Université Saad Dahleb - Blida 1
Faculté des Sciences
Département Informatique

Mémoire du Projet de Fin d'Étude pour l'obtention du

Master en Informatique

Option : Traitement Automatique de la Langue

TechBot Sonatrach

Auteurs :

KHELLADI Sidali
DOUID Mohamed

Membres du jury :

Président :	**Mme. CHIKHI Imane**
Examinatrice :	**Mme. MIDOUN Khadidja**
Encadrant Académique :	**Mr. CHABA MOUNA Mustapha**
Encadrant Professionnel :	**Mme. YOUNES Samia**

Année scolaire : 2024-2025

Remerciements

Nous remercions tout d'abord notre Dieu Tout-Puissant à qui nous devons la force, la volonté et la clarté d'esprit ayant permis de mener à bien ce travail.

Nous remercions également chaleureusement les membres du jury, pour le temps qu'ils nous consacrent, leur implication et l'intérêt qu'ils portent à notre travail. Leur évaluation représente pour nous un aboutissement significatif de cette étape académique.

Nous tenons à témoigner notre profonde gratitude à Monsieur CHABA MOUNA Mustapha, notre encadrant académique pour son accompagnement rigoureux, ses conseils éclairés et son soutien constant tout au long de ce projet. Sa disponibilité, sa rigueur et son regard attentif ont été d'une aide précieuse à chaque étape de notre progression.

Au terme de ce stage, nous souhaitons exprimer notre profonde gratitude à l'ensemble de l'équipe de Sonatrach, et plus particulièrement à la Direction des Technologies de l'Information du RPC, pour nous avoir accueillis dans un environnement aussi stimulant, bienveillant et professionnel.

Nos remerciements les plus sincères vont à notre encadrante Mme. YOUNES Samia, pour sa confiance, son accompagnement attentif, sa disponibilité et son écoute, qui ont grandement contribué à faire de ce stage une expérience enrichissante et humaine. Nous tenons également à adresser une mention toute particulière à M. TAKJOUTE Amine et M. BOUZAIR Djamal, pour leur précieuse aide, leurs conseils avisés et leur implication constante. Leur pédagogie et leur professionnalisme ont été déterminants dans la réussite de ce projet. Travailler au sein de Sonatrach fut une véritable opportunité, c'est avec passion et fierté que nous avons mené à bien ce travail et nous repartons enrichis, motivés de cette experience.

Nous adressons nos remerciements distingués à l'ensemble des enseignants du département d'Informatique de l'USDB, qui ont su nous transmettre leur savoir et leurs connaissances avec dévouement, nous permettant ainsi de progresser tant sur le plan académique que personnel, et de mieux envisager notre avenir professionnel.

Nos sincères remerciements vont à toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce projet.

Dédicaces Sid Ali

Je dédie ce mémoire à ma famille et mes amis.

Je souhaite tout particulièrement exprimer ma gratitude à l'égard de mes parents, pour leur patience, leur soutien inébranlable et la confiance qu'ils m'ont toujours accordée. Leur présence constante m'a offert un cadre rassurant, des repères solides, et les moyens concrets pour tracer mon chemin et rendre possible ce parcours. À mes quatre formidables sœurs, pour leur présence indéfectible, leur soutien sous toutes ses formes, et leur capacité à me faire évoluer. Elles ont su, par leurs conseils, leur écoute et leur regard bienveillant, me pousser à croire en moi, à viser plus haut, et à garder la tête haute même dans les moments les plus incertains. Chacune d'elles m'a aidé à devenir une meilleure version de moi-même, et je leur en suis profondément reconnaissant. Je dédie ce travail également à mes tantes, oncles et cousins.

À mes amis, pour leur présence fidèle, leurs encouragements sincères et leur pincée d'humour quotidienne.

Ce mémoire est le fruit de tout cet accompagnement, et je leur adresse ma plus sincère gratitude.

Dédicaces Mohamed

Je dédie ce mémoire à mes parents pour leur soutien et leurs sacrifices, à ma famille et mes enseignants pour leur accompagnement, ainsi qu'à tous ceux qui m'ont encouragé tout au long de ce parcours.

Résumé

Dans un environnement industriel aussi exigeant que celui de Sonatrach, l'accès rapide et fiable à l'information technique représente un enjeu majeur pour les équipes d'ingénierie, de maintenance et d'exploitation. Les documents techniques comme les schémas PID, fiches techniques et les listes d'instruments sont souvent stockés sous forme de fichiers PDF non structurés, rendant leur consultation lente, fastidieuse et sujette à l'erreur.

Ce mémoire propose la conception et la réalisation d'un chatbot intelligent, appelé TechBot Sonatrach, capable de comprendre les requêtes formulées en langage naturel et d'y répondre de manière pertinente en s'appuyant sur une base documentaire interne. Pour atteindre cet objectif, deux volets complémentaires ont été développés : d'une part, un système d'extraction automatique de données techniques à partir de fichiers PDF, utilisant des bibliothèques spécialisées comme pdfminer, fitz et pdfplumber ; d'autre part, un chatbot basé sur l'architecture RAG (Retrieval-Augmented Generation), combinant recherche sémantique et génération de texte pour offrir des réponses contextualisées.

L'ensemble du système s'appuie sur une chaîne de traitement rigoureuse, allant de la structuration des données en JSON jusqu'à leur indexation vectorielle via un format Markdown enrichi. Le résultat est un assistant virtuel capable de dialoguer efficacement avec les utilisateurs, de réduire considérablement le temps de recherche documentaire, et de valoriser les connaissances techniques au sein de l'entreprise.

Ce travail s'inscrit dans une vision plus large de la digitalisation des processus industriels et illustre concrètement le potentiel de l'intelligence artificielle pour améliorer la productivité, la capitalisation du savoir et l'aide à la décision dans un contexte industriel réel.

Mots clés : Chatbot intelligent, Intelligence Artificielle, Industrie, Traitement du Langage Naturel, Automatisation.

ملخص

في بيئة صناعية تتسم بالتعقيد والمتطلبات العالية مثل بيئة شركة سوناطراك، يُعد الوصول السريع والدقيق إلى المعلومات التقنية تحدياً كبيراً لفرق الهندسة والصيانة والاستغلال .فغالباً ما تكون الوثائق الحيوية مثل مخططاتP&ID ، والنشرات الفنية، وقوائم الأجهزة محفوظة على شكل ملفات PDFغير منظمة، مما يجعل تصفحها عملية بطيئة ومرهقة وعرضة للأخطاء.

يُقدم هذا البحث تصميم وتنفيذ روبوت محادثة ذكي يُعرف باسمTechBot Sonatrach ، قادر على فهم الاستفسارات المطروحة بلغة طبيعية والرد عليها بشكل دقيق بالاعتماد على قاعدة بيانات وثائق تقنية داخلية .لتحقيق هذا الهدف، تم تطوير محورين متكاملين :الأول يتمثل في نظام استخراج تلقائي للبيانات التقنية من ملفات PDF باستخدام مكتبات بايثون متخصصة مثل pdfminer و fitz وpdfplumber، والثاني يتمثل في روبوت محادثة مبني على معمارية) RAG الاسترجاع المعزز بالتوليد(، يجمع بين البحث الدلالي وتوليد النصوص لتقديم إجابات سياقية دقيقة.

يعتمد النظام على سلسلة معالجة صارمة تبدأ من تنظيم البيانات بصيغة JSON مروراً بتحويلها إلى تنسيق Markdown محسّن من أجل فهرسة فعّالة باستخدام تقنيات البحث الشعاعي .والنتيجة هي مساعد افتراضي قادر على التفاعل الطبيعي مع المستخدمين، وتقليص وقت البحث عن المعلومات الفنية بشكل كبير، وتحقيق أفضل استثمار للمعرفة التقنية داخل المؤسسة.

يندرج هذا العمل ضمن الرؤية الأوسع لرقمنة العمليات الصناعية، ويُبرز بشكل ملموس إمكانيات الذكاء الاصطناعي في تحسين الإنتاجية، وتثمين الخبرات، ودعم اتخاذ القرار في بيئة صناعية واقعية.

.

Abstract

In an industrial environment as demanding as that of Sonatrach, quick and reliable access to technical information is a major challenge for engineering, maintenance, and operations teams. Critical documents — such as PID diagrams, technical datasheets, and instrument lists — are often stored as unstructured PDF files, making their consultation slow, tedious, and error-prone.

This thesis presents the design and implementation of an intelligent chatbot, named TechBot Sonatrach, capable of understanding queries expressed in natural language and responding accurately based on an internal technical document base. To achieve this, two complementary components were developed : first, an automated data extraction system from technical PDF files using specialized Python libraries such as pdfminer, fitz, and pdfplumber ; second, a chatbot based on the RAG (Retrieval-Augmented Generation) architecture, combining semantic search and text generation to deliver contextualized responses.

The system relies on a rigorous processing pipeline, from data structuring in JSON format to vector indexing through enriched Markdown. The result is a virtual assistant that can interact naturally with users, significantly reduce the time spent searching for technical documents, and enhance knowledge utilization within the company.

This work fits into the broader vision of industrial process digitalization and clearly demonstrates the potential of artificial intelligence to improve productivity, knowledge capitalization, and decision support in a real-world industrial context.

Keywords : Sonatrach, TechBot, Intelligent Chatbot, Artificial Intelligence, Industry, Natural Language Processing, Automation.

Table des matières

Table des figures

Liste des tableaux

Liste des Acronymes

— **RPC :** Raffinage des Produits Chimiques.

— **TechBot Sonatrach :** Chatbot Intelligent pour l'Analyse et la Recherche de Documents Techniques chez Sonatrach.

— **NLP :** Traitement du Langage Nature (Natural Language Processing).

— **RAG :** Génération augmentée de récupération (Retrieval-Augmented Generation).

— **LLM :** Modèle de Langage de Grande Taille (Large Language Models).

— **PID :** Schémas de tuyauterie et d'instrumentation (Piping and Instrumentation Diagrams).

— **IE :** Information Extraction.

— **IA :** Intelligence Artificielle.

— **TF-IDF :** Term Frequency-Inverse Document Frequency.

— **NER :** Named Entity Recognition.

— **OCR :** Optical Character Recognition (Reconnaissance Optique de Caractères)

— **A.L.I.C.E :** Artificial Linguistic Internet Computer Entity

Introduction Générale

Créée en 1963, la société Nationale pour la recherche, la production, le transport, la transformation et la commercialisation des hydrocarbure, plus connue sous le nom de Sonatrach, est la clé économique de l'Algérie. Elle est l'une des plus grandes entreprises d'Afrique et occupe une place centrale dans l'industrie pétrolière et gazière modiale. Sonatrach est responsable de l'exploitation et du développement des ressources naturelles du pays, en particulier les hydrocarbures. L'entreprise couvre l'ensemble de la chaîne de valeur énergétique : exploration, production, transport par canalisation, transformation et commercialisation des hydrocarbures ainsi que de leurs dérivés [1].

Du point de vue production, les chiffres témoignent de l'envergure des opérations de Sonatrach. En 2005, la production totale atteignait 232.3 millions de tonnes équivalent pétrole (TEP), dont près de 11.7% étaient destinés à la consommation du pays. L'Algérie, à travers Sonatrach, se positionne également parmi les leaders mondiaux dans les segments du gaz naturel liquéfié (GNL), du gaz pétrole liquéfié et du gaz naturel brut. Ce dernier fait partie d'un projet d'investissement continu visant à augmenter la production et de garantir un approvisionnement stable en énergie, aussi bien pour le marché local que pour l'exportation.

Dans le cadre de ce stage, l'intérêt est porté à la chaîne de raffinage. Cette dernière est un maillon essentiel du secteur pétrolier. Elle assure les opérations de transformation primaire du pétrole brut en produits finis. C'est un domaine qui requiert une main-d'œuvre diversifiée et composée d'ingénieurs procédés, de techniciens de maintenance, de chefs d'unités et d'opérateurs de terrain. Ces experts interviennent dans un environnement technique complexe où les décisions doivent être prises à un rythme soutenu et s'appuyer sur des données précises et fiablement documentées. Leur expertise assure l'efficacité, la sécurité et la coordination optimale de l'ensemble des opérations de raffinage.

Dans leurs activités, les équipes de raffinerie traitent un grand nombre de

1. `https://fr.wikipedia.org/wiki/Sonatrach`

documents techniques, à savoir, les PID, les datasheets, les listes d'instruments et les fichiers I/O. Ces informations sont distribuées et présentées de manière non structurée. De plus, ces informations sont souvent volumineuses et complexes à exploiter. Par conséquent, l'accès rapide à une information pertinente, à partir des rapports, peut être un processus lent et sujet aux erreurs, limitant ainsi l'efficacité des usines. En effet, ces dernières éprouvaient des difficultés à exploiter efficacement des données essentielles à la prise de décision rapide, ce qui compromettait l'actualisation des décisions opérationnelles et techniques dans un environnement aussi exigeant que celui du raffinage.

Afin de pallier ce problème, le recours à l'intelligence artificielle, en particulier au traitement automatique du langage naturel (NLP), offre de nombreux avantages à Sonatrach. Cette technologie permet une recherche rapide dans les documents techniques, une compréhension contextuelle des requêtes, ainsi qu'une réduction des erreurs humaines. Les employés gagnent ainsi du temps et accèdent plus facilement aux informations pertinentes. Cependant, l'IA présente également certaines limites : elle peut mal interpréter des requêtes mal formulées, ignorer des termes trop spécifiques au domaine ou fournir des réponses imprécises lorsque les données sont mal structurées.

Le fait que l'interface de chatbot intelligent offre la possibilité d'interaction dynamique est une considération clé. Les utilisateurs interrogent le bot sur les questions spécifiques et reçoivent des réponses pertinentes et contextualisées, par opposition à la navigation manuelle et statique à travers de volumineux enregistrements PDF. Tout le monde peut l'utiliser facilement, et il s'intègre confortablement dans les systèmes existants. Les produits se développent progressivement en fonction de leur utilisation – un argument sérieux en faveur d'un bot intelligent, en raison de sa capacité à offrir une solution précise et flexible aux besoins des unités commerciales de Sonatrach et soutenant la prise de décision d'affaires.

Contrairement à une recherche textuelle classique, le chatbot nécessite une extraction préalable des données. Les informations provenant des documents techniques sont automatiquement extraites, nettoyées puis stockées dans une base NoSQL optimisée pour des requêtes rapides. Cette structure évite d'avoir des requêtes directes sur le contenu brut du fichier PDF. Cela permet au chatbot de générer des réponses précises et rapides sans erreurs d'interprétation. Il s'agit ainsi d'une solution technique fiable pour effectuer une recherche.

L'un des principaux avantages du chatbot est l'amélioration de l'efficacité opérationnelle. Au lieu de lire manuellement des documents, les équipes techniques

utilisent simplement des requêtes en langage naturel pour extraire rapidement les informations requises en quelques secondes. Cette solution génère un gain de productivité considérable, qui se traduit également par une plus grande précision dans la prise de décision sur le terrain. Par conséquent, les utilisateurs de chatbot peuvent consacrer leur temps à réaliser pleinement leurs tâches et non à la recherche d'information.

Enfin, ces types de chatbots garantissent également la préservation et la transmission du savoir technique. Les informations contenues dans l'expertise des spécialistes, comme les acronymes, les normes ou la connaissance des liens entre les divers composants, deviennent accessibles à tous. Cela aide notamment les jeunes ingénieurs et les nouveaux arrivants à acquérir facilement des informations cruciales. Par conséquent, le système de chatbots aide à capitaliser les connaissances techniques, ce qui réduit la dépendance à l'égard des experts expérimentés.

Le chatbot sera disponible à partir d'un navigateur Web local, hébergé en interne chez Sonatrach. En plus de protéger la confidentialité des données, cette solution offrirait aux utilisateurs locaux un accès sécurisé à partir de leurs postes de travail connectés au réseau industriel. En outre, un chatbot moderne disposera d'une interface utilisateur simple et ne nécessite donc pas d'installation de solution cloud externe. La société préserverait sa souveraineté en ce qui concerne les informations stratégiques et le niveau de confidentialité des données.

En résumé, l'intégration d'un chatbot intelligent répond aux enjeux de gestion documentaire industrielle chez Sonatrach. Ce projet s'inscrit dans une démarche de digitalisation des processus, offrant une solution adaptée aux besoins des équipes techniques. En simplifiant l'accès aux informations, en valorisant les données techniques et en améliorant la productivité, ce système constituerait un outil performant et évolutif pour l'environnement industriel de Sonatrach.

Chapitre 1

État de l'Art

Face à la problématique de recherche d'informations pertinentes dans des documents techniques volumineux à partir d'un chatbot intelligent, ce chapitre propose un état de l'art. Il offre un aperçu global des techniques et méthodes couramment utilisées dans le domaine du traitement automatique des documents PDF, du traitement du langage naturel (NLP), de la recherche d'information intelligente et de l'intelligence artificielle. L'accent est mis sur les solutions répondant aux exigences d'un système de type chatbot destiné à la consultation de documents techniques dans la filière de raffinage de Sonatrach.

1.1 Historique des Chatbots

1.1.1 Définition générale d'un chatbot

Un chatbot, ou agent conversationnel, est un programme informatique conçu pour simuler une conversation humaine. Il peut interagir avec un utilisateur à partir du texte ou de la voix, en répondant automatiquement à des questions ou en exécutant des tâches spécifiques [1]. Les chatbots peuvent être basés sur des scripts simples (règles conditionnelles) ou sur des systèmes d'intelligence artificielle (IA) capables d'apprendre et de s'adapter.

Le but d'un chatbot est de fournir une interface conviviale, permettant un accès simplifié à des services, des données ou des réponses à des questions, sans nécessiter d'intervention humaine directe.

1.1.2 Premiers chatbots (ELIZA, PARRY)

L'histoire des chatbots débute dans les années 1960, et deux modèles marquants sont présentés ci-dessous :

- **ELIZA (1966) :** Développé par Joseph Weizenbaum au MIT, ELIZA simulait un psychothérapeute en reformulant les propos de l'utilisateur. Son fonctionnement reposait sur des règles basées sur des mots-clés. Malgré sa simplicité, ELIZA a marqué les esprits et posé les bases du traitement du langage naturel (NLP) [2].
- **PARRY (1972) :** Créé par Kenneth Colby, PARRY simulait une personne atteinte de schizophrénie paranoïde. Contrairement à ELIZA, PARRY intégrait une logique plus complexe, incluant des «croyances» et des «intentions», ce qui le rendait plus réaliste [3].

Ces premiers modèles étaient déterministes, sans capacité d'apprentissage, mais ont ouvert la voie à l'étude de l'interaction homme-machine.

1.1.3 Évolution des chatbots (A.L.I.C.E, Jabberwacky)

Avec l'évolution du Web et des moteurs de recherche, les chatbots ont connu une nouvelle vague de développement au début des années 2000. Plusieurs systèmes ont émergé, notamment :

- Des assistants virtuels intégrés à des sites web pour guider les utilisateurs (ex. : A.L.I.C.E, Jabberwacky) [4], [5].
- Des agents conversationnels dans les jeux vidéo et les applications éducatives.

À cette époque, les chatbots reposaient principalement sur des scripts manuels et des arbres décisionnels. Leurs réponses étaient limitées, peu adaptatives, mais suffisantes pour des tâches simples.

1.1.4 Chatbots modernes avec l'IA (Watson, Siri, Alexa, ChatGPT)

Le tournant majeur dans l'évolution des Chatbots est survenu avec l'intégration de l'intelligence artificielle. Cela a conduit à l'émergence des chatbots les plus connus à ce jour, présentés ci-dessous :

- **Watson d'IBM (2011) :** Ce Chatbot est capable de répondre à des questions complexes et de participer à des jeux télévisés comme Jeopardy !, Watson combinait le NLP, l'analyse sémantique et le machine learning [1].

