



République Tunisienne  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université de Carthage - Ecole Supérieure de la Statistique et de l'Analyse de l'Information



*Rapport de Projet de Fin d'Etudes soumis afin d'obtenir le*  
**Diplôme National d'Ingénieur en Statistique et Analyse de l'Information**



*Réalisé par*

SALMA ABDESSLAM

---

# GenAI TAILOR : Former, adapter et évaluer les grands modèles de langage (LLMs)

---

Soutenu le 19/06/2024 devant le Jury composé de :

Dr. Ghazi BELMUFTI	Président
Dr. Tasnim HAMDENI	Rapporteur
Mme Amel BEN ABID	Encadrant entreprise
M. David LUZ	Encadrant entreprise
Mme Rana BOUALLAGUE	Encadrant entreprise
Dr. Ines ABDELJAOUED-TEJ	Encadrant universitaire

*Projet de Fin d'Etudes fait à*

FORVIA

**FORVIA**  
Inspiring mobility

# Dédicace

*À mon père, étoile éternelle dans le firmament de mon cœur,*

*Même si ton absence creuse un vide immense, que seuls les souvenir parviennent à combler.*

*Je te dédie ces mots comme des fleurs déposées sur ta tombe, un hommage vibrant à l'homme extraordinaire que tu étais.*

*Ton soutien et ta sagesse bienveillante ont façonné mon être, et ton esprit indompté continue de m'inspirer chaque jour.*

À ma mère, havre de paix et source intarissable d'affection,

*Tes bras ont été mon refuge, tes paroles mon baume, et ton amour maternel, un phare qui a illuminé ma vie.*

*Je te remercie pour ton dévouement sans bornes, pour tes sacrifices silencieux et pour la force inébranlable que tu m'as transmise.*

*Grâce à toi, j'ai appris la force de l'amour, la valeur du travail et la persévérance face aux épreuves.*

*Tu m'as inculqué la conviction que tout est possible si on y croit et que l'on se bat avec détermination.*

À mes amis, compagnons de route et confidents inestimables.

*Votre amitié est un trésor précieux que je chéris chaque instant.*

Merci pour votre présence indéfectible, vos encouragements constants et votre soutien indéfectible.

*Vous avez su me faire rire aux larmes, me consoler dans les moments difficiles et célébrer mes victoires avec ferveur.*

Vous êtes ma deuxième famille, ma famille de cœur, et je suis infiniment reconnaissante et fière de vous avoir dans ma vie.

# Remerciements

Je tiens à exprimer ma profonde gratitude à toutes les personnes qui ont contribué à la réalisation de ce mémoire.

En premier lieu, je tiens à exprimer ma profonde gratitude à ma mère et à mes deux frères, pour leur amour inconditionnel, leur soutien indéfectible et leur confiance en moi tout au long de mon parcours. Ils ont toujours été présents pour moi, m'encourageant à poursuivre mes rêves et à croire en mes capacités. Je suis infiniment reconnaissant de leur présence dans ma vie.

Je remercie ensuite chaleureusement mon encadrante universitaire, Dr. Ines Abdeljaoued-Tej, pour ses précieux conseils, sa disponibilité et son soutien indéfectible tout au long de ce projet. Sa confiance en moi m'a permis de repousser mes limites et de me surpasser. Je suis reconnaissante pour sa patience, sa bienveillance et sa rigueur, qui ont grandement contribué à la qualité de ce mémoire.

J'aimerais également dire à Dr. Ghazi BELMUFTI à quel point je suis honorée pour avoir accepté de présider ce jury de PFE. Je suis infiniment gré à Dr. Tasnim HAMDENI de s'être rendu disponible et d'avoir accepté la fonction de rapporteur.

Je suis également reconnaissante envers mes encadrants de l'entreprise, M. David Luz et Mme Rana Bouallague, pour leur expertise, leur patience et leur guidance. Ils m'ont permis d'acquérir une expérience professionnelle précieuse et de mettre en pratique mes connaissances théoriques.

Je remercie également Mme Amel Ben Abid, la manager du département Data dans lequel j'ai effectué mon stage, pour son accueil chaleureux et sa bienveillance.

Elle a su me mettre à l'aise et me donner l'opportunité de développer mes compétences.

Enfin, je ne saurais oublier mes collègues, qui ont toujours été là pour moi, pour m'encourager et me faire partager leur bonne humeur.

Je n'oublie pas mes amis, Dhia, Siwar, Azza, Eya et Aziz, qui ont toujours été là pour moi, pour me soutenir, m'encourager, m'élever lorsque j'étais au plus bas, et surtout me faire rire même dans les moments les plus difficiles.

Ils ont su croire en moi et m'apporter la force nécessaire pour persévérer.

Merci du fond du coeur.

## Résumé

Le présent projet a pu réaliser deux volets. Le premier volet visait à améliorer l'exploitation de la documentation des procédures de base de FORVIA (*Forvia Core Procedures*) ou *FCP* en développant un agent conversationnel et plus précisément un module basé sur l'intelligence artificielle générative : (*AIP Widget*), capables de répondre à tout types de questions, posées par des utilisateurs en interne, concernant les documents *FCP*. Nous avons choisi des grands modèles de langage (*LLMs*) pour atteindre cet objectif. Parmi les modèles de langage envisagés, il y a **OpenAI GPT-4-32k**, **OpenAI GPT-4o** et finalement **OpenAI GPT 4-Vision-Turbo**, qui est un modèle Multi-Modal capable de traiter à la fois du texte et des images. Toutefois, ces modèles présentent des contraintes telles que des coûts élevés des appels de l'interface de programmation d'application. Par conséquent, on a eu recours à une évaluation (*Benchmarking*) dans le deuxième volet de ce projet, ce qui a permis de fixer et mesurer des indicateurs clés de performance recherchés par l'entreprise FORVIA, afin de répondre à la question : *L'investissement dans ce type de modèles de langage payants est-il vraiment nécessaire ou pas ?*

Cette analyse comparative a été précédée par l'application de la méthode de génération augmentée par récupération (*RAG*), afin d'adapter les *LLMs* aux données de l'entreprise (documentation, contrat, rapport financier, etc.). Ce projet a permis à FORVIA non seulement d'exploiter plus précisément la documentation de base, en évitant une analyse manuelle coûteuse, chronophage et sujette aux erreurs, mais également de disposer de mesures précises des indicateurs de performance clés. Ces mesures ont démontré l'efficacité des différents modèles intégrés dans sa plateforme (Palantir Foundry) pour les tâches ciblées.

**Mots clés**— Traitement Automatique de Langage, Intelligence Artificielle Générative, Génération Augmentée par Récupération, Analyse Comparative.

## Abstract

This project was able to achieve two main objectives. The first objective aimed to improve the utilization of the Forvia Core Procedures (*FCP*) documentation by developing a conversational agent, specifically a module based on generative artificial intelligence (*AIP Widget*), capable of answering all types of questions posed by internal users regarding the *FCP* documents. We selected large language models (*LLMs*) to achieve this goal. Among the considered language models were **OpenAI GPT-4-32k**, **OpenAI GPT-4o**, and finally **OpenAI GPT-4-Vision-Turbo**, which is a multi-modal model capable of handling both text and images.

However, these models present constraints such as high API call costs. Therefore, we conducted a benchmarking evaluation in the second part of this project, which allowed us to establish and measure key performance indicators sought by FORVIA, in order to answer the question : *Is the investment in these types of paid language models really necessary ?*

This comparative analysis was preceded by the application of the Retrieval-Augmented Generation (*RAG*) method to adapt the *LLMs* to the company's data (documentation, contracts, financial reports, etc.). This project enabled FORVIA not only to more precisely exploit the core documentation, avoiding costly, time-consuming, and error-prone manual analysis, but also to have precise measurements of key performance indicators. These measurements demonstrated the effectiveness of the different models integrated into its platform (Palantir Foundry) for the targeted tasks.

**Keywords**— Natural Language Processing (NLP), Generative AI, Retrieved Augmented Generation (*RAG*), Benchmarking.

# Table des matières

<b>Table des figures</b>	iii
<b>Liste des tableaux</b>	v
<b>Introduction générale</b>	1
<b>1 Introduction au TAL et à la Transformation Digitale</b>	2
1.1 Organisme d'accueil . . . . .	2
1.1.1 Adaptation de l'industrie 5.0 . . . . .	2
1.1.2 Groupe de Transformation Digitale de FORVIA (FDT) . . . . .	3
1.1.3 Plateforme Palantir . . . . .	4
1.1.4 Méthodologie de travail . . . . .	5
1.2 Problématique . . . . .	6
1.3 Etude de l'application développée sur <i>Power Apps</i> . . . . .	7
1.4 Traitement automatique de langage naturel (TAL) . . . . .	10
1.4.1 Histoire du TAL : L'aube des machines communicantes . . . . .	10
1.4.2 Apparition des Ontologies Conceptuelles - Approche de 1950-1990 . . . . .	12
1.4.3 Approche statistique (années 1990-2010) . . . . .	14
1.5 Modèles de Markov . . . . .	15
1.5.1 Chaînes de Markov . . . . .	15
1.5.2 Modèles de n-grammes . . . . .	17
1.6 Du Perceptron aux Grands Modèles de Langage . . . . .	18
1.6.1 Réseaux de neurones récurrents <i>RNN</i> . . . . .	19
1.6.2 Limitations des <i>RNN</i> classiques . . . . .	22
1.6.3 Transformateurs et Mécanismes d'Attention . . . . .	25
1.7 Conclusion . . . . .	32
<b>2 Sélection et Évaluation des Modèles pour un <i>Widget</i> d'IA Performant</b>	33
2.1 Modèles utilisés . . . . .	33
2.1.1 Grands Modèles de Langage ( <i>LLMs</i> ) . . . . .	33
2.1.2 Choix du modèle d'Encodage ( <i>Embeddings Model</i> ) . . . . .	35
2.2 Contributions . . . . .	40
2.2.1 Documents de départ . . . . .	40
2.2.2 Repérage et description d'images via <b>GPT-4-Vision-Turbo</b> . . . . .	47
2.2.3 Evaluation des Grands Modèles de Langages ( <i>LLMs</i> ) . . . . .	52
2.2.4 Calcul des KPIs . . . . .	59
2.3 Crédit à l'organisme . . . . .	65

2.4 Conclusion . . . . .	66
<b>Conclusion et Perspectives</b>	<b>69</b>
<b>Annexe</b>	<b>70</b>
<b>A Conversion de documents</b>	<b>71</b>
<b>Bibliographie</b>	<b>72</b>

# Table des figures

1.1	Investissements adaptés par FORVIA pour sa transformation digitale . . . . .	3
1.2	Intelligence artificielle sur Foundry . . . . .	4
1.3	Quelques applications de Foundry . . . . .	5
1.4	Flux de données dans une plante [Faurecia, 2024] . . . . .	7
1.5	Page d'Accueil de Power Apps ( <i>Forvia Core Procedures</i> ) . . . . .	8
1.6	Fonction de recherche plus détaillée . . . . .	9
1.7	Définitions Technologiques entre l'existant et la Nouvelle Application . . . . .	10
1.8	"Le cerveau" ou IBM 701 traduisant automatiquement du texte russe vers l'anglais . .	11
1.9	Représentation schématique d'une Ontologie Conceptuelle . . . . .	13
1.10	Comparaison entre les différents modèles de représentation sémantique . . . . .	14
1.11	Chaîne de Markov pour représenter les transitions entre un Verbe et un Nom . . . .	16
1.12	Modèles de n-grammes . . . . .	17
1.13	Composition et architecture des réseaux de neurones à propagation descendante . . . .	18
1.14	Correction d'erreur par rétropropagation . . . . .	19
1.15	Boucle de rétroaction . . . . .	20
1.16	Technique de Troncation de la Rétropropagation à travers le Temps ( <i>TBPTT</i> ) . . . .	22
1.17	Maîtrise de l'explosion des gradients ( <i>Gradient Clipping</i> ) . . . . .	22
1.18	Problème des réseaux de neurones récurrents . . . . .	23
1.19	Problème des réseaux de neurones récurrents . . . . .	23
1.20	Mécanisme clé des Réseaux de mémoire à long terme ( <i>LSTM</i> ) . . . . .	24
1.21	Architecture du Transformateur [Han et al., 2021] . . . . .	25
1.22	Obtention des matrices clé ( $K$ ), valeur ( $V$ ), requête ( $Q$ ) . . . . .	26
1.23	Mise en relation des matrices clé ( $K$ ) et requête ( $Q$ ) . . . . .	27
1.24	Traitement du produit scalaire obtenu . . . . .	27
1.25	Mécanisme d'Attention : Attribution de scores . . . . .	28
1.26	Mécanisme d'attention dans un Transformateur . . . . .	29
1.27	Processus d'encodage . . . . .	30
2.1	Architecture du décodeur uniquement ( <i>Decoder-only</i> ) . . . . .	34
2.2	Catalogue des modèles d'encodage contenant uniquement <i>text-embedding-ada-002</i> . .	36
2.3	Exemple d'utilisation du terme "Plant" au sein de FORVIA . . . . .	36
2.4	Vision Transformers (ViT) . . . . .	39
2.5	Visualisation du nombre de pages par document . . . . .	41
2.6	Schéma explicatif d'un Media Set . . . . .	41
2.7	Base de données contenant les Fichiers <i>FCP</i> . . . . .	42
2.8	Répartition du contenu textuel et visuel dans les documents PDFs . . . . .	43
2.9	Répartition des types des éléments visuels . . . . .	43

2.10	Exemples d'éléments visuels présents dans les PDFs . . . . .	44
2.11	Résultat de l'extraction du texte avec PDFplumber . . . . .	45
2.12	Résultat de l'extraction de texte à partir d'un fichier PDF avec <i>Tesseract</i> . . . . .	45
2.13	Résumé du pipeline du processus de l'extraction du texte . . . . .	46
2.14	Exemple d'un résultat de l'étape d'extraction du texte . . . . .	46
2.15	Description superficielle des images dans une page . . . . .	48
2.16	Résultat de l'application du <i>co-prompt engineering</i> . . . . .	49
2.17	Détection du logo dans la description des images . . . . .	49
2.18	Amélioration des descriptions des images . . . . .	51
2.19	Colonnes obtenues pour chaque page (Texte + Description des images) . . . . .	52
2.20	Ensemble de données contenant les questions et les réponses de références . . . . .	53
2.21	Génération augmentée par récupération ( <i>RAG</i> ) . . . . .	54
2.22	Question avec son texte de référence . . . . .	54
2.23	Exemple de diagramme dont Copilot (IA) n'est pas capable d'interpréter . . . . .	55
2.24	Question en rapport avec les images avec la description fournie au modèle . . . . .	56
2.25	Catalogue des Modèles de langages disponibles sur Foundry . . . . .	57
2.26	Modèles pris en considération dans ce projet . . . . .	57
2.27	Similarité cosinus (calculée) versus score de similarité pour <b>GPT-4-32k</b> . . . . .	59
2.28	Similarité cosinus (calculée) versus le score de similarité pour <b>GPT-4o</b> . . . . .	60
2.29	Moyennes des scores de similarité obtenues par document . . . . .	62
2.30	Moyennes des scores de similarité obtenues pour tous les documents . . . . .	63
2.31	Moyennes des scores de similarité obtenues par document pour les images . . . . .	64
2.32	Moyennes des scores de similarité obtenues pour tous les documents d'images . . . . .	64
2.33	Sélection du document proposé par l'agent conversationnel . . . . .	65
2.34	Visionneuse de documents ( <i>document viewer</i> ) . . . . .	66
2.35	Analyse des scores de similarité pour <b>GPT-4-32k</b> . . . . .	67
2.36	Analyse des scores de similarité pour <b>GPT-4o</b> . . . . .	68
A.1	Conversion d'une page PDF en JPEG ensuite en <i>base64_encoded</i> . . . . .	71

# Liste des tableaux

2.1	Classement du modèle <i>text-embedding-ada-002</i> selon le benchmark <i>MTEB</i> . . . . .	37
2.2	Similarité cosinus mesurant les éléments communs aux phrases choisies . . . . .	38
2.3	Comparaison des scores de similarité . . . . .	67

# Introduction Générale

La transformation digitale est devenue un enjeu crucial pour les entreprises de tous secteurs, leur permettant de rester compétitives sur le marché. Lorsqu'on parle de transformation digitale, nous pensons souvent à la digitalisation des archives. Cependant, cette transformation a considérablement évolué au fil du temps.

Parmi les exemples les plus emblématiques de la transformation digitale dans cette nouvelle ère, il y a des entreprises comme META, Amazone et Netflix, qui exploitent l'intelligence artificielle pour personnaliser le contenu destiné à leurs clients. Pour notre entreprise d'accueil FORVIA, la transformation digitale est encore plus importante et coûteuse, car le secteur manufacturier n'a que peu de liens avec la digitalisation jusqu'à ces dernières années.

FORVIA a été parmi les premières entreprise à adopter l'industrie 4.0, en numérisant tous les *outputs* de ses machines grâce à des capteurs. Plus récemment, avec l'industrie 5.0, l'intégration de l'intelligence artificielle permet de faire des prédictions, comme pour l'amortissement des machines et les pannes techniques.

Notre projet est en quelque sorte unique dans le secteur manufacturier en général et pour FORVIA en particulier. Cette fois-ci, les données traitées, ne sont pas quantitatives ou qualitatives (comme celles générées par les machines : pression, température, nombre de cycles opérationnels, etc.), mais **textuelles et visuelles**. Ces données proviennent de documents contenant les procédures de base de FORVIA (*Forvia Core Procedures* ou *FCP*), qui sont des fichiers PDF. Ces documents sont très importants et sensibles, car ils constituent une sorte de constitution sur laquelle se basent toutes les filiales de FORVIA à travers le monde.

Traiter ces documents demande beaucoup de précautions, d'autant plus que l'un des objectifs de ce projet est de créer un agent *AIP Widget*, capable de répondre aux questions des utilisateurs sur les documents *FCP*. Ce *widget* jouera le rôle de guide pour les collaborateurs et les employés de FORVIA, et le fait qu'il donne des réponses non pertinentes ou fasse des hallucinations serait très mal perçu.

De plus, un autre objectif de ce projet pour FORVIA est d'évaluer les outils dans lesquels l'entreprise investit afin d'assurer les meilleures conditions pour sa transformation digitale. Parmi ces outils figurent les grands modèles de langage (*LLMs*) proposés par OpenAI qui sont les modèles **GPTs** (Generative Pretrained Transformers) dans leurs dernières versions : **GPT-3.5**, **GPT-4**, **GPT-4-32k**, **GPT-4-Vision** et **GPT-4-Vision-Turbo**. Ces modèles sont relativement coûteux, par conséquent les résultats des mesures des KPIs recherchées par l'entreprise, ont été pris en compte par le groupe commercial pour décider du renouvellement ou non de leurs achats des APIs d'OpenAI.

# Chapitre 1

## Introduction au TAL et à la Transformation Digitale

Ce chapitre présente la naissance de FORVIA, notre organisme d'accueil, ainsi que le département de transformation digitale (FDT). Ensuite, nous allons présenter la problématique, ainsi que l'étude de l'application existante (*Power Apps*). La seconde partie de ce chapitre, présente une exploration théorique de l'évolution des outils du Traitement Automatique du Langage Naturel (TAL).

### 1.1 Organisme d'accueil

FORVIA représente un groupe d'excellence dans le domaine technologique automobile, au cœur d'une mobilité plus intelligente et écologique. Née de la fusion entre FAURECIA et HELLA, cette collaboration rassemble une expertise multidisciplinaire, couvrant l'électronique, la mobilité durable, l'éclairage, les intérieurs, les sièges, et les solutions de cycle de vie. FAURECIA, né de la fusion entre Bertrand Faure et ECIA en 1999, était un leader français dans la fabrication de sièges, d'intérieurs et de systèmes d'échappement. Jusqu'en 2021, le Groupe PSA, détenteur des marques Peugeot, Citroën, DS, Opel et Vauxhall, était l'actionnaire majoritaire de FAURECIA. Lorsque le Groupe PSA fusionna avec Fiat Chrysler pour former Stellantis, FAURECIA devint indépendant et acquit un intérêt majoritaire dans HELLA. En 2022, FAURECIA décida de fusionner avec HELLA, équipementier automobile allemand, en raison de leurs spécialisations complémentaires. FAURECIA était leader dans les intérieurs et les systèmes d'échappement, tandis que HELLA se spécialisait dans l'éclairage et l'électronique. Cette fusion a créé FORVIA, un groupe plus diversifié et compétitif. FAURECIA et HELLA partageaient une vision commune de l'industrie automobile, centrée sur la connectivité, l'électrification et la conduite autonome.

#### 1.1.1 Adaptation de l'industrie 5.0

Avant leur fusion en 2022 pour créer FORVIA, HELLA et FAURECIA étaient déjà deux acteurs majeurs de l'industrie 4.0, ayant chacun développé des initiatives et des solutions innovantes pour adapter leurs opérations aux technologies numériques. En 2016, HELLA a lancé un programme de transformation digitale appelé "HELLA 4.0", visant à digitaliser l'ensemble de la chaîne de valeur, de la conception des produits à la production et à la distribution. Quant à FAURECIA elle a initié en 2017, un plan similaire baptisé "FAURECIA 4.0", dans le but d'améliorer la performance et l'agilité du groupe en tirant parti des technologies numériques. Après cette fusion, FORVIA a com-

mencé à travailler sur l'industrie 5.0 marquée par l'intégration de l'intelligence artificielle dans divers processus, favorisant ainsi la Transformation Digitale de FORVIA.

### 1.1.2 Groupe de Transformation Digitale de FORVIA (FDT)

La mission de FORVIA est de concevoir des technologies pour une mobilité sûre, durable, avancée et personnalisée. Avec une équipe de 112 000 employés, comprenant plus de 35 000 ingénieurs répartis dans plus de 40 pays, FORVIA figure parmi les dix principaux fournisseurs mondiaux de l'industrie automobile. Organisée en 6 Business Groups, chargés du développement et de la commercialisation de 24 lignes de produits et services, FORVIA bénéficie du soutien de 5 groupes fonctionnels ,qui fournissent des ressources et une expertise essentielles aux Business Groups pour les aider à atteindre leurs objectifs. Parmi ces groupes fonctionnels, on retrouve le *FORVIA Digital Transformation Group (FDT)* créé en 2017, dont la mission est de perfectionner les processus métier grâce aux technologies numériques, **principalement l'intelligence artificielle**. Son objectif est d'accroître l'efficacité et de renforcer la compétitivité de FORVIA, positionnant ainsi l'entreprise en tant que leader de l'industrie.

