-publications, identifier les institutions et leurs rôles, ainsi que pour isoler le contenu principal des publications. Ces techniques précises permettent une extraction efficace des données.

1. **Segmentation par Titres et Métadonnées :** Les expressions régulières sont employées pour identifier et extraire les titres des publications, qui sont souvent des points d'entrée clés pour obtenir des informations structurées. Par exemple, le regex a été configuré pour détecter des motifs spécifiques indiquant le type de publication (tel que : loi, décret), le pouvoir émetteur (tel que : présidentiel, ministériel), et le numéro de la publication.

2. **Capture des Dates :** Les dates, essentielles pour la contextualisation temporelle des publications, sont extraites en prenant en compte à la fois le calendrier hégirien et le calendrier Julien-Grégorien, grâce à des expressions régulières adaptées à ces formats.

3. **Gestion des Références et Liens :** Les regex sont aussi utilisés pour détecter les références à d'autres publications au sein des textes, permettant ainsi de construire des relations entre les différentes entités légales.

4. **Identification des Institutions et Contenus :** Les mentions des institutions recommandant ou édictant les publications sont également ciblées à l'aide de motifs regex spécifiques. De même, le contenu principal des publications ainsi que les articles associés sont extraits en appliquant des expressions régulières adaptées.

La figure 5.4 illustre comment le modèle regex isole les différentes informations :

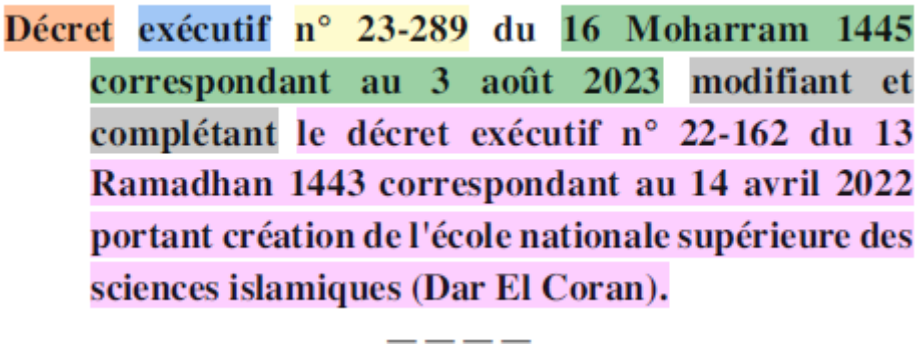


FIGURE 5.4 – Exemple de publication légale structurée par expressions régulières.

Flexibilité du Modèle Regex

Une des forces majeures de cette approche réside dans sa flexibilité à traiter différents cas de figure. Le modèle regex a été conçu pour prendre en compte la variabilité naturelle des publications légales, y compris l'absence partielle ou totale de certaines informations attendues. Par exemple :

- La non-présence de certains éléments tels que les numéros de publication ou les dates dans certains contextes est gérée par des expressions régulières qui tolèrent ces variations.

- Les différentes formulations syntaxiques utilisées pour décrire les références à d'autres lois ou décrets sont prises en compte pour garantir une capture robuste des relations entre les publications.

Importance de l'Utilisation des Expressions Régulières

L'emploi des expressions régulières revêt une importance capitale dans ce contexte. Cette approche permet une extraction précise des informations pertinentes à partir de textes bruts, facilitant ainsi la structuration automatisée des données légales. La flexibilité du modèle regex garantit une couverture étendue des diverses formes que peuvent prendre les publications légales, ce qui est crucial pour assurer la fiabilité et la complétude des données utilisées dans la construction potentielle d'un graphe de connaissance. La Table 5.1 présente les publications extraites.

Nature de publication	**Quantité**
Décret	7184
Arrêté	1351
Loi	609
Décision	260
Ordonnance	90
Règlement	32
Avis	9

TABLE 5.1 – Publications extraites

5.4 Construction du graphe

La construction du graphe constitue une étape fondamentale de notre méthodologie, offrant une représentation structurée des relations entre les différentes publications légales extraites du corpus législatif algérien. Cette section met en lumière l'importance de la modélisation en graphe et présente les composantes clés de ce graphe.

L'Importance de la Modélisation en Graphe

La modélisation en graphe revêt une grande importance dans notre contexte, offrant une approche visuelle et intuitive pour représenter les relations complexes entre les diverses publications légales. En utilisant les nœuds pour symboliser les publications, les articles et les institutions, et en reliant ces entités par des arêtes, le graphe devient un outil puissant pour analyser la structure du corpus législatif.

En plus, cette modélisation est importante en vue de la conception du système RAG qui alimentera notre assistant juridique. En utilisant les relations capturées dans le graphe, le système RAG peut enrichir la génération de réponses en répondant de manière contextuelle et précise aux requêtes des utilisateurs sur la législation algérienne.

Les Composantes du Graphe

Dans notre modèle de graphe, les nœuds représentent les différentes publications légales telles que les lois, décrets, et arrêtés, les articles associés, ainsi que les institutions comme les différents ministères, la présidence, et les secrétariats généraux. Les publications constituent les entités fondamentales de notre graphe, capturant les divers aspects du corpus législatif. Les arêtes du graphe représentent les liens entre ces publications, reflétant les relations de modification, de complétion, d'abrogation et d'autres interconnexions entre les textes, ainsi que les liens avec d'autres types de nœuds. Le schéma du graphe est illustré dans la Figure 5.5

Il est important de souligner que notre graphe est orienté, ce qui signifie que les arêtes ont une direction spécifique, indiquant la relation entre deux publications. Cette orientation des arêtes apporte une dimension logique au graphe, reflétant la la causalité des relations juridiques.

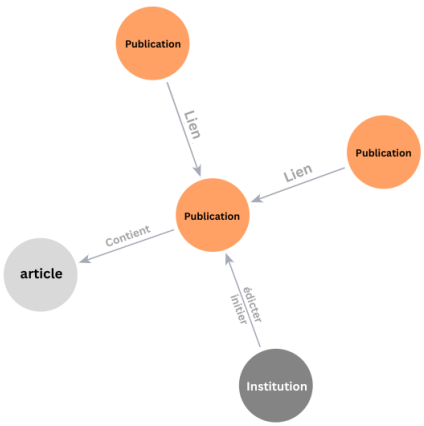


FIGURE 5.5 – Un schéma qui illustre la structure du graphe

Les types de liens entre les publications

Dans le cadre de notre analyse exploratoire des publications légales algériennes, nous avons identifié plusieurs types de liens entre ces publications.

Les types de liens que nous avons identifiés sont les suivants :

- **Annexe et Liste :** Certaines publications sont liés par des annexes ou des listes complémentaires. Ces ajouts spécifiques peuvent fournir des détails supplémentaires, des classifications ou des références utiles pour une compréhension approfondie du texte principal.

- **Modifications Législatives :** Les modifications législatives représentent les changements apportés à une publication existante. Ces modifications ont un impact direct sur la signification et l'application de la publication d'origine.

- **Abrogations et Annulations :** Certaines publications peuvent être annulées ou abrogées par d'autres, ce qui entraîne la suppression ou la mise hors vigueur de la législation précédente. Comprendre ces relations est essentiel pour suivre l'évolution du cadre juridique. Il faut mentionner que dans certains cas, la publication annule quelques articles de la publication d'origine et garde les autres.

- **Approbations :** Certaines publications nécessitent des approbations ou des endorsements pour entrer en vigueur. Ces approbations peuvent provenir d'autres publucations et sont cruciales pour garantir la validité du texte original.

- **Contrôle de Conformité et Constitutionnalité :** Certains textes peuvent faire l'objet d'un contrôle de conformité et de constitutionnalité. Ces évaluations visent à garantir que les lois respectent les principes fondamentaux de la constitution du pays.

- **Extensions et Applications :** Certaines publications peuvent être étendus ou appliqués à des situations spécifiques. Comprendre ces extensions est essentiel pour interpréter correctement la portée d'une loi dans divers contextes. Ces liens indiquent l'élargissement de la portée ou l'application de certaines dispositions à de nouveaux domaines ou groupes.

- **Liens "Vu" :** Un autre type de lien significatif que nous avons identifié est celui mentionnant "Vu". Ce lien se produit lorsque l'on fait référence à une publication légale spécifique dans un autre texte, indiquant ainsi une interconnexion entre les deux. Par exemple, "Vu la loi n° 22-24 du Aouel Joumada Ethania 1444 correspondant au 25 décembre 2022 portant loi de finances pour 2023" établit un lien entre la loi actuelle et celle mentionnée.

En intégrant ces divers types de liens dans notre analyse, nous visons à créer une représentation complète et précise du corpus législatif algérien, facilitant ainsi une exploration approfondie des interactions entre les textes législatifs.

De plus, notre graphe a été déployé sur Neo4j, une base de données de graphes populaire et performante, ce qui facilite la gestion, l'interrogation et la visualisation des données complexes contenues dans notre corpus législatif. La Figure 5.6 représente une visualisation graphique d'une partie du graphe des connaissances juridiques.

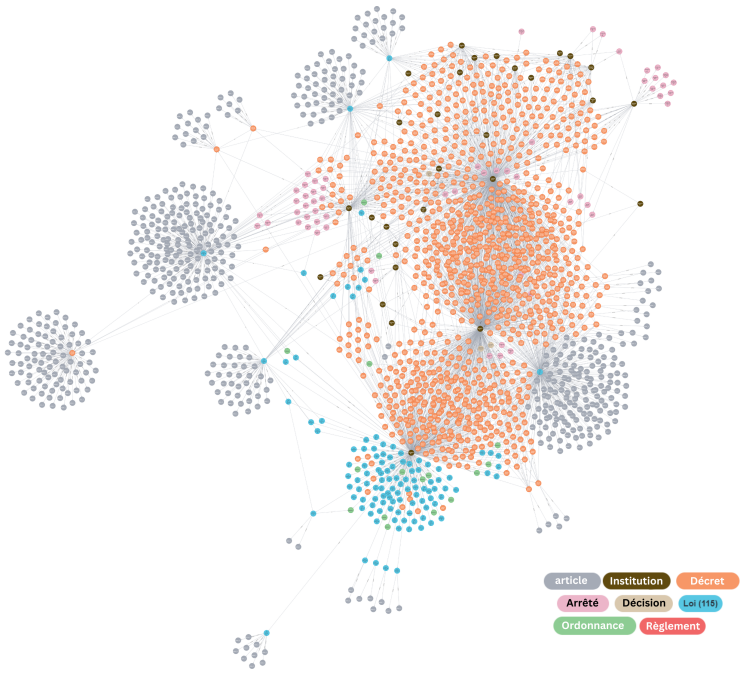


FIGURE 5.6 – Visualisation graphique d'une partie du graphe des connaissances juridiques. Les nœuds correspondent aux publications légales présentées dans le tableau 5.1, aux articles et aux institutions selon les couleurs présentées dans le coin inféreur droit de la figure. Les arêtes représentent les différentes relations entre les nœuds.

5.5 Conclusion

Ce chapitre a détaillé les méthodes et algorithmes utilisés pour l'extraction et la structuration des publications légales à partir des Journaux Officiels algériens. Nous avons commencé par présenter l'algorithme général d'extraction, abordant des techniques spécifiques. Chaque étape a été analysée en profondeur, mettant en évidence les défis rencontrés et les solutions adoptées.

L'utilisation des expressions régulières a été examinée comme un outil essentiel pour la structuration des publications. Nous avons démontré la flexibilité du modèle regex et son importance pour la précision et l'efficacité de l'extraction des informations juridiques.

En conclusion, la construction du graphe de connaissances a été discutée, en soulignant son rôle crucial dans la représentation des relations complexes entre les textes législatifs. La modélisation en graphe permet une navigation intuitive et une meilleure compréhension des interactions entre les différents textes, facilitant ainsi l'accès à l'information juridique.