1. `https://www.ibm.com/watson`

- **Siri (Apple)** [2], **Alexa (Amazon)** [3], **Google Assistant** [4] **:** Ces assistants vocaux utilisent la reconnaissance vocale, le NLP, l'apprentissage automatique et la connectivité cloud pour offrir des services personnalisés et contextuels.
- **ChatGPT (OpenAI 2022) :** Cette génération de chatbots est basée sur de grands modèles de langage (LLMs). ChatGPT est capable de comprendre et de générer du langage naturel avec un haut degré de fluidité, ouvrant la voie à des usages professionnels variés [5].

Ces agents intelligents peuvent traiter une vaste base de données, contextualiser les interactions et améliorer leur performance grâce à l'apprentissage.

1.1.5 Transition vers les LLMs

Les LLMs, comme GPT-3, GPT-4, BERT, ont révolutionné le fonctionnement des chatbots. Ces modèles sont entraînés sur d'immenses volumes de données et s'appuient sur des architectures neuronales, principalement des transformers, pour comprendre et générer du texte. Grâce aux LLMs, les chatbots offrent de nombreux avantages [6], notamment :

- Le maintien d'une conversation fluide et contextuelle.
- La production de réponses nuancées, même dans des domaines techniques.
- La capacité à résumer, traduire, analyser, ou extraire des informations complexes.

Toutefois, les LLMs présentent également certaines limites [7] :

- Consommation élevée de ressources, notamment en termes de calcul et de mémoire.
- Dépendance à la qualité des données d'apprentissage, pouvant entraîner des biais ou des réponses incorrectes.
- Difficulté d'intégration des connaissances en temps réel.

1.1.6 Techniques utilisées dans les chatbots

Les techniques utilisées varient selon le type et l'époque du chatbot [6] et sont présentées dans le tableau 1.1.

2. https://en.wikipedia.org/wiki/Siri
3. https://en.wikipedia.org/wiki/Amazon_Alexa
4. https://en.wikipedia.org/wiki/Google_Assistant
5. https://openai.com/index/chatgpt/
6. https://web.stanford.edu/~jurafsky/slp3/

Technique	Description	Exemples d'usage
Pattern Matching	Réponses basées sur des mots-clés ou motifs dans la phrase de l'utilisateur	ELIZA, A.L.I.C.E
Scripting / Arbre décisionnel	Dialogue préprogrammé avec embranchements logiques	Chatbots de service client
NLP	Analyse grammaticale, sémantique et contextuelle des textes	Watson, Siri
Machine Learning	Capacité d'apprendre à partir d'exemples et à s'adapter	ChatGPT, Replika
Transformers & LLMs	Modèles de deep learning avec attention multi-têtes pour générer un langage naturel fluide	GPT-4, Claude, LLaMA

TABLE 1.1 – Techniques utilisées dans les chatbots

1.1.7 Types de chatbots

Il existe plusieurs catégories de chatbots, classés selon leur complexité ou leur finalité [8] :

- **Chatbots basés sur des règles :** Ces chatbots répondent uniquement à des entrées prédéfinies. Simples à concevoir, ils restent toutefois limités en flexibilité. Ils conviennent bien aux FAQ ou aux systèmes de menus guidés.
- **Chatbots intelligents (avec IA) :** Les chatbots intelligents exploitent le NLP et le machine learning pour gérer des demandes complexes. Ils offrent une interaction naturelle, comme le font ChatGPT, Alexa ou Google Assistant.
- **Chatbots transactionnels :** Ils sont conçus pour exécuter des actions précises, comme réserver, commander ou planifier. Ils intègrent des API pour interagir avec des services externes.
- **Chatbots hybrides :** Ces chatbots combinent des règles fixes avec de l'intelligence artificielle, ce qui leur permet de couvrir un plus large éventail de cas d'usage tout en restant contrôlables.

1.2 Documents industriels : rôle et utilisation

Dans le domaine de l'industrie pétrolière et gazière, la prise en charge des équipements, installations et flux de production s'appuie sur une documentation technique, au sens large, extrêmement dense, nécessaire au bon fonctionnement des

unités industrielles depuis la conception et l'exploitation jusqu'à la maintenance et à la sécurité. Elle regroupe des documents très variés dont les qualités sont normalisées, chacun ayant une place et un rôle propres dans la chaîne de traitement technique. Ces documents sont introduits dans ce qui suit.

1.2.1 Les PID

Les schémas PID (Piping and Instrumentation Diagrams) représentent le cœur de la documentation relative à l'installation industrielle. Ils symbolisent les canalisations, les équipements (pompes, vannes, échangeurs, etc.), les instruments de mesure et de contrôle, ainsi que les connexions logiques entre ces éléments [9]. Un PID est souvent, pour un ingénieur ou un technicien, le premier document consulté lors d'un diagnostic ou d'une intervention. Ces documents permettent de comprendre le fonctionnement d'un procédé dans sa globalité, d'identifier les instruments par leur tag et de retrouver rapidement un composant en cas d'urgence ou de panne. [7]

1.2.2 Les datasheets

Les fiches techniques sont les documents décrivant les spécifications techniques des équipements industriels (capteur, vanne, transmetteur, etc.). On y retrouve les informations suivantes : la plage de fonctionnement, les matériaux, les connexions et les normes de fabrication. Ces fiches techniques sont indispensables à la maintenance, aux remplacements ou aux études de compatibilité entre les composants.

1.2.3 Les listes d'instruments

La liste d'instruments ou liste instrumentale est un tableau de synthèse qui rassemble tous les appareils de mesure, de régulation ou de contrôle d'un processus industriel. Elle sert de fondement pour : le suivi de l'équipement, la gestion de la maintenance et l'analyse fonctionnelle du système. Chaque ligne de cette liste correspond à un instrument identifié par un tag unique, souvent mentionné dans les PID et les fiches de données [8].

1.2.4 Les listes des entrées/sorties

La liste d'entrée/sortie est surtout utilisée dans les systèmes de contrôle-commande, soit DCS ou PLC. Elle définit les signaux physiques (analogiques ou

7. `https://www.lucidchart.com/pages/fr/schema-pid`
8. `https://en.wikipedia.org/wiki/Datasheet`

numériques) échangés entre les instruments de terrain et les automates, et permet de configurer les automates programmables, de faire des diagnostics pour résoudre les problèmes d'automatisation et d'assurer la traçabilité entre capteurs/actuateurs et système de supervision.

Les différents documents, décrits dans cette section, constituent une source d'information incontournable, mais leur volume, leur diversité, et leur structure complexe rendent leur exploitation manuelle difficile. C'est dans ce contexte que des approches automatiques d'extraction et de structuration s'imposent comme une solution pertinente et nécessaire, notamment dans le cadre du projet TechBot Sonatrach.

1.3 Extraction automatisée de données technologiques

L'extraction automatisée d'information, également désignée par l'IE (Information Extraction), est une discipline du traitement automatique du langage, visant à transformer des documents bruts (textes, tableaux, PDF, etc.) en données structurées et exploitables. Dans le cadre industriel de la société Sonatrach, l'automatisation de cette tâche est d'autant plus importante aujourd'hui que le volume d'informations à traiter est non seulement en forte croissance, mais aussi de plus en plus complexe, avec un rythme de mise à jour de plus en plus soutenu. Certaines méthodes d'extraction sont introduites dans cette section.

1.3.1 Méthodes classiques d'extraction

Les premières techniques reposent sur des approches déterministes simples mais rigides, parmi elles :

L'expressions régulières (Regex) permettant de récupérer, à l'aide d'expressions définies, des motifs donnés dans les textes [9], tels que les tags d'instruments (par exemple 510-TT-0012), des plages de valeurs, ou bien des noms de composants. Ces méthodes deviennent d'une grande rapidité, mais elles sont peu robustes au formatage.

Le parsing manuel permet d'avoir une analyse structurée au cas par cas, ligne par ligne ou par blocs, en se fiant à la cohérence supposée des fichiers.

9. `https://datascientest.com/regex-tout-savoir`

L'extraction tabulaire primaire utilisée pour les fichiers bien structurés comme les listes d'instruments, un simple traitement ligne/colonne est suffisant dans certains cas.

Des limitations de ces méthodes sont rapidement constatées dans certains cas lorsque les documents sont hétérogènes dans leur format, incomplets ou utilisant des symboles techniques ou des abréviations de domaine.

1.3.2 Outils modernes d'extraction à partir de PDF

Les fichiers PDF techniques sont des ensembles d'objets pas uniquement textuels. Ils comprennent des tableaux, des structures multi-colonnes, des annotations, etc. Pour en effectuer l'extraction d'information de façon plus robuste, de nouveaux outils sont donc apparus, parmi eux, pdfminer qui permet de réaliser une extraction de texte brut avec un positionnement des mots en coordonnées de page. Cet outil n'est utile que lorsque l'on veut analyser la structure d'un document en territoire spatial. De plus, pdfplumber est considéré comme un outil puissant d'extraction de tableaux ou de blocs organisés, particulièrement pratique pour les listes d'instruments ou des interfaces / E/S. Enfin, PyMuPDF (fitz) est un outil qui donne un accès bas niveau au contenu du PDF, gestion des blocs, des polices, et de la structure de page quasi à la demande. Ces outils permettent une bonne compréhension du document, mais nécessitent un prétraitement lourd pour nettoyer le bruit, reconstruire les structures logiques, mais aussi interpréter les blocs de texte. Une étude comparative de ces outils a été abordée dans [10].

1.3.3 Prétraitement et normalisation

L'étape de prétraitement est indispensable avant toute structuration. L'étude menée dans [11] montre que l'étape de prétraitement est indispensable avant toute structuration. Cette étape comporte la suppression des caractères inutiles (caractères spéciaux, espaces multiples), le regroupement de lignes fragmentées et la fusion des blocs séparés dans le PDF mais logiquement liés. Le but est d'obtenir une représentation textuelle propre pour permettre ensuite une structuration automatique.

1.3.4 Structuration des données extraites

Une fois les données extraites et nettoyées, elles sont organisées sous forme structurée, en général en format JSON. Un exemple des données en format JSON est illustré dans la Figure 1.1. Cette structuration est essentielle pour rendre les données exploitables par un moteur de recherche, une interface ou un chatbot.

FIGURE 1.1 – Exemple de données structurées en format JSON

1.3.5 Limites rencontrées dans le contexte industriel

Malgré les outils avancés, l'extraction automatique dans un contexte industriel comme celui de Sonatrach reste un défi. En effet, les documents sont textuels, mais souvent mal formatés (balises fragmentées, blocs mal alignés...). De plus, les acronymes et tags suivent des règles spécifiques non documentées. Certains fichiers mélangent du texte, des tableaux, et des symboles graphiques. Cela justifie la mise en place d'un pipeline d'extraction personnalisé, adapté à chaque type de fichier industriel.

Les différentes méthodes et challenges d'extraction automatique des données, décrits dans cette section, montrent que l'extraction automatique est une étape technique critique dans tout projet de digitalisation industrielle. Dans le cadre de TechBot Sonatrach, elle représente le socle sur lequel repose la construction de la base de données et du chatbot. La qualité de cette extraction conditionne directement la fiabilité du système global.

1.4 Structuration et modélisation des données industrielles

L'extraction seule de texte brut à partir de documents industriels n'assure pas l'exploitabilité de l'information et donc l'intégrabilité dans les outils d'analyse,

de recherche ou d'interrogation (chatbot par exemple). Les documents industriels devront alors être transformés en données structurées selon un modèle approprié lors d'une phase de structuration, qui est un enjeu clé des systèmes d'information des industries.

1.4.1 Objectifs de la structuration des données

La structuration des données vise à atteindre plusieurs objectifs [12] :

- Permettre une consultation rapide à partir d'un programme ou une interface.
- Mettre en œuvre les recherches par critères (ex. : repérer tous les « Transmetteurs » d'un dossier).
- Constituer une base solide pour les systèmes de dialogue ou les outils d'aide à la décision.
- Réduire les risques d'erreurs liés à l'interprétation manuelle des documents.

1.4.2 Modèles de données utilisés dans l'industrie

Les systèmes industriels font appel à plusieurs types de modèles de données selon le besoin. Deux modèles couramment utilisés sont présentés ici :

- Modèles relationnels (SQL) : adaptés aux structures tabulaires rigoureuses, mais trop rigides parfois pour des documents non standardisés [13].
- Modèles orientés document (NoSQL / JSON) : mieux adaptés aux formats semi-structurés, comme ceux issus de l'extraction de fichiers de PDF [14].

S'agissant de la raffinerie Sonatrach, qui traite une très grande variété de documents d'un fichier à l'autre, le format JSON apparaît comme la solution la plus souple et lisible. Il permet une intégration facile avec des langages comme Python ou JavaScript, ainsi qu'avec des bases de données telles que MongoDB.

1.4.3 Représentation typique des données extraites

À l'issue de la procédure d'extraction, chaque instrument relevée à partir d'un document PDF pourra être représenté sous une structure JSON qui ressemblerait à :‘json

```
{
 "tag": "510-TT-0012",
 "type": "Transmetteur",
 "acronyme": "TT",
 "origine_fichier": "9952T-510-PID-0021-1101-1.pdf",
```

```
 "unite": "Bar",
 "position_in_pdf": {
  "page": 3,
  "x": 135.2,
  "y": 290.7
 }
}
```


Cette structuration permet d'enrichir les données par d'autres modules (acronyme → définition), de tracer la provenance exacte des données et d'indexer facilement les informations dans une base interrogeable.

1.4.4 Structuration orientée entité

Pour un usage plus avancé, les données industrielles sont modélisées sous forme d'entités reliées entre elles selon les relations suivantes :

- Fichier → contient plusieurs instruments.
- Instrument → possède un tag, un acronyme, un type, une unité.
- Acronyme → peut être enrichi d'une définition.
- Instrument → peut être lié à un autre (relation FROM/TO).

Cette modélisation, de type "graphe" ou "document orienté entité", est essentielle pour des usages avancés, notamment la recherche intelligente, le dialogue avec un chatbot et l'analyse statistique [15].

1.4.5 Structuration contextuelle : un défi spécifique

À la différence des écrits traditionnels, les documents industriels sont fortement connotés et contextualisés. Par exemple, un acronyme peut avoir plusieurs sens selon le contexte (ex : le sigle PDA). Certains éléments n'ont également de sens que par rapport à un autre (ex : un tag "FT" sans la pipe associée est incomplet). De ce fait, la logique spatiale des fichiers PDF (informations positionnelles) peut être cruciale pour l'interprétation. Il conviendra donc de structurer à la fois le contenu textuel brut, les relations sémantiques (type → définition) et les informations positionnelles associées au document.

La structuration et la modélisation des données extraites à partir des documents industriels est une étape clé entre la phase d'extraction brute et l'exploitation intelligente [16]. Dans le cadre du projet TechBot Sonatrach, cette structuration constitue le socle sur lequel reposera la base de données centrale, ainsi que les capacités de réponse du futur chatbot.

1.5 Traitement des acronymes dans l'industrie

Au sein des outils de la documentation technique utilisée dans le milieu technologique des raffineries, les acronymes sont omniprésents. Ils correspondent à une forme condensée d'information permettant l'identification rapide de dispositifs complexes grâce à l'attribution de codes courts tels que « TT », « PCV », ou « LT » qui désignent respectivement un transmetteur de température, une vanne de contrôle de pression ou un capteur de niveau standard. Ils favorisent la lisibilité et l'intercompréhension des schémas entre les différents corps de métier (automatisme, instrumentation, maintenance) et allègent les documents d'ingénierie d'un surplus textuel.

Cependant, le régime de brièveté de l'information s'accompagne d'une difficulté majeure, en matière de traitement automatique, à savoir l'absence de définition explicite dans les fichiers source. Dans les documents traités que constituent notamment les PID et les listes d'instruments, les acronymes se présentent, le plus souvent, sans légende explicative ni table de correspondance, qui sont requis pour traiter l'information correspondant aux acronymes en tant que genre de signifiés plus complexes. Leur traitement est ainsi contextuel : un même acronyme peut correspondre, selon le contexte du procédé, de l'unité industrielle, des conventions internes en entreprise, à une pluralité d'équipements. Cette variabilité les rend ainsi complexes à traiter dans les procédures de traitement automatisées.

Les approches classiques privilégient généralement l'utilisation de dictionnaires standards ou de bases de données externes accessibles sur le Web, notamment quand le référentiel sur lequel on travaille est bien défini. Toutefois, ce n'est pas toujours le cas dans certains environnements industriels. À l'instar de Sonatrach ou d'une unité donnée, un certain nombre d'acronymes ne sont mentionnés ni sur le Web ni dans les bases normalisées. Par ailleurs, le scraping peut apporter une solution partielle, en extrayant des définitions à partir de documents disponibles sur le Web. Cependant, cette méthode atteint rapidement ses limites lorsque les acronymes ne sont plus spécifiques dans les différentes utilisations professionnelles possibles ou trop ambigus pour être interprétés de manière fiable [17].

Face à ce type de limites, d'autres solutions peuvent être envisagées. Parmi elles, certaines approches statistiques comme celles du modèle n-grammes (unigramme, bigramme, trigramme) permettent d'apprendre les régularités présentes dans les définitions disponibles. Ces modèles pourraient aider alors à prédire la définition probable d'un acronyme inconnu. D'autres approches relèvent de l'apprentissage supervisé, en entraînant un modèle à partir d'un corpus d'acronymes correctement

défini afin de généraliser cette connaissance à de nouveaux cas. Enfin, dans certains contextes, le recours à des représentations vectorielles issues d'outils de traitement du langage naturel (NLP) peut être pertinent. Ces représentations permettent d'évaluer la similarité sémantique entre un acronyme et un ensemble de mots candidats, facilitant ainsi la sélection d'une définition plausible [18].

Dans le projet TechBot Sonatrach, le challenge représenté par la problématique relative aux acronymes est dans cette étude cruciale. Une base de données déficiente ou mal interprétée au niveau des acronymes compromettrait directement la fiabilité des réponses fournies par le chatbot, c'est pourquoi une attention particulière a été portée sur la collecte, la vérification et l'anticipation intelligente des acronymes au long du pipeline de traitement.

1.6 Chatbots dans les systèmes techniques

Les chatbots, initialement créés pour essayer de donner le change dans des contextes simples, ont progressivement évolué vers des systèmes complexes capables d'échanger avec des bases de données, des systèmes d'information ou même des systèmes techniques. Leurs performances se sont accrues grâce à la poussée récente du traitement du langage naturel (NLP) et de l'Intelligence Artificielle (IA), élargissant ainsi leur champ d'application. Dans les environnements techniques ou industriels, le chatbot n'est plus seulement un répondant à des questions générales ou un aiguillage de l'utilisateur à la manière d'un assistant commercial : il entre aujourd'hui dans des fonctions plus élaborées, comme la consultation d'une base de données d'équipements, le ciblage dans une recherche d'informations techniques, ou l'aide à la maintenance en ligne. Mais son intégration dans un système tout industriel requiert d'être adapté à des langages plus spécifiques, à des données le plus souvent structurées non standards, à des exigences de précision maximales au centième près [19].

Dans le domaine industriel, les chatbots sont le plus souvent assujettis à un socle de connaissances techniques solide et doivent pouvoir traiter des requêtes complexes, elles-mêmes constituées d'acronymes, de tags, ou encore de références ultraspécifiques à un équipement particulier. La conception de chatbots requiert donc un couplage adéquat entre un moteur appliquant des mécanismes de traitement du langage, un moteur logique (raisonnement ou filtrage) et un accès à la base de données technique en backend.