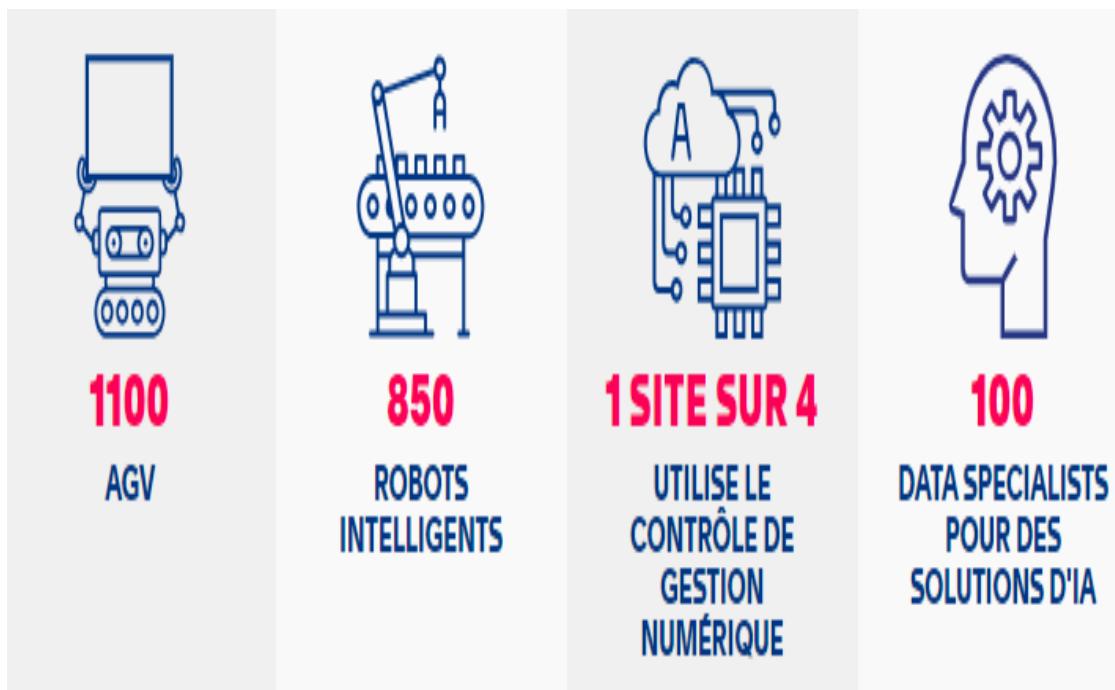


FIGURE 1.1 – Investissements adaptés par FORVIA pour sa transformation digitale

Faurecia a entamé la transformation digitale de sa production en 2017, avec l'objectif d'améliorer la qualité industrielle, l'efficacité et la compétitivité des coûts grâce à une chaîne de production et d'approvisionnement réactive. Cette stratégie industrielle a mené à l'adoption d'outils d'automatisation intelligents et de nouvelles solutions numériques de contrôle de gestion - transformant l'assemblage et la manipulation des produits, ainsi que le suivi de la performance opérationnelle. Comme indiqué dans la Figure 1.1. À la fin de 2021, plus de 2 500 robots intelligents et véhicules à guidage automatique (AGV) étaient utilisés dans nos usines. Les systèmes d'analyse axés sur les données visent à accroître encore l'agilité de la gestion des usines, à améliorer la maintenance prédictive et à stimuler l'efficacité des processus industriels. Plus de 15 projets de cas d'utilisation axés sur les données ont été lancés en 2021.

Le **FORVIA Digital Transformation Group (FDT)** repose sur deux piliers essentiels : le **Data Office** et le **Data Program**. Le **Data Office** centralise et organise les données de FORVIA, collectées à partir des systèmes de traçabilité. Il est géré par des ingénieurs data qui rendent les données exploitables par le **Data Program**, qui composé d'experts scientifiques, techniques et de consultants externes. Le **Data Program** exploite ces données pour créer de la valeur commerciale, notamment grâce aux avancées en intelligence artificielle.

Ces processus sont régis par le **Data Governance Network**, une communauté de 70 membres travaillant à garantir la cohérence, la qualité, l'accessibilité et la sécurité des données, tout en les alignant sur les besoins métier. La gouvernance de la donnée est cruciale pour surmonter les défis liés à la centralisation et à l'exploitation des données pour générer de la valeur. Parmi ces défis figurent la mauvaise capture des données, la complexité de la gestion des données de base, l'absence d'identification des sources de données, le manque de discipline dans la gestion des données, ainsi que les problèmes liés à la multiplicité des rapports et des formules des indicateurs clefs de performances KPIs. Le **Data Program** collabore étroitement avec le groupe d'intelligence artificielle (*IA*) pour intégrer les dernières avancées technologiques dans les activités de FORVIA. Cette collaboration se manifeste par le développement d'applications de vision artificielle pour les sites de production, l'amélioration des produits intégrés dans les véhicules et l'analyse des solutions des fournisseurs externes en utilisant une approche intitulée *make vs buy*.

Les piliers d'intelligence artificielle sur lesquels travaillent ces deux groupes peuvent être résumé dans la Figure 1.2.

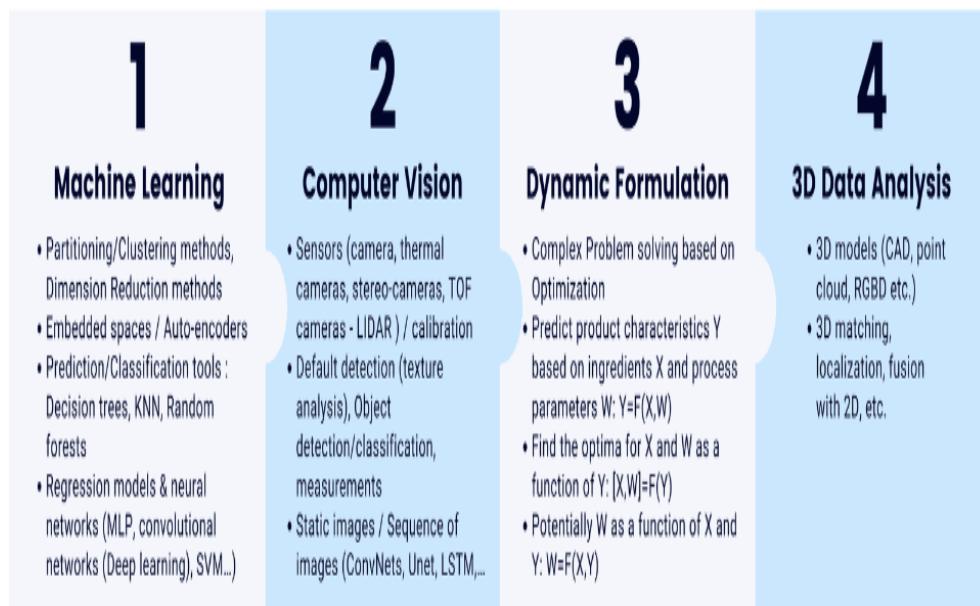


FIGURE 1.2 – Intelligence artificielle sur Foundry

### 1.1.3 Plateforme Palantir

FORVIA a fait le choix de la plateforme Foundry de Palantir pour mener à bien sa transformation digitale. Palantir, une entreprise américaine fondée en 2003 par Peter Thiel et Alex Karp, est basée à Denver, dans le Colorado. Elle développe et commercialise deux logiciels spécialisés dans la fusion,

la visualisation et le traitement opérationnel des données : Palantir Gotham et Palantir Foundry.

Gotham, initialement conçu pour les agences gouvernementales et les services de renseignement, permet aux analystes de données de relier et d'analyser des informations disparates provenant de multiples sources. Parmi ses clients figurent la communauté du renseignement des États-Unis, ainsi que les services de police du NYPD et du LAPD, et ceux de plusieurs pays européens, dont la France.

FORVIA a commencé à utiliser la plateforme Foundry en 2021, dans le cadre d'un partenariat annoncé en février de la même année. Cette collaboration vise à accélérer la transformation digitale de FORVIA et son objectif de devenir neutre en CO<sub>2</sub>. Bien que peu d'entreprises aient opté pour Foundry en raison de son coût élevé, quelques-unes, telles qu'Airbus, IBM et Pfizer, l'ont choisie pour leurs propres transformations digitales.

Foundry est une plateforme d'exploitation conçue pour aider les organisations à utiliser les données pour résoudre des problèmes concrets. Elle propose une collection d'outils puissants pour la création d'applications exploitant les primitives de sécurité, de lignage, de données et de calcul de Foundry.

Les applications de Foundry, construites à partir de l'ontologie, permettent l'accumulation des connaissances à mesure que les flux de travail opérationnels se développent au sein de l'entreprise. Parmi ces applications, on trouve Contour, Quiver, Code Repos<sup>1</sup>, Fusion, Pipeline Builder et Data Lineage, qui répondent à différents besoins analytiques et opérationnels.

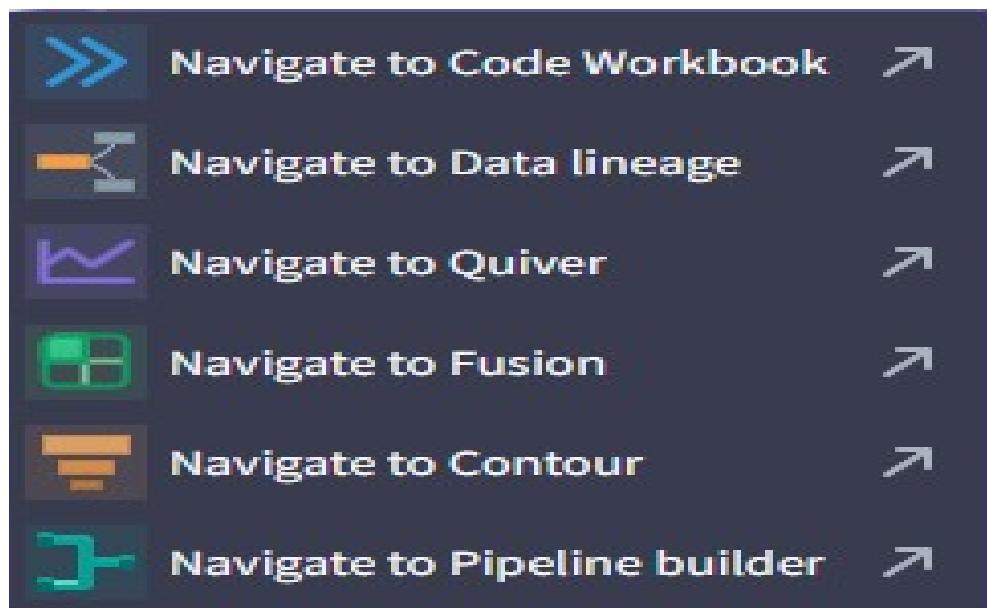


FIGURE 1.3 – Quelques applications de Foundry

La Figure 1.3 représente les applications les plus importantes présentes dans Foundry.

#### 1.1.4 Méthodologie de travail

De nos jours, les sociétés adoptent une variété de méthodologies pour gérer leurs projets. Parmi les plus répandues, on retrouve l'approche Agile, qui favorise la livraison incrémentale des projets,

1. Une application de programmation sous le langage PySpark [Drabas and Lee, 2017].

Waterfall, et la méthodologie Hybride.

L'entreprise FORVIA favorise l'utilisation de la méthodologie agile : SCRUM dans la réalisation de ce projet, c'est d'ailleurs la méthodologie adaptée pour la majorité de ses projets. SCRUM est une méthode de gestion de projet itérative et incrémentale, largement utilisée dans le développement logiciel.

Elle se caractérise par des cycles de développement courts appelés *sprints*, durant lesquels des solutions sont développées. Les résultats obtenus sont livrés et présentés lors de réunions journalières (*Daily meetings*) ou hebdomadaires (*Weekly meetings*), où l'on consacre plus de temps à résoudre les problèmes éventuels et à tester les résultats. SCRUM favorise la collaboration au sein de l'équipe, la transparence et l'adaptation aux changements tout au long du projet.

## 1.2 Problématique

Après l'inauguration de l'industrie 5.0, les usines intelligentes de FORVIA produisent d'énormes quantités de données, réparties en deux principales catégories : les données structurées et les données non structurées.

Les données structurées comprennent des informations sur la production, telles que la quantité produite, les rebuts, le temps de cycle et l'efficacité, ainsi que des données sur la maintenance, comme les interventions, les pannes, le temps d'arrêt, les stocks de pièces détachées, la température et la pression. La chaîne d'approvisionnement est également couverte, avec des données sur les délais de livraison, la qualité des produits, les prix et les négociations. Les données commerciales, telles que le chiffre d'affaires, les marges, les parts de marché et les clients, ainsi que les données financières, y compris le bilan, le compte de résultat et les flux de trésorerie, sont également recueillies. Malgré leur diversité, ces données peuvent être facilement interrogées grâce aux différentes applications de Foundry.

Cependant, un autre type de données, les données non structurées, reste largement inexploité. Il s'agit principalement de **contrats** et de **documents** constitutionnels contenant les procédures de base de FORVIA ou *Forvia Core Procedures* au format PDF. Ces documents, qui contiennent des informations textuelles et aussi sous forme d'images, sont souvent volumineux, disparates et difficiles à traiter. Leur exploitation nécessite l'utilisation de l'intelligence artificielle, en particulier dans le domaine du traitement du langage naturel (TAL) ou (*Natural Language Processing (NLP)*).

Les procédures de base de FORVIA (*Forvia Core Procedures*), qui comprennent des manuels détaillant les processus essentiels définissant le fonctionnement de l'entreprise, représentent une ressource largement sous-exploitée jusqu'à présent par FORVIA. Ces documents couvrent pratiquement tous les départements comme indiqués dans la Figure 1.4. Ils vont de la finance avec ses rapports et états financiers permettant d'identifier les tendances et les anomalies, à l'achat grâce aux contrats avec les fournisseurs et aux données de performance, facilitant ainsi la sélection et la gestion des fournisseurs. De même, ils incluent les données de vente et les commentaires des clients dans le département des ventes, permettant d'identifier les opportunités de croissance et d'amélioration, ainsi que des documents juridiques réduisant le besoin de révisions manuelles et garantissant la conformité.

Chaque type de document nécessite des experts dans le domaine correspondant pour en tirer des

informations (*insights*) pertinentes.

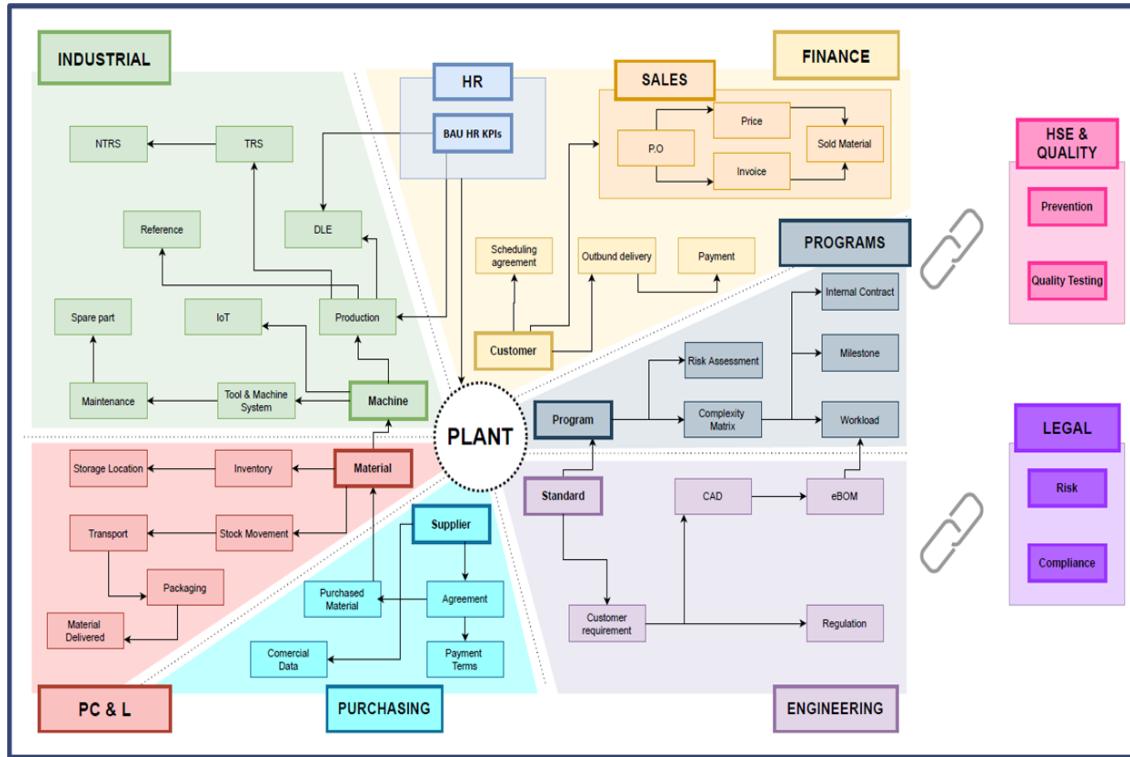


FIGURE 1.4 – Flux de données dans une plante [Faurecia, 2024]

Quelle approche optimale permettrait d'adapter les grands modèles de langage génériques (*Large Language Models ou LLMs*) déjà intégrés dans Foundry à savoir les modèles **GPT** (*Generative Pre-trained Transformers*), aux documents de FORVIA, afin de pouvoir les interroger automatiquement, sans nécessiter l'intervention d'experts ?

Un tel investissement de souscription à des interfaces de programmation d'applications *APIs* payantes pour intégrer les *LLMs* conçus par OpenAI dans Panatir Foundry et les utiliser par la suite, en vaut-il vraiment la peine ?

En d'autres termes, ces *LLMs* à accès restreint offrent-ils des réponses pertinentes lorsqu'il s'agit d'interroger les données de FORVIA ? Quels indicateurs de performances clés ou (KPIs) choisir pour pouvoir évaluer, les performances des différents *LLMs* en question et leurs capacités à atteindre les finalités métiers souhaitées par l'entreprise ?

Quels indicateurs de performances clés ou (KPIs) choisir pour pouvoir évaluer et par conséquent classer, la performance des différents *LLMs* et leurs capacités à atteindre les finalités métiers souhaitées par l'entreprise ?

### 1.3 Etude de l'application développée sur *Power Apps*

Dans cette section, nous allons explorer une application considérée comme "non intelligente" développée sous *Power Apps* et conçue pour le traitement des documents *FCP*.

Cette application, s'avère un peu rudimentaire, vu qu'elle n'offre qu'une seule solution : permettre aux utilisateurs d'appliquer des filtres en se basant sur les titres des documents recherchés. Cependant, elle présente des limitations importantes qui entravent son efficacité et sa flexibilité.

FORVIA a déjà développé une application *Power Apps* [Microsoft, 2024b] pour gérer les fichiers des procédures de base de FORVIA (*Forvia Core Procedures*) et en faciliter l'accès pour les employés internes. Cette application vise également à simplifier la recherche et la compréhension de ces documents grâce à une interface utilisateur intuitive.

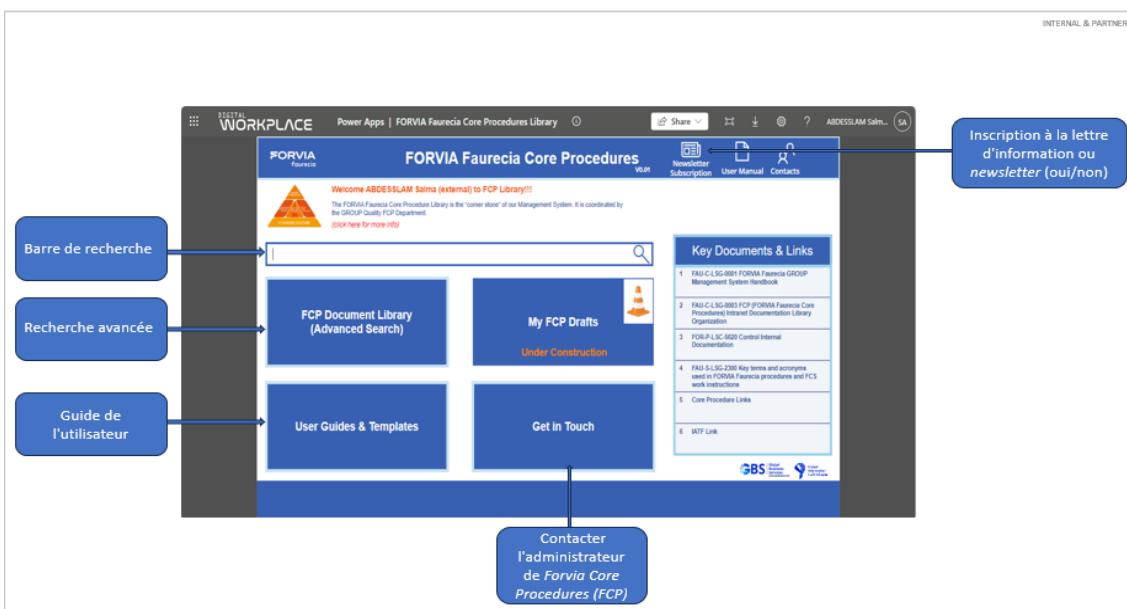


FIGURE 1.5 – Page d’Accueil de Power Apps (*Forvia Core Procedures*)

La page d'accueil de Power Apps pour la Gestion des Fichiers de Procédures de Base de FORVIA (*Forvia Core Procedures*) est présentée dans la Figure 1.5 comprend plusieurs fonctionnalités clés :

- Une fonction de recherche globale permettant de rechercher par nom de document ou par ID de référence.
- Une fonction de recherche avancée qui mène vers la page représentée dans la Figure 1.6, où l'utilisateur a la possibilité de rechercher selon des critères plus détaillés tels que l'entité, le processus clé, etc.
- Des guides utilisateur et des modèles : des modèles *FCP*, un guide du propriétaire du document *FCP* et un guide de l'utilisateur du document *FCP*.
- Un accès rapide à certains documents clés.
- Un lien pour contacter l'administrateur du département *FCP* via le bouton "nous contacter".
- Un lien vers la bibliothèque de brouillons des documents *FCP* (bibliothèque contenant les documents en cours de construction ou de validation).

Certes cette application permet d'appréhender les documents des procédures de base de FORVIA, mais cela reste une approche superficielle, elle ne permet pas à l'utilisateur de poser des questions directement en lien avec le contenu de ces documents sans avoir besoin d'indiquer (qu'il s'agisse de texte ou d'images) ou de résumer le contenu de ce fichier. En outre, un utilisateur qui n'a pas une

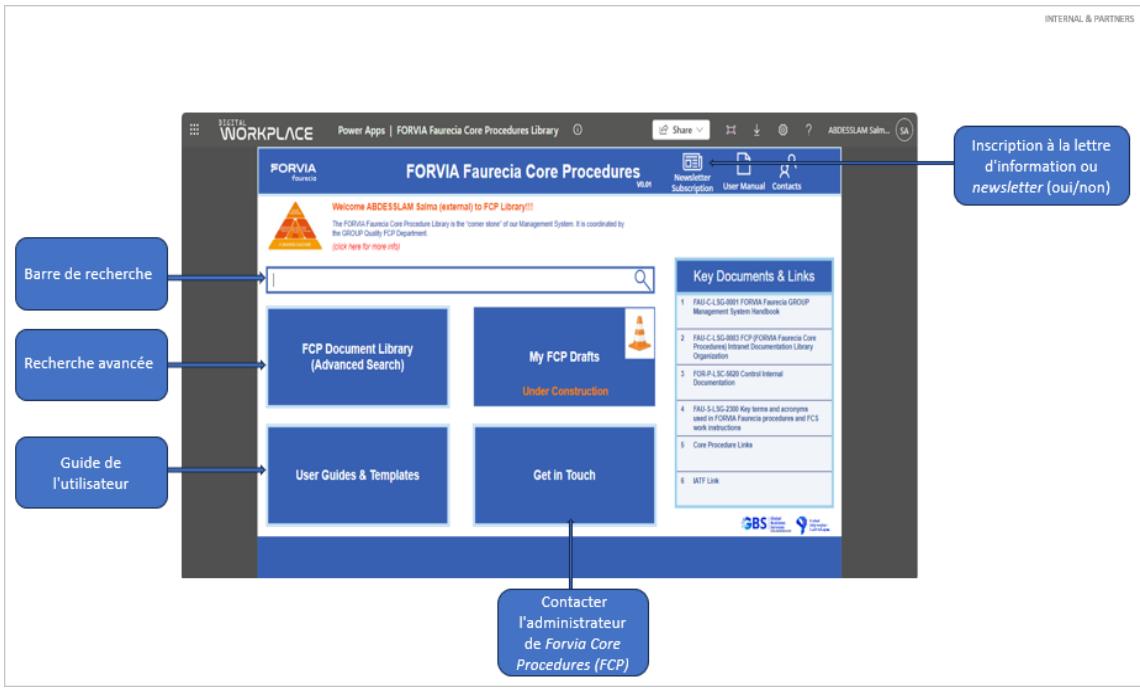


FIGURE 1.6 – Fonction de recherche plus détaillée

idée préalable de l'endroit où trouver la réponse à sa question, va perdre beaucoup de temps. En effet, une recherche "classique" pour cibler les documents dans lesquels on va mener sa recherche, nécessite le parcours d'un grand nombre de pages que contient chaque document.

Cette application n'est pas considérée comme "intelligente", contrairement à celle développée dans le cadre du projet GenAI Tailor. Ce dernier projet intègre plusieurs *LLMs* qui, dans notre cas, se basent sur **GPT-4** (Generative Pre-trained Transformer 4) [Alto, 2023].