Les techniques et approches décrites dans ce chapitre constituent les fondations solides sur lesquelles repose le développement de notre assistant juridique intelligent. Dans le chapitre suivant, nous aborderons le développement du chatbot qui permettera de transformer les données structurées en un outil interactif et performant, capable de répondre aux requêtes juridiques de manière précise et contextuelle.

Chapitre 6

Développement du Chatbot Juridique : Workflow et Technologies

6.1 Introduction

Dans ce chapitre, nous allons explorer en détail le processus de développement du chatbot juridique, en mettant en avant le workflow et les technologies utilisées. Le but de ce chapitre est de fournir une compréhension claire et structurée des étapes suivies depuis la structuration initiale des données légales jusqu'à l'interaction finale avec l'utilisateur.

Le développement d'un chatbot juridique efficace et précis nécessite une série d'étapes interconnectées, allant de l'extraction et la structuration des publications légales, à leur vectorisation et indexation, jusqu'à la gestion des requêtes utilisateur et la génération de réponses pertinentes. Chaque étape du workflow est cruciale pour assurer la qualité et la fiabilité des réponses fournies par le chatbot.

Nous commencerons par une vue d'ensemble du workflow, où nous présenterons brièvement chaque étape clé et les technologies sous-jacentes. Ensuite, nous plongerons dans les détails de chaque composante, expliquant les méthodologies, les outils et les défis rencontrés. En fin de chapitre, nous aborderons les méthodes d'évaluation et d'optimisation utilisées pour améliorer continuellement les performances du chatbot.

Ce chapitre est conçu pour offrir une compréhension globale et détaillée du workflow de développement, permettant aux lecteurs de saisir les aspects techniques et méthodologiques impliqués dans la création de cet assistant juridique basé sur l'intelligence artificielle.

6.2 Vue d'Ensemble du Workflow

Le workflow du chatbot juridique se compose de plusieurs modules interconnectés, chacun jouant un rôle crucial dans la transformation de la requête utilisateur en une réponse informative et contextuelle. Ce processus se déroule en plusieurs étapes, depuis la réception de la question de l'utilisateur jusqu'à la génération de la réponse par un modèle de langage. La Figure 6.1 illustre ce workflow à travers un diagramme détaillé.

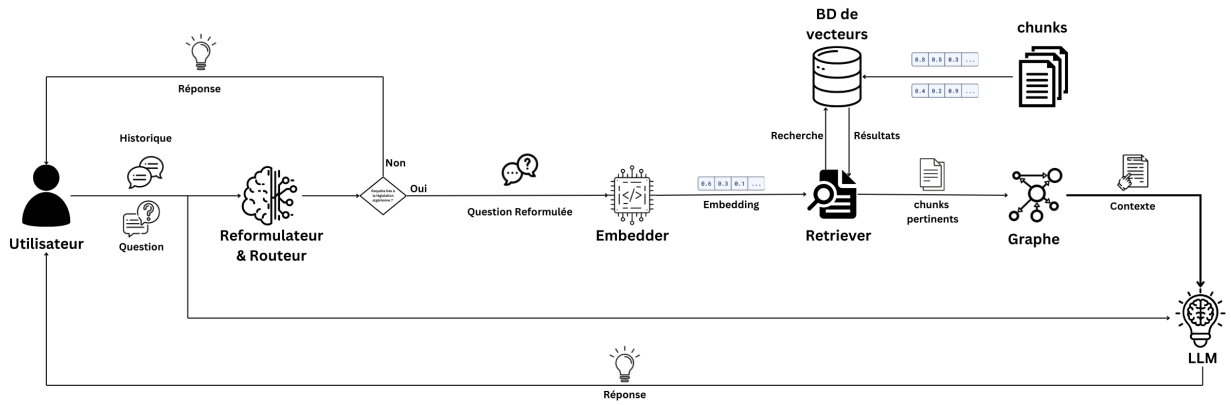


FIGURE 6.1 – Diagramme illustrant le workflow du chatbot.

Interaction de l'Utilisateur

Le processus commence avec l'utilisateur qui soumet une question. Si l'utilisateur a un historique de conversation, celui-ci est également utilisée pour améliorer la compréhension de la question actuelle. Cette question est ensuite transmise au module de reformulation et de routage.

Module de Reformulation et de Routage

Le module de reformulation et de routage joue un rôle crucial dans l'amélioration de la clarté et de la pertinence de la question soumise. Utilisant un modèle de langage, ce module reformule la question initiale en tenant compte de l'historique de conversation pour fournir une version optimisée de la question. De plus, ce module agit comme un routeur : il évalue si la question est liée à la législation algérienne. Si tel est le cas, la question est reformulée et transmise au module d'embedding. Si ce n'est pas le cas, le module de routage explique ses limites à l'utilisateur en générant une réponse telle que : "Je suis désolé, mais je suis conçu pour répondre à des questions spécifiques à la législation algérienne." Cela permet d'assurer que le chatbot se concentre sur les requêtes pertinentes à son domaine d'expertise, tout en clarifiant ses limites lorsqu'il est confronté à des questions en dehors de ce cadre.

Embedding de la Question

La question reformulée est ensuite transformée en une représentation vectorielle par le module d'embedding. Ce processus consiste à convertir la question en un vecteur de dimensions fixes qui capture le sens sémantique de la question.

Recherche dans la Base de Données de Vecteurs

Le vecteur obtenu est utilisé pour effectuer une recherche dans la base de données de vecteurs. Cette base de données contient des embeddings pré-calculés de segments (chunks) des articles contenus dans les publications légales extraites et structurées à partir des journaux officiels algériens. Le module de recherche (retriever) identifie les chunks les plus pertinents en termes de similarité cosinus avec le vecteur de la question.

Sélection et Structuration des Chunks Pertinents

Les chunks pertinents identifiés sont ensuite structurés et envoyés au module de graphe. Ce graphe de connaissances relie les textes légaux et les articles entre eux, facilitant ainsi la contextualisation des informations. Chaque chunk est enrichi par ses relations avec d'autres textes, permettant une compréhension approfondie et contextuelle de la question posée.

Génération de Contexte et Réponse

Le module de graphe fournit un contexte structuré qui est transmis au modèle de langage (LLM). Ce contexte comprend les chunks pertinents et leurs relations, ce qui aide le LLM à comprendre la question dans un cadre légal précis. Le LLM utilise ce contexte et les interactions de l'utilisateur pour générer une réponse précise et informative, qui est ensuite renvoyée à l'utilisateur.

Boucle de Rétroaction

Enfin, la réponse est présentée à l'utilisateur. L'interaction de l'utilisateur avec la réponse, ainsi que toute nouvelle question, réintègre le workflow, enrichissant ainsi l'historique de conversation et améliorant les futures reformulations et recherches.

Ce workflow sophistiqué assure une interaction fluide et efficace entre l'utilisateur et le chatbot juridique, garantissant des réponses précises et contextuelles aux questions légales posées. Chaque module, de la reformulation à la génération de réponses, est conçu pour maximiser la pertinence et la clarté des informations fournies, en utilisant des techniques avancées d'intelligence artificielle et de traitement du langage naturel.

6.3 Vectorisation des Données

La vectorisation des données constitue une étape clé dans le workflow du chatbot juridique. Elle permet de transformer les segments de textes légaux en représentations numériques qui peuvent être efficacement manipulées pour la recherche et la génération de réponses. Voici une description détaillée du processus de vectorisation des données mis en place dans ce projet.

Prétraitement des Données

Avant de vectoriser les textes, il est crucial de passer par une étape de prétraitement. Le prétraitement des données assure la propreté et la normalisation des textes, facilitant ainsi leur utilisation ultérieure.

1. **Suppression des éléments non pertinents :** Enlever les caractères spéciaux, les chiffres inutiles et les espaces superflus.
2. **Normalisation :** Convertir tout le texte en minuscules pour uniformiser les données.

Division en Chunks

Pour rendre les textes légaux plus gérables et exploitables, le contenu de chaque article a été divisé en segments plus petits appelés "chunks". Ce processus a suivi plusieurs étapes méthodiques :

1. **Segmentation des Articles :** Chaque article légal a été analysé et divisé en chunks. L'objectif était d'obtenir des segments de texte qui ne dépassent pas 500 tokens, une taille optimale pour garantir une bonne performance lors de la vectorisation et de la recherche.

2. **Utilisation de la Structure des Textes :** La division en chunks a été réalisée en respectant la structure intrinsèque des contenus. Les paragraphes, les tirets, les sections et autres éléments structurants ont été utilisés pour découper le texte de manière logique et cohérente. Cela permet de préserver le sens et le contexte des informations dans chaque chunk.

Vectorisation des Chunks

Une fois les chunks obtenus, la prochaine étape a consisté à les transformer en vecteurs. Le modèle qui a été utilisé pour la vectorisation est "text-embedding-3-large" de OpenAI. Ce modèle représente l'état de l'art en matière d'embedding et offre une excellente capacité à capturer le sens sémantique des textes. Chaque chunk a été passé à travers ce modèle pour obtenir une représentation vectorielle de dimension fixe.

Intégration des Chunks dans le Graphe

Les chunks vectorisés ont ensuite été intégrés dans le graphe de connaissances, ce qui permet de lier les représentations vectorielles à leurs contextes d'origine :

1. **Ajout des Chunks comme Nœuds :** Chaque chunk a été ajouté en tant que nœud dans le graphe déployé sur Neo4j. Ces nœuds sont reliés à leurs articles d'origine par des relations spécifiques, ce qui permet de conserver une trace claire de leur provenance et de leur contexte.

2. **Attributs d'Embedding :** Les vecteurs obtenus lors de la vectorisation ont été stockés comme des attributs des nœuds de type "chunk". Cela permet de facilement accéder aux embeddings pour des opérations de recherche et d'analyses ultérieures.

Création du Vector Store

Pour permettre une recherche rapide et efficace parmi les chunks vectorisés, une fonctionnalité spécifique de Neo4j a été utilisée.

Neo4j offre des fonctionnalités avancées d'indexation qui ont été exploitées pour créer un vector store. Un index a été créé sur l'attribut "embedding" des nœuds de type "chunk", ce qui permet de réaliser des recherches basées sur la similarité cosinus de manière très efficace.

La vectorisation des données est un élément fondamental du workflow. En divisant les articles légaux en chunks structurés et en utilisant des modèles d'embedding pour les vectoriser, nous avons créé une base solide pour des recherches précises et contextuelles. L'intégration des chunks vectorisés dans le graphe de connaissances et l'utilisation de la fonctionnalité d'indexation de Neo4j pour créer un vector store assurent une efficacité et une performance optimale lors de la recherche et de la génération de réponses.

6.4 Traitement des Requêtes Utilisateur

Le traitement des requêtes utilisateur est une étape cruciale dans le workflow du chatbot juridique. Cette section détaille les processus et technologies utilisés pour reformuler les questions des utilisateurs afin d'améliorer la précision et la pertinence des réponses fournies par le système.

Module de Reformulation et de Routage

Le module de reformulation et de routage, également appelé "Reformulateur et Routeur", est responsable de prendre la requête utilisateur, de la reformuler en une version plus claire et précise, et de déterminer si la requête concerne la législation algérienne. Ce processus est essentiel pour traiter correctement les requêtes complexes ou mal formulées initialement. La Figure 6.2 fait un zoom pour illustrer fonctionnement de ce module.