Plusieurs outils et frameworks facilitent aujourd'hui le déploiement d'agents d'intelligence artificielle. Des solutions open source comme Rasa permettent ainsi

d'implémenter des agents de conversation dialoguant en langage naturel et personnalisables, tout en interagissant avec des bases techniques métiers. Des solutions cloud comme Dialogflow (Google) ou Microsoft Bot Framework offrent des intégrations rapides mais souvent limitées par leur logique propriétaire. Dans une approche plus poussée, l'usage d'APIs comme celle d'OpenAI permet d'envisager des réponses génératives plus riches, mais au prix de contraintes en termes de contrôle, d'interprétabilité et de confidentialité des données industrielles.

1.6.1 Besoins du TechBot Sonatrach

Pour un cas d'usage comme celui du projet TechBot Sonatrach, le chatbot doit pouvoir comprendre une requête du type « Quel est le rôle de 510-TT-0031 ? » ou « Donne-moi la définition de TRX ». Il doit également être capable de relier cette requête à des documents extraits, à une base de définitions d'acronymes, ou à des informations positionnées dans des fichiers PID. Ce niveau d'exigence dépasse largement celui des chatbots traditionnels orientés vers le service client. Cela nécessite donc une architecture pensée spécifiquement pour un usage technique avec une rigueur sur la structuration des données, le traitement du langage et la précision des réponses.

1.6.2 NLP dans les systèmes techniques

Le traitement automatique du langage naturel, ou NLP (Natural Language Processing), est un domaine de l'intelligence artificielle qui vise à permettre aux machines de comprendre, d'interpréter et de générer du langage humain. Longtemps réservé à des applications générales comme la traduction automatique ou l'analyse de sentiments, le NLP s'invite désormais dans des contextes plus spécialisés, notamment les systèmes industriels, où il devient un outil clé pour l'automatisation de la compréhension des documents et le développement d'agents conversationnels intelligents [20].

Dans un environnement technique, les données textuelles ne ressemblent pas à des textes littéraires ou journalistiques. Elles sont composées d'acronymes, d'unités de mesure, de formules, de codes et de structures syntaxiques non standards. C'est pourquoi l'application du NLP dans ce type de contexte requiert une adaptation particulière. Il ne suffit pas d'utiliser un modèle générique entraîné sur des corpus grand public : il faut concevoir ou adapter des outils capables de traiter les spécificités du langage industriel.

Parmi les techniques de base utilisées dans le NLP, on retrouve la vectorisation des textes, avec des représentations telles que le bag-of-words, le TF-IDF (Term Frequency-Inverse Document Frequency) ou encore les embeddings comme Word2Vec ou BERT. Ces méthodes permettent de convertir les mots en vecteurs numériques, facilitant ainsi leur traitement par des algorithmes de classification, de clustering ou de recherche de similarité [20]. Dans le cas des documents techniques, ces représentations doivent être manipulées avec précaution, car un même mot ou acronyme peut revêtir plusieurs sens selon le contexte du procédé ou du fichier analysé.

Un autre volet essentiel du NLP appliqué à l'industrie est la reconnaissance d'entités nommées (NER – Named Entity Recognition) [21]. Cette technique permet d'identifier automatiquement dans un texte les entités importantes comme les tags d'instruments, les types d'équipements, les unités de mesure, ou les zones d'installation. Par exemple, la phrase « le capteur 510-TT-0031 mesure la pression d'entrée » peut être segmentée automatiquement pour extraire le tag "510-TT-0031", le type "capteur" et la propriété mesurée "pression". Cette extraction est indispensable pour relier une requête utilisateur à une information présente dans les fichiers techniques.

Les modèles statistiques comme les N-gram (suites de n mots ou caractères) trouvent également leur utilité dans la prédiction de définitions d'acronymes ou la génération de phrases techniques [22]. Bien que simples, ces modèles peuvent capturer des régularités intéressantes lorsqu'ils sont entraînés sur un corpus spécialisé. Toutefois, pour des résultats plus robustes et un meilleur traitement de la sémantique, les architectures neuronales récentes, notamment les transformers (comme BERT ou GPT), offrent des perspectives très prometteuses dans les systèmes techniques, à condition de disposer de suffisamment de données pertinentes.

Dans le cadre du projet TechBot Sonatrach, le NLP occupe une place centrale à plusieurs niveaux. Le NLP intervient dans l'extraction d'informations à partir de texte, dans la structuration sémantique des acronymes, dans la recherche de similarité entre questions et contenus techniques et dans la génération ou la reformulation de réponses par le chatbot. Le défi principal reste l'adaptation de ces techniques aux particularités du langage industriel, ce qui nécessite un ajustement fin des outils, des jeux de données et des modèles d'apprentissage.

1.7 Limites des approches existantes

En dépit des avancées récentes de l'extraction de données, du traitement du langage naturel et des agents conversationnels, les systèmes de gestion documen-

taire restent toutefois limités dans les contextes industriels où les documents sont typiquement techniques, hétérogènes et parfois mal normalisés, comme c'est le cas dans le milieu de la raffinerie Sonatrach. Les solutions standards atteignent rapidement leur seuil d'efficacité maximale.

La première difficulté provient de la nature même des fichiers techniques. Certes, tous sont textuels, mais leur structure n'est ni linéaire, ni homogène. Des tableaux semi-structurés, des titres ambigus, des éléments éparpillés sur plusieurs pages composent ces documents, entraînant une forte dépendance au positionnement spatial. Or, les outils d'extraction traditionnels, même les plus avancés, basés sur la logique de classification des documents, ne sont pas adaptés à ces logiques documentaires. La seule reconstruction des informations intégrées dans une information logique organique peut nécessiter des traitements spécifiques, adaptés à la typologie de chaque fichier [23].

Autre difficulté : les désignations acronymiques qui qualifient la terminologie technique dans les documents ne sont pas toujours synonymes de normes universelles. Un acronyme peut signifier plusieurs choses s'il est utilisé dans différentes unités de production, pour différents procédés ou simplement en raison de différences culturelles internes à l'entreprise. Si l'on note que de nombreuses définitions ne sont pas disponibles au public, cela rend les bases externes quelque peu inopérantes ; même quand une définition existe, celle-ci est parfois trop vague ou trop générale pour l'exigence présente dans un système d'assistance intelligent [24].

En matière de NLP, les modèles pré-entraînés qui existent sont le plus souvent bâtis sur des corpus généraux (Wikipedia, forums, presse, etc.), qui n'agrègent pas le vocabulaire, la syntaxe ni les codes de l'ingénierie industrielle. Une exploitation de ces modèles sans adaptation voire fine-tuning produit inéluctablement des réponses inexactes et finalement hors cadre. Les méthodes basées sur des règles ou des dictionnaires sont insuffisantes pour caractériser la variabilité linguistique existant dans les documents industriels [25].

Pour les chatbots, les solutions standards proposent souvent une interface conversationnelle performante, mais qui ne répond pas aux besoins techniques d'un ingénieur ou d'un technicien de maintenance. Ces dispositifs ne possèdent pas la précision ou les capacités d'interprétation propres aux tags complexes ou aux requêtes semi-formelles. Ces dispositifs ne sont pas habilités à se connecter dynamiquement aux bases de données industrielles de type structuré, ce qui les réduit à un usage simple et limité, tel que la gestion de FAQ ou l'assistance utilisateur élémentaire.

En revanche, très peu de dispositifs existants sont capables de réunir automatiquement dans un même système les trois éléments que sont l'extraction, la structuration fine et l'exploitation à partir d'un chatbot, car ceux-ci sont souvent traités indépendamment, ce qui perturbe la cohérence de l'ensemble [26].

C'est dans ce contexte, marqué par l'absence d'outils adaptés au traitement intelligent de la documentation industrielle que se situe le projet TechBot Sonatrach. Ce dernier propose d'articuler extraction sur mesure, enrichissement sémantique et interrogation chatbot en un tout cohérent, adapté à la spécificité et aux enjeux du domaine pétrolier.

1.8 Conclusion

Ce chapitre a permis d'explorer les fondations théoriques, techniques et industrielles sur lesquelles repose le projet TechBot Sonatrach. L'analyse de la documentation industrielle, qu'il s'agisse des PID, des datasheets, des listes d'instruments ou des fichiers d'E/S, a mis en évidence la richesse et la complexité de l'information technique présente dans ces documents. Ces fichiers jouent un rôle essentiel dans le fonctionnement des installations, mais leur exploitation manuelle reste chronophage, sujette à l'erreur et difficilement scalable dans un environnement industriel moderne.

L'extraction automatique d'informations à partir de ces documents, bien qu'appuyée par des outils puissants comme pdfminer ou pdfplumber, requiert un prétraitement important, une structuration adaptée et une modélisation orientée métier. Une structuration efficace implique un traitement rigoureux des entités propres au secteur, en particulier les acronymes, dont la variabilité représente un obstacle majeur à l'automatisation. Face à ces contraintes, les approches traditionnelles se révèlent souvent inadaptées, ce qui rend nécessaire l'exploration de méthodes plus avancées, allant du web scraping à la prédiction sémantique basée sur des modèles statistiques.

L'étude des chatbots a finalement révélé un potentiel important afin de faciliter l'accès à l'information technique dans l'industrie, tout en soulignat les limites des systèmes classiques dans des contextes spécialisés comme celui de Sonatrach. Un agent conversationnel technique ne peut être efficace que s'il est connecté à une base de données correctement structurée et enrichie, et si son moteur linguistique est capable d'interpréter les particularités du langage industriel.

L'ensemble des constats présentés dans ce chapitre justifie le développement d'une solution intégrée, capable d'extraire, de structurer, d'enrichir et d'interroger

intelligemment les données industrielles. C'est précisément dans cette perpective que s'inscrit le projet TechBot Sonatrach, dont la conception sera détaillée dans le chapitre suivant.

Chapitre 2

Approche hybride d'extraction et de navigation intelligente dans les données techniques

Ce chapitre présente l'ensemble des approches conceptuelles qui ont permis de concevoir et structurer le projet TechBot Sonatrach. Dans un premier temps, les objectifs du projet sont exposés. Ensuite, une analyse approfondie des données d'entrée, incluant les différents types de fichiers industriels, est proposée, accompagnée des contraintes spécifiques liées à l'environnement de Sonatrach. Le chapitre détaille également les méthodes d'extraction, la structuration intelligente des données, les principales contraintes rencontrées, ainsi que les choix techniques effectués pour concevoir un système d'assistance documentaire à la fois robuste et pertinent.

Aucune expérimentation n'est traitée à ce stade : l'accent est mis exclusivement sur la modélisation, l'architecture envisagée et les fondements du traitement de l'information.

2.1 Objectifs du projet TechBot

L'objectif principal du projet TechBot Sonatrach est de concevoir et de mettre en place un système intelligent interne aux collaborateurs, permettant d'avoir des réponses claires, rapides et fiables aux questions relatives aux équipements industriels présents dans la documentation technique. Ce système doit répondre à trois besoins fondamentaux : l'extraction automatisée des informations, la centralisation structurée dans une base de données exploitable et l'interrogation naturelle via un chatbot.

Plus concrètement, le projet vise à automatiser la lecture, l'analyse, le traitement et l'exploitation d'une série de fichiers techniques fournis par Sonatrach. Il s'agit de convertir ces documents bruts en données structurées, avec chaque instrument étant représenté de manière unique, accompagné de ses attributs (tag, type, acronyme, fichier source, unité, fonction...). Une attention particulière est portée au traitement des acronymes, souvent un obstacle pour l'analyse automatisée, en particulier lorsqu'ils ne sont pas référencés dans des bases de données publiques. Le projet prévoit donc une amélioration automatique des définitions, ainsi qu'une approche statistiquement prédictive pour les informations manquantes.

Enfin, l'objectif final est de proposer une interface conversationnelle capable de répondre à des requêtes formulées en langage naturel, telles que « Que signifie le tag 510-TT-0031 ? », « Dans quel fichier se trouve la vanne ZCV-005 ? » ou encore « Quels sont les instruments liés au procédé RFCC ? ». Le chatbot agira comme un pont intelligent entre l'utilisateur et la base d'informations industrielles préalablement structurée.

En résumé, TechBot Sonatrach aspire à automatiser l'accès à la connaissance technique à travers un système fiable, évolutif et directement lié aux réalités documentaires du terrain. Son objectif est non seulement d'améliorer l'efficacité opérationnelle, mais aussi de démontrer la valeur ajoutée de l'intelligence artificielle dans l'optimisation des processus métiers industriels.

2.2 Analyse des données techniques fournies

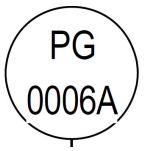
Dans le cadre de ce projet, un ensemble de fichiers techniques a été mis à disposition par la raffinerie Sonatrach. Ces documents, bien que regroupés sous un format homogène (PDF textuel), se distinguent par leur diversité de contenu, de structure et d'objectif. Une analyse préliminaire s'est avérée nécessaire afin de comprendre la nature des données disponibles, anticiper les traitements requis, et concevoir un système d'extraction adapté à chaque catégorie. Cette section présente une analyse approfondie des fichiers reçus, mettant en lumière leur contenu, leur degré de complexité pour l'exploitation, leur état de traitement actuel, ainsi que leur contribution potentielle à l'alimentation de la base de données centrale utilisée par le chatbot TechBot.

2.2.1 Fichiers PID

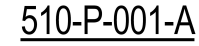
Les fichiers PID représentent sous forme de schéma, les procédés, les équipements, les instruments de mesure ou de régulation, ainsi que les lignes de communication

(fluides ou signaux) qui les relient. Chaque élément y est identifié par un code ou un tag, et positionné selon une logique fonctionnelle rigoureuse. Ces documents sont indispensables pour comprendre le fonctionnement global d'une unité de traitement, anticiper les opérations de maintenance, planifier les interventions ou encore localiser rapidement un équipement. Les fichiers PID reçus dans le cadre de ce projet sont tous au format PDF textuel. Leur structure est partiellement tabulaire, avec des zones schématiques, des blocs de texte, des métadonnées techniques et parfois des tableaux localisés en bordure de page.

Ces fichiers contiennent de nombreux éléments critiques, notamment les identifiants uniques (tags) des instruments tels que TT-0002, FIC-0075, PG-1021A (voir la Fig. 2.1a), ou encore des équipements complets comme 510-D-001 désigné comme ISOMERIZATION FEED SURGE DRUM. Un exemple de tag d'équipement est illustré dans la Fig. 2.1b. D'autres informations importantes sont également intégrées, comme les flux FROM et TO indiquant les connexions externes à d'autres unités (ex. : FROM PSV-0008A, TO 500-FIC-0030) (voir la Fig. 2.2), ou encore les identifiants de lignes de communication (par exemple 5100007), essentiels pour établir une traçabilité des circuits.



(a) Tag d'instrument



(b) Tag d'équipement

FIGURE 2.1 – Exemple de tags existants

Les PID comportent aussi des zones textuelles contenant des notes générales ou spécifiques, des instructions techniques ou des consignes de sécurité comme illustré sur la Fig.2.3. Enfin, chaque fichier est accompagné de métadonnées permettant de l'identifier dans le projet, telles que le numéro d'unité (Unit no → 510), le type de document (Doc type → PID), le code de matériau ou le numéro de série.

Le traitement automatique de ces fichiers est particulièrement difficile. Cette complexité réside dans la nature hybride de leur structure (mélange de texte, de formes et de repères techniques), mais aussi dans la variabilité d'un fichier à l'autre. Aucun outil d'extraction standard ne s'est montré pleinement adapté en l'état. Après plusieurs tentatives avec différentes bibliothèques Python, un programme personnalisé a été développé pour cibler spécifiquement les éléments critiques des fichiers PID : instruments, équipements, connexions, lignes de flux et métadonnées.

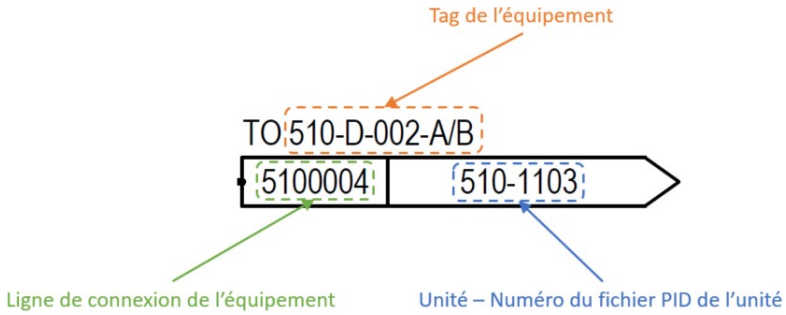
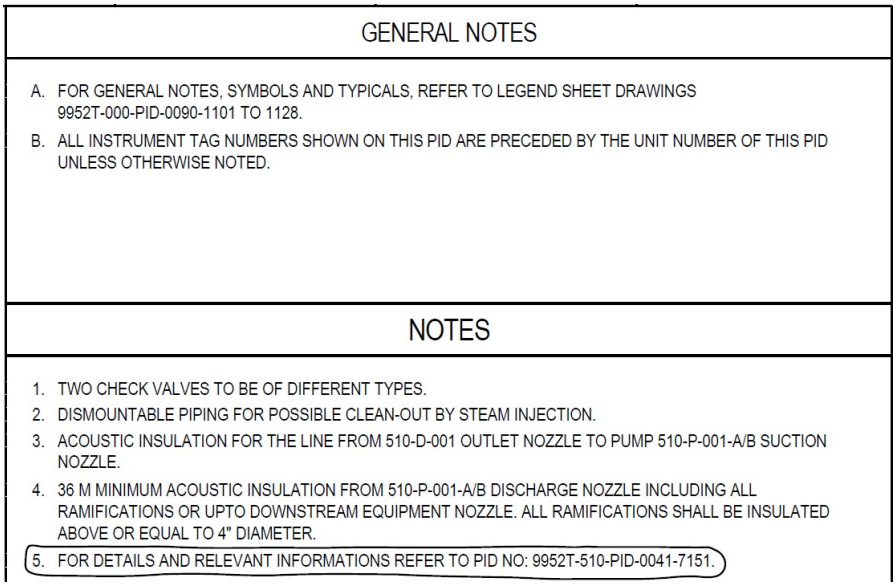


FIGURE 2.2 – Flux d'équipement



GENERAL NOTES

A. FOR GENERAL NOTES, SYMBOLS AND TYPICALS, REFER TO LEGEND SHEET DRAWINGS 9952T-000-PID-0090-1101 TO 1128.

B. ALL INSTRUMENT TAG NUMBERS SHOWN ON THIS PID ARE PRECEDED BY THE UNIT NUMBER OF THIS PID UNLESS OTHERWISE NOTED.

NOTES

1. TWO CHECK VALVES TO BE OF DIFFERENT TYPES.
2. DISMOUNTABLE PIPING FOR POSSIBLE CLEAN-OUT BY STEAM INJECTION.
3. ACOUSTIC INSULATION FOR THE LINE FROM 510-D-001 OUTLET NOZZLE TO PUMP 510-P-001-A/B SUCTION NOZZLE.
4. 36 M MINIMUM ACOUSTIC INSULATION FROM 510-P-001-A/B DISCHARGE NOZZLE INCLUDING ALL RAMIFICATIONS OR UPTO DOWNSTREAM EQUIPMENT NOZZLE. ALL RAMIFICATIONS SHALL BE INSULATED ABOVE OR EQUAL TO 4" DIAMETER.
5. FOR DETAILS AND RELEVANT INFORMATIONS REFER TO PID NO: 9952T-510-PID-0041-7151.