Le modèle est utilisé dans un module ou *widget* de plateforme d'intelligence artificielle (*AIP Widget*) permettant de réaliser diverses tâches (résumé, reconnaissance d'entités nommées, etc.) et de répondre à tout type de questions, non seulement sur le contenu textuel des documents, mais aussi sur les images. Cela est rendu possible grâce à un processus préalable de prétraitement et de description contextuelle des images avec **GPT-4-Vision** [Achiam et al., 2023], qui est une version avancée du modèle **GPT-4**, qui intègre des capacités de traitement d'images, en plus de ses fonctions traditionnelles de traitement du langage naturel (*NLP*).

La recherche et l'identification des documents les plus pertinents pour trouver les réponses sont automatisées grâce à une fonction *TypeScript* [Bierman et al., 2014]. Cette fonction adapte la logique du plus proche voisin (*Nearest neighbor*) pour identifier les documents cibles avant de rechercher l'information pertinente et de produire une réponse logique et précise à la question ou la requête de l'utilisateur. Cette fonction de recherche sémantique sera abordée avec plus de détails dans les prochains chapitres.

La Figure 1.7 montre l'apport technologique en terme d'intégration de l'intelligence artificielle générative et d'utilisation des *LLMs*.

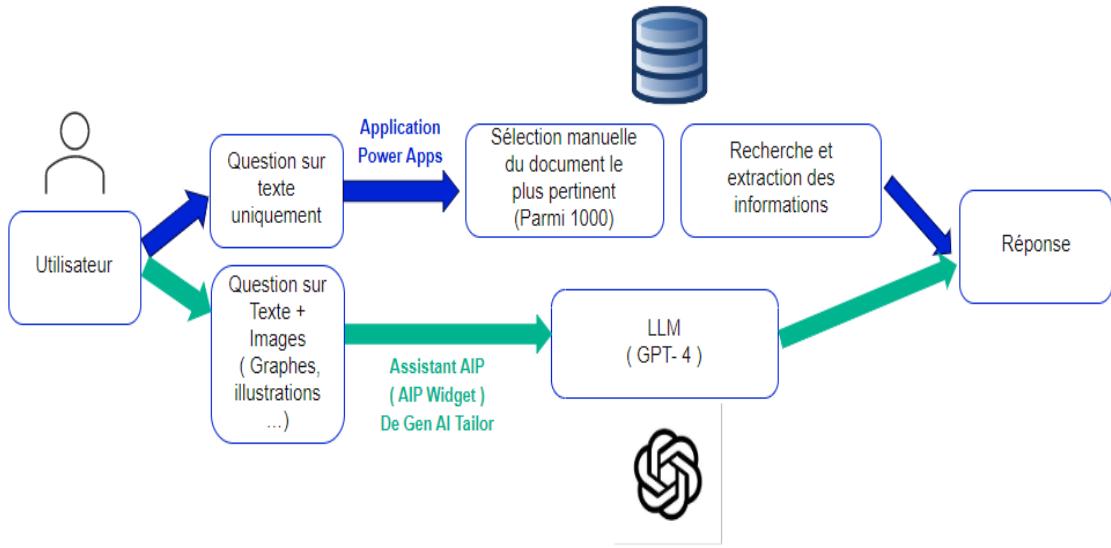


FIGURE 1.7 – Différences Technologiques entre l'existant et la Nouvelle Application

## 1.4 Traitement automatique de langage naturel (TAL)

Le traitement automatique du langage naturel (TAL), également connu sous le nom de *Natural Language Processing* (NLP), est né au milieu du XXe siècle et s'est imposé comme un domaine d'étude fascinant à la croisée du langage et de l'informatique. Ce domaine englobe des champs de recherche variés, les premières tâches recherchées par les chercheurs étaient la traduction automatique, la prédiction du mot suivant ou masqué, la classification de texte dont les réseaux neuronaux profonds (que ce soit *RNNs* ou *LSTMs*) étaient capables de réaliser. Puis avec la nouvelle ère des grands modèles de langage (*LLMs*) est apparu l'analyse sémantique, la génération de texte, et bien d'autres encore. Les tâches effectuées par les modèles de langages sont en constante évolution.

### 1.4.1 Histoire du TAL : L'aube des machines communicantes

Les premiers travaux en traitement automatique du langage naturel (TAL) remontent aux années 1950, principalement aux États-Unis, dans le contexte de la **guerre froide**. L'intérêt pour la **traduction automatique** était particulièrement vif, car il était considéré comme un moyen crucial de recueillir et de comprendre les renseignements étrangers.

En 1954, la machine IBM 701 indiquée dans la Figure 1.8, un système rudimentaire capable de traduire un nombre limité de phrases russes spécifiques en anglais vit le jour. Limitée à un vocabulaire de 250 mots et à 6 règles grammaticales, elle ne pouvait traduire qu'un nombre restreint de phrases russes spécifiques.

Prenons l'exemple de la règle suivante : "Les verbes doivent être conjugués selon le temps de l'action qu'ils expriment (passé, présent, futur)". Si la machine pouvait appliquer cette règle à une phrase simple comme "I studied French yesterday" (J'ai étudié le français hier), elle ne pouvait pas la traduire dans des contextes plus complexes ou avec des structures de

phrases plus élaborées.

Néanmoins, cette machine, fruit de l'expérience Georgetown-IBM, fut saluée comme une avancée majeure. Cette percée suscita un vif intérêt pour le traitement automatique du langage naturel, ouvrant la voie à de nombreuses recherches et développements ultérieurs. Cette période coïncide avec les succès soviétiques dans *la course à l'espace*, attisant l'intérêt des militaires américains pour *la traduction des publications techniques soviétiques*, sans pour autant investir dans la formation linguistique de leurs ingénieurs.

Le communiqué de presse publié le 7 janvier 1954 par IBM pour annoncer l'événement disait : « *Une jeune femme qui ne comprend pas un mot de la langue des Soviets a poinçonné des messages en russe sur des cartes perforées IBM. La machine IBM 701, surnommée « le cerveau » capable de traduire du texte, a livré ses traductions en anglais sur une imprimante automatique à la vitesse vertigineuse de deux lignes et demie par seconde.*



FIGURE 1.8 – "Le cerveau" ou IBM 701 traduisant automatiquement du texte russe vers l'anglais

Les années qui suivirent, furent marquées par un optimisme débordant et un foisonnement de recherches ,soutenus par des investissements massifs. Les travaux de l'époque se concentraient sur la création de *dictionnaires électroniques* contenant des règles grammaticales, expressions idiomatiques et des correspondances lexicales. Ces dictionnaires étaient parfois complétés par un réarrangement syntaxique manuel, un processus long, laborieux et finalement peu efficace. En outre, ces dictionnaires, utilisés comme base pour les systèmes de traduction automatique, présentaient des limites :

- **Rigidité** : Les règles étaient souvent trop rigides et ne pouvaient s'adapter aux nuances et aux ambiguïtés du langage naturel.
- **Manque de compréhension du contexte** : Ces systèmes ne prenaient pas en compte le contexte de la conversation ou de la phrase, ce qui limitait leur capacité à produire un langage naturel fluide et cohérent.

- **Difficulté d'acquisition de connaissances linguistiques** : Développer et maintenir des ensembles de règles complexes et des dictionnaires exhaustifs était un processus long et laborieux.

L'illustration la plus frappante de ces limites, évoquées précédemment, est la célèbre anecdote de la phrase :

"The spirit is willing but the flesh is weak"  
("L'esprit est fort mais la chair est faible").

Traduite en russe puis retraduite en anglais, elle devient "The vodka is strong but the meat is rotten" ("La vodka est forte mais la viande est pourrie"). Cet exemple met en lumière la **nécessité de connaissances contextuelles** et encyclopédiques approfondies pour une traduction adéquate.

Un autre défi majeur, apparu moins de dix ans après les débuts de la recherche, est la représentation et l'exploitation des connaissances. Prenons l'exemple du mot "pen" dans les phrases "(a) the box is in the pen" et "(b) the pen is in the box". La signification du mot dépend du contexte, ce qui pose des problèmes importants pour les systèmes de traduction automatique. Face à ces obstacles, Bar-Hillel déclara que le problème de la traduction automatique était *probablement insoluble*.

Un groupe d'experts, *l'Automatic Language Processing Advisory Council* (ALPAC), rédigea en 1966 un rapport accablant, concluant que la traduction automatique était **deux fois plus onéreuse** que la traduction humaine et produisait des résultats nettement inférieurs. Cette réalité économique entraîna un tarissement des financements publics dans ce domaine, tant aux États-Unis qu'en Europe. Dans ce cadre, le rapport de l'ALPAC constate qu'en dix ans de recherches, les objectifs n'ont pas été atteints. Cette prise de conscience de l'extrême complexité des langues a considérablement réduit l'ambition des travaux de recherche. Certains chercheurs ont même prédit la fin du domaine.

#### 1.4.2 Apparition des Ontologies Conceptuelles - Approche de 1950-1990

L'approche linguistique du Traitement Automatique du Langage (TAL) a apporté des contributions majeures à la compréhension et à la génération du langage naturel par les machines. Parmi ces contributions, les ontologies conceptuelles jouent un rôle crucial.

Imaginons un dictionnaire pour les ordinateurs. Les *ontologies conceptuelles* dont un exemple est introduit dans la Figure 1.9 sont des structures organisées qui représentent le sens du langage naturel d'une manière compréhensible par les ordinateurs. Elles fonctionnent comme des dictionnaires spécialisés, mais au lieu de simplement définir des mots, elles décrivent des concepts et les relations entre eux. Ces ontologies permettent de modéliser des aspects complexes du langage naturel, tels que :

- **Les scripts** : Des séquences d'actions typiques dans des situations données, comme aller au restaurant ou prendre l'avion.
- **Les plans d'action** : Des ensembles d'étapes nécessaires pour atteindre un objectif, comme préparer un gâteau ou organiser un voyage.
- **Les structures narratives** : Les éléments clés d'une histoire, comme les personnages, les lieux et les événements.

Des exemples notables incluent MARGIE (Schank, 1975), SAM (Cullingford, 1978), PAM (Wilensky, 1978), TaleSpin (Meehan, 1976), SCRUPULE (Lehnert, 1977), Politics (Carbonell, 1979), et Plot Units (Lehnert, 1981).

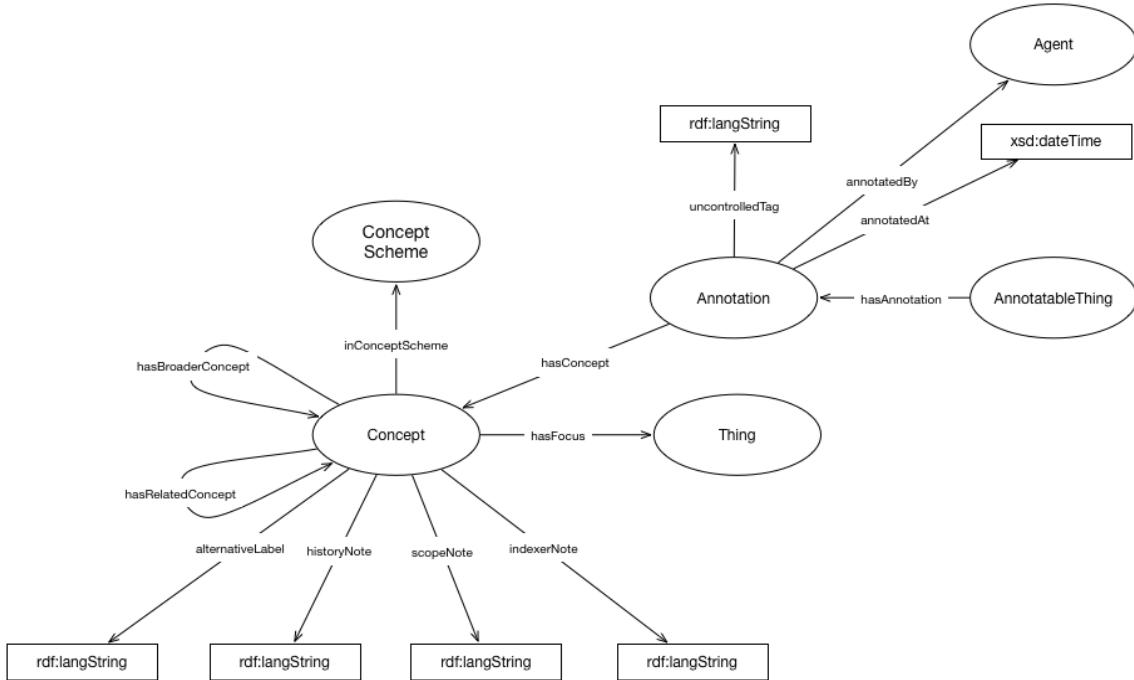


FIGURE 1.9 – Représentation schématique d'une Ontologie Conceptuelle

La Figure 1.9 montre une représentation simplifiée d'une ontologie conceptuelle. Les concepts sont représentés par des nœuds, et les relations entre les concepts sont représentées par des liens. Les flèches indiquent la direction de la relation.

Par exemple, dans cette figure, le concept "Animal" est lié aux concepts "Mammifère", "Oiseau" et "Poisson" par une relation de type "est-un". Cela signifie que tous les mammifères, oiseaux et poissons sont des types d'animaux.

De même, le concept "Mammifère" est lié aux concepts "Chat", "Chien" et "Baleine" par une relation de type "est-un". Cela signifie que tous les chats, chiens et baleines sont des types de mammifères.

Les « ontologies conceptuelles » peuvent être beaucoup plus complexes que cette illustration simplifiée. Elles peuvent inclure des centaines, voire des milliers de concepts et de relations. Elles peuvent également être organisées en hiérarchies, avec des concepts plus généraux au sommet et des concepts plus spécifiques en bas.

Ces « ontologies conceptuelles » étaient fondamentales pour représenter le sens des phrases en langage naturel et ont été adaptées par divers **modèles de représentation sémantique** pour interpréter et générer du langage naturel.

Bien que tous ces modèles utilisent des symboles pour représenter les concepts et les relations du langage naturel, chacun offre une représentation différente pour une même phrase.

Par exemple, la phrase "Le chat est assis sur le tapis" est souvent utilisée pour illustrer les différences entre les différents modèles de représentations sémantiques.

Modèle de représentation sémantique	MARGIE (Schank, 1975)	SAM (Cullingford, 1978)	PAM (Wilensky, 1978)
Structure de la représentation	(SIT (AGENT CHAT) (OBJET TAPIS) (LOCATION SUR))	Chat(x) & tapis(y) & sur (x, y) & être (x, assis)	Chat(agent) --> être(état) --> assis(position) --> tapis(objet) --> sur(location)
Éléments clés	Agent (CHAT), Objet (TAPIS), Lieu (SUR), Action (SIT)	Entités (x, y), Prédicats (chat, tapis, sur, être, assis), Connecteurs (&)	Objet (Chat), Attribut (être(état)), Valeur (assis(position)), Objet (tapis), Attribut (sur(location))
Avantages	Capture bien les relations sémantiques, facile à comprendre	Représentation précise et formelle	Structure hiérarchique claire, permet le raisonnement
Inconvénients	Moins expressif que d'autres formalismes	Peut être difficile à écrire et à interpréter	Peut ne pas capturer toutes les nuances du sens

FIGURE 1.10 – Comparaison entre les différents modèles de représentation sémantique

Les « ontologies conceptuelles » ont permis de développer **des formalismes pour la représentation du sens et du savoir**, améliorant ainsi la compréhension du contexte et des intentions des utilisateurs par les systèmes de traitement automatique du langage (TAL). Elles ont également jeté les bases pour le développement d'applications plus sophistiquées en TAL, telles que les systèmes de dialogue, les générateurs de texte, et les outils d'analyse de contenu, ainsi que pour les recherches ultérieures dans le domaine de la représentation du savoir et de l'intelligence artificielle. Cependant, elles présentent certaines limitations :

- **Hétérogénéité des ontologies** : Différentes communautés et organisations peuvent développer des ontologies pour le même domaine, ce qui entraîne des incohérences terminologiques et sémantiques.
- **Complexité des ontologies** : Les ontologies détaillées et complètes peuvent être difficiles à créer et à maintenir, limitant leur adoption à grande échelle.
- **Évolution des ontologies** : Le monde réel est en constante évolution, et les ontologies doivent être mises à jour en conséquence, ce qui peut être un processus laborieux et coûteux.

### 1.4.3 Approche statistique (années 1990-2010)

L'avènement de grands corpus de texte et l'essor de l'apprentissage automatique ont conduit à l'adoption d'une approche statistique en TAL. Dans cette approche, les systèmes apprennent à partir de données d'observation, **en identifiant des patterns statistiques dans les corpus**. Cette approche a permis d'obtenir des résultats bien meilleurs que l'approche linguistique, en particulier pour des tâches comme la traduction automatique et la reconnaissance vocale.

## 1.5 Modèles de Markov

Dans l'analyse de Markov, introduite pour la première fois par Andrey Markov au début du XXe siècle, nous supposons **une condition initiale ou un état de départ pour le système**.

Par exemple, un état initial pourrait être une paire de fabricants rivaux de pièces automobiles : l'une des entreprises concurrentes de FORVIA et FORVIA, **l'une avec 40 % de part de marché et l'autre avec 60 %**. Au fil du temps, les parts de marché pourraient changer pour être de **45 % et 55 %**.

Pour prédire un tel résultat, nous devons savoir quelle est la probabilité que le système passe de l'état initial à l'état final. Ces probabilités peuvent être établies grâce à l'expertise humaine ou par le biais d'algorithmes d'apprentissage automatique.

Nous pouvons prendre toutes les probabilités dans un problème spécifique et les placer dans un tableau ou **une matrice des probabilités de transition**, indiquant à quelle fréquence un changement de système se produira au fil du temps.

C'est ainsi que fonctionne le processus de Markov, nous permettant de faire des prédictions concernant un état ou une condition au fil du temps.

Les modèles de Markov, sont des outils puissants pour prédire des événements futurs en se basant sur des événements passés. Ils sont largement utilisés dans divers domaines, tels que l'économie, la finance, la météorologie et, le Traitement Automatique du Langage (TAL).

### 1.5.1 Chaînes de Markov

Dans le contexte du TAL, on utilise plus précisément les chaînes de Markov, qui sont des modèles de Markov discrets où l'état actuel du système détermine de manière probabiliste l'état suivant.

Ce sont des modèles stochastiques continuellement changeants utilisés pour prédire ou estimer le résultat d'événements linguistiques spécifiques en ne se basant que sur l'état précédent et les observations actuelles.

Dans ce contexte, un "*état ou évènement linguistique*" fait référence à une unité ou une caractéristique linguistique spécifique sur laquelle se base le modèle de Markov pour effectuer ses prédictions.

Par exemple, dans le cas de **la génération de texte**, un état linguistique pourrait représenter **un mot individuel dans une phrase**. Dans le cas de **l'analyse grammaticale**, un état linguistique pourrait représenter une partie du discours telle que **nom, verbe, adjectif**, etc. Dans d'autres applications de TALN, les états linguistiques pourraient être définis différemment en fonction de la tâche spécifique à accomplir.

Supposons que nous voulions modéliser la probabilité de transition entre deux types de mots dans une phrase : **les verbes et les noms**. Nous pourrions utiliser une chaîne de Markov similaire pour représenter ces transitions.

La Figure 1.11 représente une chaîne de Markov où les États sont : Verbe et Nom ; les Probabilités de Transition sont La probabilité qu'un verbe soit suivi d'un autre verbe : 30%, la probabilité qu'un verbe soit suivi d'un nom : 70%, la probabilité qu'un nom soit suivi d'un autre nom : 80% et la probabilité qu'un nom soit suivi d'un verbe : 20%.

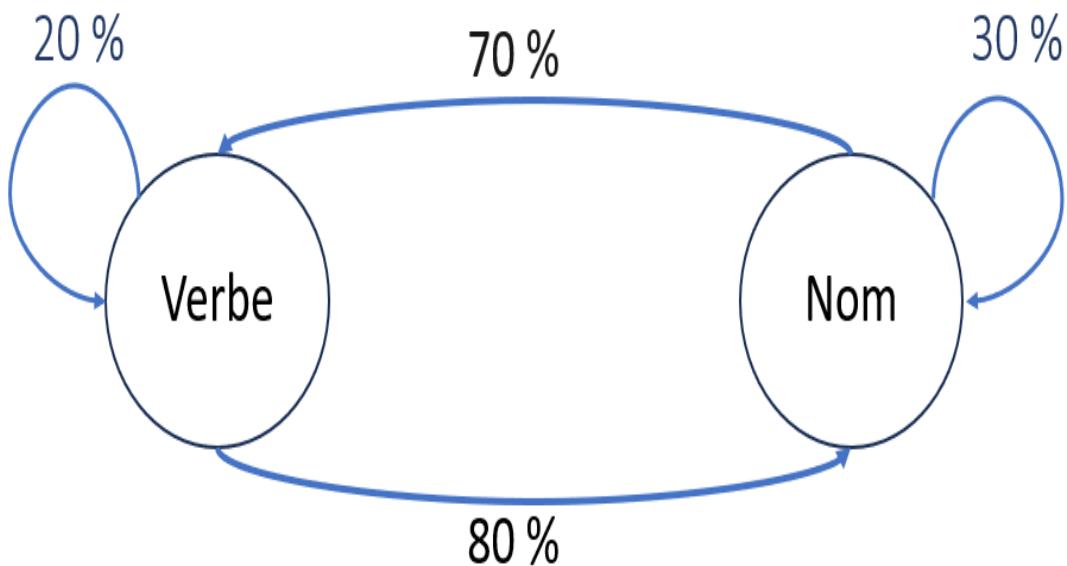


FIGURE 1.11 – Chaîne de Markov pour représenter les transitions entre un Verbe et un Nom

Cette chaîne de Markov pourrait être appliquée sur l'exemple de phrase suivant :

- Phrase : "Les enfants jouent." ;
- Types de mots : Nom ("Les enfants"), Verbe ("jouent").

Dans ce cas, si nous avions un verbe ("jouent"), la chaîne de Markov nous dit qu'il y a 70% de chances que le prochain mot soit un nom et 30% de chances que ce soit un autre verbe.

Parfois, les chaînes de Markov ne sont pas suffisantes car certaines informations importantes ne sont pas directement observables. Par exemple, dans de nombreux cas, les événements d'intérêt ne sont pas visibles et ne peuvent pas être directement observés : un lecteur humain ne distingue généralement pas les étiquettes de parties du discours lorsqu'il lit un texte, il infère les étiquettes « cachées » à partir de la séquence de mots. Dans ce cas, il est crucial d'avoir recours aux Modèles de Markov Cachés (*Hidden Markov Models : HMM*).

Les Modèles de Markov - qu'ils soient cachés ou pas - prennent en compte seulement l'évènement précédent. Cependant il existe un modèle beaucoup plus sophistiqué qui prend en considération les *n-1 évènements précédents* pour prédire l'évènement suivant. Cela permet une modélisation plus précise, en particulier pour les langues naturelles où les relations entre les mots peuvent s'étendre sur plusieurs mots consécutifs.

### 1.5.2 Modèles de n-grammes

Les modèles de n-grammes reposent sur l'hypothèse selon laquelle **la probabilité d'un mot dépend des  $n$  mots qui le précédent**. Il y a les unigrammes, bigrammes, trigrammes, etc., comme le montre la Figure 1.12.