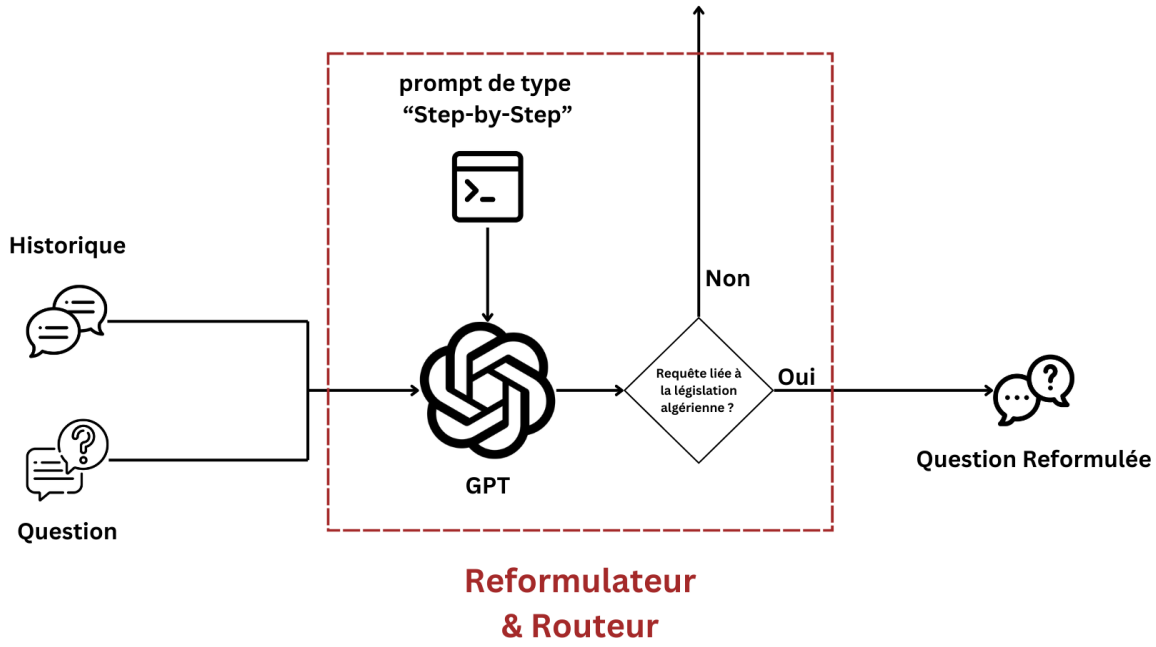


FIGURE 6.2 – Diagramme illustrant le fonctionnement du reformulateur.

1. **Prise en Compte de l'Historique de Conversation :** Lorsque l'utilisateur envoie une requête, cette dernière est combinée avec l'historique de discussion. L'historique comprend les messages précédents échangés entre l'utilisateur et l'assistant, fournissant un contexte riche pour comprendre la nouvelle requête.

2. **Prompt Engineering et Technique "Step-by-Step" :**
   - **Technique "Step-by-Step" :** Pour reformuler efficacement la question, la technique "Step-by-Step" est utilisée. Cette technique consiste à demander au llm de suivre des étapes bien précise pour arriver au résultat attendu.

   - **Prompt Engineering :** Cette technique est mise en œuvre à travers un prompt sophistiqué. Le prompt est conçu pour guider le modèle de langage à produire une reformulation optimale.
3. **Utilisation du LLM (GPT) :**
   - **Modèle GPT :** Le composant de reformulation utilise un modèle de langage de pointe, tel que GPT qui est particulièrement bien adapté à cette tâche en raison de ses capacités avancées de compréhension et de raisonnement.

   - **Output du modèle :** Le modèle décide si la question de l'utilisateur est pertinente et si oui génère une version reformulée de la question qui est claire, concise et contextuellement pertinente.
4. **Routage de la Requête :**
   - **Évaluation de la Requête :** Le module évalue si la question est liée à la législation algérienne. Cette évaluation est basée sur le raisonnement du LLM.

   - **Décision de Routage :**
     - **Requête Pertinente :** Si la question est jugée pertinente pour la législation algérienne, elle est transmise au module d'embedding pour les étapes de vectorisation et de recherche.
     - **Requête Non Pertinente :** Si la question n'est pas liée à la législation algérienne, le module de routage n'essaie pas de fournir une réponse directe. Au lieu de cela, il explique ses limites à l'utilisateur. Par exemple, le modèle pourrait générer une réponse du type : "Je suis désolé, mais je suis conçu pour répondre à des questions spécifiques à la législation algérienne."

Importance du module

La reformulation de la question utilisateur est essentielle pour plusieurs raisons :

1. **Clarté et Précision :**
   - **Amélioration de la Clarté :** Une question bien formulée est plus facile à comprendre et à traiter pour les étapes suivantes du workflow. Cela réduit le risque de mauvaise interprétation de la requête utilisateur.

   - **Précision de la Recherche :** Une question reformulée avec précision conduit à des résultats de recherche plus pertinents lors de la vectorisation et de la recherche dans la base de données de vecteurs.

2. **Optimisation de la Vectorisation :**
   - **Qualité de l'Embedding :** Une question claire et bien formulée permet une vectorisation plus efficace. Le modèle d'embedding peut ainsi générer des vecteurs plus représentatifs du sens réel de la question.

- **Recherche Efficace :** Avec des vecteurs de meilleure qualité, la recherche dans la base de données de vecteurs est plus précise, ce qui améliore les performances globales du chatbot.

3. **Efficacité du Routage :**
   - **Évaluation de la Pertinence :** Le module de routage détermine si la question est pertinente pour la législation algérienne. Cela garantit que seules les questions pertinentes passent par le processus de vectorisation et de recherche, optimisant ainsi les ressources et le temps de traitement.

   - **Gestion des Questions Hors Contexte :** Pour les questions non liées à la législation algérienne, le routage permet de clarifier les limites du chatbot. Cela permet de gérer les attentes des utilisateurs et de maintenir le focus du chatbot sur son domaine d'expertise.

Exemples et Cas d'Usage

Pour illustrer l'importance du traitement des requêtes, considérons les exemples suivants :

- **Exemple 1 :**
  - **Question Initiale :** "Quels sont les droits des employés ?"
  - **Reformulation :** "Quels sont les droits légaux des employés en Algérie ?"
  - **Décision de Routage :** La question est pertinente pour la législation algérienne, elle est donc envoyée au module d'embedding et suit le processus de recherche.

- **Exemple 2 :** Question Complexe avec Historique :
  - **Historique :** L'utilisateur a posé plusieurs questions sur les obligations des entreprises en matière de conformité anti-corruption
  - **Question Initiale :** "Et pour celles concernant les données personnelles ?"
  - **Reformulation :** "Quelles sont les obligations des entreprises en matière de protection des données personnelles en Algérie ?"
  - **Décision de Routage :** La question est pertinente pour la législation algérienne, elle est donc envoyée au module d'embedding et suit le processus de recherche.

- **Exemple 3 :** Question non pertinante :
  - **Question Initiale :** "Comment devenir riche ?"
  - **Reformulation :** "Comment devenir riche ?"
  - **Décision de Routage :** La question n'est pas liée à la législation algérienne. Le module de routage explique les limites du chatbot à l'utilisateur : "Je suis désolé, mais je suis conçu pour répondre à des questions spécifiques à la législation algérienne. Pour toute autre question, veuillez consulter une source appropriée."

Le traitement des requêtes utilisateur via le module de reformulation et de routage est une composante essentielle pour garantir la pertinence et la précision des réponses fournies par le chatbot juridique. En utilisant des techniques avancées de prompt engineering et en exploitant la puissance des modèles GPT, ce module transforme des questions parfois ambiguës ou mal formulées en requêtes claires et contextuellement enrichies, optimisant ainsi chaque étape subséquente du workflow.

6.5 Recherche et Récupération d'Information

La recherche et la récupération d'information sont des étapes centrales dans le fonctionnement du chatbot juridique. Ces étapes permettent de trouver les chunks de texte les plus pertinents à partir de la base de données vectorielle, en réponse à une question utilisateur. Ce processus repose sur des algorithmes avancés de recherche et de similarité, garantissant ainsi la précision et la pertinence des résultats. Le but de cette étape est de récupérer les chunks les plus pertinents qui fourniront le contexte nécessaire à la génération de la réponse finale.

Algorithmes de Recherche

La recherche dans la base de données vectorielle repose sur des algorithmes qui évaluent la similarité entre les vecteurs. Voici les détails des méthodes et algorithmes utilisés :

1. **Utilisation de Neo4j :** Neo4j, la base de données graphique utilisée, offre des fonctionnalités d'indexation des vecteurs. Un index a été créé sur l'attribut "embedding" des nœuds de type "chunk", permettant ainsi des recherches rapides et efficaces.

2. **Approximate Nearest Neighbor :** Pour améliorer l'efficacité des recherches, Neo4j utilise l'algorithme Approximate Nearest Neighbor (ANN). ANN permet de trouver rapidement les vecteurs les plus proches, même dans des ensembles de données très volumineux. Cet algorithme fonctionne en sacrifiant une petite quantité de précision pour un gain substantiel en vitesse, ce qui est crucial pour les applications en temps réel.

3. **Processus de Recherche :** Lorsqu'une question est vectorisée, une requête de recherche est effectuée sur cet index pour trouver les vecteurs les plus similaires. Les résultats sont classés en fonction de leur score de similarité cosinus.

Sélection des Chunks Pertinents

Une fois que les résultats de la recherche sont obtenus, il est crucial de sélectionner les chunks les plus pertinents pour former le contexte nécessaire à la génération de la réponse :

1. **Critères de Sélection :**
   - **Pertinence Sémantique :** Les chunks sont sélectionnés principalement sur la base de leur similarité cosinus avec la question reformulée. Un seuil de similarité est défini pour exclure les chunks moins pertinents.

   - **Contexte Juridique :** En plus de la similarité cosinus, des règles contextuelles spécifiques sont appliquées pour s'assurer que les chunks choisis contiennent des informations juridiques pertinentes et actuelles. Par exemple, les articles de lois abrogées ne sont pas pris en considération.
2. **Agrégation du Contexte :**
   - **Combinaison des Chunks :** Les chunks sélectionnés sont combinés pour former un contexte cohérent. Ce contexte est ensuite utilisé pour enrichir la réponse générée par le modèle de langage.

- **Optimisation de la Longueur :** Il est important que le contexte ne dépasse pas une certaine longueur afin de rester compatible avec les limites des modèles de langage (comme GPT), tout en fournissant suffisamment d'information pour une réponse complète.

Implémentation Technique

L'implémentation de la recherche et de la récupération d'information implique plusieurs composants techniques et étapes :

1. **Requête de Recherche :**
   - **Formulation de la Requête :** Une fois la question vectorisée, une requête de recherche est formulée pour interroger l'index des vecteurs dans Neo4j. La requête spécifie le vecteur de la question et demande les vecteurs les plus similaires.

   - **Exécution de la Requête :** Neo4j exécute la requête en utilisant son index de vecteurs pour retourner les résultats les plus pertinents.
2. **Traitement des Résultats :**
   - **Analyse des Scores de Similarité :** Les résultats sont analysés en fonction de leurs scores de similarité cosinus. Les chunks avec les scores les plus élevés sont considérés comme les plus pertinents.

   - **Filtrage et Sélection :** Les chunks sont ensuite filtrés et sélectionnés en fonction des critères de pertinence définis précédemment.
3. **Construction du Contexte :**
   - **Agrégation des Chunks :** Les chunks sélectionnés sont agrégés pour former un contexte complet et cohérent. Ce contexte est ensuite passé au modèle de langage pour la génération de la réponse.

La recherche et la récupération d'information sont des processus essentiels qui déterminent la pertinence et la précision des réponses fournies par le chatbot juridique. En utilisant des algorithmes avancés de similarité cosinus et en exploitant les fonctionnalités d'indexation de Neo4j, le système est capable de trouver et de sélectionner les chunks de texte les plus pertinents. Ces chunks sont ensuite combinés pour former un contexte riche qui alimente la génération de réponses, assurant ainsi que les utilisateurs reçoivent des informations juridiques précises et utiles.

6.6 Génération de Réponses par le Modèle de Langage

La génération de réponses est une étape cruciale du workflow du chatbot juridique, où le modèle de langage génère des réponses pertinentes et précises en se basant sur le contexte fourni. Dans ce projet, nous utilisons le modèle GPT-3.5-turbo pour cette tâche, en veillant à minimiser les erreurs et à fournir des réponses utiles aux utilisateurs.