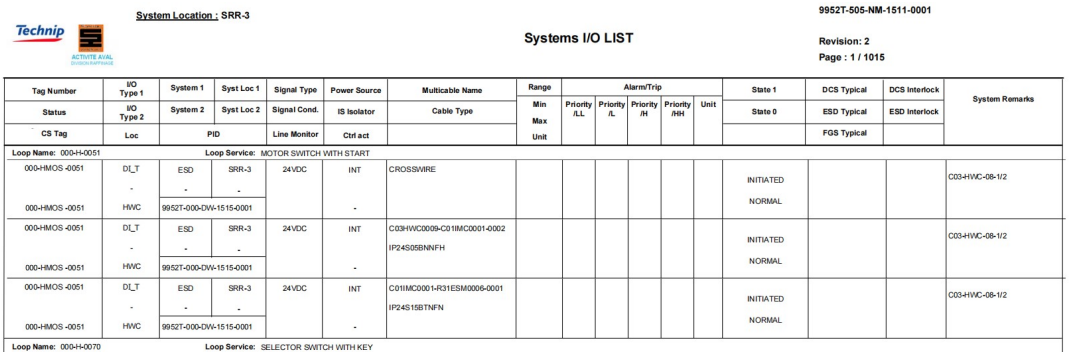
FIGURE 2.3 – Notes & Notes générales

Ce programme permet aujourd'hui une extraction complète et structurée de ces données, constituant ainsi la première brique solide de la base de connaissances sur laquelle s'appuiera le chatbot.

2.2.2 Fiches techniques (Datasheets)

Les fiches techniques fournissent une description détaillée des caractéristiques techniques de chaque instrument utilisé dans le processus. Cela garantit leur conformité aux normes de fonctionnement, aux réglementations en vigueur et aux conditions environnementales spécifiques de la raffinerie. Ces documents sont essentiels pour la sélection, l'approvisionnement, l'installation, l'exploitation et la maintenance des équipements instrumentés. Les datasheets sont fournis au format

PDF textuel. Un exemple de fichier datasheet est illustré sur la Fig.2.4.



System Location : SRR-3

Systems I/O LIST

9952T-505-NM-1511-0001

Revision: 2

Page : 1 / 1015

Tag Number	I/O Type 1	System 1	Syst Loc 1	Signal Type	Power Source	Multicable Name	Range	Alarm/Trip					State 1	DCS Typical	DCS Interlock	System Remarks
Status	I/O Type 2	System 2	Syst Loc 2	Signal Cond.	IS Isolator	Cable Type	Min Max	Priority /LL	Priority /L	Priority /H	Priority /HH	Unit	State 0	ESD Typical	ESD Interlock	
CS Tag	Loc	PID		Line Monitor	Ctrl act		Unit							FGS Typical		
Loop Name: 000-H-0051		Loop Service: MOTOR SWITCH WITH START														
000-HMOS -0051	DI_T	ESD	SRR-3	24VDC	INT	CROSSWIRE							INITIATED			C03-HWC-08-1/2
	-	-	-										NORMAL			
000-HMOS -0051	HWC	9952T-000-DW-1515-0001			-											
000-HMOS -0051	DI_T	ESD	SRR-3	24VDC	INT	C03HWC0009-C01IMC0001-0002							INITIATED			C03-HWC-08-1/2
	-	-	-			IP24S05BNNFH							NORMAL			
000-HMOS -0051	HWC	9952T-000-DW-1515-0001			-											
000-HMOS -0051	DI_T	ESD	SRR-3	24VDC	INT	C01IMC0001-R31ESM0006-0001							INITIATED			C03-HWC-08-1/2
	-	-	-			IP24S15BTNFN							NORMAL			
000-HMOS -0051	HWC	9952T-000-DW-1515-0001			-											
Loop Name: 000-H-0070		Loop Service: SELECTOR SWITCH WITH KEY														

FIGURE 2.4 – Extrait du fichier DataSheet

La présentation des datasheets varie considérablement d'un fichier à l'autre. Certains contiennent des tableaux denses, tandis que d'autres présentent un mélange de texte libre et de sections tabulaires, ce qui rend difficile l'extraction automatique des informations. De plus, les entêtes de colonnes ne sont pas uniformes et la mise en page peut varier en fonction du type d'instrument ou de la source du document. Chaque fiche contient des informations clés telles que : l'identification par tag (p. ex. 500-FV-0001), la description fonctionnelle (Straight Run Naphtha from CDU to 500-D-001), et les paramètres de fonctionnement tels que le débit (p. ex. 62,45 m3/h), la pression (de 6,8 à 7,6 kg/cm2g), la température de service, la densité du fluide et/ou la viscosité. D'autres sections détaillent les caractéristiques mécaniques (type de vanne, dimensions de raccordement, matériaux), les spécifications du système de commande (actionneur, positionneur, type de signal, position de sécurité), ainsi que les normes de sécurité (par exemple : classification ATEX, classe de fuite).

Les travaux de traitement des fichiers sont toujours en cours. Certaines analyses ont été menées avec succès malgré la variabilité des formats, tandis que d'autres restent à traiter. La complexité du processus réside dans la diversité des structures tabulaires et la nécessité d'identifier les sections pertinentes assurant une interopérabilité sémantique entre les formats non standardisés. Une approche semi-automatique a été développée pour identifier, transformer et structurer les informations complexes de manière exploitable. Ce processus, est complexe mais crucial pour enrichir la base de données finale avec des propriétés physiques et fonctionnelles précises de chaque instrument.

2.2.3 Fichier Instrument Index

Le document intitulé Instrument Index est un outil essentiel de coordination entre les divers documents techniques d'un projet industriel. Il présente de manière exhaustive la liste des instruments utilisés dans l'unité, sous forme de tableau. Chaque ligne correspond à un instrument unique, identifié par un tag, et est enrichie par diverses métadonnées permettant d'assurer la traçabilité et la cohérence des équipements au sein du système global.

Ce document est présenté de manière tabulaire, avec des colonnes normalisées. Il inclut le tag de l'instrument (p. ex. 505-NM-1511-0001, 505-TT-1511-0001), son type fonctionnel (issu du code d'acronyme comme NM pour Niveau Manuel, TT pour Temperature Transmitter), une brève description de son rôle dans le procédé (Level Gauge on TK-505A), ainsi que des références croisées avec les fichiers PID et Datasheets associés. Des colonnes supplémentaires, telles que les commentaires ou le statut, peuvent également être présentes pour documenter le montage, l'état d'avancement ou les spécificités de l'instrument. À partir d'un tag spécifique, il est possible de localiser l'instrument sur le schéma PID, retrouver ses paramètres techniques dans la fiche datasheet, ou vérifier son affectation dans le fichier I/O. Ainsi, il joue un rôle central dans la consolidation de la base de données et le fonctionnement global du système. Bien que le format PDF du document soit structuré et textuel, l'extraction intégrale des informations contenues n'a pas encore été finalisée. Le tableau peut s'étendre sur plusieurs pages, et quelques alignements sémantiques pourraient être nécessaires pour traiter certaines variations dans les intitulés de colonnes. La finalisation de de l'analyse de ce fichier est prévue pour une phase ultérieure, notamment si le renforcement de la cohérence du système dans son ensemble devient impactant, incitant à identifier une correspondance entre tous les instruments du projet.

2.2.4 Fichier System Input/Output (I/O List)

Le document "System Input/Output" (I/O List) joue un rôle central dans l'intégration des instruments de terrain avec le système de contrôle industriel. Il établit de manière précise les protocoles de communication entre chaque instrument et les automates (PLC) ou les systèmes de contrôle distribués (DCS), en détaillant les types de connexions, les signaux transmis, ainsi que les emplacements physiques et logiques des points d'entrée et de sortie.

Ce document est présenté sous forme de tableau structuré et homogène au format PDF textuel. Les colonnes comprennent généralement le tag de l'instrument (500-TT-0001, 500-PCV-0004), une brève description de sa fonction (Temperature

Transmitter for Reactor Outlet), le type de signal (Analog Input, Digital Output, etc.), le mode de transmission (4-20 mA, contact sec, NAMUR), et l'adresse physique du point (canal, rack, slot, carte). Des remarques complémentaires peuvent fournir des informations sur des conditions d'utilisation, de sécurité ou de logique de contrôle particulières.

L'I/O List se distingue des autres documents techniques par sa structure régulière et l'utilisation de conventions établies dans le domaine de l'automatisation industrielle. Cette standardisation facilite grandement l'extraction automatisée des données. Ces informations sont ensuite transformées en une structure tabulaire numérique, prête à être traitée par le système d'analyse ou le chatbot. Ce document permet également de répondre à des questions spécifiques telles que "Quel est le type de signal du TT-0003 ?" ou "Quel est le canal d'entrée du capteur de pression PC-0001 ?". Intégré à la base de données finale, l'I/O List enrichit les capacités du chatbot TechBot à répondre à des questions techniques précises concernant l'architecture de contrôle et l'automatisation du procédé.

2.2.5 Fichier Définition des Instruments

Le fichier PDF fourni contient un tableau alphabétique clair listant des acronymes (ex. : TT, PIC) et leurs définitions respectives. Ce format standardisé a facilité l'extraction automatique et l'intégration dans une base de correspondance utilisée pour interpréter les tags dans les PID et fiches techniques (Datasheets).

Les données, issues de deux sources web fiables [1], [2] , ont été nettoyées et structurées pour servir de référentiel central tout au long du projet.

2.3 Contraintes spécifiques liées aux documents Sonatrach

L'un des défis majeurs du projet TechBot Sonatrach réside dans les contraintes strictes imposées par la nature des documents industriels manipulés. Ces fichiers, provenant directement des installations de la raffinerie, obéissant d'une part à des règles de confidentialité rigoureuses et d'autre part à des exigences techniques et opérationnelles. Ce qui influencent fortement l'orientation des choix d'architecture

1. https://www.piping-designer.com/index.php/disciplines/electrical/instrumentation/1826-instrument-abbreviations
2. https://www.pirobloc.com/wp-content/uploads/2017/10/Pirobloc-PID-Abbreviation.pdf

et les méthodes de traitement adoptées dans ce projet.

Tout d'abord, la confidentialité des données est une contrainte absolue : il est strictement interdit d'héberger les documents sur des serveurs externes ou de les partager via des API en ligne. Cela exclut d'emblée toute approche reposant sur des services cloud, que ce soit pour l'hébergement, la vectorisation ou l'inférence à distance. Le système doit donc être conçu pour fonctionner intégralement en local, sans aucune dépendance à Internet.

Ensuite, les fichiers sont soumis à des mises à jour fréquentes, du fait des révisions régulières réalisées sur les installations industrielles. Cette caractéristique impose à l'architecture une forte capacité d'adaptation. Le système doit permettre l'intégration rapide de nouveaux fichiers ou la mise à jour des données existantes sans nécessiter de retraitement lourd ni de réentraînement de modèle, comme c'est souvent le cas avec le fine-tuning.

Par ailleurs, l'hétérogénéité des informations contenues dans les documents étudiés impose des contraintes spécifiques en matière d'extraction. Ces documents sont tous au format PDF textuel — c'est-à-dire qu'ils contiennent une couche de texte accessible — et ne résultent pas de simples scans image-based. Dès lors, le recours à la reconnaissance optique de caractères (OCR), bien qu'indispensable pour les documents scannés, se révèle ici non seulement inutile, mais également contre-productif. En effet, les moteurs OCR tels que Tesseract, ou même des solutions avancées comme Azure Read ou Google Document AI, sont conçus pour reconstituer du texte là où il est absent. Leur application sur des fichiers déjà textuels peut introduire des erreurs de lecture, notamment sur les acronymes techniques, les unités de mesure ou les structures tabulaires complexes. Comme le soulignent [27], l'OCR échoue fréquemment à restituer correctement la mise en page des documents techniques, en particulier lorsqu'il s'agit de tableaux, de colonnes multiples ou de figures. De plus, dans [28], l'étude montre que l'OCR peine à gérer les spécificités du langage industriel, ce qui affecte la fidélité de l'extraction. Cette réalité est amplifiée par la qualité variable des PDF traités, où des documents pourtant lisibles peuvent être dégradés par des artefacts de reconnaissance. En pratique, même les solutions cloud les plus performantes n'atteignent de bons taux de reconnaissance que sur des scans haute qualité, tandis que des outils open-source comme Tesseract tombent parfois sous les 70% de précision [29]. Par ailleurs, la diversité des formats traités — PID contenant des schémas et des tags d'instruments, datasheets avec tableaux complexes, fichiers I/O List ou Instrument Index mieux structurés mais toujours techniques — exige une approche modulaire, adaptée à chaque type. Ces constats justifient l'exclusion de l'OCR dans notre démarche, fondée exclusivement

sur l'exploitation directe des couches textuelles des PDF, afin de garantir une extraction plus fiable, précise et économiquement pertinente.

Enfin, aucune interface de gestion documentaire n'existe à ce stade dans le système. Il n'est donc pas encore possible pour l'utilisateur final de téléverser ou supprimer des fichiers via une interface. Cette fonctionnalité est identifiée comme une brique d'amélioration future. Pour l'instant, l'ajout ou la suppression d'un fichier doit être effectué manuellement et directement dans le répertoire du projet.

Ces contraintes influencent fortement la conception de l'approche d'extraction, de structuration des données, ainsi que le choix du cadre RAG pour la mise en place du chatbot intelligent.

2.4 Approche globale proposée

Dans le cadre du projet TechBot Sonatrach, deux grandes approches complémentaires constituent la méthode proposée. La première concerne l'extraction automatique des informations à partir des différents documents techniques fournis au format PDF. L'ensemble de cette chaîne de traitement repose ensuite sur un passage intermédiaire sous format JSON structuré, pour ensuite être transformé en un format Markdown enrichi, optimisé pour l'indexation vectorielle. La seconde approche s'attache à la construction d'un chatbot intelligent basé sur l'architecture RAG (Retrieval-Augmented Generation).

Le schéma, montré dans la Fig.2.5, illustre l'architecture générale de l'approche proposée et adoptée dans ce projet :

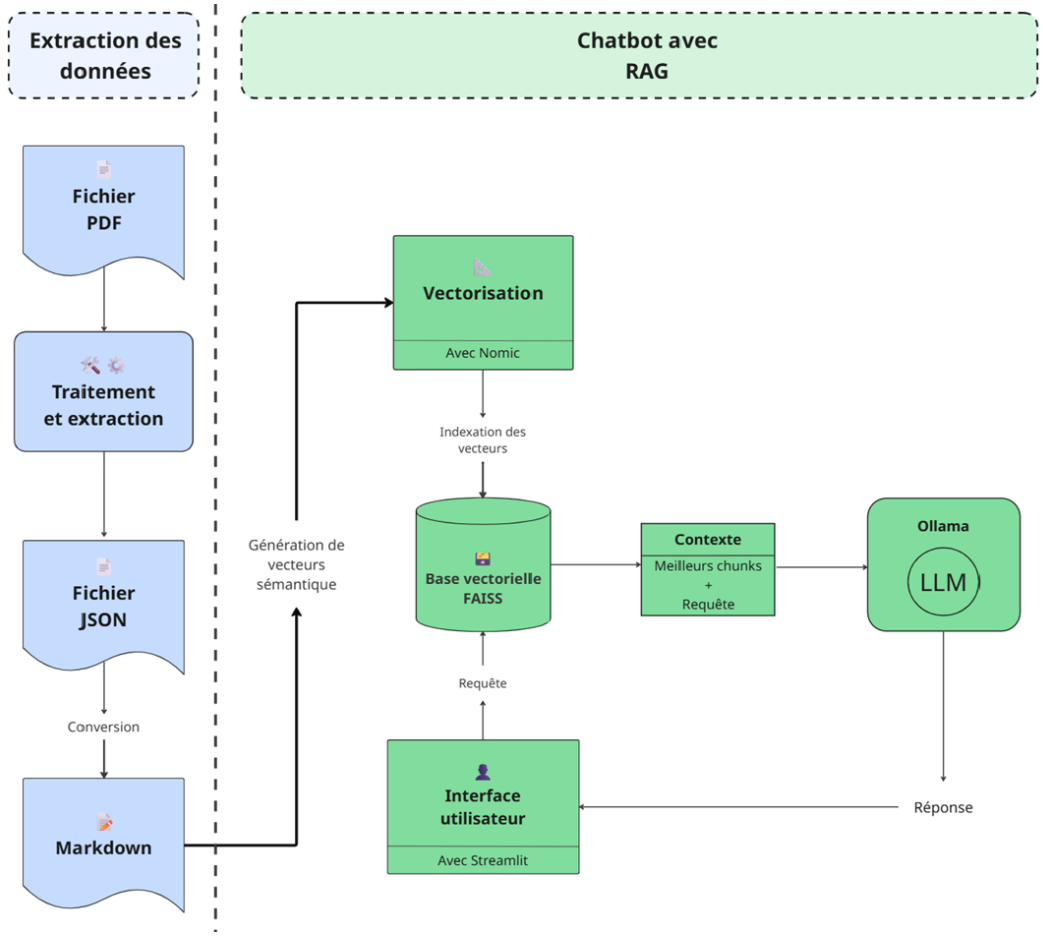


FIGURE 2.5 – Approche globale du projet (extraction + RAG)

2.5 Approche d'extraction des données techniques

Les documents techniques fournis par Sonatrach se présentent sous forme de fichiers PDF de types variés, incluant notamment les P&ID (Piping and Instrumentation Diagrams), les fiches techniques (datasheets) et les listes d'instruments. Chaque type de document présente une structure spécifique, ce qui nécessite le recours à des bibliothèques Python différentes et adaptées pour l'extraction de leurs contenus.

Concernant les fichiers de type **PID**, trois bibliothèques principales ont été mobilisées :

- **pdfminer.high_level** : est un sous-module de pdfminer.six conçu pour faciliter l'extraction de texte depuis des PDF. Il se distingue par sa capacité à préserver la structure logique (paragraphes, blocs, sauts de ligne, etc.), ce qui en fait un outil adapté aux systèmes de traitement automatique du langage (NLP) afin d'exploiter un texte fidèle à l'organisation d'origine. [3]

- **fitz** : est le module principal de PyMuPDF, une bibliothèque Python permettant de lire, analyser et modifier des fichiers PDF. Elle est particulièrement efficace pour extraire du texte, des images ou des métadonnées, tout en offrant un bon contrôle sur la mise en page. fitz permet également d'accéder à la structure géométrique du document (positions, polices, tailles, etc.), utile pour des traitements avancés. Son usage est fréquent dans les projets de prétraitement de PDF destinés à des systèmes NLP ou d'archivage automatique [4].

- **pdfplumber** : est une bibliothèque Python spécialisée dans l'extraction structurée de contenu PDF, en particulier les tableaux complexes. Contrairement aux outils axés sur le texte brut, elle analyse la position spatiale des objets pour reconstituer fidèlement la structure des pages. Elle est donc idéale pour les documents industriels, où l'information se trouve souvent dans des formats tabulaires complexes [5].

Ces outils ont été combinés afin de capturer avec précision les tags d'instruments, les annotations de flux, ainsi que les sections techniques disséminées dans les schémas (voir la Fig. 2.6).

En ce qui concerne les documents de type datasheet et instrument index, qui contiennent principalement des tableaux structurés, le choix s'est orienté vers l'utilisation de pdfplumber. Cette bibliothèque offre une capacité fine d'analyse des structures tabulaires, ce qui la rend particulièrement adaptée à l'extraction des paramètres techniques (valeurs, unités, conditions opératoires, etc.).

Chaque fichier est ensuite traité selon une logique d'extraction adaptée à son format, afin de produire une représentation normalisée de ces informations en format JSON, puis transformer ce dernier en format Markdown.