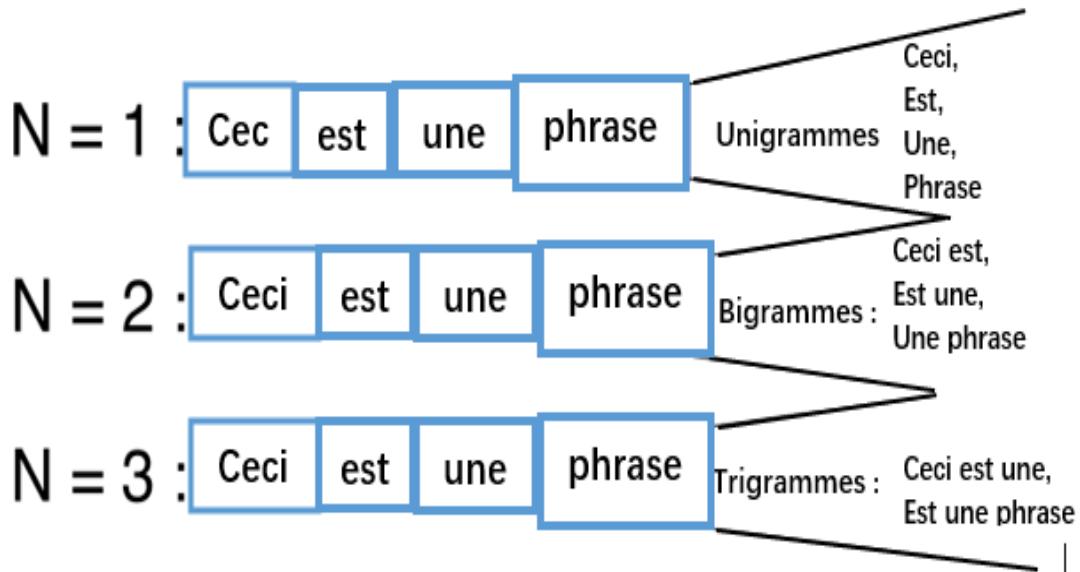


FIGURE 1.12 – Modèles de n-grammes

Pour identifier les patterns, ce type de modèles représentés dans la Figure 1.12 analysent la fréquence d'apparition des séquences de  $n$  mots dans un corpus de texte. En comptant ces occurrences, les modèles évaluent la probabilité d'observer ces séquences dans la suite du texte.

**Modèle unigramme :**  $P(w_n)$  ou  $P(w_n | \text{contexte})$  Dans **un modèle unigramme**, la probabilité de chaque mot est indépendante de son contexte. Il suppose que chaque mot apparaît indépendamment des mots environnants. Par exemple, la probabilité de rencontrer "pomme" est  $P(\text{"pomme"}) = 0.3$ .

**Modèle bigramme :**  $P(w_n | w_{n-1})$  Dans **un modèle bigramme**, la probabilité d'un mot dépend uniquement du mot précédent. Il calcule la probabilité d'un mot étant donné le mot immédiatement précédent.

Par exemple, supposons que le mot "pomme" suive "je" dans le corpus 10 fois, et que "je" apparaisse 20 fois. Alors  $P(\text{"pomme"} | \text{"je"}) = \frac{10}{20} = 0.5$ .

**Modèle trigramme :**  $P(w_n | w_{n-1}, w_{n-2})$  Dans **un modèle trigramme**, la probabilité d'un mot dépend des deux mots précédents. Il calcule la probabilité d'un mot étant donné les deux mots précédents dans la séquence.

Par exemple, si "pomme" suit "je" et "aime" dans le corpus 5 fois et que "j'aime" apparaît 10 fois. Alors  $P(\text{"pomme"} | \text{"je"}, \text{"aime"}) = \frac{5}{10} = 0.5$ .

Les inconvénients des modèles n-grammes incluent plusieurs aspects. Tout d'abord, il y a la nécessité de **stocker de grandes quantités** de données pour des ***n* plus grands**. Ensuite, ces modèles sont sensibles aux erreurs introduites par les co-occurrences rares. De plus, ils rencontrent **des difficultés à capturer les dépendances à long terme** entre les mots et les motifs linguistiques nuancés. Pour répondre à ces limites, nous nous tournons vers les réseaux de neurones.

## 1.6 Du Perceptron aux Grands Modèles de Langage

En s'inspirant du fonctionnement des neurones biologiques, *Warren McCulloch* et *Walter Pitts* posent les jalons des réseaux de neurones artificiels en 1943 avec le Perceptron. Ce modèle mathématique simple, considéré comme la brique élémentaire des réseaux de neurones, ouvre la voie à l'exploration de l'apprentissage automatique et de l'intelligence artificielle.

Cependant, le livre "**Perceptrons**", écrit par Frank Rosenblatt en 1962, met en exergue les limitations du Perceptron, notamment son incapacité à résoudre des problèmes non linéaires complexes comme la fonction XOR. Cette découverte freine temporairement l'enthousiasme pour les réseaux de neurones, plongeant le domaine dans une période d'incertitude.

Les **réseaux de neurones à propagation descendante**, composés de plusieurs perceptrons interconnectés, comme indiqué dans la Figure 1.13 émergent ensuite comme une solution plus puissante et flexible. Ces réseaux, dotés de l'aptitude à apprendre des relations non linéaires, jettent les bases de l'apprentissage automatique moderne.

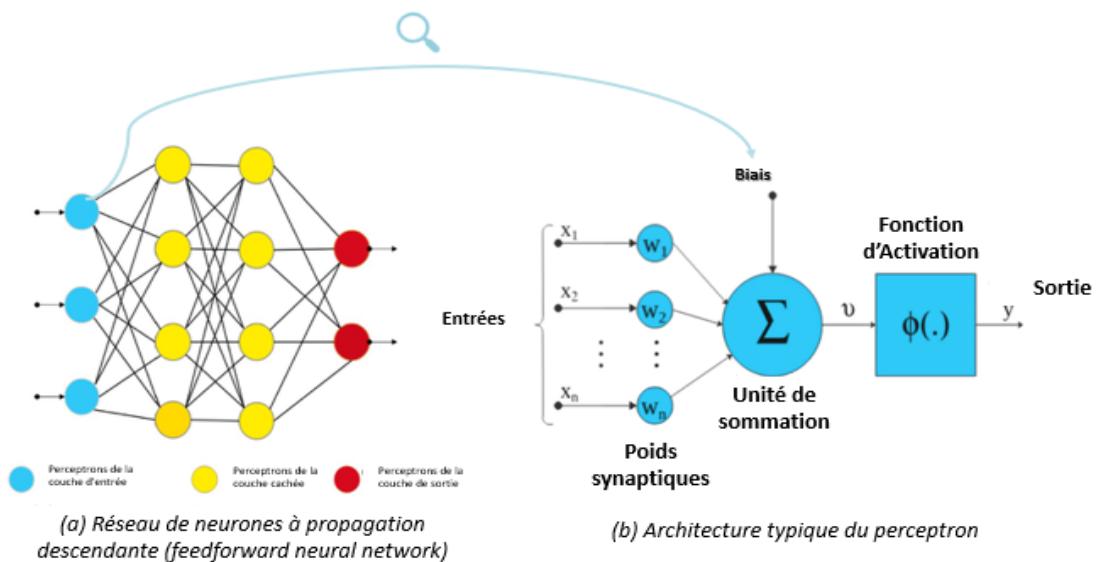


FIGURE 1.13 – Composition et architecture des réseaux de neurones à propagation descendante

En 1986, David E. Rumelhart, Geoffrey Hinton et Ronald J. Williams ont joué un rôle crucial dans la renaissance des réseaux de neurones artificiels en introduisant l'algorithme de rétropropagation (*backpropagation*). Cette avancée majeure a permis de surmonter les limitations des perceptrons simples et d'ouvrir la voie à l'entraînement efficace des réseaux de neurones multicouches, aujourd'hui connus sous le nom de réseaux de neurones profonds.

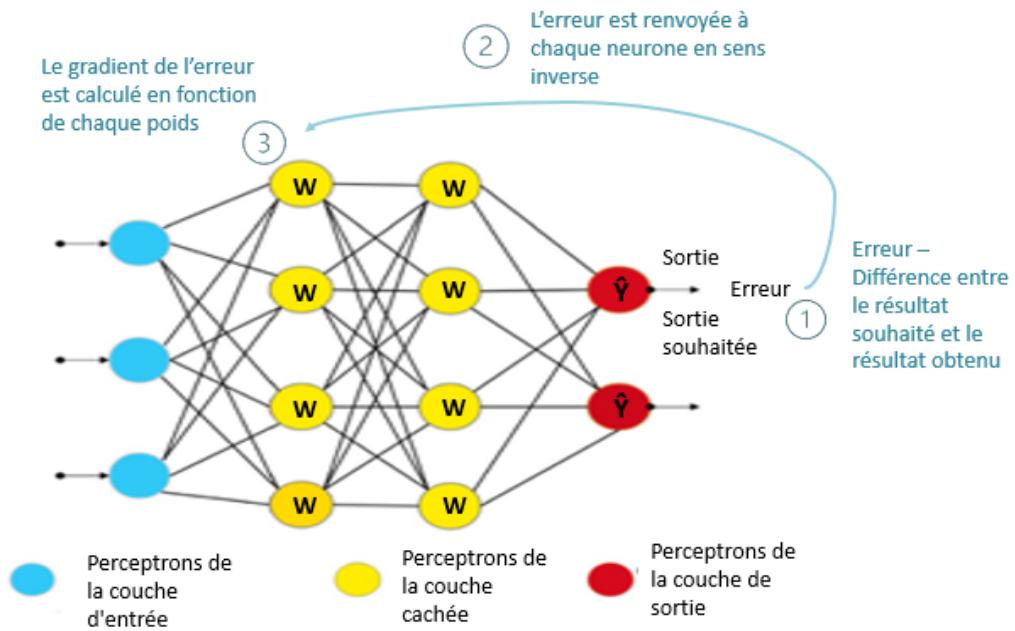


FIGURE 1.14 – Correction d'erreur par rétropropagation

La rétropropagation, abréviation de "propagation d'erreurs vers l'arrière", comme indiqué dans la Figure 1.14, est un élément essentiel de l'apprentissage et du raffinement des modèles de réseaux de neurones. Son principe consiste à ajuster les connexions internes du réseau, appelées poids, afin de minimiser l'erreur qu'il commet pendant le processus d'apprentissage.

### 1.6.1 Réseaux de neurones récurrents *RNN*

Depuis les années 2000, la capacité de calcul des ordinateurs a progressé de manière vertigineuse. Cette augmentation exponentielle de la puissance computationnelle, combinée à l'accessibilité croissante du matériel informatique, a permis le **développement de réseaux de neurones aux performances nettement supérieures à celles des générations précédentes**. En conséquence, les centres de recherche, tant publics que privés, ont intensifié leurs efforts en matière de recherche et développement dans le domaine de l'intelligence artificielle.

Cette nouvelle ère de l'intelligence artificielle est principalement dominée par les avancées liées aux **réseaux de neurones récurrents (RNN)**, une catégorie de réseaux de neurones artificiels conçus pour traiter *des données séquentielles*, telles que les séquences de mots et les séries temporelles.

Les *RNN* peuvent mémoriser les entrées précédentes en utilisant une **boucle de rétroaction** comme indiqué dans la Figure 1.15. Ils sont conçus pour traiter des séquences de données où l'information d'une étape précédente est importante pour la prédiction de l'étape suivante. Cela signifie qu'ils reçoivent non seulement des entrées de la couche précédente, mais aussi de leur propre sortie à l'étape temporelle précédente.

Cette structure permet aux neurones récurrents de conserver une mémoire de l'information passée, ce qui est crucial pour traiter des séquences de données.

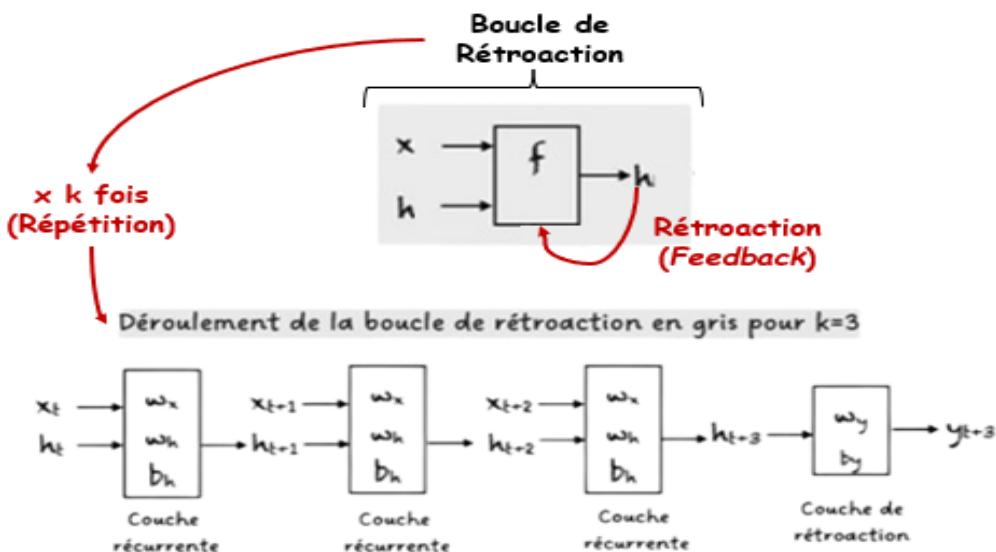


FIGURE 1.15 – Boucle de rétroaction

La boucle de rétroaction représentée dans le rectangle gris peut être déroulée en trois étapes temporales pour produire le deuxième réseau présenté juste en dessous. Chaque couche récurrente de cette boucle possède **deux ensembles de poids** : l'un pour l'entrée et l'autre pour l'unité cachée. La dernière couche récurrente, qui calcule la sortie finale pour le  $k$ -ème pas de temps, est comme une couche ordinaire d'un réseau recurrent traditionnel.

- $X \in \mathbb{R}$  est l'entrée au pas de temps  $t$ . Pour simplifier, on suppose que  $x$  est une valeur scalaire avec une seule caractéristique. L'idée peut être étendue à un vecteur de caractéristiques de dimension  $d$ .
- $y_t \in \mathbb{R}$  est la sortie du réseau au pas de temps  $t$ . Le réseau peut produire plusieurs sorties, mais on suppose ici qu'il y en a une seule.
- $h_t \in \mathbb{R}^m$  est un vecteur stockant les valeurs des unités/états cachés au pas de temps  $t$ . Il représente également le contexte actuel.  $m$  désigne le nombre d'unités cachées. Le vecteur  $h_0$  est initialisé à zéro.
- $\mathbf{W}_x \in \mathbb{R}^m$  sont les poids associés aux entrées dans la couche récurrente.
- $\mathbf{W}_h \in \mathbb{R}^{m \times m}$  sont les poids associés aux unités cachées dans la couche récurrente.
- $\mathbf{W}_y \in \mathbb{R}^m$  sont les poids associés aux unités cachées pour les unités de sortie.
- $\mathbf{b}_x \in \mathbb{R}^m$  est le biais associé à la couche récurrente.
- $b_y \in \mathbb{R}$  est le biais associé à la couche de propagation anticipée (*feedforward*).

À chaque pas de temps, on peut dérouler le réseau sur  $k$  pas de temps pour obtenir la sortie au pas de temps  $k+1$ . Le réseau déroulé est très similaire à un réseau de neurones de propagation anticipée.

Le rectangle dans le réseau déroulé montre une opération en cours. Ainsi, par exemple, avec une fonction d'activation  $f$  :

$$h_t = f(\mathbf{W}_x x_t + \mathbf{W}_h h_{t-1} + \mathbf{b}_x). \quad (1.1)$$

La sortie  $y$  au temps  $t$  est calculée comme suit :

$$y_t = f(\mathbf{W}_y h_t + \mathbf{b}_y). \quad (1.2)$$

La formation d'un *RNN* n'est pas triviale, car nous propageons les gradients à travers les couches et également à travers le temps. Ainsi, à chaque étape temporelle, nous devons additionner toutes les contributions précédentes jusqu'à celle actuelle, comme le montre l'équation suivante :

$$\frac{\partial L}{\partial \mathbf{W}} \approx \sum_{i=0}^T \left( \prod_{j=i+1}^T \frac{\partial h_j}{\partial h_{j-1}} \right) \frac{\partial h_i}{\partial \mathbf{W}}. \quad (1.3)$$

Dans cette équation, la contribution d'un état à l'étape de temps  $k$  au gradient de l'ensemble de la fonction de perte  $L$ , à l'étape de temps  $t = T$ , est calculée. Le défi pendant l'apprentissage réside dans le rapport de l'état caché :

$$\frac{\partial L}{\partial W} \approx \sum_{i=0}^T \left( \prod_{j=i+1}^T \frac{\partial h_j}{\partial h_{j-1}} \right) \frac{\partial h_k}{\partial W}. \quad (1.4)$$

Le problème des gradients qui disparaissent et qui explosent sont deux problèmes courants qui surviennent lors de la rétropropagation des données de séries temporelles sont les gradients qui disparaissent et ceux qui explosent. L'équation ci-dessus présente deux cas problématiques : la disparition du gradient et l'explosion du gradient. Dans le premier cas, le terme tend vers zéro de manière exponentielle, ce qui rend difficile l'apprentissage de certaines dépendances à long terme. Ce problème est appelé le gradient qui disparaît. Dans le second cas, le terme tend vers l'infini de manière exponentielle, et sa valeur devient NaN en raison du processus instable. Ce problème est appelé le gradient qui explose.

La disparition du gradient est caractérisée par  $\left\| \frac{\partial h_i}{\partial h_{i-1}} \right\|_2 < 1$ .

L'explosion du gradient est, au contraire, caractérisée par  $\left\| \frac{\partial h_i}{\partial h_{i-1}} \right\|_2 > 1$ .

Dans les deux sections suivantes, nous passerons en revue deux approches pour traiter ces problèmes. Les deux solutions les plus utilisées pour faire face aux deux principaux problèmes des *RNN*, à savoir la disparition et l'explosion des gradients, sont :

- La troncation de la rétropropagation à travers le temps (*TBPTT*) : Plutôt que de propager les gradients sur toute la séquence temporelle, la *TBPTT* limite la propagation à un nombre fixe d'étapes, comme indiqué dans la Figure 1.16. Cela réduit la complexité du calcul tout en préservant l'information à court terme.

**Cependant, elle sacrifie la capacité à capturer des dépendances à long terme, ce qui peut être crucial pour certaines tâches séquentielles.**

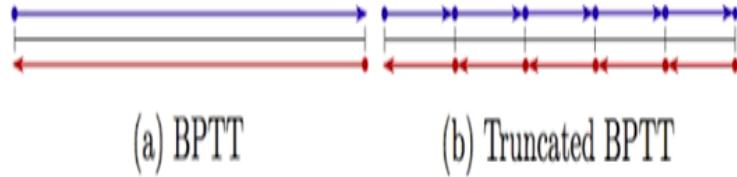


FIGURE 1.16 – Technique de Troncation de la Rétropropagation à travers le Temps (*TBPTT*)

- L'écrêtage de gradient ou *Gradient Clipping* est une méthode pour **contrôler** le problème de l'**explosion des gradients** en définissant une limite supérieure pour leur valeur. Ainsi, si un gradient dépasse cette limite, il est simplement réduit à cette valeur, empêchant ainsi l'instabilité du processus d'apprentissage, comme le montre la Figure 1.17.

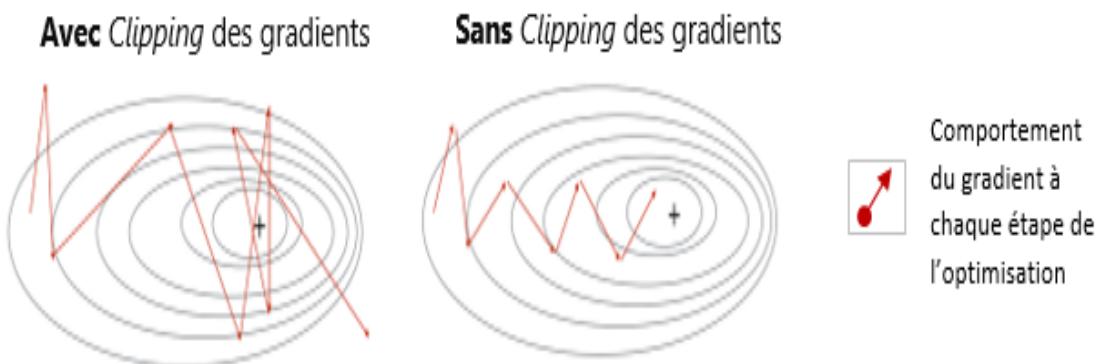


FIGURE 1.17 – Maîtrise de l'explosion des gradients (*Gradient Clipping*)

### 1.6.2 Limitations des *RNN* classiques

L'un des attraits des *RNN* est leur capacité potentielle à relier des informations passées à la tâche présente, comme utiliser des images vidéo précédentes pour comprendre l'image actuelle. Si les *RNN* pouvaient le faire, ils seraient extrêmement utiles. Mais en sont-ils capables ? Cela dépend.

Parfois, il suffit de considérer **les informations récentes** pour accomplir la tâche actuelle. Par exemple, un modèle de langage essayant de prédire le prochain mot basé sur les précédents.

Si l'on essaie de prédire le dernier mot dans "les nuages sont dans le ciel", le contexte récent suffit – il est évident que le prochain mot sera "ciel".

Dans ces cas, où l'écart entre l'information pertinente et l'endroit où elle est nécessaire est **petit**, les RNN peuvent apprendre à utiliser l'information passée, comme illustré dans la Figure 1.18.

Mais il y a aussi des cas où nous avons besoin de plus de contexte. Par exemple, essayer de prédire le dernier mot dans la phrase "J'ai grandi en France... je parle couramment le français.".

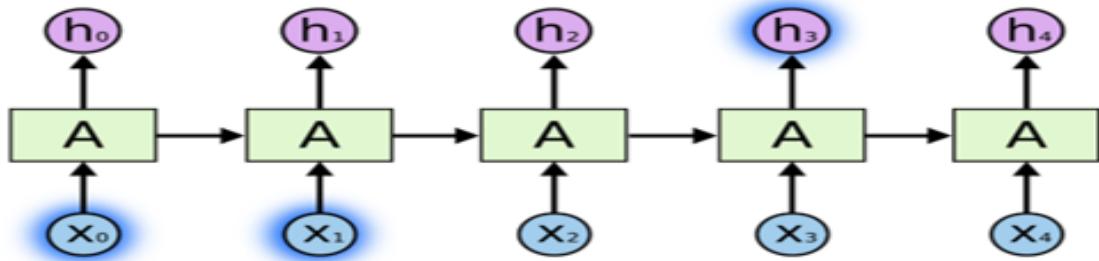


FIGURE 1.18 – Problème des réseaux de neurones récurrents

L’information récente suggère que le prochain mot est probablement le nom d’une langue, mais pour préciser laquelle, nous avons besoin du contexte "France" provenant de plus loin dans le texte. Il est tout à fait possible que l’écart entre l’information pertinente et le point où elle est nécessaire devienne très grand comme le montre la Figure 1.19.

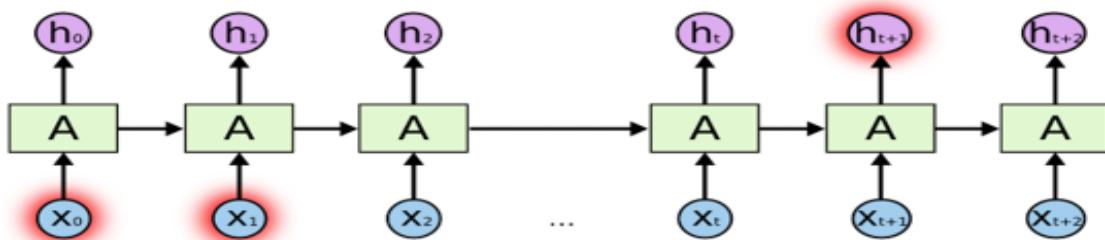


FIGURE 1.19 – Problème des réseaux de neurones récurrents

Malheureusement, à mesure que cet écart grandit, les *RNN* deviennent incapables d’apprendre à connecter les informations.

**Réseaux de mémoire à long terme (*LSTM*)** C’est dans ce contexte que les *LSTM* (ou *Long Short-Term Memory*) ont été introduits par Hochreiter et Schmidhuber en 1997. Les *LSTM* ont été perfectionnés et popularisés par de nombreux chercheurs par la suite.

Les *LSTM* intègrent **des mécanismes de mémoire** comme indiqué dans la Figure 1.20 et de régulation des gradients qui surmontent les limitations des *RNN* traditionnels, offrant ainsi une solution plus efficace et robuste pour traiter les séquences temporelles complexes.

La Figure 1.20 illustre la structure interne d’un bloc *LSTM*, un type de cellule utilisée dans les réseaux de neurones récurrents pour gérer des dépendances à long terme dans les séquences de données.