Processus de Génération de Réponses

Le processus de génération de réponses suit plusieurs étapes, de la formulation du prompt à la génération de la réponse finale :

1. **Formulation du Prompt :**
   - **Intégration du Contexte :** Le contexte récupéré lors des étapes précédentes est intégré dans le prompt. Cela inclut les chunks pertinents qui ont été sélectionnés et combinés pour former un contexte cohérent.

   - **Ajout des Instructions :** Les instructions spécifiques sont ajoutées au prompt pour guider le modèle. Cela inclut l'instruction de se baser uniquement sur le contexte fourni et de rappeler à l'utilisateur de vérifier les informations.
2. **Appel au Modèle :**
   - **Envoi du Prompt :** Le prompt, comprenant le contexte, les instructions, l'historique de la discussion et le dernier message représentant la question de l'utilisateur, est envoyé au modèle GPT-3.5-turbo via l'API d'OpenAI. L'inclusion de l'historique de la discussion permet au modèle de mieux comprendre la séquence des interactions et d'assurer la continuité et la pertinence de la réponse.

   - **Réception de la Réponse :** Le modèle génère une réponse en se basant sur le prompt enrichi. Cette réponse est ensuite renvoyée au workflow pour être présentée à l'utilisateur.
3. **Post-traitement de la Réponse :**
   - **Formatage et Présentation :** La réponse est formatée de manière appropriée pour être présentée à l'utilisateur de manière claire et compréhensible.

Gestion des Hallucinations

Les hallucinations, où le modèle de langage pourrait générer des informations non vérifiées ou inventées, sont un risque important dans la génération de réponses. Pour minimiser ce risque, plusieurs mesures sont prises :

1. **Instruction de Contexte Strict :** En demandant explicitement au modèle de se baser uniquement sur le contexte fourni, nous réduisons la probabilité que le modèle génère des informations non vérifiées.

2. **Rappel à l'Utilisateur :** En incluant une instruction demandant à l'utilisateur de vérifier les informations auprès des sources officielles, nous renforçons la fiabilité des réponses générées et encourageons une utilisation **responsable**.

Importance de la Responsabilité Utilisateur

Il est essentiel que les utilisateurs ne comptent pas exclusivement sur les réponses générées par le chatbot. En incluant une instruction qui rappelle aux utilisateurs de vérifier les informations auprès des sources officielles, nous visons à :

1. **Promouvoir la Vérification des Sources :** Encourage les utilisateurs à vérifier les informations fournies, augmentant ainsi la confiance et la crédibilité des réponses du chatbot.

2. **Éducation et Sensibilisation :** Aide à éduquer les utilisateurs sur l'importance de la vérification des informations, contribuant à une utilisation plus responsable et informée du chatbot.

La génération de réponses par le modèle de langage GPT-3.5-turbo est une étape critique qui combine les techniques de prompt engineering et des mesures strictes pour minimiser les hallucinations. En intégrant des instructions claires et en rappelant aux utilisateurs de vérifier les informations auprès des sources officielles, le chatbot juridique offre des réponses pertinentes et fiables, tout en responsabilisant les utilisateurs dans l'utilisation des informations fournies.

6.7 Adaptabilité et Gestion des Connaissances

L'adaptabilité et la gestion des connaissances sont essentielles pour assurer que le chatbot juridique reste à jour avec les évolutions législatives. Pour ce faire, un processus automatisé est mis en place pour vérifier et intégrer les nouvelles données chaque semaine.

Processus de Vérification Hebdomadaire

Chaque semaine, un processus automatisé est déclenché pour vérifier la publication de nouveaux journaux officiels (JOs). Ce processus inclut les étapes suivantes :

1. Un script automatisé interroge les sources officielles pour vérifier la présence de nouveaux JOs publiés au cours de la semaine écoulée.

2. Si de nouveaux JOs sont détectés, ils sont automatiquement téléchargés pour être traités.

Extraction des Données

Une fois les nouveaux JOs identifiés et téléchargés, le pipeline d'extraction de données est activé pour traiter ces documents :

- Le contenu de chaque JO est analysé pour extraire les publications légales pertinentes. Cela inclut la détection des nouvelles publications légales, leurs articles et les interactions entre eux.

- Les publications légales sont ensuite formatées et structurées pour faciliter leur intégration dans le graphe de connaissances.

- Chaque publication est décomposée en articles individuels.

- Les interactions entre les nouvelles publications et celles existantes sont identifiées.

Mise à Jour du Graphe de Connaissances

Après l'extraction, les nouvelles données sont intégrées dans le graphe de connaissances existant :

1. **Intégration des Nouvelles Publications :**
   - Les nouvelles publications légales et leurs articles sont ajoutés au graphe de connaissances déployé sur Neo4j.
   - Les relations entre les nouvelles publications et celles existantes sont mises à jour pour refléter les interactions et les dépendances.
2. **Chunking et Vectorisation des Articles :** Les articles des nouvelles publications sont divisés en chunks de moins de 500 tokens. Chaque chunk est ensuite vectorisé et ajouté en tant que nœud dans le graphe, avec l'attribut "embedding" associé.

3. **Mise à Jour de la Base de Données de Vecteurs :** Enfin, les nouvelles données vectorisées sont intégrées dans la base de données de vecteurs

Ce processus rigoureux permet au chatbot juridique de rester constamment à jour avec les dernières évolutions législatives. En intégrant de nouvelles publications légales de manière systématique et en maintenant un graphe de connaissances dynamique et précis, le chatbot offre des réponses pertinentes et actualisées aux utilisateurs, tout en garantissant une gestion efficace des connaissances juridiques.

6.8 Conclusion

La conception et le développement de l'assistant juridique basé sur l'IA ont impliqué une série de processus complexes et interconnectés, visant à offrir un outil robuste et fiable pour répondre aux requêtes juridiques des utilisateurs en Algérie. Dans ce chapitre, nous avons détaillé les différentes étapes du pipeline, depuis l'extraction et la structuration des données jusqu'à la génération de réponses par le modèle de langage. Nous avons commencé par une vue d'ensemble du workflow, décrivant la vectorisation des données, le traitement des requêtes utilisateur, et enfin, la génération des réponses. Chaque composant joue un rôle essentiel dans le fonctionnement global du chatbot, assurant une précision et une pertinence maximales dans les réponses fournies. En maintenant une veille constante sur les évolutions législatives et en intégrant les dernières technologies d'IA, ce projet se positionne comme un outil indispensable pour les professionnels du droit et le grand public en Algérie. Le succès de ce projet ouvre la voie à de nouvelles opportunités pour l'amélioration continue et l'expansion des capacités de l'assistant juridique.

Quatrième partie

Mise en œuvre et Résultats

Chapitre 7

Mise en œuvre pratique du système

7.1 Introduction

Dans ce chapitre, nous nous concentrons sur la mise en œuvre pratique de notre assistant juridique artificiel, en détaillant les différentes étapes et les choix techniques qui ont été effectués pour créer un système fonctionnel et robuste.

Nous commençons par la description de l'architecture du système et de l'intégration des différents composants, tels que l'API backend, la base de données Neo4j et les services d'OpenAI. Nous décrivons ensuite le déploiement du système sur AWS, en utilisant des outils tels que Docker et AWS Lambda pour assurer une haute disponibilité et une scalabilité automatique. Enfin, nous présentons une série d'études de cas pour évaluer la performance de notre assistant, en testant l'exactitude des informations fournies, la pertinence et la clarté des réponses, ainsi que la capacité à gérer les reformulations et les questions complexes.

7.2 Implémentation du système

Dans cette section, nous décrivons la mise en œuvre pratique de notre assistant juridique artificiel. Nous détaillons comment les différentes parties théoriques et techniques ont été intégrées pour former un système fonctionnel et robuste, en mettant l'accent sur l'architecture API, le déploiement sur AWS, l'intégration de Neo4j et la gestion des logs.

Intégration des Composants

L'architecture de notre assistant juridique est basée sur une API centrale qui interagit avec plusieurs composants principaux : l'instance Neo4j pour le stockage et la gestion du graphe de connaissances et du vector store, et les services d'OpenAI pour la génération de réponses.

- **API Backend :** L'API est développée pour gérer les requêtes du frontend et orchestrer les interactions avec Neo4j et OpenAI. Cette API est conçue pour être stateless [1], facilitant ainsi son déploiement et sa scalabilité.

1. ça siginifie que le serveur traite chaque requête de manière isolée, sans se souvenir des requêtes précédentes.

- **Base de données Neo4j :** Utilisée pour stocker et interroger le graphe de connaissances, Neo4j est dockerisé et hébergé sur un serveur dédié. L'API interagit avec Neo4j via des requêtes Cypher pour accéder aux données juridiques et les relations entre elles.

- **Services OpenAI :** Les requêtes utilisateur sont traitées par l'API qui appelle les services d'OpenAI pour la génération des réponses. Ces appels sont enrichis par le contexte extraits de Neo4j. Enfin, les résultats vont être retournés au frontend.

Le schéma d'architecture intégré est illustré dans le diagramme présenté dans la Figure 7.1.

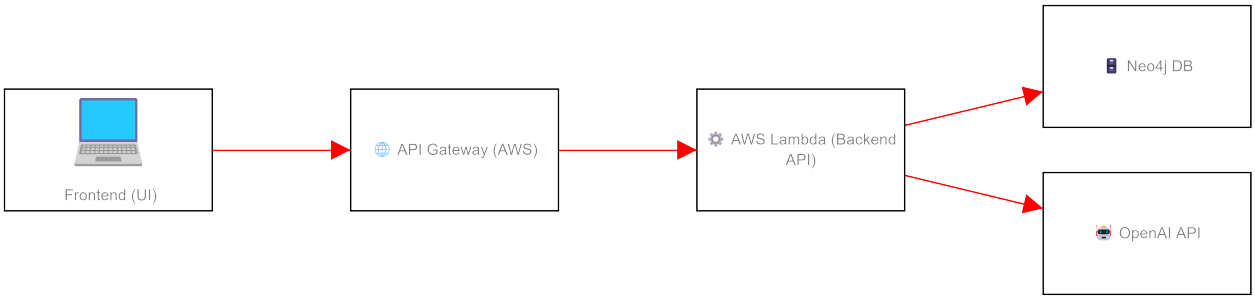


FIGURE 7.1 – Architecture du système.

Déploiement

Le déploiement du système s'appuie sur les services AWS pour assurer une haute disponibilité et une scalabilité automatique :

- **AWS Lambda :** L'API backend est déployée en tant que fonction AWS Lambda, permettant un déploiement serverless qui s'adapte automatiquement à la charge. AWS Lambda facilite également la gestion des versions et les déploiements continus.

- **Docker et Neo4j :** Neo4j est dockerisé et hébergé sur un serveur dédié, offrant une isolation et une portabilité accrues. Le serveur est configuré pour assurer la persistance des données et les backups réguliers.

Le déploiement se fait en plusieurs étapes :

1. **Développement local :** Les fonctionnalités sont développées et testées localement en utilisant Docker pour simuler l'environnement de production.

2. **Déploiement sur AWS Lambda :** Une fois les tests validés, le code est déployé sur AWS Lambda.

3. **Déploiement de Neo4j :** L'instance Neo4j est gérée via Docker Compose, facilitant le déploiement et la gestion des mises à jour.

Maintenance et Surveillance

Pour assurer la performance et la fiabilité du système, nous utilisons plusieurs outils de surveillance comme **CloudWatch (AWS)** qui est utilisé pour surveiller les logs des fonctions AWS Lambda.

La performance du système est constamment optimisée grâce à :

- **Profiling et Tuning :** Analyse régulière des performances des fonctions Lambda et des requêtes Neo4j pour identifier les goulots d'étranglement.