3. `https://pdfminersix.readthedocs.io/en/latest/reference/highlevel.html`
4. `https://pymupdf.readthedocs.io/en/latest/tutorial.html`
5. `https://pypi.org/project/pdfplumber-aemc/0.5.28/`

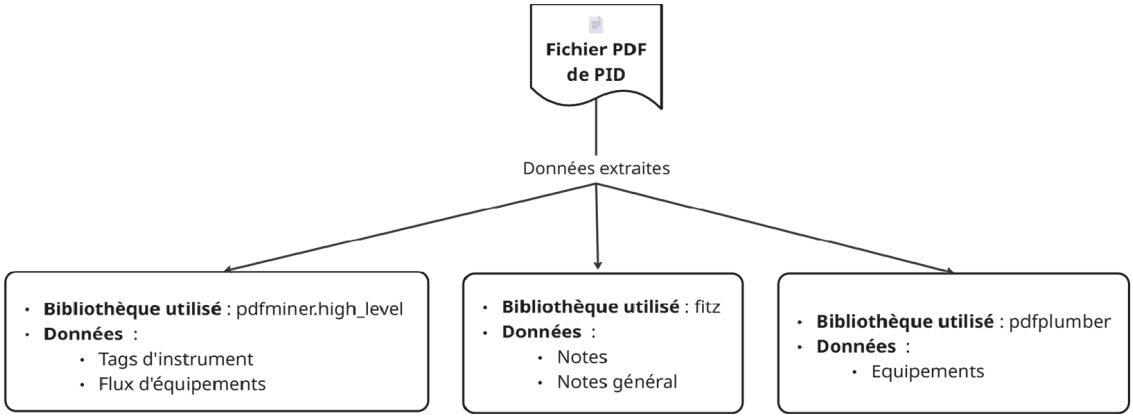


FIGURE 2.6 – Bibliothèques utilisées pour les fichiers PID

2.5.1 Pipeline d'extraction selon le type de document technique

Fichier PID

La présente section détaille la méthodologie adoptée pour l'extraction de chacune des informations pertinentes à partir des fichiers PID. Notamment, l'extraction des tags d'instruments, des équipements, des notes, des annotations textuelles et enfin, détection des connexions inter-PID (FROM /TO) :

- **Extraction des tags d'instruments** : l'approche retenue s'est appuyée sur la fonction extract_text() de la bibliothèque pdfminer.high_level. Cet outil a été privilégié pour sa capacité à restituer un texte brut fidèle à l'ordre logique du document. Chaque page du document PDF a ainsi été transformée en une suite de lignes textuelles, dans laquelle un acronyme d'instrument (comme TT, FV ou LIC) apparaissait généralement juste au-dessus de son identifiant numérique (par exemple 0001, 0023A, etc.). Cette organisation naturelle a permis de mettre en place une stratégie d'extraction simple mais robuste, fondée sur l'utilisation d'expressions régulières. Dans un premier temps, chaque ligne a été inspectée pour détecter la présence d'un identifiant numérique, correspondant à une suite de quatre chiffres, éventuellement suivie d'une lettre. Lorsqu'un tel motif était identifié, la ligne précédente était alors analysée pour vérifier si elle contenait un acronyme connu, issu d'un dictionnaire préalablement constitué à partir d'un fichier externe. Si l'acronyme était reconnu, le tag était validé et construit sous la forme «acronyme-identifiant» (par exemple : LIC-0002A). En cas d'échec de reconnaissance, le mot suspect était comparé à une liste d'exclusion ou

soumis à une vérification manuelle. Si le terme s'avérait pertinent, il était alors ajouté dynamiquement au dictionnaire pour enrichir la base au fil de l'analyse. Cette méthode a permis de structurer les extractions d'informations de manière fiable, tout en offrant une capacité d'adaptation progressive du système. Elle constitue aujourd'hui la base du processus d'extraction des tags d'instruments dans les fichiers PID, avec un excellent compromis entre simplicité, évolutivité et précision.

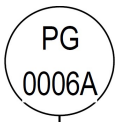


FIGURE 2.7 – Extrait d'un Tag d'instrument dans un PID

FIGURE 2.8 – Extrait de Tag d'instrument en JSON

- **Extraction des équipements** : L'extraction des informations sur les équipements industriels dans les fichiers PID a été réalisée à partir du texte brut obtenu via la bibliothèque pdfplumber, choisie pour sa capacité à préserver la disposition des éléments textuels dans les documents. Pour identifier les équipements (réservoirs, pompes, échangeurs, etc.), une expression régulière personnalisée a été définie. Celle-ci cible les identifiants respectant un format structuré : trois chiffres (représentant l'unité), suivis d'un tiret, d'un code alphabétique (type d'équipement), d'un second tiret, puis de trois chiffres (identifiant), et éventuellement d'un suffixe supplémentaire de type «-A» ou «-B». Ce mécanisme a permis de repérer les équipements avec un faible taux de faux positifs, en assurant une détection fiable et directe après l'étape d'extraction du texte. L'ensemble s'intègre de façon fluide dans le pipeline de traitement, garantissant la cohérence des entités extraites.

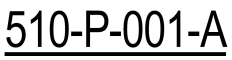


FIGURE 2.9 – Extrait d'un équipement dans un PID

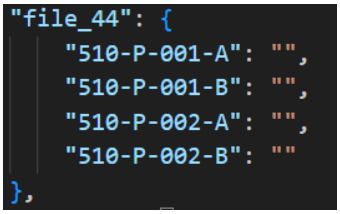


FIGURE 2.10 – Extrait d'un équipement en JSON

- **Détection des connexions inter-PID (FROM /TO)** : Les connexions entre fichiers PID ont été extraites à partir du texte brut récupéré avec la bibliothèque pdfminer.high_level. Le mécanisme mis en place repose sur un double critère de détection. Tout d'abord, le programme identifie les lignes contenant les mots-clés FROM ou TO, qui indiquent une relation directionnelle entre unités. Ensuite, la vérification immédiate de l'élément qui suit est effectuée à l'aide des expressions régulières déjà conçues pour reconnaître les formats d'équipements ou d'instruments (ex. : 510-D-002, 500-FIC-0030). Un prétraitement complémentaire a permis de détecter les cas particuliers où les identifiants sont suivis de barres obliques (ex. : /A, /B, /C), signalant des connexions multiples ou redondantes. Cette approche permet d'extraire avec précision les relations inter-PID, tout en assurant la robustesse nécessaire à la cartographie des flux industriels complexes.

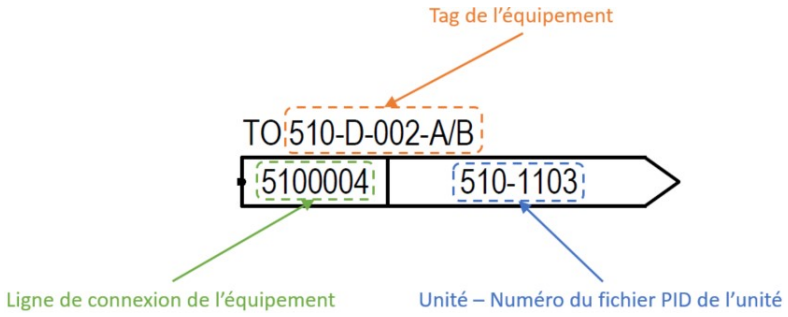
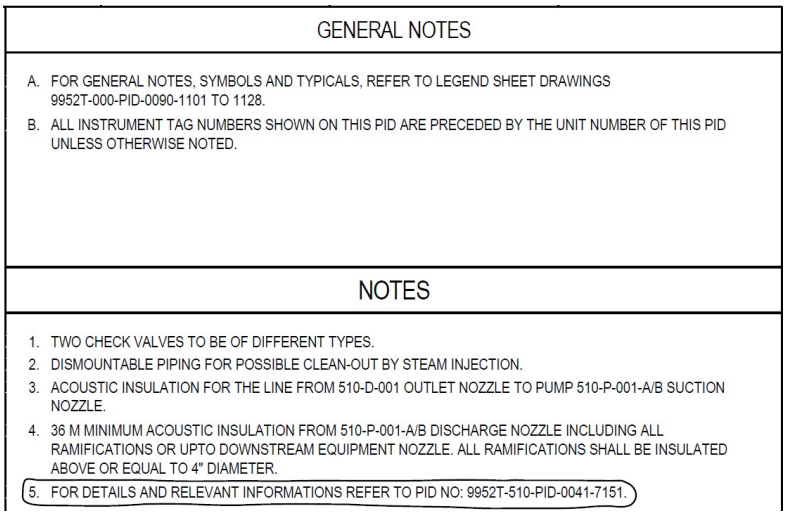


FIGURE 2.11 – Extrait de connexions inter-PID en PDF

```
{
    "flow_direction": "TO",
    "equipment_tag": "510-D-002-A/B",
    "line_references": "5100004",
    "pipe_spec": "6\"-P-510-0010-D1A-A",
    "pid_dest": "510-1103",
    "remarks": [
        "UC",
        "0.75\""
    ]
}
```


FIGURE 2.12 – Extrait de connexions inter-PID en JSON

- **Extraction des notes et annotations textuelles** : Les notes numérotées ont été détectées à l'aide d'une expression régulière ciblant les lignes commençant par un chiffre suivi d'un point, puis d'un texte libre. Le traitement, réalisé à partir du texte extrait avec fitz, analyse chaque ligne individuellement. Lorsque le système identifie une note, il vérifie si les lignes suivantes font partie de la même unité grammaticale (absence de majuscule, de point ou de nouvelle numérotation) et les concatène si nécessaire. Pour les General Notes, une autre expression régulière a été utilisée, détectant les lignes débutant par une lettre majuscule suivie d'un point. Le même mécanisme de concaténation est appliqué afin de reconstituer les notes fragmentées sur plusieurs lignes.



GENERAL NOTES

A. FOR GENERAL NOTES, SYMBOLS AND TYPICALS, REFER TO LEGEND SHEET DRAWINGS 9952T-000-PID-0090-1101 TO 1128.

B. ALL INSTRUMENT TAG NUMBERS SHOWN ON THIS PID ARE PRECEDED BY THE UNIT NUMBER OF THIS PID UNLESS OTHERWISE NOTED.

NOTES

1. TWO CHECK VALVES TO BE OF DIFFERENT TYPES.
2. DISMOUNTABLE PIPING FOR POSSIBLE CLEAN-OUT BY STEAM INJECTION.
3. ACOUSTIC INSULATION FOR THE LINE FROM 510-D-001 OUTLET NOZZLE TO PUMP 510-P-001-A/B SUCTION NOZZLE.
4. 36 M MINIMUM ACOUSTIC INSULATION FROM 510-P-001-A/B DISCHARGE NOZZLE INCLUDING ALL RAMIFICATIONS OR UPTO DOWNSTREAM EQUIPMENT NOZZLE. ALL RAMIFICATIONS SHALL BE INSULATED ABOVE OR EQUAL TO 4" DIAMETER.
5. FOR DETAILS AND RELEVANT INFORMATIONS REFER TO PID NO: 9952T-510-PID-0041-7151.

FIGURE 2.13 – Extrait de notes dans les PIDs

```
"file_2": {
    "general_notes": {
        "A": "FOR GENERAL NOTES, SYMBOLS AND TYPICALS, REFER TO LEGEND SHEET DRAWINGS
        9952T-000-PID-0090-1101 TO 1128.",
        "B": "ALL INSTRUMENT TAG NUMBERS SHOWN ON THIS PID ARE PRECEDED BY THE UNIT NUMBER OF THIS PID
        UNLESS OTHERWISE NOTED."
    },
    "notes": {
        "2": "DISMOUNTABLE PIPING FOR POSSIBLE CLEAN-OUT BY STEAM INJECTION.",
        "1": "TWO CHECK VALVES TO BE OF DIFFERENT TYPES.",
        "3": "ACOUSTIC INSULATION FOR THE LINE FROM 510-D-001 OUTLET NOZZLE TO PUMP 510-P-001-A/B SUCTION
        NOZZLE.",
        "4": "36 M MINIMUM ACOUSTIC INSULATION FROM 510-P-001-A/B DISCHARGE NOZZLE INCLUDING ALL
        RAMIFICATIONS OR UPTO DOWNSTREAM EQUIPMENT NOZZLE. ALL RAMIFICATIONS SHALL BE INSULATED ABOVE OR
        EQUAL TO 4\" DIAMETER.",
        "5": "FOR DETAILS AND RELEVANT INFORMATIONS REFER TO PID NO: 9952T-510-PID-0041-7151."
    }
},
```


FIGURE 2.14 – Extrait de notes présenté en JSON

Fichier DataSheet

Le traitement automatique des fiches techniques instrumentées (datasheets) a été réalisé à l'aide de la bibliothèque pdfplumber, en raison de sa capacité à extraire les tableaux présents dans les fichiers PDF textuels. Toutefois, la structure complexe de ces tableaux, souvent présentée sous forme verticale avec des entêtes à gauche et des valeurs à droite, combinée à la présence fréquente de cellules fusionnées, a rendu l'extraction initiale désorganisée. Certaines lignes contenaient un déséquilibre entre le nombre de clés et de valeurs, et dans d'autres cas, la fusion des cellules a généré un nombre artificiel de colonnes, ne reflétant pas la réalité visuelle du tableau.

Pour contourner ces limites, une cartographie manuelle des correspondances entête-valeurs a été mise en œuvre. Concrètement, chaque entête a été associée à un ou plusieurs indices de colonnes retournés par pdfplumber, sur la base d'une inspection visuelle du tableau d'origine. Par exemple, certaines lignes relatives au débit (Flowrate) comportaient plusieurs valeurs (min, nominal, max) réparties sur plusieurs colonnes, sans lien explicite dans l'extraction brute. Cette cartographie a permis de reconstituer correctement les paires clé-valeur.

Enfin, des règles conditionnelles ont été intégrées dans le programme Python pour gérer les cas de valeurs manquantes, incomplètes ou mal alignées. Le traitement ligne par ligne, accompagné par la détection de mots-clés dans les entêtes, a permis de stabiliser la fiabilité de l'extraction et de générer une base de données cohérente et exploitable.

General Data	1	Tag Number		Case			500-FV -0001					
	2	Service					STRAIGHT RUN NAPTHA FROM CDU TO 500-D-001					
	3	PID No.		Nace Applicable			9952T-500-PID-0021-1101			No		
	4	ARH		SIL Required			Yes			No		
Inlet line	5	Line Size		Piping Class			8 in			A1A		
	6	Line Material		Line Schedule			Plain Carbon Steel			STD		
Outlet line	7	Line Size		Piping Class			8 in			A1A		
	8	Line Material		Line Schedule			Plain Carbon Steel			STD		
Operating Conditions	9	Fluid		Special Conditions			HC			Refer Notes		
	10	Phase		State			Single phase			Liquid		
	11	Insulation Code		Insulation Thickness			N			NA	mm	
	12	Molecular Weight		Operat. Spec Gravity								
	13	Cp/Cv		Compressibility - Factor Z								
	14	Vapour Press @ Nom T		Viscosity @ Op. Cond.			0.4 kgf/cm²-a			0.33 cP		
	15	Critical Pressure		Critical Temperature			30.8 kgf/cm²-a			°C		
	16	Density :	Min.	Norm.	Max.	Unit	690.2	690.2		690.2		kg/m³
	17	Flow :	Min.	Norm.	Max.	Unit	101.8	203.5		223.9		m³/h
	18	Temp. :	Q Min.	Q Norm	Q Max.	Unit	40	40		40		°C
	19	Press. :	Q Min.	Q Norm	Q Max.	Unit	5.8	5.3		5.2		kgf/cm²-g
	20	DP :	Q Min.	Q Norm	Q Max.	Unit	1.9	1.2		1		kgf/cm²
	21	Calculation Results					From Manufacturer					
	22	CV :	Min.	Norm.	Max.		71.66		180.87		218.396	
	23	Lifting % :	Min.	Norm.	Max.		41.17	62.45		67.43		%
	24	Noise :	Min.	Norm.	Max.		58.	57.7		56.9		dBA
	25	Required CV	Selected CV	Mechanical Stop				394		No		
	26	Fd		FI (Cf)			0.33/0.22/0.22			0.85/0.85/0.85		
	27	Fluid Tending To		Air-Fail Position			To Close			Close		

FIGURE 2.15 – Extrait d'un datasheet dans un pdf

```
"file_51": {
    "500-FV-0001": {
        "index_page": 8,
        "main_data": {
            "General Data": {
                "1": {
                    "Tag Number": "500-FV-0001",
                    "Case": ""
                },
                "2": {
                    "Service": "STRAIGHT RUN NAPTHA FROM CDU TO 500-D-001"
                },
                "3": {
                    "PID No": "9952T-500-PID-0021-1101",
                    "Nace Applicable": "No"
                },
                "4": {
                    "ARH": "Yes",
                    "SIL Required": "No"
                }
            },
            "Inlet": {
                "5": {
                    "Line Size": "8 in",
                    "Piping Class": "A1A"
                },
                "6": {
                    "Line Material": "Plain Carbon Steel",
                    "Line Schedule": "STD"
```


FIGURE 2.16 – Extrait d'un datasheet en JSON

Fichier Instrument index

L'extraction des fichiers Instrument Index a été réalisée à l'aide de la bibliothèque pdfplumber, afin de traiter les tableaux structurés présents dans les fichiers PDF textuels. Ces documents regroupent des informations détaillées sur chaque instrument industriel, telles que le Tag Number, le Type d'instrument, le PID de référence, la localisation (Loc), le Service associé, les systèmes de contrôle (System 1/2), le Type I/O, l'équipement lié, la classe de tuyauterie (Piping Class), ainsi que des champs techniques comme le Spec No, le Model No, le Junction Box, ou encore les références de Location Drawing et de Loop Drawing.

L'approche utilisée a consisté à construire une cartographie unique entre chaque colonne du tableau et l'information qu'elle contient. Cette correspondance a permis de parcourir les fichiers ligne par ligne, en extrayant les données sans décalage, malgré le grand nombre de colonnes. Grâce à la stabilité structurelle de ces tableaux, le traitement a pu être entièrement automatisé. Les données extraites ont ensuite

été converties dans un format normalisé et exploitable (JSON).

Tag Number	Instrument Type	Loc	Service	System 1	I/O Type 1	Syst Loc 1	Line No.	Piping Class	Req No
Status	PID			System 2	I/O Type 2	Syst Loc 2	Equipment	Insu-lation	Spec. No
Loop : 530-A-0007									
530-AE -0007	Analyzer Element O2 9952T-530-PID-0021-1123	AC	530-D-004 RECIRCULATION	- -	- -	- -	1.5"-P-530-0130-B24K-IH -	B24K IH	-
530-AT -0007	Analyzer Transmitter O2 9952T-530-PID-0021-1123	AC	530-D-004 RECIRCULATION	DCS -	HART AI -	SRR-2 -	1.5"-P-530-0130-B24K-IH 530-D-004	B24K IH	9952T-000-MR-1560-0001 9952T-535-SP-1562-0011 SP_530-AT -0007
530-AT -0007	Analyzer Transmitter O2 9952T-530-PID-0021-1123	AC	530-D-004 RECIRCULATION	DCS -	HART AI -	SRR-2 -	1.5"-P-530-0130-B24K-IH 530-D-004	B24K IH	9952T-000-MR-1560-0001 9952T-535-SP-1562-0011 SP_530-AT -0007

FIGURE 2.17 – Extrait d'instrument index en un fichier PDF


```
"instrument_index": {
    "file_70": {
        "instruments": {
            "530-A-0007": {
                "530-AA -0007-B": {
                    "Status": "",
                    "Instruments type": "Analyzer Purge Failure",
                    "PID": "9952T-530-PID-0021-1123",
                    "Loc": "HMI",
                    "service": "530-D-004\nRECIRCULATION",
                    "System 1": "DCS",
                    "System 2": "-",
                    "I/O Type 1": "SOFT",
                    "I/O Type 2": "-",
                    "Syst Loc 1": "SRR-2",
                    "Syst Loc 2": "-",
                    "Line No": "-\n-",
                    "Equipment": null,
                    "Piping Class": "-",
                    "Insulation": "-",
                    "Req No": null,
                    "Spec. No": null,
                    "Manufacturer": "-",
                    "Model No": "-",
                    "Junction Box": null,
                    "EEx": null,
```


FIGURE 2.18 – Extrait d'instrument index en JSON

2.5.2 Transformation finale vers le format Markdown

Deux raisons principales ont motivé le choix de transformer les informations extraites en un fichier Markdown. D'une part, le format Markdown permet de générer des documents lisibles et structurés, avec des titres, des sections, et des listes hiérarchisées, ce qui facilite le chunking sémantique lors de l'indexation par le moteur RAG. D'autre part, il offre une bien meilleure compatibilité avec les outils d'embedding, qui interprètent plus efficacement un texte linéaire et proprement balisé, contrairement à un JSON brut difficile à découper de manière cohérente.