- Entrée : Les données d’entrée au bloc *LSTM*.
- Porte d’entrée (en bleu) : Contrôle quels éléments de l’entrée actuelle sont utilisés pour mettre à jour l’état de la cellule.
- Cellule (au centre) : Représente l’état de la cellule qui maintient des informations à travers les étapes de temps.
- Porte d’oubli (en orange) : Détermine quelles parties de l’état de la cellule doivent être conservées ou oubliées.

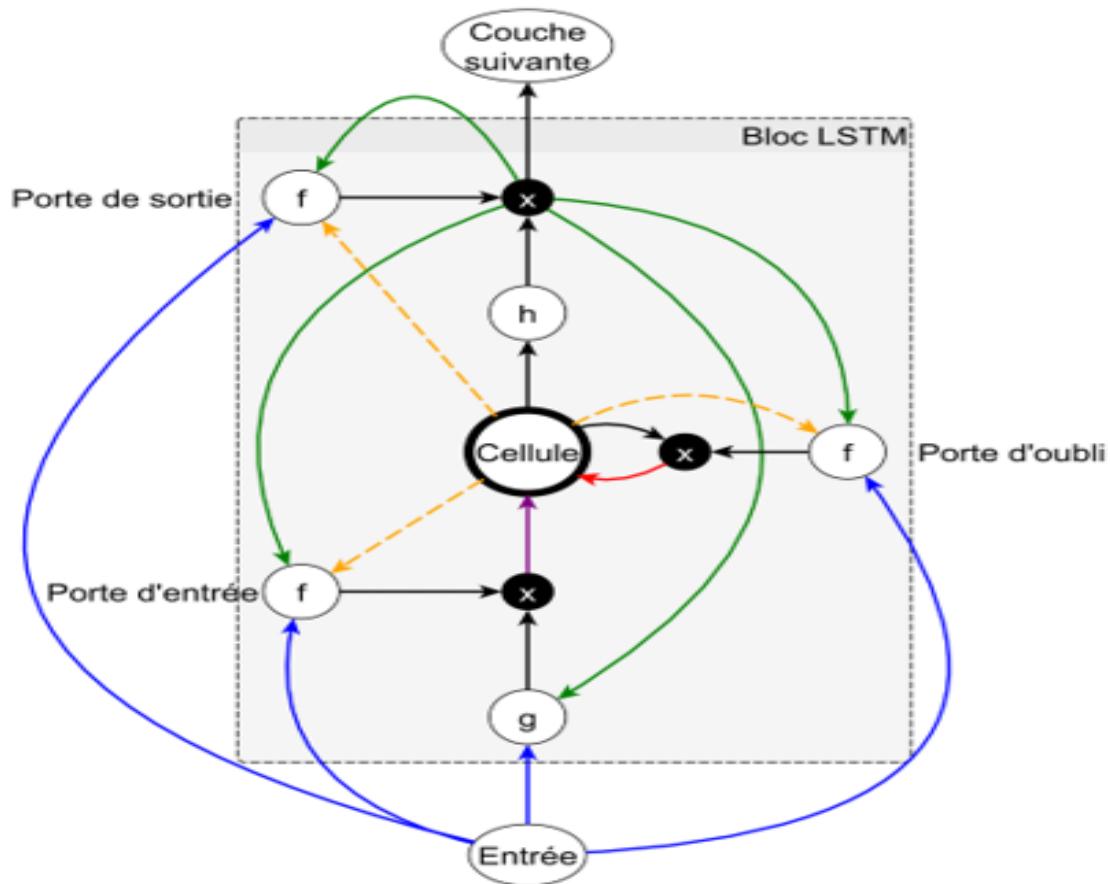


FIGURE 1.20 – Mécanisme clé des Réseaux de mémoire à long terme (*LSTM*)

- Porte de sortie (en vert) : Décide quelles parties de l'état de la cellule doivent être transmises à la sortie du bloc.
- Fonction d'activation  $f$  généralement une fonction sigmoïde, qui transforme les valeurs d'entrée pour aider à réguler le flux d'informations.
- $g$  et  $h$  : Fonctions d'activation qui peuvent être des  $tanh$  ou d'autres fonctions non linéaires, aidant à transformer les données d'entrée et l'état de la cellule pour la prochaine étape ou la sortie.
- Couche suivante : L'état de sortie qui est transmis soit à une autre couche du réseau, soit utilisé comme sortie finale.

Chaque porte a pour rôle de **filtrer les informations** de manière adaptative, permettant au *LSTM* de **maintenir des informations pertinentes sur de longues périodes et d'ignorer les détails non pertinents**.

Cependant, les *LSTM* sont limités par leur **incapacité à paralléliser les opérations** et leur tendance à **oublier les informations** dans les **séquences très longues**, problèmes que les architectures de (*Transformers*) ont adressé avec leur mécanisme **d'attention globale** qui permet une contextualisation efficace sur l'ensemble de la séquence et une plus grande scalabilité.

### 1.6.3 Transformateurs et Mécanismes d'Attention

Le transformateur, une architecture d'apprentissage profond, introduite pour la première fois dans l'article « Attention is all you need »<sup>2</sup> par des chercheurs de Google en 2017, a deux composantes principales : l'encodeur (*Encoder*) et le décodeur (*Decoder*), voir Figure 1.21.

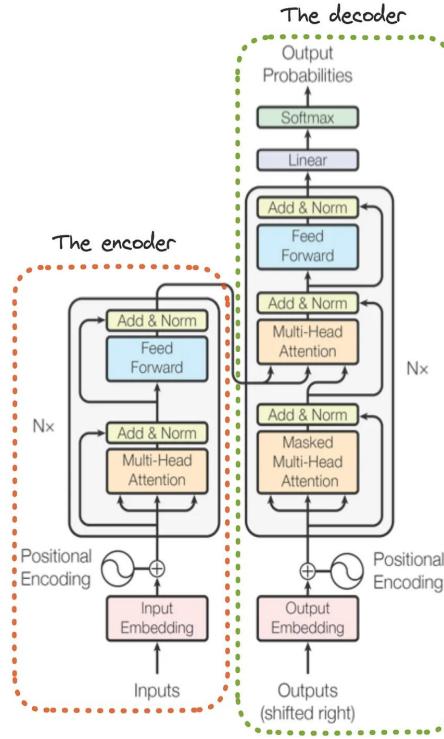


FIGURE 1.21 – Architecture du Transformateur [Han et al., 2021]

L'encodeur (*Encoder*) examine la séquence complète de texte fournie en entrée et construit une représentation contextuelle de ce texte. Ensuite, le décodeur (*Decoder*) prend cette représentation en entrée et l'utilise pour produire, une séquence de sortie, répondant à la tâche qui lui est demandée (Génération, traduction ou classification de texte, etc.).

Par exemple, si un modèle adaptant l'architecture du transformateur est utilisé pour générer une phrase, il effectuera les étapes de traitement suivantes :

- L'encodeur ingère une phrase d'entrée, qui peut être une amorce ou un contexte donné.
- L'encodeur produit une représentation contextualisée de cette amorce.
- Le décodeur reçoit la représentation contextualisée et commence la génération de texte.
- À chaque étape de la génération, le décodeur prend en compte à la fois l'entrée initiale et les mots déjà générés pour prédire un mot unique, utilisant son mécanisme d'attention pour pondérer l'importance relative des différents éléments du contexte.
- Ce processus se poursuit jusqu'à ce que le décodeur produise un marqueur de fin de séquence ou atteigne une limite prédéfinie de longueur de texte.

L'intuition derrière l'encodeur est qu'à travers une série de couches, nous construisons des représentations contextualisées de plus en plus riches des significations des mots ou des jetons d'entrée.

2. "L'attention, c'est tout ce dont vous avez besoin".

Les jetons obtenus, grâce à la phase de prétraitement (*Preprocessing*) des séquences de texte à traiter, sont passés à travers un mécanisme d'Attention qui permet à un modèle de pondérer l'importance de différents mots dans une séquence par rapport à un mot donné.

**Attention Scalée par Produit Scalaire (Scaled Dot Product Attention)** Dans ce processus, chaque mot dans une séquence est transformé en trois vecteurs différents : *une clé (K)*, *une valeur (V)*, et *une requête (Q)*, comme le montre la Figure 1.22.

Ces matrices font partie des **paramètres du modèle** et sont apprises pendant le processus d'entraînement. Cela signifie que **leurs valeurs sont ajustées au fil de l'apprentissage** pour optimiser la performance du modèle sur une tâche donnée.

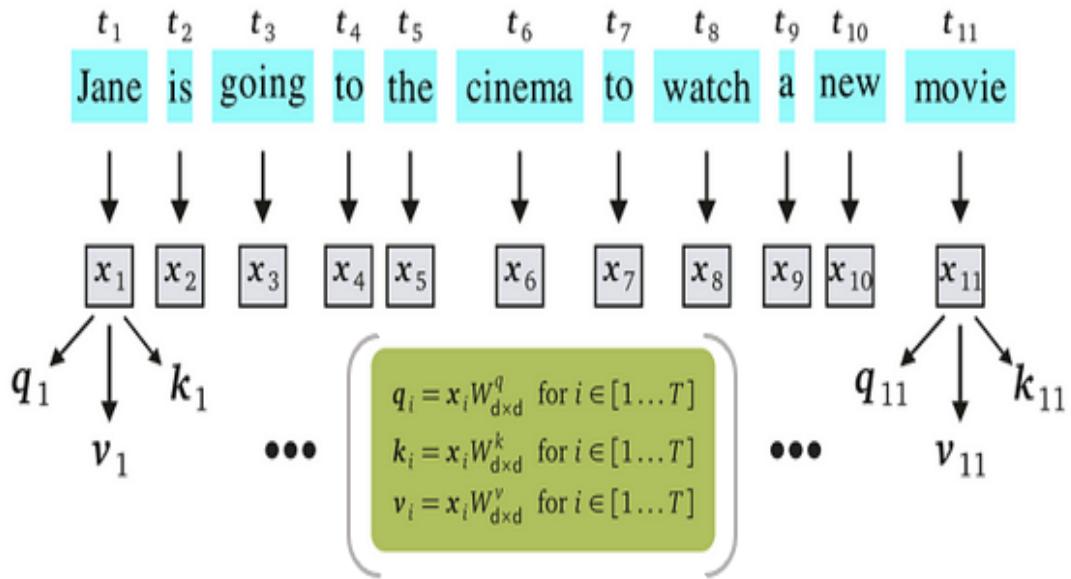


FIGURE 1.22 – Obtention des matrices clé (K), valeur (V), requête (Q)

Le calcul de l'attention est ensuite effectué, par :

- **Le produit scalaire** des vecteurs de requête ( $Q$ ) avec tous les vecteurs de clé ( $K$ ) comme le montre la Figure 1.23, afin de déterminer l'importance des mots dans le contexte de chaque mot spécifique.

Pour chaque mot de la séquence d'entrée (représenté par un vecteur requête  $q_i$ ), on calcule le produit scalaire avec tous les vecteurs clés de la séquence ( $k_j$  pour tous les  $j$  de 1 à la longueur de la séquence). Ce produit scalaire mesure la similitude entre le mot actuel (représenté par  $q_i$ ) et chaque autre mot (représenté par  $k_j$ ) dans la séquence.

Puis le résultat obtenu est passé à travers un cycle composé de 3 parties essentielles comme indiqué dans la Figure 1.24.

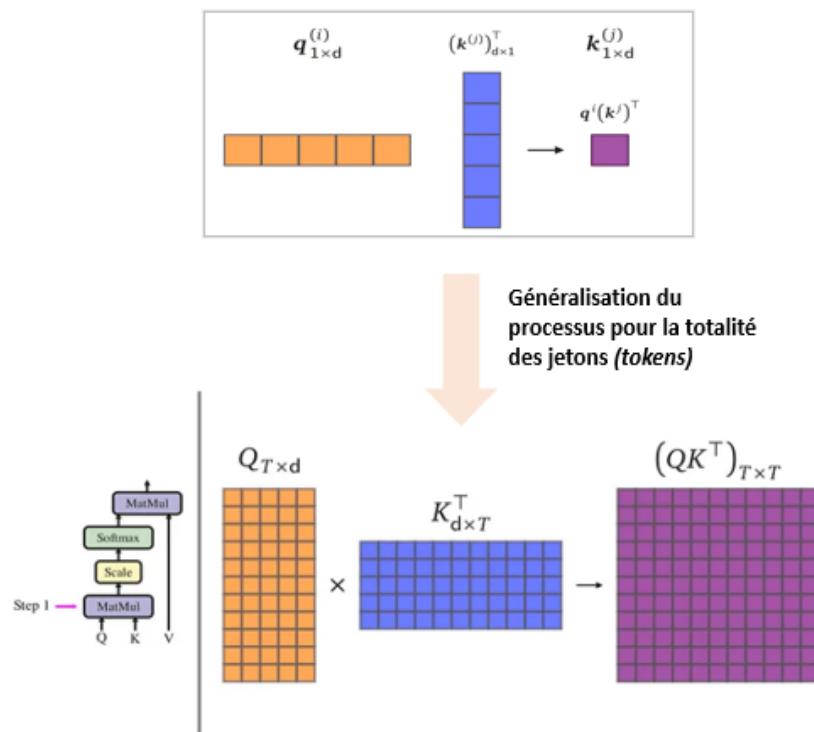


FIGURE 1.23 – Mise en relation des matrices clé ( $K$ ) et requête ( $Q$ )

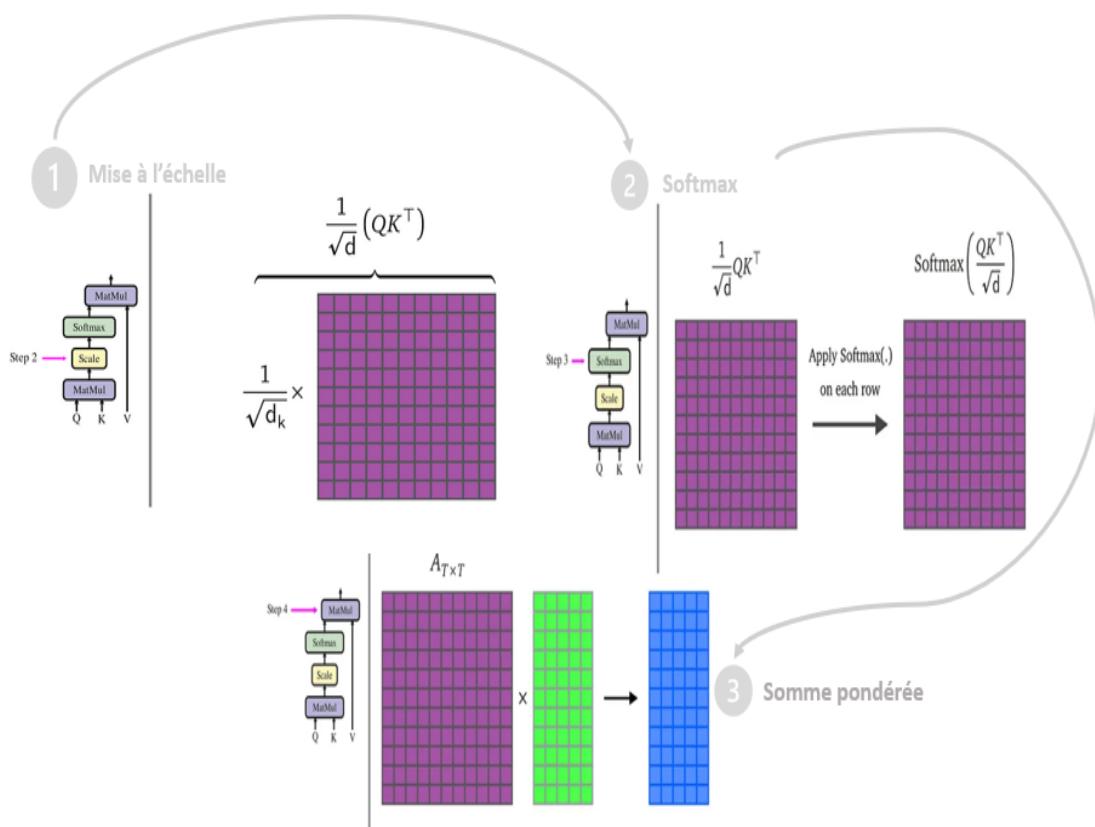


FIGURE 1.24 – Traitement du produit scalaire obtenu

Les étapes indiquées dans la Figure 1.24 ci-dessus sont :

- La Mise à l'échelle : Les valeurs du produit scalaire sont divisées par la racine carrée de la dimension des vecteurs clés. Cela aide à stabiliser le calcul et à prévenir l'explosion des gradients pendant l'apprentissage.
- Fonction softmax : On applique une fonction softmax aux valeurs du produit scalaire mis à l'échelle. Celà convertit les valeurs en une distribution de probabilités sur tous les mots de la séquence. ***L'interprétation est que chaque valeur représente l'importance relative du mot correspondant ( $k_j$ ) pour le mot actuel ( $q_i$ ) dans le contexte.***
- Somme pondérée : On multiplie chaque vecteur **Valeur**  $v_j$  par son poids d'attention correspondant (calculé à l'étape de la mise en échelle) et on somme les vecteurs ainsi pondérés. Cela donne un vecteur de contexte  $r_i$ , qui représente les informations les plus pertinentes de la séquence, pour le mot actuel ( $q_i$ ).

Plus simplement, le processus de *scaled dot product* attribue à chaque mot pris en considération, un ensemble de scores qui modélisent sa relation avec les différents mots présents dans la séquence. Plus ce score est élevé, plus le mot auquel il est associé est important pour le mot actuel dans le contexte. L'intuition de base de ce mécanisme d'attention, est l'idée de comparer un **élément d'intérêt** à une collection d'**autres éléments de manière à révéler leur pertinence dans le contexte actuel**. La Figure 1.25, schématisé le processus évoqué précédemment.

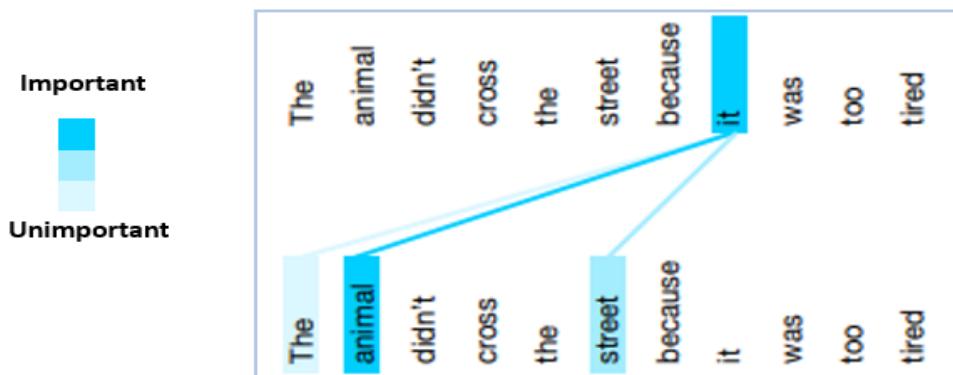


FIGURE 1.25 – Mécanisme d'Attention : Attribution de scores

Ensuite, vient le processus de *multihead attention* qui utilise le produit du scale dot product attention, comme indiqué dans la Figure 1.26, permettant de mieux capturer les relations sémantiques entre des mots qui ne sont pas nécessairement adjacents dans la séquence. Cela permet au modèle de mieux comprendre la structure globale de la phrase et d'extraire des informations plus contextuelles.

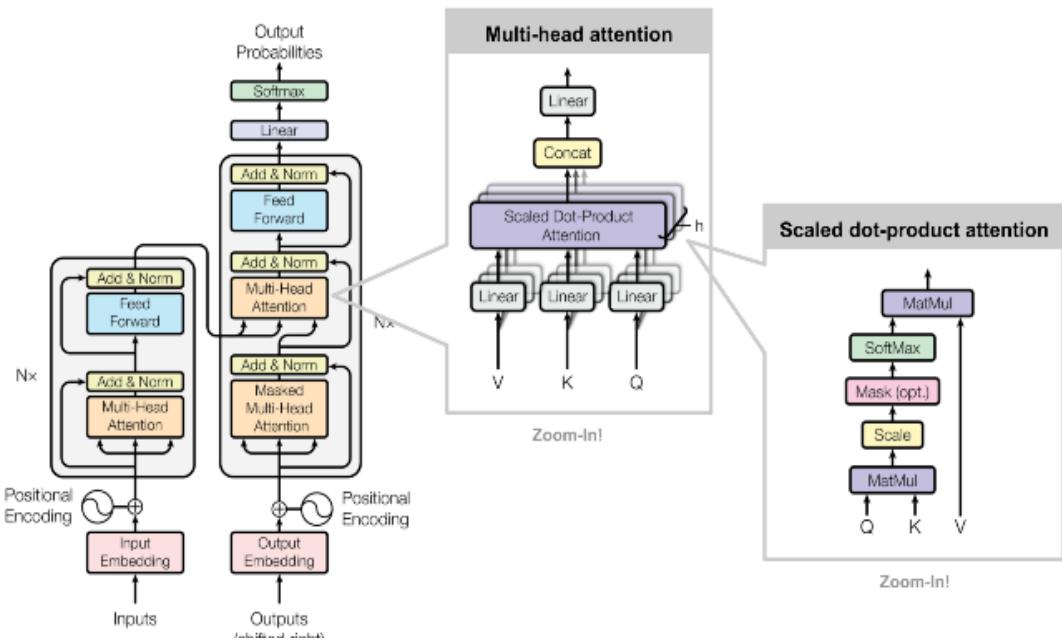


FIGURE 1.26 – Mécanisme d’attention dans un Transformateur

A l’issue de l’application de *dot product attention*, le processus de Multi-Head Attention applique les deux étapes suivantes :

- Concaténation des sorties : Les sorties de chaque application du dot scaled product attention sont ensuite concaténées, créant une représentation multidimensionnelle riche des relations entre les mots dans la séquence.
- Application d’une transformation linéaire finale : La concaténation est ensuite projetée sur un nouvel espace vectoriel via une transformation linéaire finale. Cette étape permet d’ajuster la dimension de la sortie et de l’adapter aux besoins spécifiques de la tâche à accomplir.

Ces scores obtenus sont pris en considération durant le processus d’encodage (*Embeddings*), permettant une meilleure compréhension des **relations sémantiques entre les mots**.

Dans la partie décodeur (Decoder), un autre type de mécanisme d’attention est pris en considération : Contrairement au "Multi-Head Attention" vu dans l’encodeur, le "Masked Multi-Head Attention" est spécialement conçu pour le décodage séquentiel, où le modèle génère la sortie mot par mot.

Le Masked Multi-Head Attention a pour but principal d’empêcher le modèle de langage d’anticiper les mots à venir lors de la génération du texte. En d’autres termes, il garantit que le modèle se base uniquement sur les mots déjà générés pour déterminer le mot suivant à produire.

Imaginons un modèle de langage apprenant à générer une phrase.

- **Sans Masked Multi-Head Attention**, le modèle pourrait "tricher" en regardant les mots qui n’ont pas encore été générés. Cela pourrait le conduire à produire des phrases incohérentes ou grammaticalement incorrectes.
- **Avec Masked Multi-Head Attention**, le modèle est contraint de se concentrer uniquement sur les mots déjà générés. Cela l’oblige à raisonner de manière logique et à produire un texte qui a du sens dans le contexte.

Pour illustrer ce concept, prenons l'exemple de la phrase suivante : "Le chat marcha dans la rue.". Supposons que le modèle ait déjà généré les mots "Le chat" et qu'il cherche à déterminer le mot suivant. Sans Masked Multi-Head Attention, le modèle pourrait envisager des mots comme "courut", "mangea" ou "joua", qui ne correspondent pas au contexte de la phrase. Avec le Masked Multi-Head Attention, le modèle ne pourra considérer que les mots qui suivent "chat" dans la séquence, tels que "marcha", "bondit" ou "s'assit". Cela le guidera vers le choix du mot correct, "marcha", qui donne du sens à la phrase.

**Encodage** Dans le domaine du traitement automatique du langage (TAL), l'encodage (*Embeddings*) joue un rôle fondamental. Il consiste à transformer des **données textuelles brutes**, non structurées, en une **représentation numérique dense et vectorielle** que les modèles d'apprentissage automatique en particulier les *grands modèles de langages (LLMs)* peuvent comprendre et traiter efficacement. La Figure 1.27 schématise ce processus au plus simple.