- **Feedback des Utilisateurs :** Les retours des utilisateurs sont collectés et analysés pour améliorer l'expérience utilisateur et la précision des réponses.

En résumé, l'implémentation du système a été réalisée en intégrant diverses technologies modernes pour assurer une haute disponibilité, une scalabilité et une facilité de maintenance. Les choix techniques adoptés, tels que l'utilisation d'AWS Lambda, de Docker et de Neo4j, combinés avec une surveillance rigoureusee, ont permis de créer un système efficace et robuste.

7.3 Études de cas et résultats préliminaires

Dans cette section, nous présentons une série d'études de cas pour évaluer la performance de notre solution. Ces études de cas couvrent différentes situations et types de requêtes, permettant de tester l'exactitude des informations légales fournies, la pertinence et la clarté des réponses, ainsi que la capacité du système à gérer les reformulations et les questions complexes.

Étude de Cas 1 : Consultation d'une loi spécifique

1. **Scénario :** Un utilisateur souhaite consulter la loi de la protection des données personnelles.

2. **Requête :** "Je voudrais connaître les détails de la loi sur la protection des données personnelles en Algérie."

3. **Résultats :**
   - **Exactitude des informations légales fournies :** Le chatbot, comme illustré dans la Figure 7.2, a renvoyé les articles pertinents de la loi de la protection des données personnelles, citant les définitions importantes.

   - **Pertinence et clarté des réponses :** La réponse était claire, bien structurée, et contenait des extraits pertinents de la loi avec des explications supplémentaires lorsque nécessaire.

   - **Capacité à gérer les reformulations :** Lorsqu'on a reformulé la question, par exemple "Quels sont les points principaux de la loi sur la protection des données ?", le chatbot a fourni les mêmes informations de manière concise, démontrant une bonne compréhension des variations linguistiques.

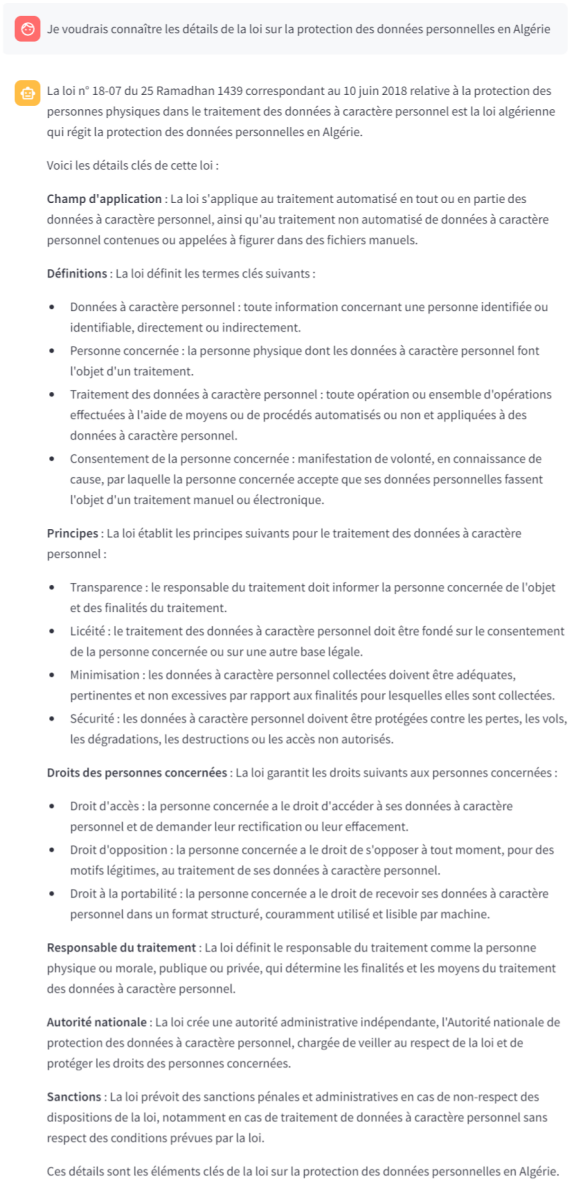


FIGURE 7.2 – Capture d'écran illustrant la réponse de l'assistant

Étude de Cas 2 : Création d'entreprise en Algérie

1. **Scénario :** Un entrepreneur souhaite connaître les étapes pour créer une entreprise en Algérie et les différentes formes juridiques disponibles.

2. **Requête :** "Quelles sont les étapes pour créer une entreprise en Algérie et quelles sont les différentes formes juridiques disponibles ?"

3. **Résultats :**
   - **Exactitude des informations légales fournies :** Le chatbot, comme illustré dans la Figure 7.3, a énuméré les étapes détaillées pour la création d'une entreprise, incluant l'enregistrement, les formalités administratives, et les régulations spécifiques.

   - **Pertinence et clarté des réponses :** La réponse a inclus une comparaison claire des différentes formes juridiques disponibles avec des avantages et inconvénients pour chacune.

- **Capacité à gérer les reformulations :** En réponse à des questions reformulées comme "Comment puis-je démarrer une entreprise en Algérie ?", le chatbot a fourni des informations cohérentes et pertinentes.

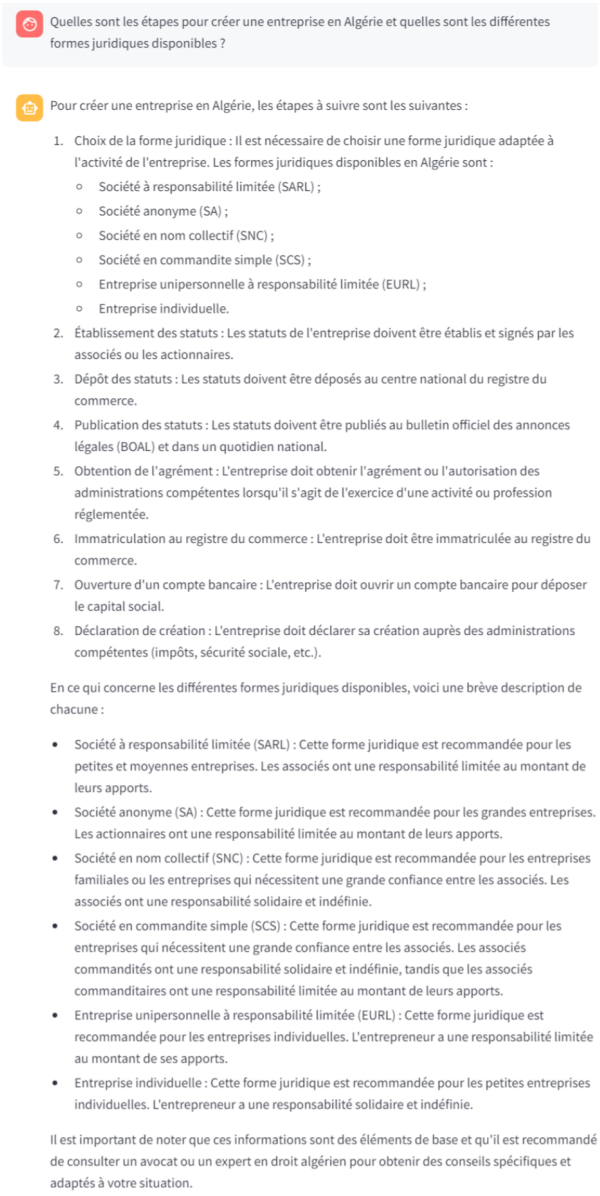


FIGURE 7.3 – Capture d'écran illustrant la réponse de l'assistant

Étude de Cas 3 : Chitchat

1. **Scénario :** Un utilisateur engage une conversation informelle avec le chatbot.

2. **Requête :** "Bonjour, comment ça va ?"

3. **Résultats :**
   - **Exactitude des informations légales fournies :** Non applicable dans ce contexte.

   - **Pertinence et clarté des réponses :** Le chatbot, comme illustré dans la Figure 7.4, a répondu de manière appropriée avec des réponses standardisées telles que "Bonjour ! Je suis ici pour vous aider avec vos questions juridiques. Que puis-je faire pour vous ?".

- **Capacité à gérer les reformulations :** Le chatbot a montré la capacité de répondre de manière cohérente à plusieurs variations de salutations et de questions informelles.

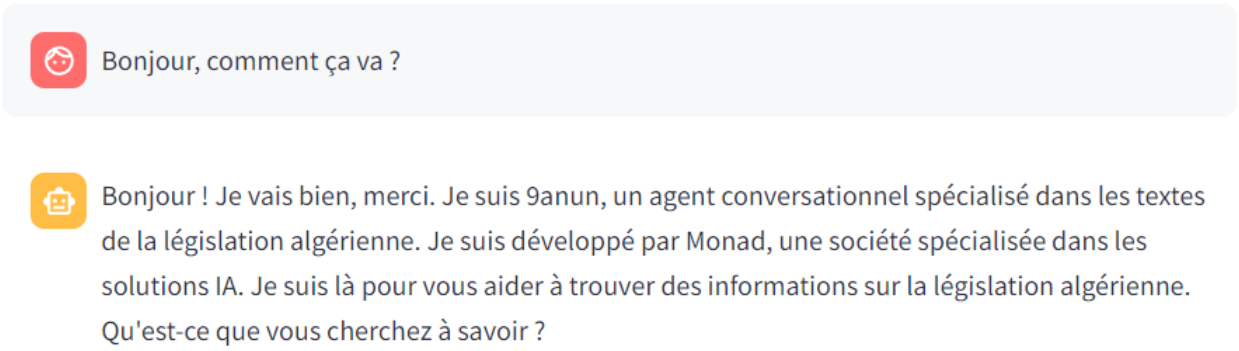


FIGURE 7.4 – Capture d'écran illustrant la réponse de l'assistant

Les résultats des études de cas montrent que notre assistant juridique artificiel est capable de fournir des informations légales précises et pertinentes, de répondre de manière claire et structurée, et de gérer efficacement les reformulations et les questions complexes. Ces résultats préliminaires démontrent la robustesse et l'utilité pratique de notre système pour les utilisateurs finaux.

7.4 Conclusion

En conclusion, notre assistant juridique artificiel a été conçu et mis en œuvre avec succès en utilisant des technologies modernes et des pratiques de développement éprouvées. L'architecture du système, basée sur une API centrale, permet une intégration facile des différents composants et une scalabilité automatique. Le déploiement sur AWS, en utilisant des outils tels que Docker et AWS Lambda, assure une haute disponibilité et une maintenance simplifiée. Les études de cas présentées dans ce chapitre ont démontré la capacité de notre système à fournir des informations juridiques précises et pertinentes, ainsi qu'à gérer efficacement les reformulations et les questions complexes. Ces résultats préliminaires sont encourageants et ouvrent la voie à de futures améliorations et à une utilisation pratique de notre assistant juridique.

Chapitre 8

Évaluation et Validation

8.1 Introduction

Dans ce chapitre, nous nous concentrons sur l'évaluation et la validation de notre système d'assistant juridique artificiel. Nous détaillons les méthodes d'évaluation utilisées pour mesurer l'efficacité et les performances de notre système à travers plusieurs aspects clés, tels que l'extraction de textes, la construction et l'analyse du graphe de connaissances, ainsi que les performances du retriever. En appliquant ces méthodes d'évaluation, nous avons pu obtenir une vue d'ensemble précise et détaillée des performances de notre système, ce qui constitue une base solide pour les améliorations futures et la validation continue de notre assistant juridique artificiel.

8.2 Méthodes d'évaluation

Dans cette section, nous présentons les méthodes utilisées pour évaluer l'efficacité et la performance de notre système à travers plusieurs aspects clés : l'extraction de textes, la construction et l'analyse du graphe de connaissances, ainsi que les performances du retriever.