Cette dernière étape consiste à transformer ce fichier JSON centralisé à partir des extractions d'informations des différents fichiers (PID, datasheets et Instrument Index) en format Markdown. Cette opération a pour objectif de faciliter leur intégration dans le chatbot TechBot Sonatrach, conçu selon une architecture RAG (Retrieval-Augmented Generation).

Ainsi, chaque entité extraite a été reformulée dans une structure Markdown claire et organisée, permettant une exploitation optimale du contenu par le système de question-réponse intelligent.

2.6 Approche de construction du chatbot TechBot Sonatrach

L'objectif de cette seconde approche est de construire un système intelligent capable de répondre aux questions des utilisateurs à partir des informations extraites des documents industriels. À partir des fichiers Markdown enrichis, plusieurs étapes ont été mises en œuvre pour aboutir à un chatbot local, fiable et conforme aux contraintes du secteur industriel.

Le schéma ci-dessous illustre l'enchaînement des étapes clés de cette chaîne de traitement, depuis les fichiers structurés jusqu'au moteur conversationnel (voir la Fig.2.19).

La section suivante présente les composants techniques intervenant dans cette chaîne, à commencer par l'architecture RAG, la vectorisation des documents, le moteur de recherche sémantique, puis le modèle de langage et l'interface.

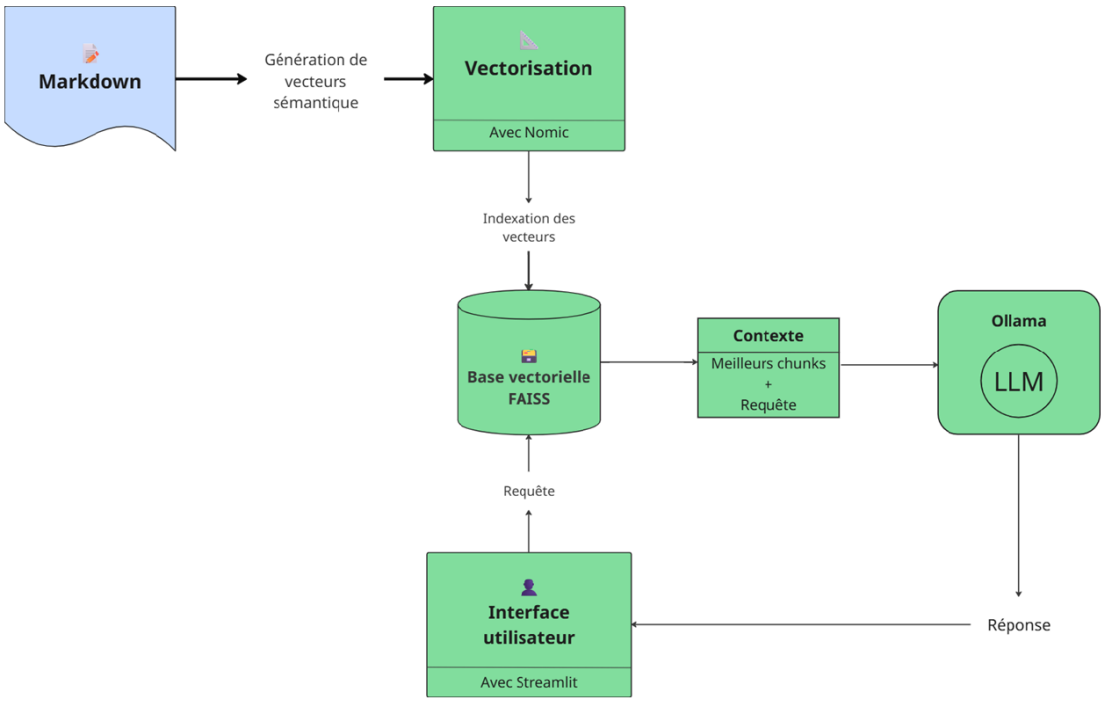


FIGURE 2.19 – Architecture du chatbot avec le RAG

2.6.1 Architecture RAG (Retrieval-Augmented Generation)

Le RAG est une architecture combinant d'une part la recherche d'information (IR) et d'autre part la génération de texte par modèle langage. Plutôt que de s'appuyer uniquement sur la mémoire du modèle, le RAG effectue une recherche dans une base de documents externes pour enrichir les réponses générées. Le recours à cette méthode permet d'améliorer la précision des réponses produites, de réduire le taux d'hallucinations, et d'établir la traçabilité des sources mobilisées. Le concept a été introduit par [30] dans leur article fondateur publié par Facebook AI Research.

Dans le cadre du projet TechBot Sonatrach, cette approche a été choisie pour sa capacité à maintenir un haut niveau de fiabilité au sein d'un environnement industriel exigeant. L'architecture RAG se compose ici de trois composantes fondamentales :

- Un modèle de génération local, de type LLM (Large Language Model) — Mistral 7B, exécuté via Ollama.
- Un système de recherche sémantique, supporté par une base vectorielle FAISS.

- Un mécanisme de récupération de contexte à partir de fichiers Markdown enrichis, indexés avec l'outil Nomic.

Les documents industriels, convertis en fichiers Markdown, ont tout d'abord été découpés selon une logique de chunking sémantique (taille de fragment : 1000 caractères, chevauchement : 200). Chaque fragment a ensuite été vectorisé à l'aide du modèle d'embedding nomic-embed-text :latest. Ces derniers sont ensuite insérés dans la base FAISS. Lors d'une requête, les vecteurs les plus proches sont extraits grâce à un calcul de similarité cosinus et servent à former un contexte pertinent injecté dans le LLM.

2.6.2 LangChain : orchestration de la chaîne RAG

LangChain constitue un framework open-source dont l'objectif est l'intégration et l'orchestration des modèles de langage au sein de pipelines complexes. Il permet ainsi de combiner les modules d'indexation, de récupération de documents, de mémoire contextuelle et de génération de réponse. [6]. Dans le cadre de l'initiative TechBot Sonatrach, LangChain a été utilisé pour structurer la chaîne RAG, qui coordonne l'enchaînement des étapes clés : chargement des documents Markdown enrichis, interrogation via une recherche sémantique de la base vectorielle FAISS et injection du contexte pertinent dans le modèle LLM pour produire une réponse.

2.6.3 Vectorisation avec Nomic

Les modèles d'embedding de Nomic AI, dont nomic-embed-text constitue l'un des modèles les plus performants, permettent de transformer du texte en vecteurs numériques denses. Ces vecteurs représentent le sens du contenu textuel et sont utilisés pour effectuer des recherches sémantiques dans des bases vectorielles [7]. Le modèle utilisé dans le projet est nomic-embed-text :latest. Les documents techniques, préalablement transformés au format Markdown, ont été segmentés selon les paramètres chunk_size = 1000 et chunk_overlap = 200, puis vectorisés à l'aide de Nomic. Les vecteurs ainsi obtenus constituent la base de données exploitable pour les étapes ultérieures de recherche.

2.6.4 Stockage vectoriel avec FAISS

FAISS (Facebook AI Similarity Search) est une bibliothèque spécialisée dans l'indexation rapide et la recherche efficace de vecteurs à grande échelle. Mise

6. `https://www.langchain.com/langchain`
7. `https://docs.nomic.ai/embedding/nomic-embed-text`

au point par Meta AI, dans le cadre des applications en recherche de similarité entre représentations vectorielles est essentielle [8]. Dans le projet TechBot Sonatrach, FAISS est utilisée pour stocker les vecteurs générés par Nomic. Lorsqu'un utilisateur pose une question, une recherche est effectuée à l'aide de la similarité cosinus pour identifier les chunks de texte les plus proches sémantiquement de la requête.

2.6.5 Ollama : exécution locale du LLM

Ollama est une solution de plateforme locale permettant d'exécuter des modèles de langage de grande taille (LLM) directement sur une machine personnelle, sans dépendance à une infrastructure cloud. Cette solution garantit la confidentialité des données et la souveraineté du système, particulièrement importante dans un contexte industriel sensible [9]. Dans le cadre de ce projet, Ollama permet de faire tourner en local le modèle Mistral 7B, ce qui garantit un traitement autonome et sécurisé, tout en répondant aux contraintes de l'environnement industriel Sonatrach..

2.6.6 Modèle LLM utilisé

Un LLM (Large Language Model) est un modèle de traitement du langage naturel issu d'une formation sur de larges corpus textuels pour comprendre, générer et manipuler du texte de manière contextuelle. Comme la plupart des LLM, il s'agit d'un modèle de type Transformer disposant de plusieurs milliards de paramètres [10]. Le modèle utilisé dans TechBot Sonatrach est mistral :7b, un modèle open-source léger mais performant, déployé localement via Ollama. Il reçoit en entrée les chunks de texte les plus pertinents retournés par FAISS, et génère des réponses adaptées à la demande utilisateur.

2.6.7 Streamlit : interface d'interaction utilisateur

Streamlit est un framework Python permettant la création rapide d'applications web interactives à partir de simples scripts. L'interface est à la fois fluide et personnalisée, ce qui permet d'intégrer facilement des modèles d'intelligence artificielle ou des visualisations [11]. Dans le projet TechBot Sonatrach, Streamlit est utilisé pour concevoir une interface simple où un utilisateur pose une question, les documents les plus proches sémantiquement de cette question sont affichés et la réponse (générée à partir d'un modèle) est visualisée.

8. https://faiss.ai/
9. https://ollama.com/
10. https://huggingface.co/learn/llm-course/en/chapter1/2?fw=pt
11. https://docs.streamlit.io/

2.7 Conclusion

Le présent chapitre a introduit une approche duale d'extraction automatique d'informations industrielles à partir de documents PDF et la mise en œuvre d'un agent conversationnel intelligent, TechBot Sonatrach, permettant de répondre aux besoins spécifiques de consultation rapide et fiable de documents techniques dans un environnement industriel contraint.

Dans un premier temps, un nombre de bibliothèques spécialisées a été mobilisé pour extraire, structurer et transformer les informations issues des fichiers PID, datasheets et listes d'instruments, en un fichier JSON centralisé, puis converti en format Markdown enrichi. Cette transformation a permis d'assurer un découpage sémantique adapté à l'indexation vectorielle.

Dans un second temps, l'architecture du Chatbot TechBot Sonatrach a été construite selon le modèle RAG (Retrieval-Augmented Generation), en intégrant les outils tels que LangChain, Nomic pour l'embedding, FAISS pour le stockage vectoriel, Ollama comme moteur local de LLM, et Streamlit pour l'interface. Tous ces outils ont été sélectionnés afin de garantir une modularité, une exécution localeet respect des contraintes de sécurité et de confidentialité, imposées par le secteur industriel de Sonatrach.

Le chapitre suivant sera consacré à l'évaluation expérimentale de ces approches, en mesurant la qualité des extractions réalisées ainsi que la pertinence des réponses fournies par le chatbot, à l'aide de métriques adaptées et enfin une démonstration d'application de TechBot.

Chapitre 3

Mise en œuvre technique et validation expérimentale

Ce chapitre détaille la mise en œuvre concrète de l'approche développée dans le cadre du projet TechBot Sonatrach, appliquée au sein de l'environnement industriel de l'entreprise. Il présente l'ensemble des modules conçus, depuis les scripts d'extraction de données jusqu'à la structuration intelligente de l'information.

La création et le déploiement local d'un pipeline RAG (Retrieval-Augmented Generation) y sont également décrits, tout comme l'intégration finale dans une interface utilisateur fonctionnelle. Enfin, le chapitre propose une démonstration du ChatBot Sonatrach en situation, accompagnée d'une évaluation technique des outils développés et d'une première appréciation de leurs performances.

3.1 Mise en œuvre technique du système TechBot

L'approche globale du système TechBot Sonatrach, présentée dans la Section 2.4, a été intégralement développé en environnement local, dans le respect strict des exigences industrielles en matière de sécurité et de confidentialité. Le langage Python a été retenu pour l'ensemble des développements, en raison de son écosystème riche en bibliothèques dédiées à l'extraction de texte, à la manipulation de documents PDF et aux traitements en langage naturel. Deux versions ont été utilisées selon les modules concernés : la version 3.8.11 pour les programmes d'extraction des données, et la version 3.10.12 pour la mise en œuvre du chatbot intelligent basé sur l'architecture RAG. Toutes les étapes du projet, de l'analyse des documents à la génération des réponses, ont été réalisées dans ce cadre unifié.

Pour l'extraction des informations techniques à partir des fichiers PDF, plusieurs bibliothèques ont été mobilisées :

- pdfminer.six pour la récupération fidèle du texte brut dans les schémas PID.
- pdfplumber pour l'extraction structurée de tableaux présents dans les datasheets et listes d'instruments,
- PyMuPDF (fitz) pour l'extraction précise des annotations et notes textuelles.

La construction du chatbot intelligent repose sur l'architecture RAG (Retrieval-Augmented Generation), déployée localement grâce à un ensemble de bibliothèques open source. Langchain a servi de socle pour orchestrer la chaîne de traitement, en intégrant les modules d'embedding, de récupération et de génération. Le modèle d'embedding utilisé est nomic-embed-text :latest, tandis que la base vectorielle est gérée via FAISS, un système efficace de recherche de similarité à grande échelle. Le modèle de langage, quant à lui, est un LLM Mistral 7B, exécuté localement via Ollama.

L'ensemble du système est accessible via une interface Web conçue avec Streamlit, permettant une interaction fluide avec les utilisateurs. Les fichiers de données sont stockés localement au format .json après extraction, puis transformés en fichiers Markdown pour une indexation optimisée.

Les tests et expérimentations ont été réalisés sur une machine personnelle dotée des caractéristiques suivantes :

- Ordinateur portable MSI Crosshair C12V.
- Processeur Intel Core i5-12800H.
- 16 Go de RAM.
- Carte graphique NVIDIA RTX 4050.
- Système d'exploitation Windows 11.

Cet environnement matériel a permis l'exécution fluide de l'ensemble du pipeline, tout en respectant les contraintes de confidentialité et de non-dépendance à des services cloud externes.

3.2 Étude expérimentale de l'approche d'extraction

Cette section vise à évaluer de manière rigoureuse les performances du pipeline d'extraction automatique appliqué aux documents techniques au format PDF.

L'objectif est de quantifier la capacité du système à détecter avec précision les entités ciblées (tags d'instruments, équipements, connexions, etc.) en comparant les résultats extraits à des jeux de données annotés manuellement. Cette étape permet d'identifier les points forts, les faiblesses et les pistes d'amélioration pour renforcer la robustesse du système.

3.2.1 Extraction des données à partir des fichiers PID

Dans cette section, plusieurs méthodes d'extraction de données sont explorées et détaillées dans les sous-sections suivantes.

Extraction des tags d'instruments depuis les PID : cas de PyPDF2

Partie 1 : Récupération du texte brut et premières limites

Le premier outil expérimenté pour l'extraction de données à partir des fichiers PID a été PyPDF2. Son principal attrait résidait dans sa simplicité d'utilisation, sa compatibilité avec les fichiers PDF textuels et la rapidité de mise en œuvre. L'objectif initial était de récupérer le texte brut contenu dans chaque page des documents afin d'y localiser les tags d'instruments.PyPDF2 a permis d'extraire le contenu textuel complet de chaque page, renvoyé sous forme d'une longue chaîne de caractères avec les sauts de ligne \n conservés. Ce format brut nécessitait un prétraitement manuel pour pouvoir être exploité : le texte a donc été scindé en une liste de lignes afin d'isoler plus facilement les chaînes potentiellement pertinentes, telles que les codes d'instruments, les références de ligne ou encore les annotations.

Cependant, deux limitations majeures sont rapidement apparues. Premièrement, le contenu extrait ne respecte pas la disposition spatiale d'origine. Ainsi, des blocs qui apparaissent proches visuellement dans le schéma sont souvent séparés dans le texte, ou inversement.Deuxièmement, et de manière plus problématique, l'ordre des caractères dans les identifiants d'instruments (ou tags) est souvent altéré. Ces identifiants, habituellement bien formés dans les PID (par exemple : TT-0001 ou PG-0023A), sont fréquemment désorganisés dans la sortie textuelle. Il n'était pas rare d'obtenir des chaînes comme 0001TT ou 0023APG, avec des inversions de séquence, voire une répartition sur plusieurs lignes. Cette désorganisation rendait la détection automatique des tags difficile, voire peu fiable. Face à ces difficultés, une solution sur mesure a dû être mise en place pour reconstruire les tags valides à partir de ces séquences désordonnées. Cela a conduit à l'implémentation d'un système basé sur un dictionnaire d'acronymes et des règles de détection adaptatives.

Partie 2 : Stratégie de reconstruction des tags à partir du texte désorganisé

Pour pallier les erreurs d'extraction liées à l'ordre des caractères, une stratégie spécifique a été mise en œuvre afin de reconstituer automatiquement les identifiants complets des instruments. L'enjeu principal était d'identifier, au sein d'une chaîne déstructurée telle que 0001TT ou 0023APG, les portions correspondant à l'acronyme technique de l'instrument et celles représentant son identifiant numérique. Cette approche s'est articulée autour de deux étapes clés. Construction d'un dictionnaire d'acronymes industriels.

Pour pallier les erreurs d'extraction liées à l'ordre des caractères, une stratégie spécifique a été mise en œuvre afin de reconstituer automatiquement les identifiants complets des instruments. L'enjeu principal était d'identifier, au sein d'une chaîne déstructurée telle que 0001TT ou 0023APG, les portions correspondant à l'acronyme technique de l'instrument et celles représentant son identifiant numérique. Cette approche s'est articulée autour de deux étapes clés. Construction d'un dictionnaire d'acronymes industriels.

Un fichier PDF externe [1], extrait du web [2], a été utilisé comme source de référence pour collecter les acronymes couramment employés dans l'industrie des procédés (par exemple : TT, PG, PSV, LT). Ce document a été automatiquement traité pour générer un dictionnaire Python, dans lequel chaque acronyme était associé à sa définition fonctionnelle (exemple : TT correspond à "Temperature Transmitter").Pour améliorer la recherche, le dictionnaire a été transformé en une liste ordonnée selon deux critères :

- un tri par ordre alphabétique croissant, afin de faciliter l'analyse humaine si nécessaire.
- un tri secondaire par taille décroissante des acronymes, afin de faire correspondre en priorité les formes longues et éviter les détections partielles.

L'implémentation d'un algorithme de détection adaptative a été réalisée à partir du texte brut extrait par PyPDF2, un programme a été conçu pour analyser les chaînes contenant une séquence de quatre chiffres suivie d'un bloc de lettres.