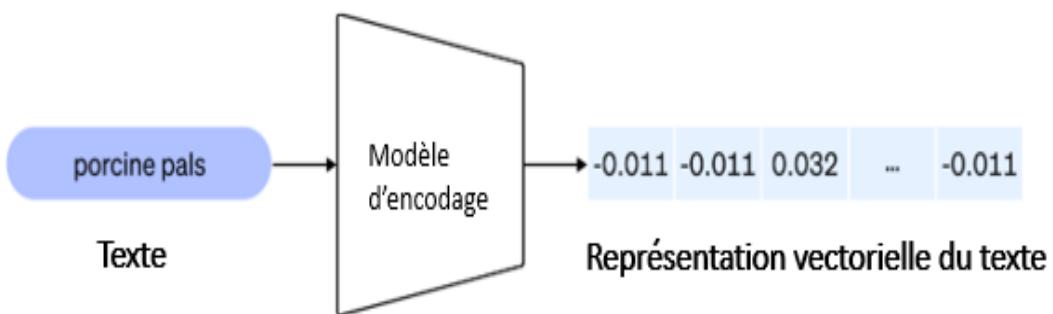


FIGURE 1.27 – Processus d'encodage

L'évolution des techniques d'encodage pour le traitement automatique du langage naturel (TAL) a permis aux modèles de mieux comprendre et de traiter le langage humain. Ce domaine a connu des progrès majeurs, notamment en ce qui concerne la prise en compte de **la sémantique des mots dans leur contexte**.

**Passage du Bag-of-Words aux modèles contextuels** Le *Bag-of-Words (BoW)*, une approche initiale d'encodage, représentait les mots comme de simples entités indépendantes, ignorant leur contexte dans une phrase ou un document. Chaque mot est associé à une valeur binaire indiquant sa présence ou son absence dans le document.

Cette limitation empêchait les modèles de capturer les nuances du langage et les relations entre les mots. De plus, cette technique d'encodage (*Embeddings*) s'est avérée sensible à la présence de mots rares ou non pertinents.

L'avènement des modèles contextuels a révolutionné l'encodage en prenant en compte le contexte des mots dans leur environnement linguistique.

Des modèles comme Continuous Bag-of-Words (CBOW) et Skip-gram dans **Word2Vec** ont permis aux modèles d'apprendre des relations sémantiques entre les mots, améliorant ainsi leur capacité à représenter le sens du texte. Cependant ils restent limités :

- **Représentation statique des mots** : ils représentent les mots par des vecteurs statiques, ne tenant pas compte de leur sens variable en fonction du contexte. Cela peut limiter leur capacité à comprendre des phrases complexes ou des expressions idiomatiques.
- **Incapacité à capturer des relations à longue distance** : Les modèles traditionnels n'arrivent pas à capturer des relations sémantiques à longue distance entre les mots dans une phrase ou un document, ce qui peut affecter leur compréhension globale du sens du texte.

L'essor des réseaux de neurones profonds, en particulier les réseaux de neurones récurrents (*RNN*) et les *Long Short-Term Memory (LSTMs)*, a permis de créer des modèles d'encodage encore plus sophistiqués. Ces modèles peuvent capturer des patterns complexes dans les données textuelles et apprendre des relations sémantiques à longue distance entre les mots grâce à leurs constituants évoqués dans les sections précédentes : les cellules de mémoire et de les boucles de rétroaction.

**Modèles d'encodage basés sur l'architecture du Transformateur** L'arrivée de vastes corpus de texte et des grands modèles de langage (*LLMs*) a mis en lumière les limites des encodeurs traditionnels, peinant à s'adapter au traitement de textes volumineux et complexes. Face à ce défi, les modèles d'encodage basés sur l'architecture Transformer se sont imposés attirant l'attention et devenant le centre d'intérêt de tous les chercheurs.

Leurs avantages qui ont le plus marqué et révolutionné le domaine de l'encodage sont :

### 1. Parallélisation et scalabilité pour traiter de gros volumes de données

En comparaison avec les modèles récurrents classiques, l'architecture Transformer tire parti d'un mécanisme d'attention ingénieux, qui lui permet de traiter les séquences de mots de manière parallèle, exploitant pleinement la puissance des architectures de calcul modernes comme les GPU.

Cette approche se traduit par une exécution bien plus rapide et une scalabilité accrue, rendant les modèles Transformer parfaitement adaptés au traitement de gros volumes de données textuelles.

### 2. Flexibilité et adaptabilité pour répondre à des besoins spécifiques

L'atout majeur de l'architecture Transformer réside dans sa modularité. Sa structure flexible permet d'intégrer facilement de nouveaux modules d'attention, répondant ainsi à des exigences et des tâches spécifiques.

Cette personnalisation ouvre un champ de possibilités infinies pour l'adaptation des modèles Transformer à divers domaines d'application.

### 3. Une meilleure compréhension du langage naturel pour des résultats de pointe

Grâce à leur capacité à capturer des relations à longue distance entre les mots et à comprendre le contexte avec finesse, les modèles Transformer offrent une représentation plus riche et plus précise du sens du texte. Cette compréhension approfondie du langage naturel se traduit par des résultats de pointe dans une large gamme de tâches de TAL, telles que la traduction automatique, la génération de texte ou la question-réponse.

## 1.7 Conclusion

À travers ce chapitre, nous avons pu découvrir FORVIA et son importance sur le marché de la production automobile. Nous avons également exploré la plateforme Palantir, avec laquelle FORVIA collabore, et jeté un coup d’œil sur l’histoire du traitement automatique du langage (TAL) et les premières avancées dans ce domaine.

Nous avons ensuite effectué une étude sur la solution existante, considérée comme très limitée. Nous avons examiné les différentes approches adaptées, en les présentant de manière chronologique, tout en mettant en avant les limitations de chacune et en expliquant comment l’approche suivante a pu les surmonter. Cette exploration a couvert l’approche statistique, en détaillant les aspects mathématiques considérés, jusqu’à l’approche intégrant l’apprentissage profond, en passant par les premières avancées dans le domaine des réseaux neuronaux.

# Chapitre 2

## Sélection et Évaluation des Modèles pour un *Widget d'IA Performant*

Ce chapitre présente les architectures des modèles sélectionnés dans le cadre de ce projet, à savoir des modèles **GPT** proposés par OpenAI. Ensuite, les bonnes performances du modèle d'encodage *Text-embedding-ada-002*, qui nous ont convaincus de l'adopter. La section 2.2 présente nos contributions.

### 2.1 Modèles utilisés

Dans cette section, nous aborderons d'abord les modèles utilisés pour le traitement des données textuelles, en mettant particulièrement l'accent sur l'architecture des grands modèles de langage, notamment l'architecture "*Decoder only*" (décodeur uniquement) et les mécanismes clés qui lui permettent d'exceller dans la génération de texte.

Nous justifierons également les choix effectués concernant les modèles d'encodage (*embedding models*).

Ensuite, nous expliquerons l'architecture du modèle Multi-Modal : **GPT-4-Vision**, qui est le modèle utilisé pour traiter les images et en extraire des descriptions, ainsi que la manière dont cette architecture permet de traiter à la fois, des données textuelles et des éléments visuels.

#### 2.1.1 Grands Modèles de Langage (*LLMs*)

Dans le cadre de ce projet, nous nous sommes intéressés plus aux grands modèles de langage (*ou Large Language Models : LLMs*), qui se situent à la croisée de deux domaines en plein essor actuellement : le Traitement Automatique du Langage Naturel TAL ou (*NLP*) et l'Intelligence Artificielle Générative (GenAI).

Ces modèles représentent une **nouvelle génération de l'architecture du Transformateur**, offrant une performance nettement supérieure en matière de génération de texte. En adoptant une architecture exclusivement basée sur des décodeurs, comme illustré dans la Figure 2.1, ces modèles ont atteint des niveaux de performance considérés comme révolutionnaires dans le domaine de la génération de texte.

Contrairement aux architectures traditionnelles où le texte brut est prétraité par l'encodeur pour être transformé en représentations vectorielles exploitable par le décodeur, dans le modèle que nous décrivons ici, illustré dans la Figure 2.1, les entrées sont directement des vecteurs d'*embeddings*.

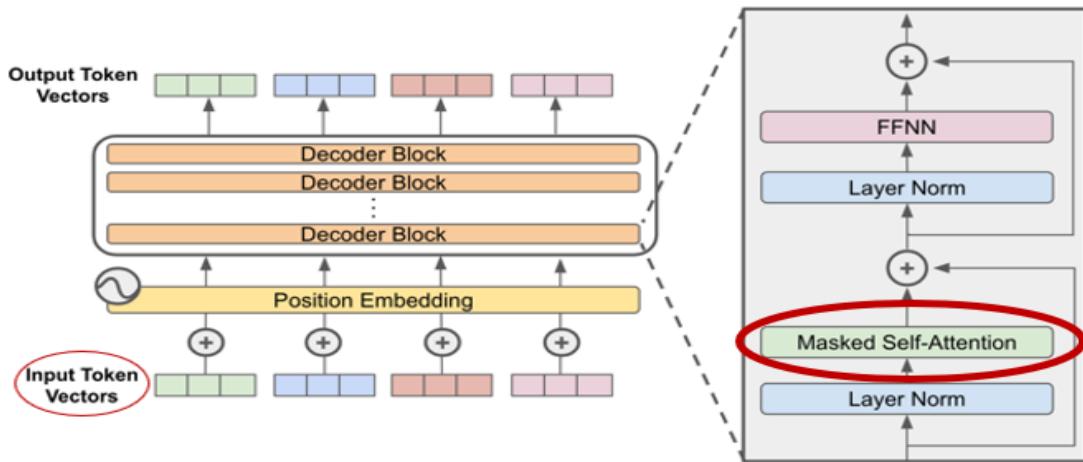


FIGURE 2.1 – Architecture du décodeur uniquement (*Decoder-only*)

**Avantages de l'architecture du décodeur uniquement (*Decoder-only*)** Les avantages apportés par cette architecture sont principalement :

- L'architecture est plus adaptée à l'affinage (*fine-tuning*) puisque cette architecture contient moins de couches.  
En effet, l'absence de la partie encodeur réduit la mémoire et le temps de calcul nécessaires, ce qui est particulièrement avantageux pour l'affinage (*fine-tuning*) sur des ressources limitées.
- Performances meilleures pour la génération de texte grâce au **mécanisme d'attention masquée** (*Masked self-attention*). En effet, ce mécanisme de masquage empêche le modèle de prendre en compte les mots qui ont été déjà générés, lors de la génération du mot suivant. Cela garantit que le texte généré est cohérent, grammaticalement correct et pertinent pour le contexte.

Imaginons par exemple, que le modèle "*Decoder-only*" génère la phrase suivante :

"Le chat **court** dans le jardin."

La formule de masquage adaptée par le modèle est la suivante :

$$g(x) = f(x) \cdot \mathbb{1}_M(x)$$

$f(x)$  représenterait le mot "**court**" dans cet exemple. Le mécanisme d'attention calculerait les **scores d'attention** pour chaque mot de la phrase. Le **mécanisme d'attention masquée** joue le rôle d'une fonction indicatrice  $\mathbb{1}_M(x)$  appliquée avant le calcul des scores d'attention. Il adapte dynamiquement sa valeur à 1 pour les mots pertinents que le modèle doit prendre en considération et à 0 pour les mots à négliger.

Cela garantit que les scores d'attention sont basés uniquement sur les mots pertinents pour la prédiction du mot suivant, c'est-à-dire les mots déjà générés qui sont dans notre cas : "le", "chat" et "jardin".

Par conséquent, les scores d'attention seraient plus élevés pour ces mots comme, car ils sont plus pertinents pour le sens du mot "**court**".

Le contexte pondéré serait calculé en utilisant les scores d'attention et l'information précédente. Cela **donnerait une représentation pondérée du contexte, en se concentrant davantage sur les mots les plus pertinents.**

Le modèle utiliserait, ensuite, le contexte pondéré et l'état caché du réseau neuronal pour générer le mot suivant. Dans cet exemple, le modèle pourrait générer le mot "**rapidement**", car il suit logiquement le mot "**court**".

Ces vecteurs d'embeddings sont obtenus par des **processus de prétraitement**, tels que la tokenisation et l'encodage, **effectués en amont**, en dehors du modèle de langage lui-même.

Par exemple, dans le cas d'un texte, chaque mot ou phrase est d'abord converti en un **token** unique. Ces tokens sont ensuite encodés en **vecteurs numériques** qui capturent des aspects sémantiques et syntaxiques du langage.

Ces vecteurs **prétraités** sont alors utilisés comme **entrées pour le modèle**, permettant ainsi une intégration plus directe et efficace des données.

Le choix du modèle d'encodage offre aux chercheurs et développeurs plusieurs options, nécessitant une analyse approfondie et diverses considérations.

### 2.1.2 Choix du modèle d'Encodage (*Embeddings Model*)

Le modèle d'encodage *text-embedding-ada-002* a été sélectionné par le département de la Transformation Digitale de FORVIA, pour être intégré dans la plateforme Palantir Foundry, en raison de sa compatibilité optimale avec les modèles de type **GPT** (Generative Pre-trained Transformer), principalement utilisés tout au long de ce projet.

Bien qu'il existe des modèles d'encodage préentraînés sur des textes manufacturiers, qui se sont montrés très efficaces, ces modèles ne sont pas accessibles à travers la plateforme Palantir Foundry .

Nous avions donc uniquement le choix entre *text-embedding-ada-002* ou avoir recours aux techniques d'encodage traditionnelles telles que **Word2Vec**<sup>1</sup>.

Il n'était clairement pas envisageable pour nous de recourir à ces techniques, vu qu'elles se sont révélées très limitées pour le traitement de grandes quantités de données textuelles, notamment en raison de leur faible capacité à gérer **la sémantique et le contexte**, des aspects cruciaux non seulement pour ce projet mais également pour la transformation digitale de l'entreprise FORVIA.

La Figure 2.2, ci-dessous montre le catalogue des modèles d'encodages offert par la société Forvia, incluant uniquement *text-embedding-ada-002*.

Nous avons donc décidé de creuser un peu plus pour pouvoir comprendre les raisons qui ont poussé les experts scientifiques de la société Forvia, à choisir *text-embedding-ada-002* parmi des milliers de modèles, pour l'intégrer dans palantir Foundry.

---

1. Le modèle "sac de mots" ou *Bag of Words*, voir Section 1.6.3

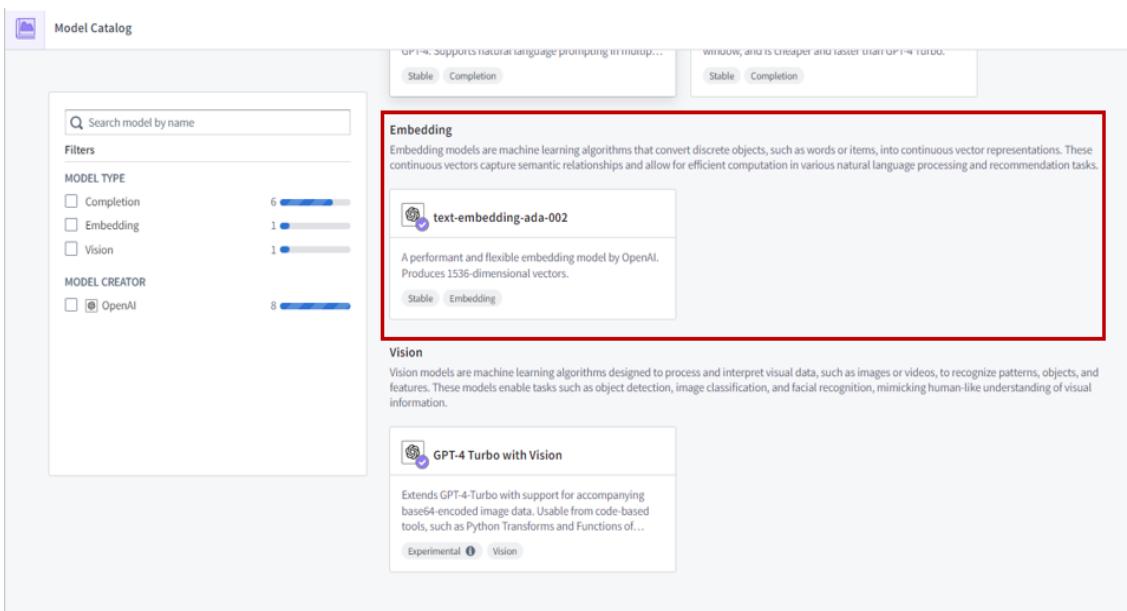


FIGURE 2.2 – Catalogue des modèles d’encodage contenant uniquement *text-embedding-ada-002*

L’élément le plus important pris en considération était un indicateur clé de performance (KPI) incontournable dans le cadre de notre projet, la **similarité cosinus** ou (*cosine similarity*), qui permet d’évaluer la capacité du modèle à capturer la sémantique et à différencier les contextes, même s’il s’agit du même terme.

The FORVIA logo is at the top right. To its left are five circular icons representing different business areas: a car wheel, a gear, a document, a key, and a camera. Below these icons, the word 'Leadership' is written. Further down, the word 'Finance' is written. A table follows:

Title of the document	Property, Plant and Equipment
Reference	<b>FAU-I-LSG-3502</b>
Purpose	To define the accounting principles and policies applicable to Property, Plant and Equipment for consolidated reporting purposes based on the IFRS norms (International Financial Reporting Standards).
Scope	GROUP

FIGURE 2.3 – Exemple d’utilisation du terme "Plant" au sein de FORVIA

En effet, dans le secteur de la production où opère FORVIA, la signification d’un terme peut varier considérablement selon le contexte, rendant ces approches classiques inadaptées pour une interprétation précise et efficace des données.

Prenons l’exemple du mot "Plant", un mot très fréquent dans un contexte manufacturier. Dans le langage courant, "Plant" pourrait simplement désigner une plante, c’est d’ailleurs la signification la plus courante de ce mot. Cependant, dans un contexte de production, comme celui indiqué dans la Figure 2.3, le même mot pourrait se référer à une usine.

**Evaluation de *text-embedding-ada-002*.** Dans l'étude du choix de ce modèle, nous avons eu recours à l'évaluation **MTEB**.

Le **Massive Text Embedding Benchmark (MTEB)** est considéré comme l'un des meilleurs *benchmarks* pour évaluer la similarité sémantique, un indicateur clé que nous considérons comme le plus important dans notre étude.

Le *MTEB* a gagné l'unanimité et l'accord de la communauté scientifique pour plusieurs raisons :

1. **Diversité des Données** : *MTEB* couvre une grande variété de domaines et de types de textes, incluant des *textes scientifiques*, des *nouvelles*, des *discussions en ligne*, etc. Cette diversité assure que les modèles sont évalués sur une large gamme de contextes sémantiques.
2. **Qualité des Annotations** : Les paires de phrases utilisées dans *MTEB* sont annotées par des **experts humains**, ce qui garantit la **précision et la fiabilité des scores de similarité de référence**.
3. **Grande Échelle** : *MTEB* utilise **un ensemble de données de grande taille**, ce qui permet de tester les modèles sur un nombre significatif de paires de phrases. Cela aide à assurer que les résultats sont **statistiquement significatifs** et représentatifs des capacités du modèle.
4. **Métriques Complètes** : Le *benchmark MTEB* utilise des métriques d'évaluation robustes telles que la **corrélation de Pearson** et de **Spearman** pour comparer les scores de similarité prédits par les modèles avec les annotations humaines. Ces métriques sont reconnues pour leur efficacité à mesurer la performance des modèles de similarité sémantique.

Le benchmark *MTEB* a révélé que le modèle *text-embedding-ada-002* se classe au premier rang en termes de similarité de phrases [Malode, 2024]. Ce résultat met en évidence l'efficacité et la précision de ce modèle dans la capture des similarités sémantiques entre les phrases.

La similarité des phrases est un score calculant la similarité cosinus (ou cosine similarity) entre deux phrases incluant un ou plusieurs termes en commun mais dans deux contextes différents. Ce score permet d'évaluer la capacité de ce modèle à capturer la sémantique et à différencier les contextes, même s'il s'agit du même terme.

La Table 2.1 montre le classement du modèle *text-embedding-ada-002* par rapport à d'autres modèles, illustrant clairement sa supériorité en termes de similarité de phrases.

Model	Performance
<i>text-embedding-ada-002</i>	81.5
<i>text-similarity-davinci-001</i>	80.3
<i>text-similarity-curie-001</i>	80.1
<i>text-similarity-babbage-001</i>	80.1
<i>text-similarity-ada-001</i>	79.8

TABLE 2.1 – Classement du modèle *text-embedding-ada-002* selon le benchmark *MTEB*

Pour garantir que ce modèle d'*embedding* est performant non seulement en termes de compréhension du contexte et de la sémantique générale, mais surtout dans le **contexte de production** (le contexte des données textuelles de notre projet), nous avons effectué une étude détaillée suivant les mêmes étapes adaptées dans le *MTEB Benchmark*, visant à évaluer la capacité du modèle *text-embedding-ada-002* à capturer des nuances sémantiques spécifiques et à différencier des contextes variés.

Nous avons sélectionné des exemples de phrases comportant des mots-clés communs mais utilisés dans deux contextes très différents : le contexte manufacturier et un autre contexte variable selon la phrase.

Ces paires de phrases permettent d'évaluer la robustesse du modèle dans des situations où le même terme peut avoir des significations différentes selon le contexte.

Ensuite, nous avons procédé à :

- L'encodage de ces phrases en utilisant *text-embedding-ada-002*. Chaque phrase a été transformée en une représentation vectorielle (*embeddings*), qui permet de comparer quantitativement leur similarité sémantique.
- Le calcul de la **similarité cosinus** (ou *cosine similarity*) pour mesurer la distance entre les représentations vectorielles des phrases. Cette mesure nous a permis d'évaluer la **capacité du modèle à différencier les contextes** même lorsque les phrases partagent les **mêmes mots-clés**.

La Table 2.2 représente les phrases prises en considération (de référence) pour notre étude, et les scores de similarité sémantique obtenus.

Mot_clé	Phrase_1	Phrase_2	Phrase_2 encodage	Phrase_1 encodage	cosine_similarity
Plant	The new car manufacturing plant...	The botanical garden features...	[-0.008997, -0.003328, ...]	[-0.011166, -0.001953, ...]	0.743358
Cell	The production cell was upgraded...	The scientist examined the...	[-0.004842, 0.003801, ...]	[0.017892, -0.004040, ...]	0.784127
Line	The assembly line was shut down...	He waited in line for hours...	[-0.020327, 0.019215, ...]	[0.029534, 0.011008, ...]	0.770964
Mold	The mold used in the injection...	The bread was covered in mold...	[0.007743, -0.022787, ...]	[0.013845, -0.006895, ...]	0.802035

TABLE 2.2 – Similarité cosinus mesurant les éléments communs aux phrases choisies

Nous avons obtenu des scores de similarité sémantique faibles (inférieurs à 0.8), mettant en évidence la capacité du modèle *text-embedding-ada-002* à discerner que, même si les phrases utilisent les mêmes termes du vocabulaire du *manufacturing* (tels que **Plant**, **cell**, **line** et **mold**), elles demeurent **sémantiquement distinctes**.

**Vision** Pour le traitement des éléments visuels (principalement des graphes, des images et des dessins techniques), on a eu recours à **GPT-4-Vision-Turbo**. Ce modèle est Multi-Modal, il est capable de traiter du texte et des éléments. Dans le cadre de ce projet, nous avons eu recours à ce modèle pour générer des descriptions aux images contenues dans les documents PDFs traités. Les composantes clés incluses dans **GPT-4-Vision** qui lui permet de réaliser la vision par ordinateur et de traiter des images sont :

- **Modèles Convolutionnels (CNN)** : Traditionnellement, les CNN sont utilisés pour le traitement d’images. Ils sont particulièrement efficaces pour extraire des caractéristiques visuelles à partir des images grâce à leurs *couches de convolution, de pooling et de fully connected*. Chaque couche de convolution applique un ensemble de filtres pour détecter des motifs tels que des bords, des textures, et des formes.
- **Vision Transformers (ViT)** : Plus récemment, les Vision Transformers ont gagné en popularité pour les tâches de vision par ordinateur. Les ViT **divisent une image en une série de patchs**, puis les transforment en une séquence linéaire de tokens grâce à une projection linéaire et un encodage positionnel (*Position Embeddings*), cette procédure est **similaire à la façon dont les mots sont traités dans les modèles de transformateurs textuels**. Ces tokens passent ensuite par des couches de transformateurs qui permettent au modèle de capturer les relations spatiales et contextuelles entre les différentes parties de l’image. La schématisation de ces étapes adaptées par les Vision Transformers (ViT) est indiquée dans la Figure 2.4.