Évaluation de l'extraction

Pour évaluer l'efficacité de notre méthodologie d'extraction de textes à partir des journaux officiels algériens, nous avons procédé à un échantillonnage aléatoire de documents. Cette approche nous a permis de vérifier la précision et la complétude de l'extraction des publications légales. Les critères d'évaluation comprenaient :

- **Précision :** La proportion des textes extraits correctement identifiés par rapport à l'ensemble des textes présents dans les journaux.

- **Complétude :** La couverture des publications légales extraites par rapport au total attendu.

- **Qualité des textes extraits :** L'exactitude et la clarté des textes après extraction, en vérifiant la fidélité par rapport aux documents sources.

Évaluation du graphe de connaissances

Pour évaluer le graphe de connaissances construit à partir des publications légales extraits, nous avons effectué plusieurs analyses quantitatives et qualitatives :

- **La densité du graphe :** Elle est définie comme le rapport entre le nombre d'arêtes réelles dans le graphe et le nombre maximal d'arêtes possible. Le calcul de cette métrique donne des informations sur le type du graphe.

- **Degré moyen :** Il représente le nombre moyen de connexions entre les publications légales. Cela permet d'évaluer la connectivité et la densité du graphe.

- **Distribution des degrés :** L'analyse de la distribution des degrés nous a permis de comprendre la structure du graphe, en identifiant les nœuds les plus connectés et les clusters denses.

- **Algorithme de Louvain :** Nous avons utilisé l'algorithme de Louvain pour détecter des communautés au sein du graphe. Cet algorithme de détection de communautés maximise la modularité et permet d'identifier des groupes de publications légales fortement interconnectées.

Il est important de noter que ces analyses ont été effectuées sur le graphe des publications légales et leurs interactions, sans inclure les nœuds représentant les institutions, les articles individuels ou les chunks.

Évaluation du retriever

Pour évaluer la performance du retriever, nous avons utilisé le "Mean Reciprocal Rank" (MRR), une métrique standard dans l'évaluation des systèmes de recherche d'information. Le MRR mesure la qualité des réponses en considérant la position de la première réponse correcte dans les résultats de recherche. Nous avons calculé le MRR pour plus de 100 questions générées à partir d'une trentaine de lois algériennes.

Les étapes spécifiques étaient :

- **Formulation des questions :** Génération de plus de 100 questions couvrant divers aspects des lois.

- **Recherche et classement :** Utilisation du retriever pour rechercher les réponses pertinentes dans le graphe de connaissances.

- **Calcul du MRR :** Évaluation de la position de la première réponse correcte pour chaque question et calcul de la moyenne des réciproques des rangs.

En combinant ces méthodes d'évaluation, nous avons pu obtenir une vue d'ensemble précise et détaillée des performances de notre système. Ces évaluations constituent une base solide pour les améliorations futures et la validation continue de notre assistant juridique artificiel.

8.3 Résultats

Dans cette section, nous présentons les résultats obtenus suite aux différentes méthodes d'évaluation appliquées à notre système. Ces résultats couvrent l'efficacité de l'extraction des textes, l'analyse du graphe de connaissances, et les performances du retriever.

Évaluation de l'Extraction

Après l'échantillonnage aléatoire de documents, nous avons comparé dans chaque document le nombre de publications extraites avec ceux présentes dans le sommaire pour évaluer à la fois la précision de l'extraction du texte et la pertinance du pattern regex. Nous avons remarqué une bonne extraction dans la plupart des cas mais cette méthode d'évaluation a aussi révélé quelques problèmes spécifiques présenté dans la Table 8.1

Problème identifié	Description	Solution proposée
Lignes après les tables	Dans certains cas, les premières lignes après une table ne sont pas correctement captées, entraînant une perte d'information dans le texte extrait.	Ajustement de l'algorithme d'extraction
Détection défaillante des lignes séparatrices	Certaines lignes séparatrices trop petites peuvent échapper à notre algorithme de détection, ce qui peut affecter la structure logique du texte extrait.	Amélioration de la détection des séparateurs

Table 8.1 – Problèmes d'extraction identifiés

En conclusion de notre évaluation, notre méthodologie d'extraction a démontré une robustesse considérable dans la majorité des cas, offrant une solution efficace pour la conversion de documents PDF complexes des journaux officiels algériens en texte et tables structurés.

Cependant, nous reconnaissons les défis rencontrés dans certaines situations particulières. Les limitations identifiée soulignent la nécessité d'améliorations spécifiques pour renforcer la fiabilité de notre méthodologie.

Analyse du graphe

Dans cette sous-section, on se consacre à une exploration approfondie du graphe des publications. En analysant la structure du graphe, nous visons à dégager des tendances significatives, à identifier les éléments clés du corpus législatif, et à mettre en évidence des relations complexes entre les publications légales. Le processus expliqué dans les chapitres précédants a abouti à un graphe dirigé non pondéré avec $N_n = 9578$ nœuds et $N_a = 37132$ arêtes.

Densité du Graphe

La densité d'un graphe est définie comme le rapport entre le nombre d'arêtes réelles dans le graphe et le nombre maximal d'arêtes possible. Pour un graphe dirigé, le nombre maximal

d'arêtes est $N_n \times (N_n - 1)$, car chaque nœud peut être relié à tous les autres nœuds.

La formule de densité (D) est présenté dans l'équation 8.1 :

$$D = \frac{N_a}{N_n \times (N_n - 1)} \tag{8.1}$$

Dans notre cas, avec $N_n = 9578$ et $N_a = 37132$, la densité du graphe serait calculée comme suit :

$$D = \frac{37132}{9578 \times (9578 - 1)} \approx 0.000405$$

Les résultats sont résumés dans la Table 8.2 :

Nœuds (Nn)	**Arêtes (Na)**	**Densité**
9578	37132	0.000405

TABLE 8.2 – Densité du Graphe

Donc, la densité de ce graphe dirigé non pondéré est d'environ **0.000405**, ce qui indique que le graphe est probablement assez clairsemé, avec relativement peu d'arêtes par rapport au nombre maximal possible d'arêtes. Cela pourrait suggérer une structure réseau plutôt éparse ou une certaine forme de hiérarchie dans le graphe.

Degrés

Le degré moyen ($\bar{d}$) d'un graphe dirigé est défini comme le rapport entre deux fois le nombre d'arêtes et le nombre de nœuds ($\bar{d} = \frac{2 \times N_e}{N_v}$). Dans le cas de notre graphe avec $N_n = 9578$ nœuds et $N_a = 37132$ arêtes, le calcul donne :

$$\bar{d} = \frac{2 \times 37132}{9578} \approx 7.754$$

Cela signifie que, en moyenne, chaque nœud est connecté à environ **7 autres nœuds** dans le graphe. Ce résultat peut indiquer une connectivité relativement modérée entre les nœuds.

En ce qui concerne l'histogramme des degrés, il peut être intéressant d'examiner la distribution des degrés des nœuds dans le graphe.

L'histogramme suit une distribution de puissance (loi de puissance), et cela pourrait indiquer la présence de nœuds fortement connectés, formant des centres d'influence, tandis que la majorité des nœuds ont un degré beaucoup plus faible. La loi de puissance est courante dans de nombreux réseaux du monde réel, tels que les réseaux sociaux, où quelques individus ont beaucoup plus de connexions que la moyenne.

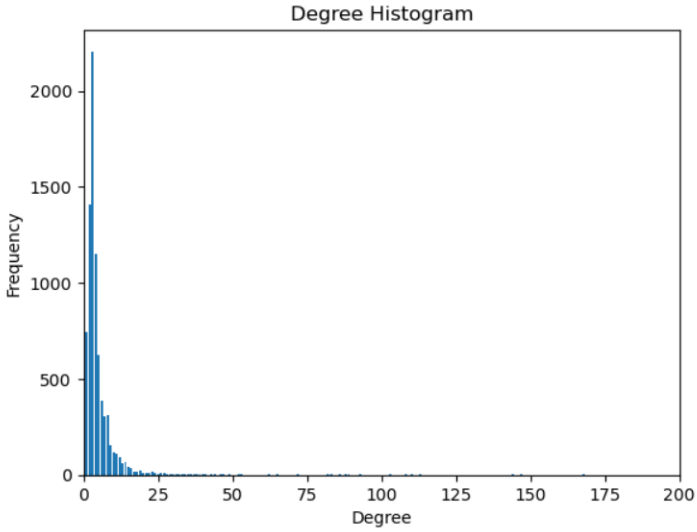


FIGURE 8.1 – Histogramme des degrés

Les noeuds avec plus de degré entrant : Les nœuds avec le plus haut degré entrant dans notre graphe jouent un rôle central car ils sont fréquemment cités, modifiés ou complétés par d'autres publications légales ce qui suggère un niveau élevé d'importance et d'influence de ces lois dans le système juridique algérien. Voici quelques exemples :

- **Loi n° 11-10 relative à la commune (70 arcs entrants) :** Cette loi joue un rôle central dans le système juridique en raison de son degré entrant élevé. Avec 70 arcs entrants, elle est fréquemment citée, modifiée ou complétée par d'autres textes juridiques. Cela indique son importance dans la régulation des affaires communales en Algérie.

- **Loi n° 12-07 relative à la wilaya (69 arcs entrants) :** La loi sur la wilaya, avec un degré entrant de 69 est également cruciale dans le paysage juridique algérien. Son influence étendue suggère qu'elle est souvent utilisée comme référence ou modifiée par d'autres textes législatifs, reflétant ainsi son rôle significatif dans la régulation des entités administratives.

Les noeuds avec plus de degré sortant : Les nœuds avec un degré sortant élevé dans notre graphe ont un impact important sur d'autres publications. Cela indique leur rôle actif dans l'évolution du système juridique algérien.

Voici quelques exemples :

- **Loi n° 84-17 modifiée et complétée, relative aux lois des finances (3513 arcs sortants) :** La loi relative aux lois des finances, avec un impressionnant degré sortant de 3513, joue un rôle central dans la régulation financière en Algérie. Cette loi a un impact étendu sur d'autres textes juridiques notamment les lois de finance annuelles, indiquant son influence dans la gestion des finances publiques et son rôle en tant que référence fréquemment utilisée.

- **Loi n° 90-30 modifiée et complétée, portant loi domaniale (949 arcs sortants) :** Cette loi domaniale, avec un degré sortant de 949, indique son rôle essentiel dans la gestion des biens publics et privés. Les modifications et compléments fréquents signalent son impact étendu sur d'autres textes liés aux propriétés foncières et immobilières en Algérie.

La Table 8.3 présente un résumé des nœuds les plus importants dans le graphe de connaissances juridiques.

Type de Degré	**Loi**	**Nombre d'Arcs**
Plus de Degré Entrant	Loi n° 11-10 relative à la commune	70
	Loi n° 12-07 relative à la wilaya	69
Plus de Degré Sortant	Loi n° 84-17 modifiée et complétée, relative aux lois des finances	3513
	Loi n° 90-30 modifiée et complétée, portant loi domaniale	949

TABLE 8.3 – Résumé des nœuds les plus importants

Cycles dans le graphe

L'identification des sous-graphes cycliques dans notre graphe dirigé revêt une importance particulière lors de l'analyse des publications légales algériennes. Les sous-graphes cycliques, également connus sous le nom de cycles, représentent des structures où une série de publications qui sont interconnectées, formant un circuit fermé. L'absence de sous-graphes cycliques est souhaitable dans les systèmes juridiques, car elle suggère une cohérence et une absence de contradictions internes.

- **Définition :** Dans un graphe orienté, on appelle circuit (ou parfois cycle) une suite d'arcs consécutifs (chemin) dont les deux sommets extrémités sont identiques.[56]

- **Identification des Cycles :** L'analyse a été réalisée en utilisant la procédure 'apoc.nodes.cycles' de 'Neo4j' qui est un outil puissant pour détecter les cycles en utilisant une approche de force brute.

- **Résultats :** Aucune structure cyclique n'a été identifiée dans notre graphe dirigé. Cela indique une cohérence relative et une absence de boucles causales contradictoires au sein de notre graphe.