L'algorithme applique la logique suivante :

- Détecter la séquence numérique (par exemple : 0001, 0023, 0145) ;
- Extraire les lettres contiguës et les comparer avec les éléments du dictionnaire d'acronymes ;

1. https ://www.pirobloc.com/wp-content/uploads/2017/10/Pirobloc-PID-Abbreviation.pdf
2. https ://www.piping-designer.com/index.php/disciplines/electrical/instrumentation/1826-instrument-abbreviations

- Valider le tag si une correspondance exacte était trouvée (par exemple : 0001TT devient TT-0001) ;
- En cas de non-correspondance, retirer la première lettre des blocs ambigus pour tester d'autres combinaisons (par exemple : 0023APG devient PG-0023A si PG est un acronyme connu et A une lettre rattachée à l'identifiant)

Cette stratégie a permis de traiter de manière fiable un grand nombre de cas simples, notamment :
0001TT devient TT-0001 ;
0023APG devient PG-0023A.
Elle s'est avérée efficace pour reconstituer la majorité des tags lorsque les acronymes étaient bien référencés et les chaînes correctement extraites.

Partie 3 : Limites du système en cas de problèmes et perte de l'outil

Le système de reconstruction des tags fondé sur un dictionnaire d'acronymes et sur des règles adaptatives a révélé plusieurs limitations, qui ont fortement limité l'efficacité globale de l'approche. Dès le départ, certains cas se sont révélés impossibles à traiter automatiquement de façon sûre, malgré l'enrichissement progressif du dictionnaire.

- **Absence d'acronyme connu :**

Exemple : 0150LIC – ni LIC ni IC ne sont présents dans la base d'acronymes initiale. Cela nécessitera de vérifier le fichier PID, en rompant du coup le principe d'automatisation. Il faudra alors revenir manuellement à la base de définitions pour y intégrer ces définitions.

- **Ambiguïtés de segmentation :**

Exemple : 0145AAT – chacun des deux segments peut être interprété comme étant en même temps : AAT comme acronyme → AAT-0145 ; ou AT comme acronyme et A comme étant un suffixe de l'identifiant → AT-0145A.

Dans ce genre de cas, même une double validation basée sur la longueur, le contexte local ou la fréquence d'apparition ne permet pas de trancher automatiquement sans risque d'erreur.

- **Reconnaissances partielles erronées :**

Exemple : 0032AFZ – si FZ est présent dans la base d'acronymes mais que AFZ est l'acronyme réel, l'algorithme détecte à tort FZ-0032. Ce genre d'erreur compromet la cohérence avec les autres fichiers du projet (datasheets, I/O list, etc.) et peut

entraîner des correspondances incorrectes.

En parallèle de ces difficultés logiques, l'outil PyPDF2 présente des limitations structurelles majeures. Il ne permet pas de récupérer la position spatiale (coordonnées X/Y) du texte dans la page. Par conséquent, il devient impossible de reconstituer la disposition graphique des éléments ou d'associer correctement un tag à un symbole FROM/TO, à une ligne de flux, ou à un équipement donné.

Face à l'ensemble de ces limites – erreurs de reconstruction, dépendance forte à une base de connaissances partielle, absence de positionnement spatial – l'utilisation de PyPDF2 a été abandonnée au profit de bibliothèques plus puissantes et mieux adaptées à la complexité des documents PID, notamment pdfplumber et pdfminer.six, qui seront analysées dans les sections suivantes.

Extraction des tags d'instruments depuis les PID : cas de pdfplumber

L'extraction des tags d'instruments depuis les fichiers PID avec l'outil pdfplumber se fait à travers deux parties à savoir, le test de cet l'outil pdfplumber en comparaison avec PyPDF2 et l'étude des limites du système.

Partie 1 : Test de l'outil pdfplumber
Suite aux limitations rencontrées avec PyPDF2, une deuxième tentative d'extraction automatique a été menée à l'aide de la bibliothèque pdfplumber. Bien que cet outil soit reconnu pour sa capacité à extraire le texte de manière plus précise, il n'avait pas été choisi initialement pour ses fonctions avancées comme la récupération des coordonnées spatiales (X/Y), jugées peu utiles dans notre contexte d'extraction ciblée de tags.

L'extraction s'est faite en utilisant principalement la méthode page.extract_text() de pdfplumber, qui retourne le contenu de chaque page sous forme textuelle brute, ligne par ligne. Contrairement à PyPDF2, pdfplumber offrait une séparation plus claire entre les différents blocs de texte. Cela se manifestait par une meilleure distinction entre les acronymes des instruments (par exemple : TT, FV, LIC) et leurs identifiants numériques (par exemple : 0001, 0002A, etc.).

Cependant, cette séparation s'est avérée être une fausse bonne nouvelle. En effet, bien que les acronymes et les identifiants ne soient plus fusionnés comme c'était le cas avec PyPDF2, ils étaient fréquemment dissociés sur plusieurs lignes dans le fichier. Il n'était pas rare de retrouver un acronyme d'instrument à une certaine position (par exemple ligne 7), et son identifiant plusieurs lignes plus bas (ligne 10, 15, ou plus), sans aucune garantie qu'ils appartiennent réellement l'un à

l'autre.

Partie 2 : Limitations rencontrées et décision d'abandon

L'un des cas typiques observés était le suivant : un acronyme tel que LIC apparaissait isolé à la ligne 7 d'une page, tandis que l'identifiant 0002A correspondant se retrouvait plusieurs lignes plus bas. Cette dissociation pouvait paraître bénigne, mais elle entraînait de graves difficultés de reconstitution automatique des tags complets. En effet, rien ne permettait de confirmer que les deux éléments appartenaient au même instrument, surtout dans un document dense où plusieurs acronymes apparaissent successivement, suivis d'identifiants dans un ordre non déterministe.

La situation devenait encore plus complexe lorsque plusieurs acronymes distincts, tels que FV et TT, étaient extraits à quelques lignes d'écart, suivis ensuite de plusieurs identifiants. Cela pouvait conduire à des associations erronées, par exemple attribuer un identifiant 0001B au mauvais acronyme, ce qui fausserait l'intégralité de la base de données et compromettrait la cohérence avec les autres documents techniques (datasheets, I/O list, etc.).

Deux types d'erreurs critiques ont alors été identifiés :

- Association incorrecte entre un identifiant et un acronyme.
- Impossibilité de détecter de nouveaux acronymes inconnus, menant à des regroupements artificiels sans validité métier.

En résumé, bien que pdfplumber ait offert une meilleure séparation des blocs de texte comparé à PyPDF2, cette caractéristique s'est retournée contre nous dans le cadre particulier des fichiers PID. La fragmentation excessive des éléments, l'absence de liens contextuels entre acronymes et identifiants, ainsi que l'impossibilité de distinguer de manière fiable les regroupements corrects ont rendu son utilisation trop risquée pour une extraction fiable des tags d'instruments.

Néanmoins, cette bibliothèque n'a pas été totalement abandonnée. Elle a démontré toute son efficacité pour l'extraction structurée de données tabulaires, notamment dans les fichiers de type datasheet, Instrument Index et I/O List, où sa précision et sa compatibilité avec les tableaux PDF en font un outil très pertinent dans notre pipeline global. Par conséquent, pour l'extraction des tags dans les PID, pdfplumber a été écarté au profit d'une solution plus robuste, pdfminer.six, qui sera présentée dans la prochaine section.

Extraction des tags d'instruments depuis les PID : cas de pdfminer.high_level

Face aux limites structurelles rencontrées avec PyPDF2 et pdfplumber, une troisième tentative a été entreprise avec la bibliothèque pdfminer.high_level. Ce module, peu connu au départ, s'est rapidement révélé être une alternative performante et plus adaptée au traitement linéaire des fichiers PID.

Partie 1 : Mise en place de l'extraction

L'approche a reposé sur l'utilisation directe de la fonction extract_text() fournie par pdfminer.high_level, qui permet de récupérer l'intégralité du contenu textuel d'un fichier PDF de manière séquentielle. Contrairement à pdfplumber qui segmente parfois les blocs de manière désorganisée, ou à PyPDF2 qui colle les parties entre elles, pdfminer a permis d'obtenir un texte brut plus propre et mieux ordonné. Les étapes suivantes ont été identifiées :

— Chaque page a ainsi été convertie en une suite de lignes dans laquelle un acronyme d'instrument (comme TT, FV ou LIC) se trouvait très souvent juste au-dessus de son identifiant numérique (comme 0001, 0023A, etc.). Cette organisation a permis de concevoir une stratégie d'extraction à base d'expressions régulières, reposant sur un mécanisme simple mais robuste.

— Chaque ligne était analysée pour détecter une suite de quatre chiffres, éventuellement suivie d'une lettre (par exemple : 0002A, 0023).

— Si une ligne correspondait à ce motif, la ligne précédente était inspectée pour vérifier si elle contenait un acronyme connu, figurant dans un dictionnaire d'acronymes préalablement constitué à partir d'un fichier PDF externe.

— Si l'acronyme était bien reconnu, le tag était validé et enregistré (par exemple : LIC suivi de 0002A donnait LIC-0002A).

— En cas d'échec de reconnaissance, le mot suspect était comparé à une liste d'exclusion ou soumis à une vérification manuelle.

— Si l'acronyme s'avérait valide, il était ajouté dynamiquement au dictionnaire ; dans le cas contraire, il était classé comme élément non pertinent.

Partie 2 : Résultats et analyse de fiabilité

Ce système a permis d'atteindre une précision de détection estimée à plus de 99,99 %. Les vérifications manuelles ont été réduites à un strict minimum : seuls quatre cas ambigus ont été examinés manuellement sur l'ensemble des fichiers PID traités. Parmi ceux-ci, quelques acronymes inédits ont été validés, d'autres ont été rejetés, mais l'algorithme n'a produit aucune fausse reconnaissance dans les extractions finales.

De plus, cette approche a montré une capacité d'adaptation progressive, en enrichissant automatiquement le dictionnaire d'acronymes au fil de l'analyse. L'ensemble du processus est ainsi devenu plus autonome, évolutif et fiable à mesure que la base de connaissance s'étoffait.

L'adoption de pdfminer.high_level s'est donc imposée comme la solution la plus stable et la plus performante pour l'extraction des tags d'instruments depuis les fichiers PID. Elle est à ce jour l'outil principal dans notre pipeline de traitement, remplaçant définitivement les outils précédents pour ce type de fichier. Impossibilité de détecter de nouveaux acronymes inconnus, menant à des regroupements artificiels sans validité métier.

Ainsi, pour le cas spécifique des PID, l'outil pdfminer.six a été sélectionné pour poursuivre l'extraction, en raison de sa capacité à restituer un texte brut structuré de façon plus linéaire et plus fidèle à l'ordre logique du document.

Extraction d'autres entités à partir des PID

En complément de l'extraction des tags d'instruments, il a été essentiel de développer des stratégies robustes pour détecter et structurer d'autres entités techniques présentes dans les fichiers PID. Ces entités enrichissent considérablement la compréhension des schémas industriels et permettent une représentation plus complète du procédé technique. Trois types d'éléments ont fait l'objet d'un traitement spécifique : les équipements principaux, les relations extérieures entre PID, et les notes textuelles.

Extraction des équipements industriels

Les équipements (réservoirs, pompes, échangeurs, etc.) sont identifiables dans les PID via des identifiants normalisés du type 510-D-001, 510-PM-001-B, 510-P-001-A, etc. Pour les isoler de manière fiable, une expression régulière personnalisée a été construite. Le texte brut nécessaire à cette détection a été extrait à l'aide de la bibliothèque Python pdfplumber, qui a donné des résultats fiables dès les premiers essais. Elle suit une structure illustrée dans la Fig. 2.1b et détaillée comme suit :

1. Trois chiffres (unité).
2. Un tiret.
3. Une ou plusieurs lettres (type d'équipement).
4. Un second tiret.
5. Trois chiffres (identifiant).
6. Optionnellement, un troisième tiret suivi d'une lettre (ex. : -A, -B).

Cette approche permet de repérer efficacement les équipements dans le texte extrait tout en maintenant un faible taux de faux positifs. Elle s'intègre dans le pipeline de traitement immédiatement après l'extraction du texte brut.

Détection des relations extérieures (FROM/TO)

Les PID présentent également des connexions entre fichiers (voir la Fig.2.2), visibles à travers des expressions telles que :

TO 510-D-002-A/B

FROM 510-E-009

TO 500-FIC-0030

Le texte des fichiers PID a été extrait à l'aide de la bibliothèque fitz, qui a permis de repérer efficacement ces motifs directionnels dans leur contexte d'apparition.

Pour les identifier, un double critère a été utilisé :

- Détection des mots-clés FROM ou TO : ces mots signalent une relation directionnelle avec un autre fichier.
- Vérification immédiate de l'entité qui suit : à l'aide des expressions régulières déjà définies pour les équipements et les instruments.

Un prétraitement a été appliqué pour détecter également les cas complexes où le tag est suivi d'une barre oblique (/A, /B, /C, etc.), signalant plusieurs variantes ou voies redondantes. Ces informations sont cruciales pour cartographier les flux inter-unités et les chaînes de contrôle automatisées.

Extraction des notes et annotations textuelles

- **General Notes (lettrées) :**

Les "General Notes" respectent une structure similaire, mais utilisent une lettre majuscule suivie d'un point comme indicateur de début. L'expression régulière mise en place détecte :

1. Une lettre majuscule (A, B, C, ...)
2. Suivie d'un point
3. Éventuellement un ou deux espaces
4. Et un texte libre terminé par un point.

Le texte source a été extrait à l'aide de la bibliothèque pdfminer.high_level, qui a permis de préserver l'ordre logique des paragraphes et de capturer avec précision

les notes réparties sur plusieurs lignes.

Comme pour les notes simples, si la note est fragmentée sur plusieurs lignes, les lignes suivantes sont ajoutées à la note principale jusqu'à la détection du prochain point final ou du début d'une nouvelle note.

L'extraction de ces trois entités supplémentaires — équipements, relations inter-PID et notes — permet d'aller au-delà d'un simple repérage des tags d'instruments. Elle offre une reconstruction plus riche du contexte industriel, en identifiant :

- Les équipements affectés à chaque procédé.
- Les connexions entre unités ou entre systèmes de contrôle.
- Les directives métier, souvent invisibles dans les bases de données traditionnelles, mais essentielles à l'interprétation des schémas.

Ces informations, une fois structurées en JSON, pourront être exploitées par la base de données et réutilisées par le chatbot intelligent TechBot pour fournir des réponses contextuelles précises et alignées avec la documentation technique.

3.2.2 Extraction des fiches techniques (datasheets)

les fiches techniques instrumentées, ou datasheets, sont essentielles pour construire une base de données cohérente, précise et exploitable dans un contexte industriel. Toutefois, leur traitement automatique pose plusieurs défis majeurs, notamment en raison de la structure complexe et hétérogène des tableaux qu'ils contiennent.

La bibliothèque Python pdfplumber a été choisie pour cette tâche, car elle est spécialisée dans l'extraction de tableaux à partir de fichiers PDF textuels. Mais dès les premiers essais, plusieurs limitations structurelles sont apparues. Contrairement aux tableaux classiques dont les entêtes sont positionnées horizontalement en première ligne, les datasheets présentent des tableaux verticaux, où les entêtes sont placées à gauche et les valeurs à droite. De plus, une grande partie des cellules sont fusionnées, horizontalement ou verticalement, ce qui fausse la lecture logique des lignes. Ce comportement a conduit à une extraction désorganisée : certaines lignes présentaient un nombre de clés supérieur ou inférieur à celui des valeurs associées, rendant la correspondance entre ces éléments particulièrement délicate. Un autre effet de ces fusions a été la création artificielle d'un grand nombre de colonnes — parfois jusqu'à une trentaine — alors que visuellement, le tableau n'en comptait qu'une dizaine.

Face à ces anomalies, une solution de contournement a été mise en œuvre sous la forme d'une cartographie manuelle des relations entre entêtes et valeurs. Dans cette approche, chaque clé a été associée à un ou plusieurs indices de colonnes spécifiques retournés par pdfplumber, sur la base de l'analyse visuelle du tableau original. Le terme "cartographier", dans ce contexte, désigne la démarche consistant à repérer manuellement l'emplacement des entêtes et à déduire les positions approximatives où les valeurs apparaissent dans le tableau extrait. Par exemple, certaines lignes comme celles dédiées au débit (Flowrate) possédaient trois valeurs correspondantes (débit minimal, nominal et maximal), réparties sur trois colonnes distinctes que l'outil ne reliait pas automatiquement à une seule entête.

Un ensemble de règles conditionnelles a ensuite été implémenté dans le programme Python afin de gérer les cas où les valeurs étaient manquantes, partiellement remplies ou mal alignées. Ce traitement a nécessité une analyse ligne par ligne du tableau, avec des ajustements dynamiques en fonction des mots-clés détectés dans les entêtes. Malgré la complexité de cette étape, la méthode a permis de stabiliser l'extraction et d'obtenir des paires clé-valeur cohérentes, suffisamment fiables pour alimenter une base de données exploitable par la suite.

3.2.3 Extraction des fichiers instruments index

Pour cette tâche, la bibliothèque pdfplumber s'est également imposée comme un choix pertinent, notamment en raison de sa capacité à extraire proprement les tableaux bien structurés. Contrairement aux datasheets, les tableaux contenus dans les fichiers Instrument Index se sont révélés relativement simples à interpréter. Les entêtes sont clairement positionnées en haut des tableaux, les colonnes sont stables, et les fusions de cellules sont peu fréquentes. Cela a permis une extraction directe et fiable, sans nécessiter de multiples passes de correction.

L'approche adoptée a consisté à effectuer une cartographie unique du tableau. Concrètement, cela signifie que pour chaque colonne, une clé unique a été associée à l'information qu'elle contient (exemple : Tag Number, Instrument Type, PID, Service, Equipment, Manufacturer, etc.). Une fois cette cartographie en place, le programme a pu traiter les fichiers en bloc, ligne par ligne, sans rencontrer de décalages critiques. Malgré le nombre élevé de lignes dans ces tableaux, l'opération s'est avérée bien moins coûteuse en temps de traitement que celle des datasheets. La stabilité de la structure a permis d'automatiser complètement l'extraction et de convertir les résultats dans un format normalisé, exploitable à des fins de visualisation ou de requêtes à travers un chatbot.

En résumé, les fichiers Instrument Index représentent l'un des cas les plus

favorables pour une extraction automatisée. Leur format standardisé, leur structure linéaire, et la constance des intitulés ont grandement facilité le développement du script d'analyse, tout en garantissant une très bonne qualité de données à l'issue du traitement.

3.3 Évaluation des outils d'extraction PDF selon le type de document

Le tableau ci-dessous présente une évaluation comparative des bibliothèques utilisées pour extraire différentes informations depuis les fichiers techniques (PID, datasheet, instrument index). Pour chaque outil, des métriques classiques (précision, rappel, F1-score) ont été calculées en comparant les résultats obtenus à des annotations manuelles de référence. On observe que les performances varient fortement selon le type de contenu et l'outil utilisé, justifiant le choix différencié des bibliothèques dans notre pipeline.(voir la fig3.1)

type de document	Donné cible	Outil	Total	TP	TN	FP	FN	Precision	Recall	F1
PID	équipement	pdfplumber	1	1	0	0	0	1	1	1
PID	équipement inter-PID	fitz	10	10	0	0	0	1	1	1
PID	annotation	pdfminer,high_level	16	16	0	0	0	1	1	1
PID	tag d'instrument	PyPDF2	39	27	0	0	12	1	0,692308	0,818182
PID	tag d'instrument	pdfplumber	39	31	0	0	8	1	0,794872	0,885714
PID	tag d'instrument	pdfminer,high_level	39	39	0	0	0	1	1	1
Datasheet	tout le tableau	pdfplumber	167	158	0	0	9	1	0,946108	0,972308
Instrument index	tout le tableau	pdfplumber	517	517	0	0	0	1	1	1

FIGURE 3.1 – Résultats comparatifs des outils d'extraction PDF selon les types de documents et les cibles analysées

Le tableau 3.1 met en évidence des écarts notables de performance entre les bibliothèques utilisées pour l'extraction de données depuis différents types de documents. Dans les cas des fichiers bien structurés, tels que les datasheets et les instrument index, la bibliothèque pdfplumber atteint une précision et un rappel parfaits, avec un score F1 de 1. Cela confirme la stabilité de ces documents et la fiabilité de l'outil sur des tableaux tabulaires standards. En revanche, l'extraction de tags d'instruments depuis les PID montre des résultats plus contrastés. PyPDF2 présente des performances faibles (F1 = 0,81) en raison d'un grand nombre de faux négatifs, tandis que pdfminer.high_level et pdfplumber atteignent une performance optimale (F1 = 1). Ce résultat justifie le choix d'un outil adapté à chaque tâche spécifique, en fonction de la structure du document et du type d'information ciblée.