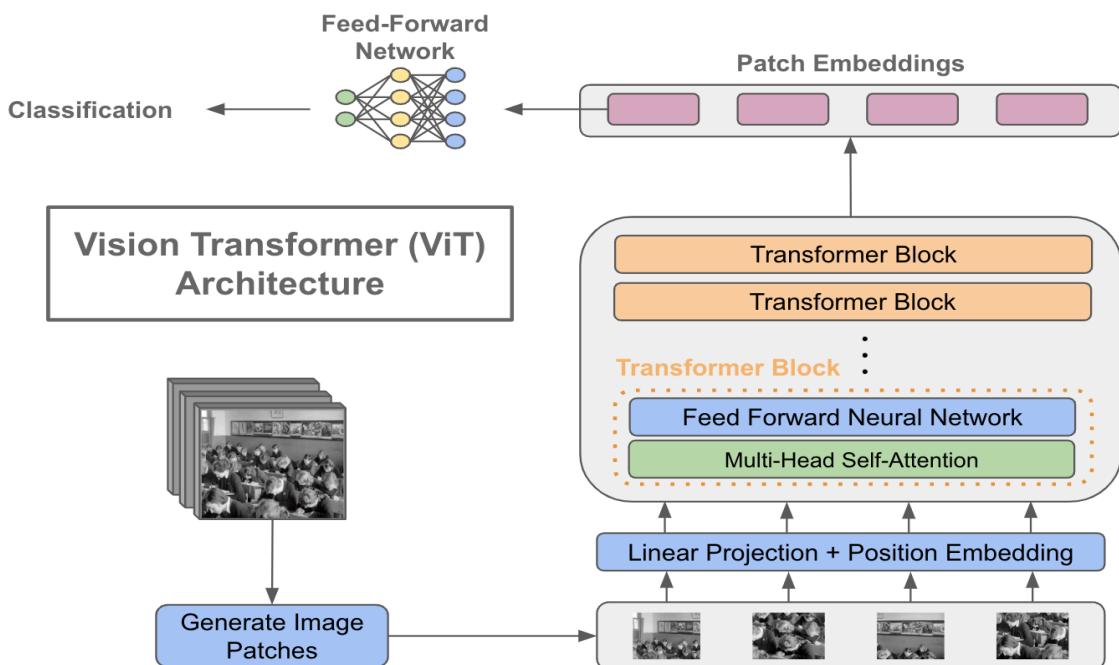


FIGURE 2.4 – Vision Transformers (ViT)

- **Fusion des Modèles Textuels et Visuels** : **GPT-4-Vision** combine ces modèles visuels (CNN ou ViT) avec un modèle de traitement du langage naturel (NLP) basé sur des transformateurs comme **GPT-4**. La fusion de ces deux types de modèles se fait généralement via une couche d’intégration ou une **couche multi-modale** qui permet d’*associer les représentations textuelles et visuelles dans un espace de caractéristiques commun*.

- **Entraînement Multi-Modal** : Pour que **GPT-4-Vision** puisse comprendre et générer du texte en fonction d'images, il est entraîné sur des ensembles de **données multi-modales qui contiennent des paires image-texte**.

Ce processus d'entraînement permet au modèle d'apprendre à associer les caractéristiques visuelles aux descriptions textuelles correspondantes.

## 2.2 Contributions

La présente section a introduit le contenu des documents *FCP*, les approches adoptées pour l'extraction du texte ainsi que pour le repérage et le traitement des images. Ensuite, nous avons présenté les indicateurs de performance clés (KPIs) sélectionnés et les performances des modèles de langage sur ces derniers.

### 2.2.1 Documents de départ

Forvia Core Procedures, connu en interne par l'abréviation *FCP Files*, désigne un ensemble de procédures standardisées et de meilleures pratiques mises en place par le groupe FORVIA, issu de la fusion de FAURECIA et HELLA.

Ces procédures forment la **constitution** sur laquelle le groupe se base pour fonctionner, tout en garantissant l'**efficacité**, la **cohérence** et la **qualité** dans l'ensemble des opérations mondiales du groupe. Elles couvrent divers domaines tels que la gestion de la qualité, la production, la logistique, les finances, les ressources humaines, etc.

Les *FCP* sont généralement documentées dans des manuels et des guides internes qui sont accessibles uniquement aux employés du groupe. Elles font également l'objet de formations régulières afin de garantir que tous les employés les connaissent et les appliquent correctement. L'adoption des *FCP* est un élément clé de la stratégie de transformation du groupe qui vise à devenir un leader mondial de la mobilité durable.

L'objectif des *FCP* est de :

- Harmoniser les pratiques au sein du groupe, en s'appuyant sur les meilleures pratiques issues des entités fusionnées.
- Améliorer l'efficacité des opérations en standardisant les processus et en éliminant les redondances.
- Renforcer la qualité des produits et services en mettant en place des contrôles qualité rigoureux.
- Réduire les coûts en optimisant les processus et en rationalisant les achats.
- Améliorer la communication et la collaboration entre les différentes entités du groupe.

Les documents de Forvia Core Procedures *FCP* considérés au début de ce projet, sont 103 fichiers PDF totalisant 3100 pages. La Figure 2.5 montre le nombre de pages par document.

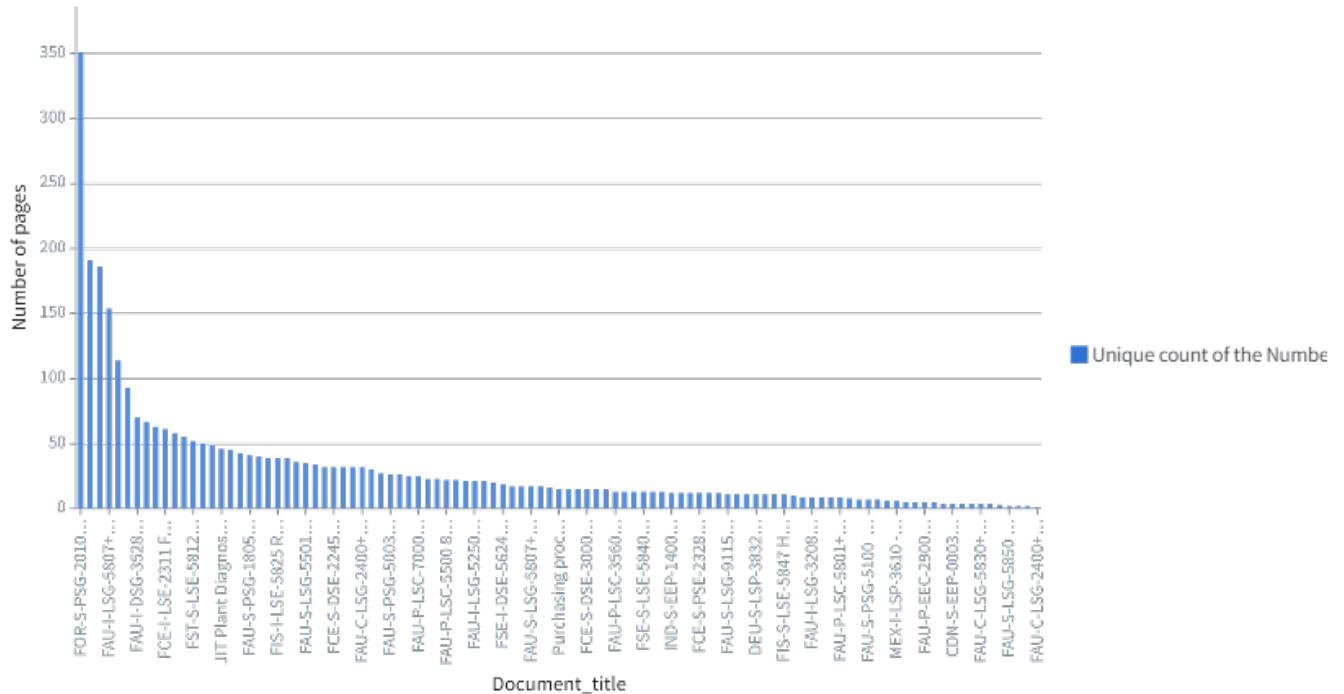


FIGURE 2.5 – Visualisation du nombre de pages par document

Ces fichiers PDF ont été intégrés dans Palantir en utilisant un ensemble de médias *Media Set*.

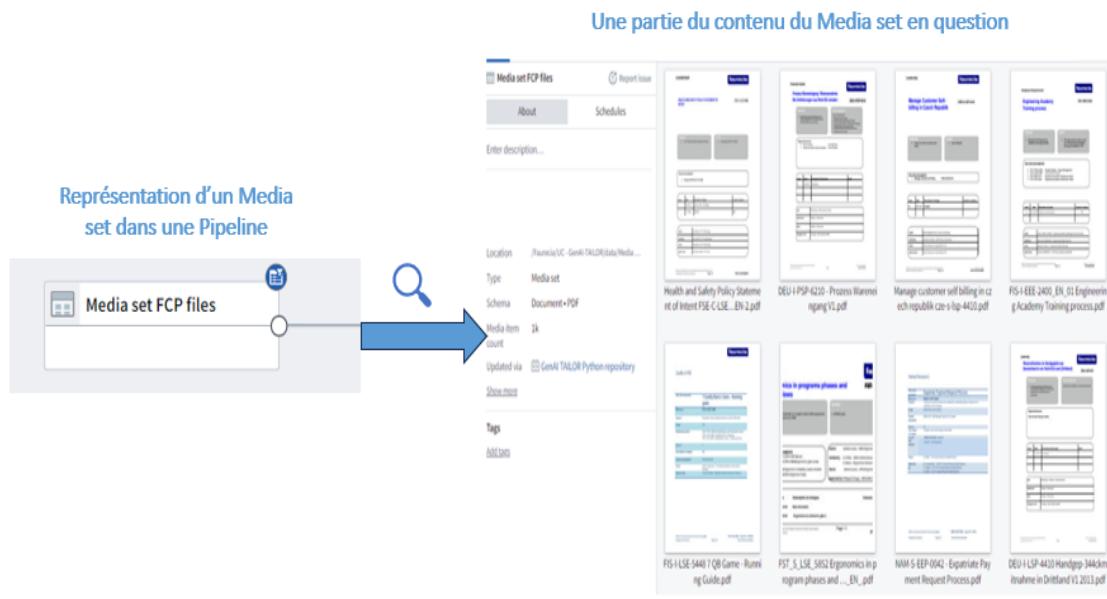


FIGURE 2.6 – Schéma explicatif d'un Media Set

Ce *Media Set* représenté dans la Figure 2.6 est une collection de fichiers multimédias partageant un schéma commun (des fichiers PDFs dans ce cas). Il est conçu pour gérer des données non structurées à grande échelle et à faciliter le traitement complexe d'éléments multimédias.

Parmi ces éléments, nous pouvons citer les fichiers audio, les images et les fichiers PDF. Cette solution permet l'accès à un stockage flexible, des optimisations de calcul et des transformations spécifiques au schéma pour améliorer les flux de travail et pipelines multimédias.

Cet ensemble de médias ou *Media Set* a été converti par la suite, en un ensemble de données incluant les *Métadonnées de base* sur chaque élément, qui sont dans notre cas : le chemin du document considéré (*path*), la référence multimédia (*Media item RID*) et l'élément multimédia (*Media item*).

Documents			Display media	Showing 300 of 1k rows	3 columns
	path String	mediaReference String	mediaItemRid String		
1	13MR_book_V2_dec_2011_fr FIS-S-LSE-5844.pdf		ri.mio.main.media-item.204df7ae-d107-453e-994d-dc909...		
2	2008 09 24 FAU-S-PSG-0667-EN-Boundary Samples.pdf		ri.mio.main.media-item.cc759d5b-87f1-4419-935d-83a39...		
3	5S Methodology FAU-S-PSG-5006-V4.pdf		ri.mio.main.media-item.2d5a5111-ce87-4c78-b918-bbf08...		
4	Assistance Agreement for Progress Actions FAU-S-LSG-404...		ri.mio.main.media-item.93bd9bc8-58ec-4908-b60b-0e710...		
5	BP tool User Guide_EN FAU-S-LSG-3233-EN-4.pdf		ri.mio.main.media-item.7a9118cb-965a-4f55-8d78-e9d6e...		
6	CDN-I-LSP-4000 Annual Supply Chain Security Verification...		ri.mio.main.media-item.335fe465-b1f9-4641-bb6c-31844f...		
7	CDN-I-LSP-4001 Supply Chain Security 5 Steps Risk Assess...		ri.mio.main.media-item.8eedef39-2d5b-4b2e-9e51-00930...		
8	CDN-I-LSP-4002 Supply Chain Security Incident Response...		ri.mio.main.media-item.a19a0367-199b-46c3-8ff3-c003eb...		

FIGURE 2.7 – Base de données contenant les Fichiers *FCP*

Les deux principales métadonnées d'un ensemble de médias ou *Media Set*, comme le montre la Figure 2.7, sont l'élément multimédia le *Media item* et la référence multimédia *Media item RID*, qui permettent de référencer un élément multimédia dans Foundry sans avoir à en faire des copies tout identifier la source de chaque document dans un format tabulaire.

Un élément multimédia (*Media item*) est un fichier unique au sein d'un ensemble de médias, qui peut être un fichier audio, image ou document, tel qu'un fichier WAV, PNG ou PDF.

Les références multimédias *Media item RIDs* peuvent être utilisées dans des ensembles de données pour associer des éléments multimédias à des métadonnées ou à d'autres informations dans un format tabulaire.

Les documents *FCP*, contiennent à la fois des données textuelles et sous formes d'images (éléments visuels). Comme on l'a déjà mentionné, les documents PDFs incluent aussi des images. La Figure 2.8 montre le pourcentage des fichiers contenant des images (à part le texte) et ceux qui contiennent du texte uniquement.

Pour avoir une idée plus claire sur les types d'éléments visuels qui vont être traités dans le cadre de ce projet, la Figure 2.9 montre les taux de chaque type d'éléments visuels présents, parmi la totalités des éléments.

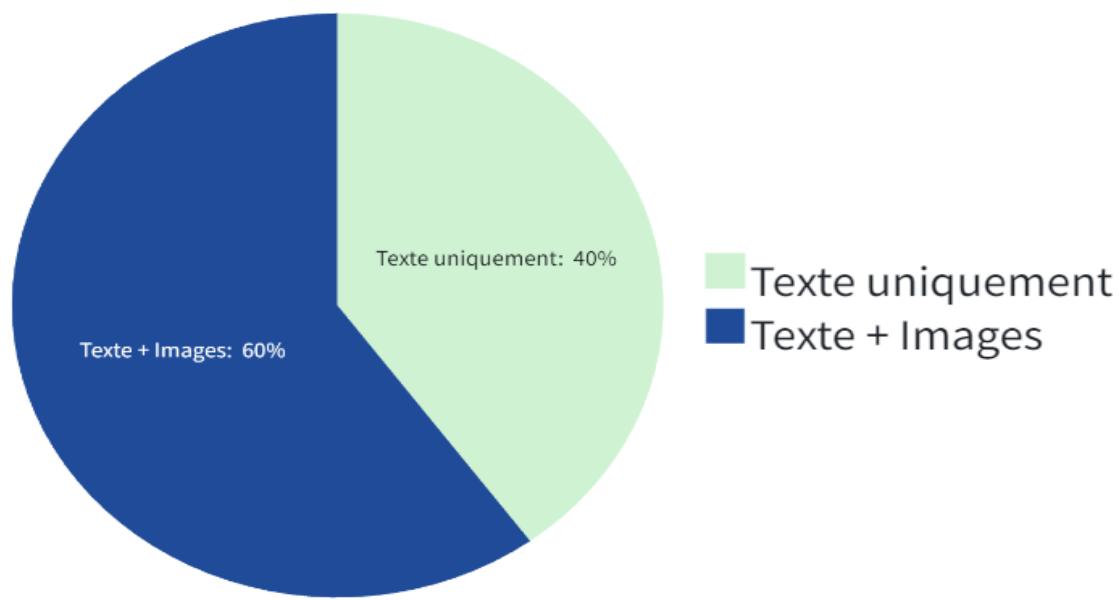


FIGURE 2.8 – Répartition du contenu textuel et visuel dans les documents PDFs

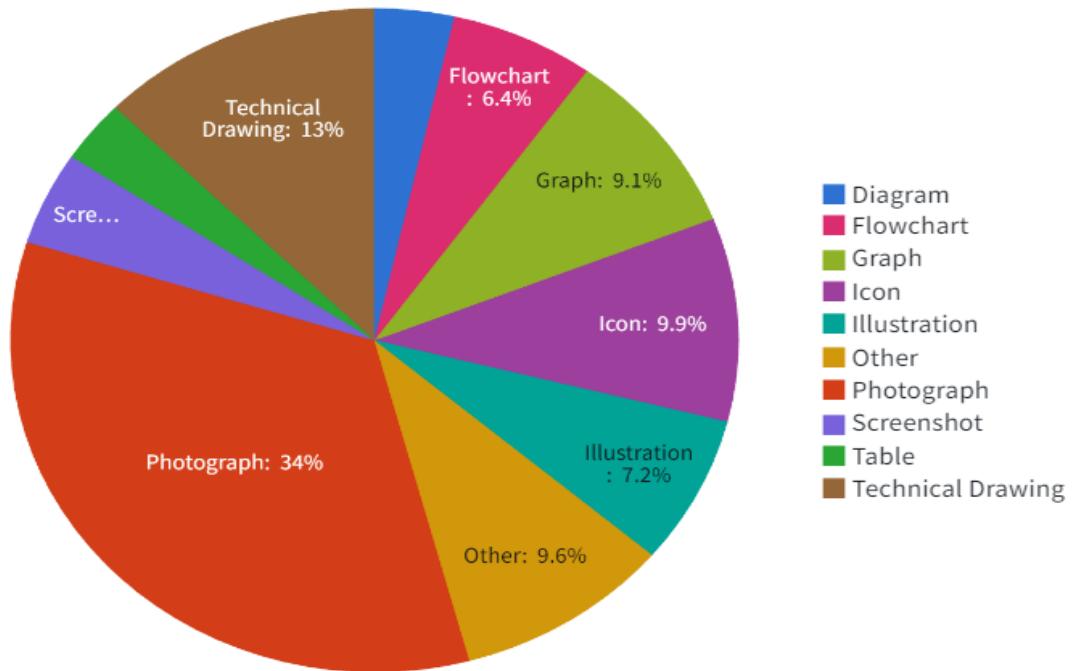


FIGURE 2.9 – Répartition des types des éléments visuels

La Figure 2.10 donne une idées sur les différents types d'éléments visuels présents dans les documents de *FCP*.

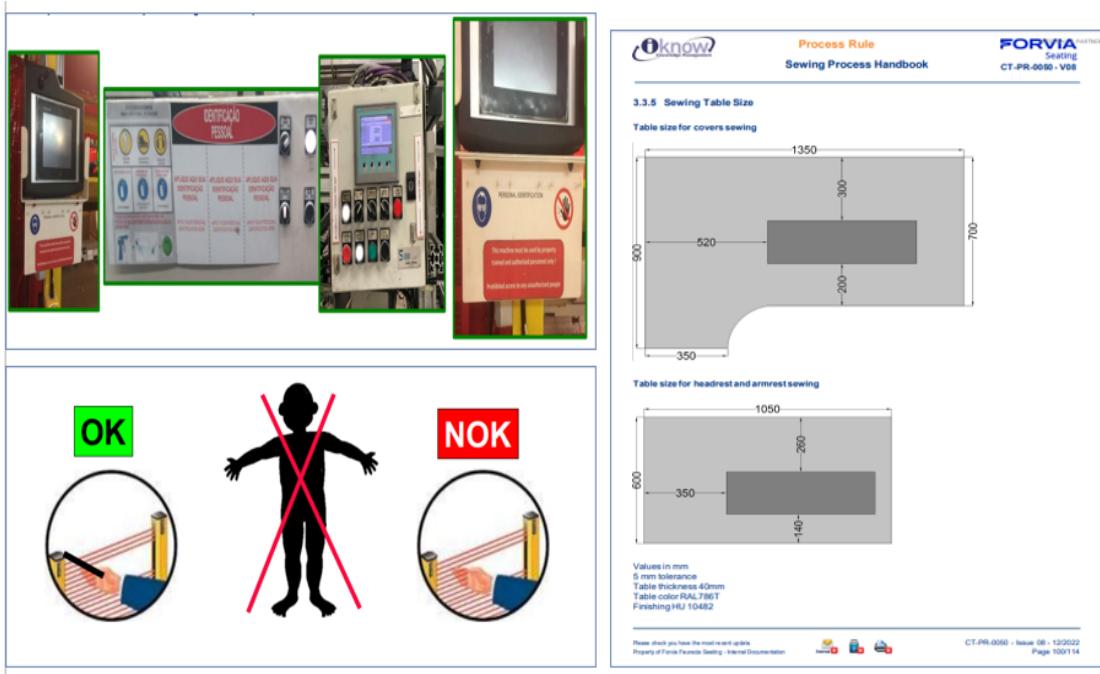


FIGURE 2.10 – Exemples d’éléments visuels présents dans les PDFs

## Extraction du texte

Au début, nous avons utilisé des bibliothèques telles que PDFplumber [pdfplumber, 2019] pour traiter les fichiers PDF. Cependant, nous avons rencontré des difficultés, notamment avec la reconnaissance des mises en page complexes comportant plusieurs colonnes. En effet, cette bibliothèque n’arrive pas à extraire le contenu de la première colonne, puis de la deuxième, dans un ordre chronologique.

Au lieu de cela, elle extrait le contenu de la première ligne entière, puis de la deuxième, sans identifier l’espace qui sépare chaque ligne en deux, qui montre que le document est organisé par colonnes et non par lignes entières. La Figure 2.11 illustre ces problèmes plus explicitement.

Pour résoudre ce problème, nous avons opté pour une approche basée sur la vision par ordinateur (*Computer Vision*) en convertissant chaque page en une image à l'aide de la bibliothèque pdf2image, puis en utilisant la reconnaissance optique de caractères (*Optical Character Recognition (OCR)*) pour extraire le texte des pages converties en images (PNG) à l'aide de Tesseract [Smith et al., 2009]. L'utilisation de cette bibliothèque permet de prendre en compte la mise en page en colonnes des pages en question et de suivre l'ordre des colonnes dans l'extraction.

Comme on peut le voir dans la Figure 2.12 ci-dessous, on a réussi à extraire le contenu de la colonne à gauche en premier, puis celle de gauche et enfin le contenu du footer, ce qui est nécessaire pour toute exploitation ultérieure de ce texte extrait.

Celà a conduit à la création d'un jeu de données composé de 16 lignes, et 8 colonnes, où chaque ligne contient les différentes caractéristiques étudiées pour un seul document PDF traité, les caractéristiques les plus importantes étant la *mediaReference* et le texte extrait de ce document.