Communautés Détectées

Dans cette partie, nous explorons les résultats de l'application de l'algorithme de Louvain à notre graphe de connaissances juridiques. Cette étape permet de révéler la structure interne de notre réseau, identifiant des regroupements significatifs au sein du système juridique algérien.

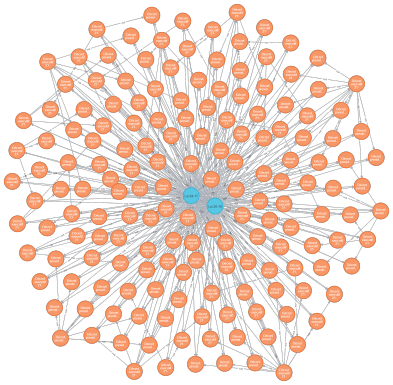
Nous avons fait un choix de deux communautés spécifiques présentant des formes attirantes, offrant une perspective visuelle sur la cohérence interne des relations entre les publications légales.

1. **Gestion Budgétaire de l'Année 2021** : Cette communauté inclut une variété de décrets portant sur la répartition et le transfert de crédits à différents ministères et départements du gouvernement.

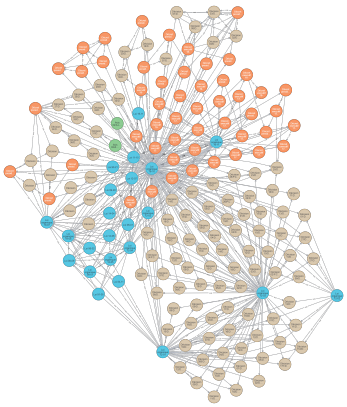
En particulier, ces décrets concernent des domaines tels que les finances, l'éducation, la santé, les affaires étrangères, la culture, l'industrie, l'environnement, les travaux publics, etc. La présence de la loi de finances pour 2021 dans cette communauté suggère que ces décrets sont liés à l'exécution et à la gestion budgétaire de cette année spécifique.

2. **Lois Electorales Algériennes :** La communauté concerne principalement les lois et les réglementations liées aux processus électoraux en Algérie. Cela englobe des lois telles que celles relatives à la prévention de la corruption, au code de procédure civile et administrative, au régime électoral, à la Haute Instance Indépendante de Surveillance des Elections, etc. Les décrets détaillent les différentes modalités et conditions liées aux élections, telles que le vote des citoyens à l'étranger, la délivrance de la carte d'électeur, la publicité des candidatures, et bien d'autres.

La Figure 8.2 présente les représentations des communautés choisies, tandis que la Table 8.4 fournit un tableau détaillé de ces communautés.



La communauté relative à la gestion budgétaire de 2021



La communauté relative aux lois électorales algériennes.

FIGURE 8.2 – Représentations des communautés choisies

Communauté	Caractéristiques Importantes
Gestion Budgétaire de l'Année 2021	- **Centralité :** La loi n° 20-16 et la loi n° 84-17 occupent une position centrale, soulignant leur importance dans l'infrastructure juridique des finances. - **Connectivité :** La communauté est dense avec de nombreux liens entre les textes, indiquant une forte interdépendance où les modifications ou abrogations d'une publication peuvent avoir des effets en cascade sur d'autres. - **Présence de Décrets :** La prédominance des décrets dans cette communauté souligne leur importance dans l'application des lois et l'adaptation du système juridique à des situations spécifiques.
Lois Électorales Algériennes	- **Centralité :** La loi organique n° 16-10 occupe une position centrale, soulignant son importance dans le système électoral algérien. - **Connectivité :** Cette communauté est dense avec de nombreux liens entre les textes, indiquant une forte interdépendance entre les publications.

TABLE 8.4 – Tableau détaillé des communautés choisies

Résultats du retriever

L'évaluation du retriever a été effectuée en utilisant le "Mean Reciprocal Rank" (MRR) pour plus de 100 questions générées à partir d'une trentaine de lois algériennes. La formule du MRR est défini comme dans l'équation 8.2 :

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \max(\frac{1}{k_i}, \text{min\_value}) \tag{8.2}$$

où k_i est la position de la première réponse correcte pour la question i, $|Q|$ est le nombre total de questions et *min_value* est la valeur minimale attribuée lorsque le chunk pertinent n'est pas retourné et nous avons choisi de la mettre "0".

Résultats Détaillés

Le résultat obtenu pour notre système est un MRR de **0.68**. Cela signifie que, en moyenne, la première réponse correcte apparaît au troisième rang dans les résultats de recherche. Voici une analyse détaillée des résultats :

- **Précision :** Le retriever a montré une grande précision, avec la plupart des réponses correctes trouvées parmi les trois premiers résultats. Cela démontre la capacité du système à localiser rapidement les informations pertinentes.

- **Répartition des Rangs :** Une analyse plus approfondie de la répartition des rangs des réponses correctes montre que dans 57% des cas, la réponse correcte est en première position, dans 22% des cas en deuxième position, dans 8% des cas en troisième position, et dans 7% des cas en quatrième position. Les 6% restants se répartissent sur des positions plus basses.

  La Table 8.5 présente la répartition détaillée des rangs des réponses correctes.

Rang	Pourcentage
1er	57%
2e	22%
3e	8%
4e	7%
5e et plus	6%

Table 8.5 – Répartition des rangs des réponses correctes

- **Temps de Réponse :** Le temps moyen de réponse pour le retriever a été mesuré à 4.5 secondes par question, ce qui montre l'efficacité du système en termes de rapidité.

Ce résultat démontre que notre retriever est efficace pour trouver les réponses pertinentes, avec la première réponse correcte apparaissant généralement dans les trois premiers résultats. Cela montre une bonne capacité du système à comprendre et à répondre aux questions des utilisateurs de manière précise et rapide.

8.4 Conclusion

Les résultats obtenus montrent que notre système est performant dans les différentes étapes cruciales de son fonctionnement. L'extraction des textes est précise et complète, le graphe de connaissances est bien structuré et offre des insights utiles, et le retriever est capable de fournir des réponses pertinentes avec une haute efficacité. Ces résultats positifs valident notre approche et montrent que notre assistant juridique artificiel est capable de répondre aux besoins des utilisateurs de manière fiable et précise.

Cinquième partie

Conclusion et Perspectives

Conclusion Générale

En conclusion, ce mémoire a mis en lumière l'importance et la faisabilité de l'application des techniques d'intelligence artificielle dans le domaine juridique, spécifiquement pour le traitement des textes législatifs en Algérie. L'objectif principal était de développer un assistant juridique intelligent capable de répondre aux requêtes des utilisateurs concernant la législation en Algérie, en s'appuyant sur un processus structuré et méthodique.

Nous avons débuté par l'extraction de texte brut à partir des journaux officiels algériens, étape cruciale pour la constitution de notre base de données. Ensuite, nous avons procédé à la détection des textes légaux présents dans chaque journal officiel, ainsi qu'à la capture des relations entre eux, permettant ainsi de construire un graphe de connaissance riche et interconnecté. Les nœuds du graphe représentant les textes légaux et les articles individuels ont été intégrés pour offrir une précision accrue.

La vectorisation des segments de texte et la création d'un vector store ont permis de faciliter la recherche et la récupération d'informations pertinentes. En optimisant le processus de reformulation des questions et de recherche dans le vector store, notre assistant juridique peut fournir des réponses précises et contextuelles basées sur l'historique de conversation et sur la question originale.

L'intégration de ces technologies d'intelligence artificielle dans le domaine juridique présente un potentiel énorme pour améliorer l'accessibilité et la compréhension de la législation en Algérie. En offrant un accès rapide et précis à des informations juridiques complexes, cet assistant juridique peut non seulement aider les professionnels du droit mais aussi les citoyens dans leurs démarches administratives et juridiques.

Néanmoins, il est crucial de continuer à développer et à affiner cet assistant en intégrant de nouvelles fonctionnalités, ainsi qu'en enrichissant continuellement la base de données avec des textes législatifs à jour. De plus, des efforts doivent être faits pour améliorer la compréhension contextuelle et la précision des réponses générées par le système.

En conclusion, ce mémoire a démontré que l'application des techniques d'intelligence artificielle dans la gestion et la recherche de textes législatifs en Algérie est non seulement réalisable mais aussi bénéfique pour la modernisation du secteur juridique. Ces avancées s'inscrivent dans une vision plus large de digitalisation et d'optimisation des processus juridiques, contribuant ainsi à une meilleure accessibilité et efficacité dans l'application de la loi. L'exploitation de ces opportunités technologiques peut significativement transformer le paysage juridique algérien, rendant la législation plus accessible et compréhensible pour tous.

Perspectives futures

Le développement de notre assistant juridique artificiel constitue une base solide pour des améliorations et des extensions futures. Afin de maximiser l'impact de notre projet et de continuer à innover dans le domaine de l'intelligence artificielle appliquée au droit, nous envisageons plusieurs axes de développement pour l'avenir.

Le plan immédiat est de diffuser l'assistant juridique tel qu'il est actuellement développé. Cette première phase de lancement permettra de tester le système en conditions réelles et d'obtenir des retours d'expérience de la part des utilisateurs. Nous prévoyons de collaborer avec divers utilisateurs cibles, y compris des avocats, des entreprises, des étudiants en droit, et des citoyens ordinaires, afin de recueillir des données variées sur l'utilisation et la performance de l'assistant. Les retours des utilisateurs seront essentiels pour identifier les points forts et les faiblesses du système. Nous analyserons ces retours de manière systématique pour orienter les futures améliorations. Cette approche nous permettra de comprendre les besoins et les attentes des utilisateurs, ainsi que les scénarios d'utilisation les plus fréquents et les plus critiques.

Pour améliorer la précision et l'efficacité de notre assistant juridique, nous envisageons de collaborer étroitement avec des avocats et des experts en droit. Leur expertise sera particulièrement précieuse pour affiner l'interprétation des relations complexes entre les textes légaux. Ces experts pourront fournir des informations sur les nuances juridiques et aider à développer des algorithmes plus sophistiqués pour l'analyse contextuelle des lois et règlements. En intégrant les connaissances des experts juridiques, nous pourrons améliorer les modèles pour une compréhension plus fine des interactions entre les différents textes de loi. Nous travaillerons également sur l'optimisation du graphe de connaissances pour représenter de manière plus précise les liens et les dépendances entre les articles et les documents légaux.

Un objectif clé pour l'avenir est d'enrichir l'assistant juridique avec des fonctionnalités avancées, telles que la rédaction et la vérification des contrats. Cette extension permettra aux utilisateurs de générer des documents juridiques conformes aux lois en vigueur, en minimisant les erreurs et les risques juridiques. Nous prévoyons d'incorporer des modèles de génération, capables de produire des clauses contractuelles et de suggérer des modifications en fonction des besoins spécifiques des utilisateurs. La vérification automatique des contrats est une autre fonctionnalité importante que nous souhaitons développer. En utilisant des techniques d'analyse sémantique, l'assistant pourra vérifier la conformité des contrats avec les lois applicables et identifier les éventuelles inconsistances ou risques juridiques. Cette fonctionnalité pourrait également inclure des suggestions pour la correction et l'amélioration des contrats, basées sur les meilleures pratiques et les régulations juridiques actuelles.

À mesure que notre assistant juridique évolue, nous nous concentrerons également sur la scalabilité et la personnalisation du système. L'objectif est de garantir que l'assistant peut être adapté à différents contextes juridiques et aux besoins spécifiques de divers utilisateurs. Des interfaces personnalisées et des options de configuration avancées seront développées pour offrir une expérience utilisateur optimale et répondre à des exigences variées. Nous mettrons en

place un processus d'évaluation continue pour surveiller la performance du système et intégrer les retours des utilisateurs et des experts de manière itérative. Des mises à jour régulières seront planifiées pour améliorer les fonctionnalités existantes et introduire de nouvelles capacités, assurant ainsi que l'assistant reste à la pointe de la technologie et des besoins juridiques.