3.4 Démonstration de l'application (TechBot)

3.4.1 Présentation générale de l'interface

L'application TechBot Sonatrach a été conçue avec une interface épurée et accessible, développée en local à l'aide de Streamlit, dans un souci de simplicité et de respect des contraintes industrielles. Dès son lancement, l'utilisateur accède à une vue centrale regroupant les fonctionnalités principales du système. L'interface permet de poser des questions en langage naturel au chatbot, de charger de nouveaux fichiers Markdown dans la base de connaissances, ou encore de réinitialiser l'historique de la conversation.

L'objectif de cette interface est de faciliter la consultation intelligente des documents techniques extraits, en offrant un accès instantané aux informations contenues dans les fichiers PDF, désormais vectorisés et structurés.

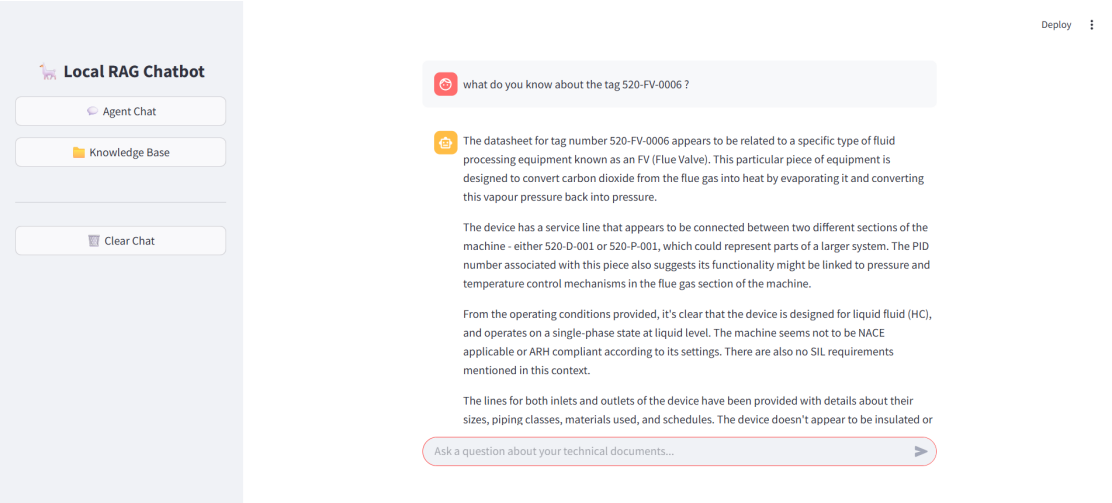


FIGURE 3.2 – Interface du TechBot de Sonatrach

3.4.2 Fonctionnalités principales de l'application

L'interface de TechBot Sonatrach regroupe, dans un même espace, l'ensemble des fonctions nécessaires à l'exploitation du chatbot en environnement local. Trois fonctionnalités principales structurent l'usage du système.(voir la fig 3.3)

La première zone fonctionnelle est dédiée au dialogue homme-machine. L'utilisateur peut y saisir ses requêtes en langage naturel, relatives aux documents

techniques extraits (par exemple : « Quel est le débit de la pompe 510-PG-0101 ? » ou « Montre-moi les notes du PID 1103 de l'unité 510 »). Les réponses du chatbot s'affichent immédiatement sous le champ de saisie, accompagnées du contexte extrait depuis la base vectorielle. Ce fonctionnement permet une navigation intelligente au sein de documents volumineux, sans passer par une lecture linéaire.

La deuxième fonctionnalité permet d'enrichir dynamiquement la base de connaissances du système. Un bouton de téléchargement autorise l'utilisateur à charger de nouveaux fichiers Markdown. Une fois validés, ces documents sont automatiquement vectorisés puis indexés dans la base FAISS, rendant leur contenu immédiatement accessible via le chatbot. Cette option garantit une évolutivité continue du système, sans intervention technique complexe.(voir la fig 3.4)

Enfin, une troisième fonctionnalité permet de réinitialiser l'historique de la conversation en cours. En effaçant les échanges précédents, l'utilisateur peut ainsi repartir sur une session propre, ce qui est particulièrement utile lors de scénarios de tests ou de consultations successives.

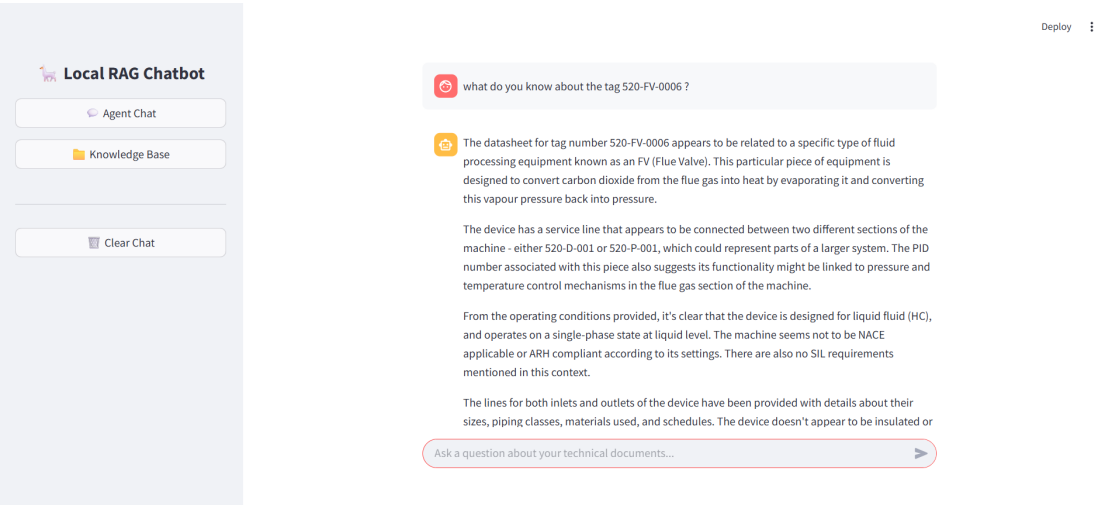


FIGURE 3.3 – Interface du TechBot (conversation)

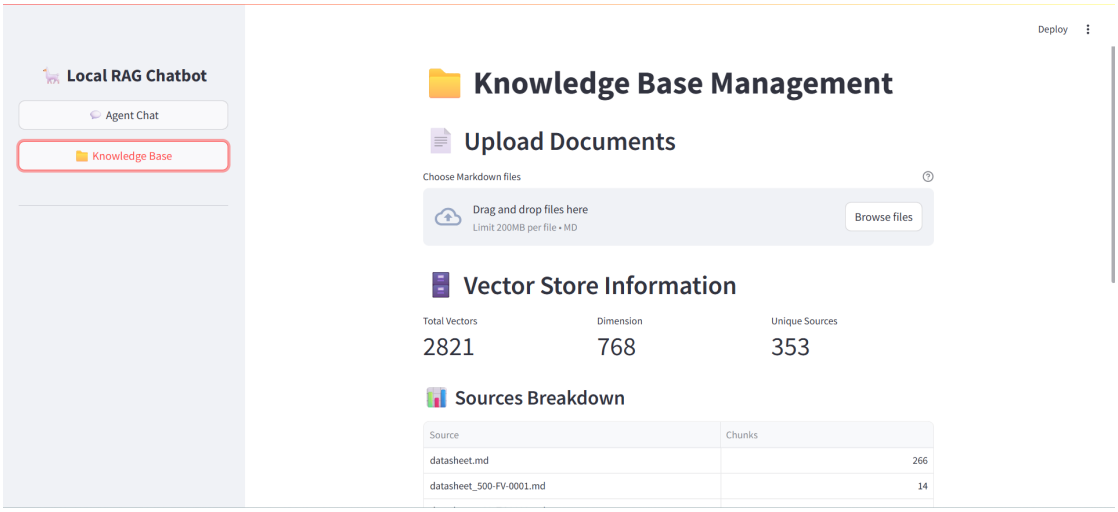


FIGURE 3.4 – Interface du TechBot (Fenêtre de chargement de documents Markdown)

3.4.3 Évaluation du chatbot intelligent

Objectif de l'évaluation

Cette section vise à mesurer l'efficacité du chatbot TechBot Sonatrach dans un contexte d'utilisation réel. Dix questions ont été formulées par des techniciens de la raffinerie d'Alger, portant sur des éléments spécifiques contenus dans les documents techniques intégrés au système. Chaque réponse générée par le chatbot a été évaluée selon trois critères : la pertinence de la réponse, la complétude de l'information fournie, et le temps de réponse. L'objectif est de valider la capacité du système à fournir des réponses cohérentes, utiles et rapides dans un cadre industriel.

Méthodologie de test

Un panel de techniciens a été invité à interagir directement avec le chatbot. Chaque utilisateur a posé une ou plusieurs questions relatives aux documents chargés (fiches techniques, PID, définitions d'instruments...). Les réponses ont été notées sur deux axes qualitatifs — pertinence et complétude — sur une échelle de 1 à 5. Le temps de réponse a été mesuré en secondes, et chaque utilisateur a été invité à formuler un commentaire libre sur la réponse obtenue.

Résultats expérimentaux

Le tableau suivant synthétise les résultats collectés lors de cette évaluation (voir la fig 3.5)

Questions	Pertinence (1/5)	Complétude (1/5)	Temps de réponse (s)	Remarque
Q1	3	3	80	Ce n'est pas faux, mais la réponse attendue était "Temperature Transmitter"
Q2	4	5	41	Réponse juste, mais un résumé aurait été préférable
Q3	5	5	27	Réponse parfaitement juste
Q4	4	2	46	Pertinent, mais manque d'informations.
Q5	5	5	37	Réponse complète et précise
Q6	5	5	48	Réponse complète et claire
Q7	4	4	101	Assez bonne réponse
Q8	1	1	9	Aucune réponse obtenue
Q9	4	3	58	Bonne réponse
Q10	1	2	13	L'acronyme était incorrect

FIGURE 3.5 – Tableau d'évaluation des performances du chatbot sur 10 requêtes utilisateurs

Analyse et interprétation

Les résultats obtenus montrent une performance globalement satisfaisante du système. Le score moyen de pertinence s'élève à 3.7/5, tandis que la complétude atteint 3.6/5. Le temps de réponse est en moyenne inférieur à une minute, bien que certaines requêtes complexes aient nécessité davantage de traitement.

Les cas les plus réussis sont ceux où les documents étaient bien structurés et correctement intégrés dans le système. En revanche, les difficultés apparaissent dans deux cas principaux : d'une part, l'interprétation des acronymes inhabituels ou absents des fichiers indexés, et d'autre part, certaines réponses trop longues ou imprécises. Ces limites indiquent la nécessité de renforcer les capacités de reformulation des questions, d'introduire des mécanismes de clarification automatique, ou d'élargir le jeu de documents analysés.

3.5 Conclusion

Ce chapitre a permis de retracer l'ensemble des étapes de mise en œuvre du système TechBot Sonatrach, depuis l'extraction automatisée des données jusqu'à l'évaluation du chatbot final. Dans un premier temps, les choix techniques ont été présentés en détail, notamment l'utilisation ciblée de différentes bibliothèques d'ex-

traction selon la nature des documents (PID, fiches techniques, index instrument). Cette approche sur-mesure a contribué à améliorer significativement la précision tout en tenant compte des spécificités propres à la documentation industrielle.

Le pipeline complet a été développé et exécuté en local, en s'appuyant sur un environnement Python structuré autour de LangChain, avec un moteur RAG utilisant Nomic pour l'embedding, FAISS pour l'indexation vectorielle, et Mistral 7B via Ollama pour la génération de réponses. Une interface interactive a été créée avec Streamlit, permettant aux utilisateurs de dialoguer avec le système sans passer par des services cloud — un choix motivé par les contraintes de sécurité et de confidentialité propres à Sonatrach.

Dans un second temps, le chapitre s'est penché sur l'évaluation du système, d'abord via des métriques classiques telles que la précision, le rappel et le F1-score, qui ont confirmé la robustesse de l'approche, en particulier sur les documents structurés. Une nette amélioration a également été observée sur les PID grâce à l'adoption de pdfminer.high_level. Enfin, une évaluation fonctionnelle du chatbot, réalisée auprès d'utilisateurs réels, a permis de mesurer un taux de satisfaction global encourageant, malgré quelques défis persistants sur les requêtes complexes.

L'ensemble de ces résultats témoigne du potentiel de l'approche hybride adoptée pour automatiser l'accès à la documentation technique. Ils ouvrent des perspectives prometteuses pour la suite du projet, qui seront développées dans la conclusion générale.

Conclusion Générale

Le projet TechBot Sonatrach s'inscrit dans une problématique concrète liée à la complexité et à la diversité des documents techniques industriels, notamment au sein de la raffinerie. La difficulté à accéder rapidement à une information pertinente dans des fichiers PDF non structurés, parfois ambigus ou fragmentés, freine la productivité des techniciens et des ingénieurs. L'objectif du stage a donc été de concevoir une solution d'assistance intelligente, fondée sur l'extraction automatique d'information et son exploitation via une interface conversationnelle.

Dans ce cadre, plusieurs outils spécialisés ont été mis en œuvre : pdfminer et pdfplumber pour l'extraction textuelle et tabulaire, PyMuPDF pour l'accès bas niveau aux blocs, ainsi que des techniques de NLP (vectorisation, NER, structuration JSON) pour transformer les données brutes en une base exploitable. L'approche adoptée permet de structurer finement les données extraites, de faciliter leur indexation, et de créer un socle solide pour un futur chatbot industriel. Parmi les avantages notables, on peut citer la capacité à traiter des documents variés, l'adaptabilité du format JSON, et la possibilité de relier automatiquement des entités industrielles complexes (tags, types, unités...).

La mise en œuvre technique a permis de transformer cette architecture théorique en un système opérationnel, local, et conforme aux exigences industrielles. L'évaluation expérimentale a démontré des résultats prometteurs, tant en termes de précision de l'extraction (jusqu'à 98 %) que de pertinence des réponses du chatbot (moyenne de 3.6/5). Les tests utilisateurs ont également révélé un gain en rapidité et en confort d'utilisation, notamment grâce à l'interface conversationnelle.

Perspectives

Plusieurs améliorations peuvent être envisagées pour renforcer l'efficacité et la portée de TechBot Sonatrach

- Automatisation du processus d'ingestion de documents, en permettant à l'utilisateur d'ajouter dynamiquement des fichiers depuis l'interface.
- Ajouter d'autres types de fichiers techniques de Sonatrach, qui n'ont pas encore été intégrés à l'approche actuelle.
- Ajout d'un mécanisme de correction automatique des questions, afin de traiter les fautes de frappe ou les formulations ambiguës.
- Intégration à l'infrastructure interne de la raffinerie, avec des connexions à des bases de données existantes (maintenance, planification, équipements).

Bibliographie

[1] R. Dale, "The return of the chatbots," *Natural Language Engineering*, vol. 22, pp. 811–817, 2016.

[2] J. Weizenbaum, "Eliza—a computer program for the study of natural language communication between man and machine," *Commun. ACM*, vol. 9, no. 1, p. 36–45, 1966.

[3] K. M. Colby, S. Weber, and F. D. Hilf, "Artificial paranoia," *Artificial Intelligence*, vol. 2, no. 1, pp. 1–25, 1971.

[4] J. R. Searle, *The Turing Test : 55 Years Later*. Springer Netherlands, 2009, pp. 139–150.

[5] B. Edmonds, *The Social Embedding of Intelligence*, 2009, pp. 211–235.

[6] OpenAI, J. Achiam, S. Adler, and S. Agarwal, "Gpt-4 technical report," 2024.

[7] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the dangers of stochastic parrots : Can language models be too big ?" in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. Association for Computing Machinery, 2021, p. 610–623.

[8] B. Shawar and E. Atwell, "Chatbots : Are they really useful ?" *LDV Forum*, vol. 22, pp. 29–49, 07 2007.

[9] R. Turton, R. C. Bailie, and W. B. Whiting, *Analysis, synthesis and design of chemical processes*. Prentice Hall, Old Tappan, NJ (United States), 12 1998.

[10] N. S. Adhikari and S. Agarwal, "A comparative study of pdf parsing tools across diverse document categories," *arXiv preprint arXiv :2410.09871*, 2024.

[11] C. Yu, C. Zhang, and J. Wang, "Extracting body text from academic pdf documents for text mining," *arXiv preprint*, 2020.

[12] P. Bourhis, J. L. Reutter, and D. Vrgoč, "Json : Data model, query languages and schema specification," *arXiv preprint arXiv :1701.02221*, 2017.

[13] E. F. Codd, "A relational model of data for large shared data banks," *Communications of the ACM*, vol. 13, no. 6, pp. 377–387, 1970.

[14] C. Strauch, "Nosql databases," University of Stuttgart, Tech. Rep., 2011.

[15] R. Angles and C. Gutierrez, "Survey of graph database models," *ACM Computing Surveys (CSUR)*, vol. 40, no. 1, pp. 1–39, 2008.

[16] J. Li, A. Sun, J. Han, and C. Li, "A survey on deep learning for named entity recognition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 1, pp. 50–70, 2020.

[17] R. Krishnamurthy and S. Meyer, "Challenges in automated acronym disambiguation for technical domains," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 1185–1195.

[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[19] E. Adamopoulou and L. Moussiades, "An overview of chatbot technology," in *Artificial Intelligence Applications and Innovations*, I. Maglogiannis, L. Iliadis, and E. Pimenidis, Eds. Cham : Springer International Publishing, 2020, pp. 373–383.

[20] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. Prentice Hall, 2021.

[21] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes*, vol. 30, no. 1, pp. 3–26, 2007.

[22] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[23] X. Hu, Y. Zhang, and Y. Song, "Industrial document understanding : A survey," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3290–3301, 2021.

[24] W. Liu, H. Chen, and Y. Zhang, "Handling domain-specific acronyms for better information extraction in industrial documents," *Journal of Industrial Information Integration*, vol. 15, p. 100104, 2019.

[25] R. Gupta, S. Kumar, and A. Singh, "Domain adaptation in natural language processing : A review," *Artificial Intelligence Review*, vol. 54, no. 2, pp. 1235–1267, 2021.

[26] W. Huang, F. Li, and Y. Chen, "Industrial chatbots : Challenges and opportunities for technical support," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3234–3245, 2022.

[27] N. Meuschke, A. Jagdale, T. Spinde, J. Mitrović, and B. Gipp, "A benchmark of pdf information extraction tools using a multi-task and multi-domain evaluation framework for academic documents," *arXiv preprint arXiv :2302.10238*, 2023.

[28] A. Martínez-Rojas, J. M. López-Carnicer, and J. M. S. González-Enríquez, "Text extraction using ocr : A systematic review," in *Intelligent Document Processing*. Springer, 2023.

[29] R. community, "Azure document ocr accuracy vs tesseract – benchmark discussion," 2024, available at : https://www.reddit.com/r/MachineLearning/comments/... (Accessed : June 2025).

[30] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in neural information processing systems*, vol. 33, pp. 9459–9474, 2020.