En conclusion, les perspectives futures de notre assistant juridique artificiel sont prometteuses et s'orientent vers une amélioration continue et une extension des fonctionnalités. En diffusant l'assistant actuel, en intégrant l'expertise des professionnels du droit, et en développant des fonctionnalités avancées comme la rédaction et la vérification des contrats, nous visons à créer une solution complète et évolutive qui réponde aux défis juridiques de manière innovante et efficace.

Bibliographie

[1] Papers with Code - Document Layout Analysis.

[2] DataCamp. Recurrent neural network tutorial (rnn), 2022.

[3] Robail Yasrab and Michael Pound. Phenomnet : Bridging phenotype-genotype gap : A cnn-lstm based automatic plant root anatomization system, 05 2020.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[5] Thierry DELPEUCH, Laurence DUMOULIN, and Claire DE GALEMBERT. Chapitre 1 - le droit dans la régulation sociale. In Thierry DELPEUCH, Laurence DUMOULIN, and Claire DE GALEMBERT, editors, *Sociologie du droit et de la justice*, Collection U, pages 27–54. Armand Colin, 2014.

[6] Vincent Ramette. Algerian legal research. Hauser Global Law School Program, février 2003.

[7] Pdf (portable document format). www.adobe.com/fr/acrobat/about-adobe-pdf.html.

[8] John Whitington. *PDF Explained*. Manning Publications, 2011.

[9] Peter W J Staar, Michele Dolfi, Christoph Auer, and Costas Bekas. Corpus conversion service : A machine learning platform to ingest documents at scale. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, page 774–782. Association for Computing Machinery, 2018.

[10] International Organization for Standardization. Document management – Portable document format – Part 1 : PDF 1.7. ISO Standard ISO 32000-1 :2008, ISO, 2008.

[11] Adobe. Convert PDF to Text Using OCR Software.

[12] H. Wang, C. Pan, X. Guo, C. Ji, and K. Deng. From object detection to text detection and recognition : A brief evolution history of optical character recognition. *WIREs Computational Statistics*, 13(e1547), 2021.

[13] R. Smith and M. J. F. Gales. Google's Tesseract Open Source OCR Engine. *In Proceedings of the International Conference on Document Analysis and Recognition*, pages 629–633, 2007.

[14] Thomas M. Breuel. The OCRopus Open Source OCR System. *In Document Recognition and Retrieval XX*, page 8, 2013.

[15] Galal M. Binmakhashen and Sabri A. Mahmoud. Document layout analysis : A comprehensive survey. *ACM Comput. Surv.*, 2019.

[16] Jilin Wang, Michael Krumdick, Baojia Tong, Hamima Halim, Maxim Sokolov, Vadym Barda, Delphine Vendryes, and Chris Tanner. A graphical approach to document layout analysis, 2023.

[17] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm : Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*. ACM, August 2020.

[18] Sanjiv Bhatia. Regular expressions. *Computer Apex*, 01 2005.

[19] Arthur L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3) :210–229, 1959.

[20] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[21] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[22] Graeme Hine and Deteri. *Neural Network Methods for Natural Language Processing*, volume 37 of *Synthesis Lectures on Human Language Technologies*. Morgan Claypool Publishers, reprint by Springer Nature Switzerland AG, 2017.

[23] Godwin Olaoye. Deep learning approaches for natural language processing : Advancements and challenges. *Machine Learning*, 02 2024.

[24] DataCamp. What is tokenization ?, 2023.

[25] Selva Birunda and R.Kanniga Devi. *A Review on Word Embedding Techniques for Text Classification*, pages 267–281. 02 2021.

[26] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove : Global vectors for word representation. volume 14, pages 1532–1543, 01 2014.

[27] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert : Pre-training of deep bidirectional transformers for language understanding, 2019.

[28] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018.

[29] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018.

[30] Kanchan M.Tarwani and Swathi Edem. Survey on recurrent neural network in natural language processing. *International Journal of Engineering Trends and Technology*, 48 :301–304, 06 2017.

[31] DataCamp. How transformers work : A detailed exploration of transformer architecture, 2024.

[32] Michael Phi. Illustrated guide to transformers- step by step explanation, 2020.

[33] Saidul Islam, Hanae Elmekki, Ahmed Elsebai, Jamal Bentahar, Najat Drawel, Gaith Rjoub, and Witold Pedrycz. A comprehensive survey on applications of transformers for deep learning tasks, 2023.

[34] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models, 2024.

[35] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models : A survey, 2024.

[36] OpenAI. Gpt-4 technical report, 2024.

[37] OpenAI. Planning for agi and beyond, 2023.

[38] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence : Early experiments with gpt-4, 2023.

[39] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models, 2023.

[40] Yihan Cao, Siyu Li, Yixin Liu, Zhiling Yan, Yutong Dai, Philip S. Yu, and Lichao Sun. A comprehensive survey of ai-generated content (aigc) : A history of generative ai from gan to chatgpt, 2023.

[41] Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual chatgpt : Talking, drawing and editing with visual foundation models, 2023.

[42] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12), mar 2023.

[43] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict : A systematic survey of prompting methods in natural language processing, 2021.

[44] Matt Crabtree. What is Prompt Engineering ? A Detailed Guide For 2024, 2024.

[45] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[46] Brown et al. Language models are few-shot learners, 2020.

[47] Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. A systematic survey of prompt engineering in large language models : Techniques and applications, 2024.

[48] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

[49] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models, 2023.

[50] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.

[51] Ravi Theja. Evaluating the Ideal Chunk Size for a RAG System using LlamaIndex — LlamaIndex, Data Framework for LLM Applications.

[52] Recursively split by character | LangChain.

[53] Sophia Yang, PhD. Advanced RAG 01 : Small-to-Big Retrieval - towards Data Science. 11 2023.

[54] Moez Ali. The top 5 vector databases, 2023.

[55] Roie Schwaber-Cohen. What is a Vector Database How Does it Work ? Use Cases + Examples, 2023.

[56] E.A.B.S.G. Williamson. *Lists, Decisions and Graphs*. S. Gill Williamson.

*

ANNEXE

Interface de l'assistant

L'interface de l'assistant juridique "9anoun" a été développée en utilisant Streamlit, un outil populaire pour la création d'applications web interactives en Python. Cette interface est conçue pour être simple, intuitive et facile à utiliser pour les utilisateurs finaux. Elle communique avec une fonction Lambda qui sert d'API backend, assurant ainsi une interaction fluide et rapide entre le front-end et les fonctionnalités backend.

Page de connexion

La première page présentée à l'utilisateur est la page de connexion. Voici les éléments clés de cette page :

- **Titre de la page :** "9anoun - L'assistant Juridique Algérien" Formulaire de connexion : La page contient un formulaire de connexion avec deux champs :
  - **Username :** Un champ de texte où l'utilisateur doit entrer son nom d'utilisateur.
  - **Password :** Un champ de mot de passe pour entrer son mot de passe.
  - **Bouton de connexion :** Un bouton "Login" qui permet de soumettre les informations de connexion.
- **Message d'information :** Un message d'information en bas de la page indique à l'utilisateur d'entrer son nom d'utilisateur et son mot de passe et affiche une erreur en cas d'informations erronées.

La Figure A.1 représente une capture d'écran de la page de connexion :

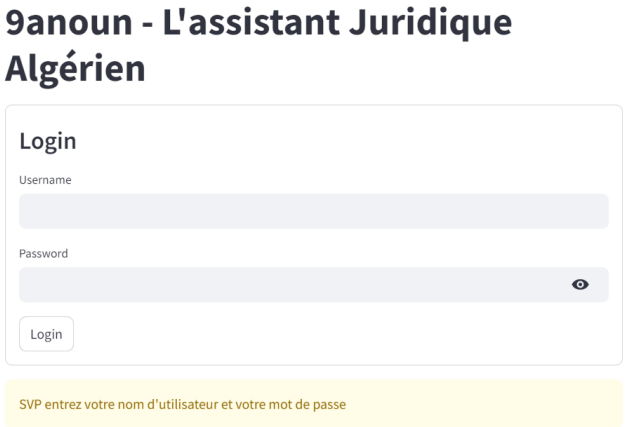


Figure A.1 - Capture d'éran illustant la page de connexion.

Page d'accueil après connexion

Après une connexion réussie, l'utilisateur est redirigé vers la page d'accueil. Les éléments clés de cette page incluent :

- **Message de bienvenue :** Un message personnalisé de bienvenue affichant le nom de l'utilisateur connecté.
- **Menu de navigation :** Un menu de navigation simple sur la gauche avec les options suivantes :
  - **Nouveau Chat :** Un bouton pour démarrer une nouvelle conversation avec l'assistant juridique.
  - **Logout :** Un bouton pour se déconnecter de l'application.
  - **Sources :** Une liste de documents et références juridiques utilisés pour fournir des réponses aux utilisateurs.
- **Champ de saisie de texte :** Au centre de la page, il y a un champ de saisie de texte où l'utilisateur peut entrer sa requête juridique.

La Figure A.2 représente des captures d'écran de la page d'accueil après connexion :

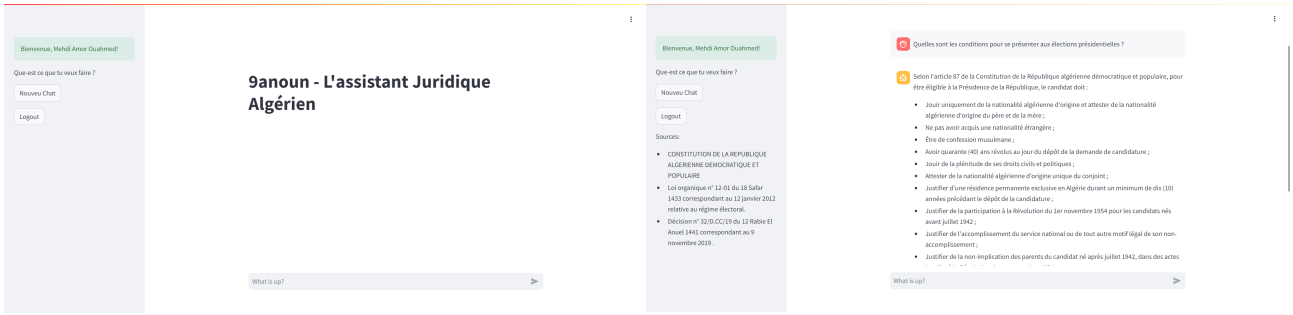


Figure A.2 - Captures d'éran illustant la page d'accueil.

Fonctionnement de l'interface

L'interface utilise Streamlit pour sa simplicité et son efficacité dans le développement d'applications web interactives. Streamlit permet de créer des éléments d'interface utilisateur de manière rapide et réactive, ce qui est idéal pour le développement d'un prototype pour les applications web. Voici comment l'interface communique avec le backend :

1. **Entrée de l'utilisateur :** Les utilisateurs interagissent avec l'interface via des champs de texte et des boutons.

2. **Envoi des données :** Les données saisies par l'utilisateur sont envoyées à une fonction Lambda via des appels API.

3. **Traitement des requêtes :** La fonction AWS Lambda traite les requêtes en passant le dernier message et l'historique de conversation pour fournir des réponses précises.

4. **Retour des résultats :** Les résultats sont ensuite renvoyés à l'interface utilisateur, où ils sont affichés de manière lisible et intuitive. Cette architecture permet une interaction rapide et efficace entre l'utilisateur et l'assistant juridique, offrant ainsi une expérience utilisateur optimisée.

Conclusion

L'interface de l'assistant juridique "9anoun" est conçue pour être intuitive et facile à utiliser, avec une navigation simple et des interactions fluides. Le choix de Streamlit pour le développement front-end, associé à une fonction Lambda pour le backend, permet de créer une application robuste et réactive, capable de répondre efficacement aux besoins des utilisateurs en matière d'assistance juridique.