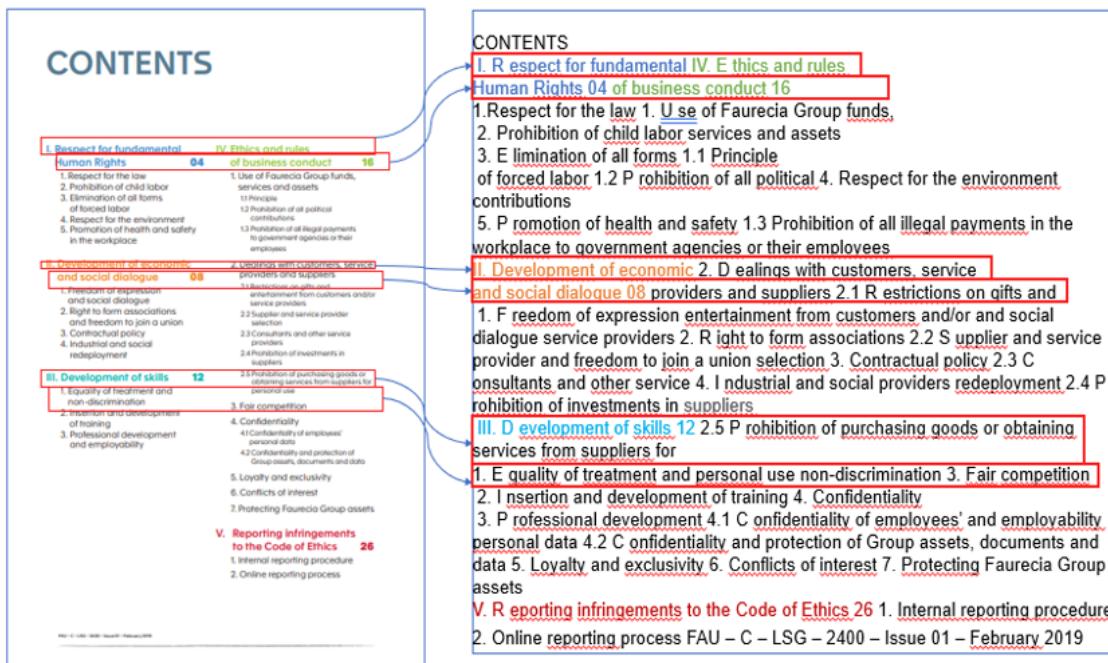


FIGURE 2.11 – Résultat de l'extraction du texte avec PDFplumber

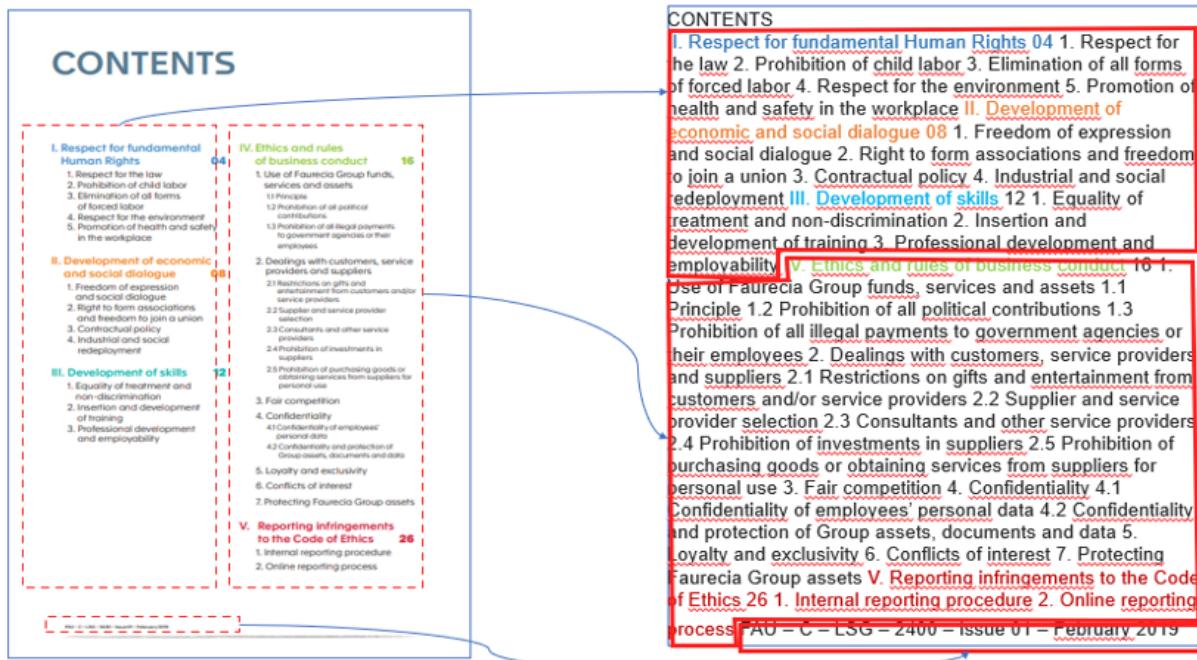


FIGURE 2.12 – Résultat de l'extraction de texte à partir d'un fichier PDF avec Tesseract

L'extraction du texte à partir des documents PDF (*Forvia Core Procedures*) a été réalisée grâce à trois transformations :

1. D'abord, l'extraction du texte par document avant de le stocker dans un ensemble de données initial. Une ligne correspond au texte de l'ensemble du document, chaque ligne étant un tableau de chaînes de caractères où chaque chaîne représente le contenu d'une page.

2. Ensuite, nous avons utilisé une transformation d'explosion pour obtenir, à partir de chaque ligne, plusieurs lignes où chaque ligne représente le texte d'une seule page.
3. Enfin, nous avons utilisé une clé primaire composée du nom du fichier, du numéro de la page et de l'orientation de la page (paysage ou portrait). Cette clé a été utile dans l'application de la génération augmentée par récupération (*RAG*).

La Figure 2.13 résume le processus de la *pipeline* d'extraction et de transformation du texte en ensembles de données exploitables.

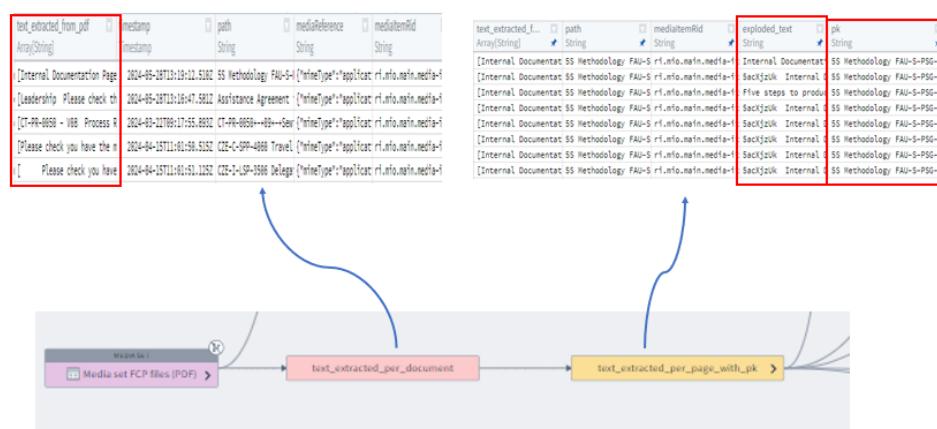


FIGURE 2.13 – Résumé du pipeline du processus de l'extraction du texte

La Figure 2.14 montre le résultat obtenu après l'extraction du texte à partir d'une page PDF et les transformations subies tout au long du processus d'extraction.

CONTENTS	
I. Respect for fundamental Human Rights	<b>04</b>
1. Respect for the law	
2. Prohibition of child labor	
3. Elimination of forms of forced labor	
4. Respect for the environment	
5. Promotion of health and safety in the workplace	
II. Development of economic and social dialogue	<b>08</b>
1. Freedom of expression and association	
2. Right to form associations and freedom to join a union	
3. Contractual policy	
4. Industrial and social redeployment	
III. Development of skills	<b>12</b>
1. Equality of treatment and non-discrimination	
2. Insertion and development of training	
3. Professional development and employability	
IV. Ethics and rules of business conduct	<b>16</b>
1. Use of Furecos Group funds, services and assets	
2. Transparency	
3. Promotion of all political parties	
4. Promotion of all legal payments to government agencies or their employees	
5. Dealing with customers, service providers and suppliers	
6. Ethical procurement from customers under normal circumstances	
7. Ethical procurement from service provider selection	
8. Ethical procurement from other service providers	
9. Ethical procurement of investments in suppliers	
10. Ethical distribution of purchasing goods or services from third parties for personal use	
11. Ethical procurement of Group assets, documents and assets	
12. Loyalty and exclusivity	
13. Confidentiality of unclassified personnel data	
14. Confidentiality of classified personnel data	
15. Protection of Furecos Group assets	
V. Reporting infringements to the Code of Ethics	<b>26</b>
1. Internal reporting procedure	
2. Online reporting process	

text_extracted_per_page_with_pk		
path	exploded_text	summary_pk
1 FAU-C-L5G-240#Code=1	FAU - C - L5G - empty atrr	empty atrr
2 FAU-C-L5G-240#Code=1 INTRODUCTION_Faurec	RESPECT FOR	empty atrr
3 FAU-C-L5G-240#Code=1 CONTENTS_I_Respect	RESPECT FOR	empty atrr
4 FAU-C-L5G-240#Code=1 RESPECT FOR	RESPECT FOR	empty atrr
5 FAU-C-L5G-240#Code=5 FAU - C - L5G - empty atrr	empty atrr	empty atrr
6 FAU-C-L5G-240#Code=6 FAU - C - L5G - empty atrr	empty atrr	empty atrr
7 FAU-C-L5G-240#Code=7 FAU - C - L5G - empty atrr	empty atrr	empty atrr
8 FAU-C-L5G-240#Code=8 FAU - C - L5G - empty atrr	empty atrr	empty atrr
9 FAU-C-L5G-240#Code=9 FAU - C - L5G - empty atrr	empty atrr	empty atrr
10 FAU-C-L5G-240#Code=10 FAU - C - L5G - empty atrr	empty atrr	empty atrr
11 FAU-C-L5G-240#Code=11 FAU - C - L5G - empty atrr	empty atrr	empty atrr
12 FAU-C-L5G-240#Code=12 DEVELOPMENT	empty atrr	empty atrr
13 FAU-C-L5G-240#Code=13 FAU - C - L5G - empty atrr	empty atrr	empty atrr
14 FAU-C-L5G-240#Code=14 FAU - C - L5G - empty atrr	empty atrr	empty atrr
15 FAU-C-L5G-240#Code=15 FAU - C - L5G - empty atrr	empty atrr	empty atrr
16 FAU-C-L5G-240#Code=16 TV ETHICS_AR empty atrr	empty atrr	empty atrr
17 FAU-C-L5G-240#Code=17 FAU - C - L5G - empty atrr	empty atrr	empty atrr
18 FAU-C-L5G-240#Code=18 FAU - C - L5G - empty atrr	empty atrr	empty atrr
19 FAU-C-L5G-240#Code=19 FAU - C - L5G - empty atrr	empty atrr	empty atrr
20 FAU-C-L5G-240#Code=20 FAU - C - L5G - empty atrr	empty atrr	empty atrr
21 FAU-C-L5G-240#Code=21 FAU - C - L5G - empty atrr	empty atrr	empty atrr
22 FAU-C-L5G-240#Code=22 FAU - C - L5G - empty atrr	empty atrr	empty atrr
23 FAU-C-L5G-240#Code=23 FAU - C - L5G - empty atrr	empty atrr	empty atrr
24 FAU-C-L5G-240#Code=24 FAU - C - L5G - empty atrr	empty atrr	empty atrr
25 FAU-C-L5G-240#Code=25 FAU - C - L5G - empty atrr	empty atrr	empty atrr
26 FAU-C-L5G-240#Code= V REPORTING INFR empty atrr	empty atrr	empty atrr

FIGURE 2.14 – Exemple d'un résultat de l'étape d'extraction du texte

## 2.2.2 Repérage et description d'images via GPT-4-Vision-Turbo

Pour obtenir des descriptions de chaque image présente dans chaque page des documents *FCP*, nous avions plusieurs options :

1. Nous pouvions utiliser des modèles de vision par ordinateur comme Detectron, capables d'extraire les images contenues dans chaque page. Cependant, nous ne pouvions pas le faire pour deux raisons principales :
  - Il n'était pas possible d'intégrer des frameworks open source externes dans Palantir.
  - L'objectif de ce projet était de tester les capacités des modèles utilisés dans Foundry, notamment **GPT-4-Vision-Turbo** pour le traitement des images.
2. Nous avons donc choisi une alternative permettant d'utiliser **GPT-4-Vision-Turbo** non seulement pour la description des images (une étape effectuée ultérieurement) mais d'abord pour détecter la présence d'éléments visuels dans chaque page et les décrire simultanément. Ce choix permet un recours unique à l'*API*, ce qui réduira les coûts pour l'entreprise.

Ce processus a été réalisé en deux étapes principales :

- La conversion de chaque page PDF en format JPEG, puis en **base64 encodé**<sup>2</sup>, car le modèle **GPT-4-Vision-Turbo** ne peut traiter les images que sous ce format.
- L'envoi d'une requête au modèle via l'*API* acquise par l'entreprise, demandant de déterminer si la page contient des éléments visuels et de fournir une description détaillée de chaque élément visuel présent, s'il existe.

La Figure A.1 présente une schématisation du processus de conversion des pages PDFs, suivi dans ce projet. Cette conversion, a été suivie par la création d'un prompt, demandant à **GPT-4-Vision-Turbo** de fournir une descriptions des images qui lui sont fournies. Le prompt utilisé est le suivant :

```
DETECT_AND_EXPLAIN = """
You'll be presented with a document page image that may include
various visual elements like illustrations, drawings, tables,
or graphs.
```

Your task is to examine these elements and provide detailed information following this format:

Determine if the image contains graphs and/or illustrations (respond with 'yes' or 'no' in lowercase).

If your return is 'yes', specify the general location of the element on the page (e.g., top right, centered bottom).

Describe the identified element. If your return is 'no', omit specifying the location and description of the element."""

On a détecté (grâce à un effort humain) que les descriptions fournies par le modèle ne prenait pas en considération le contexte dans lequel cette image est présente.

---

2. La **base64** est une méthode d'encodage qui permet de convertir des données binaires en une chaîne de caractères ASCII. Cela est particulièrement utile pour transmettre des données sur des supports qui traitent uniquement du texte, tels que les courriers électroniques ou les protocoles HTTP.

Comme on peut le distinguer sur l'exemple de la Figure 2.15, la première versions de la description fournie par le modèle **GPT-4-Vision-Turbo** est loin d'être pertinente<sup>3</sup>.

The figure shows two side-by-side screenshots. On the left is a blue-bordered box containing generated text from GPT-4-Vision-Turbo. On the right is a screenshot of the original document page from FORVIA.

**Generated Text (Left):**

```

yes

The illustrations are located at the top right and bottom right of the page.
The top right illustration is an icon of a face with safety glasses and a helmet, symbolizing personal protective equipment (PPE). The helmet portion of the illustration is in dark blue, with the safety glasses in a lighter blue color.
The bottom right illustration shows a pair of safety glasses and a welding helmet. The safety glasses are clear, placed in front of a vaguely outlined face, which demonstrates the fitting of the glasses on the face.
Below that, there's an image of a welding helmet worn on a head, positioned in a way to indicate that it is raised, showing that the wearer's eyes remain protected whether the helmet is raised or lowered.

```

**Original Document (Right):**

**Section Title:** 6. PERSONAL PROTECTIVE EQUIPMENT (PPE)

**Section Subtitle:** Safety Glasses

**Text Content:**

- Safety glasses must be defined by risk assessment. Safety glasses can be mandated either for a selection of tasks, of workstations, of GAPS, of UAPs, for the complete production area or a site or a Division.
- Vision glasses/ reading glasses are not safety glasses. Standard prescription Glasses are not deemed as Personal Protective Equipment and, in some circumstances, can even make an eye injury worse. If an employee normally wears prescription glasses prescribed by an optician or ophthalmologist, and his/her job is assessed as in need of prescription safety glasses. In these cases, it's necessary to overwear safety glasses or use safety glasses with lenses with optical correction.
- Welders should wear goggles or safety glasses with side shields under welding helmets.
  - Safety glasses under helmets are used to protect against flying metal, slag chips, grinding fragments, wire wheel bristles, and similar hazards, which can ricochet under the helmet.
  - Welder's eyes remain protected when the helmet is in the raised position.

**Image Area:** Includes three small icons (helmet, glasses, person wearing a helmet), a larger image of a welding helmet, and a photo of a person wearing a welding helmet.

**Page Footer:** Internal Documentation, Property of FORVIA France, Page 53/154, FAU-H-LSG-0007 - Issue 02 - 10/23, DLP - Data Loss Protection document, Library to ensure correct application, FORVIA feurecia

FIGURE 2.15 – Description superficielle des images dans une page

Pour surmonter ces limitations, nous avons adopté une technique avancée appelée *co-prompt engineering*.

Cette approche consiste à alimenter le modèle **GPT-4-Vision-Turbo** non seulement avec la page encodée en *base64*, mais également avec le texte extrait de la page en question. La totalité de ces informations est présente dans un ensemble de données intitulé "joined\_dataset" comme le montre la Figure 2.16.

Ceci permet au modèle de mieux comprendre le contexte dans lequel se trouve chaque image. Bien que cette méthode soit plus gourmande en temps et en ressources, elle permet à FORVIA d'obtenir des descriptions détaillées et contextualisées des éléments visuels, assurant ainsi leur pertinence pour des **utilisations futures**. Par exemple, ces descriptions serviront de base sur laquelle le modèle de langage (*LLM*) pourra s'appuyer pour répondre aux questions des utilisateurs concernant les images dans l'**application Workshop**, résultat de ce projet.

Malgré les progrès réalisés dans l'obtention de descriptions beaucoup plus contextualisées, ces descriptions étaient tellement détaillées qu'elles prenaient également en compte les logos et les filigranes de FORVIA ; chose qui allourait les descriptions obtenues sans aucune valeur ajoutée, comme indiqué dans la Figure 2.17 suivante.

3. Dans cet exemple de description, on ne fournit pas de contenu textuel de la page.



**DETECT\_AND\_EXPLAIN2 --**As an AI developed for Forvia,  
you'll be presented with a document page containing various elements like photographs, charts, plans, etc.,  
along with the document's title and extracted text.  
Your task is to analyze these elements and provide concise information.  
First, confirm if the page contains visual elements (respond with 'yes' or 'no').  
If 'yes', identify the image type (e.g., photograph, chart), its general location (e.g., top right),  
Describe the identified element in detail, focusing not only on the shape but also on the content.  
If it's a chart, describe the decisions/process mentioned. If it's a photograph describing a process,  
link between the image details (arrows, symbols, etc.) and the context.  
If your answer is 'no', omit specifying the location and description of the element.  
Use the following Forvia-related text to interpret the images in detail:"

FIGURE 2.16 – Résultat de l'application du *co-prompt engineering*

Yes, the page contains visual elements.  
The document page includes a chart that is central to the page, spanning from the top to the bottom. Above the chart, there's the FORVIA logo in the top right. To the left of the logo, there is the title of the document "FAU-P-LSC-7000 ALERT ON SENSITIVE SITUATIONS" in a bold, stylized font. Below this title, there is a horizontal section with various headings related to the strategic and operational aspects of the company: "Strategy," "Operational Systems," "PM and Sales," followed by subheadings "Leadership," "Human Resources," "Group Information Systems," "Manufacturing," "Purchasing," "Quality," "HSE," "Program Management," and "Sales."

The chart itself is divided into two main sections colored in yellow and red, indicative of different levels of alert or severity. Each section ("Yellow" for cautionary alerts and "Red" for critical alerts) lists specific scenarios under which a sensitive situation could be triggered within the context of building projects and real estate. This is detailed in bullet points titled from Y1 to Y7 for yellow alerts and R1 to R7 for red alerts, with each point outlining a type of issue, such as cost increases, project delays, legal issues, or risks to building integrity...

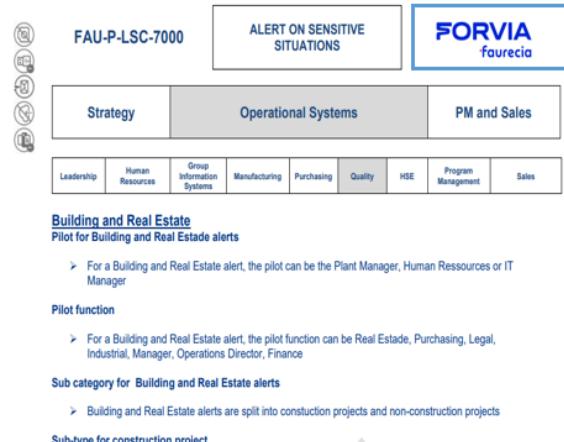


FIGURE 2.17 – Détection du logo dans la description des images

Ces éléments, présents sur chaque page, n'ont aucune utilité pour nous. Nous avons donc dû affiner d'avantage notre prompt pour ignorer ce type de contenu. La dernière version du prompt utilisé est la suivante :

DETECT\_AND\_EXPLAIN2 = As an AI developed for FORVIA, you'll be presented with a document page containing various elements like photography, charts, plans, tables, logo, icons, etc., along with the document's title and extracted text.

Your task is to analyze these elements and provide concise information.

If an element is {a logo or an icon, ignore it}, do not consider it as a visual element, do not mention its presence and omit specifying the location and description of that element.

First, confirm if the page contains visual elements (elements that are other than logos or icons), respond with 'yes' or 'no'.

If 'yes', identify the image type (e.g., photography, charts, plans, tables), its general location (e.g., top right), Describe the identified element in detail, focusing not only on the shape but also on the content.

If it's a chart, describe the decisions/process mentioned. If it's a photograph describing a process, link between the image details (arrows, symbols, etc.) and the context.

If your return is 'no', omit specifying the location and description of the element.

Use the following FORVIA-related text to interpret the images in detail.

La Figure 2.18 représente la descriptions obtenue pour la même image - indiquée dans la Figure 2.15 - mais dans sa nouvelle et dernière version obtenue après toutes les améliorations (*co-prompt engineering* et négligence des logos et éléments visuels non triviaux).

Le surlignement en jaune, montre comment le modèle *GPT-4-Vision-Turbo* a pu prendre en considération le contexte de la page, qui lui a été fourni grâce à la technique du *co-prompt engineering*.

- Yes, the page contains visual elements.
- There are two types of images present on this document page:
  - 1. **Iconography:** Top right corner of the page, which appears to symbolize "Safety Glasses," essential Personal Protective Equipment (PPE). The icon is a simplified representation of a pair of safety glasses, indicating the subject matter of the section in a glanceable format.
  - 2. **Photography:** Bottom right corner of the page, which features a person wearing a welding helmet and possibly additional safety equipment, although the face is blurred for privacy. Given the context of the document, which discusses the use of safety glasses or goggles in conjunction with welding helmets, this photograph appears to depict the correct application of PPE in a welding scenario:
    - The welder's helmet is displayed, which aligns with the text stating that "welders should wear goggles or safety glasses with side shields under welding helmets."
      - It serves as a real-life illustration of the guidelines, emphasizing the importance of protecting the eyes, as mentioned it remains protected even when the helmet is raised.
    - Both images serve to reinforce the document's focus on safety guidelines regarding eye protection in the workplace.

## 6. PERSONAL PROTECTIVE EQUIPMENT (PPE)

**Safety Glasses**




> Safety glasses must be defined by risk assessment. Safety glasses can be mandated either for a selection of tasks, of workstations, of GAPs, of UAPs, for the complete production area or a site or a Division.

> Vision glasses/ reading glasses are not safety glasses. Standard prescription Glasses are not deemed as Personal Protective Equipment and, in some circumstances, can even make an eye injury worse. If an employee normally wears prescription glasses prescribed by an optician or ophthalmologist, and his/her job is assessed as in need of prescription safety glasses. In these cases, it's necessary to overwear safety glasses or use safety glasses with lenses with optical correction.

> Welders should wear goggles or safety glasses with side shields under welding helmets.

- Safety glasses under helmets are used to protect against flying metal, slag chips, grinding fragments, wire wheel bristles, and similar hazards, which can ricochet under the helmet.
- Welder's eyes remain protected when the helmet is in the raised position.



Internal Documentation  
Property of FORVIA Faurecia  
Page 53/54  
The English master file is the official version of FCP documents. Any printed version of the document shall be compared with the latest revision published in the FCP Documentation Library to ensure correct application.

Faurecia SG-SG-907 - Issue 02 - 10/23  
DLP - Data Loss Protected document

**FORVIA**  
faurecia

FIGURE 2.18 – Amélioration des descriptions des images

## Encodage ou *Embeddings* des données obtenues

Comme mentionné précédemment, nous avons utilisé le modèle *text-embedding-ada-002* pour effectuer l'encodage des données. Ces données ne se limitent pas aux textes extraits des documents PDF, mais incluent également les descriptions textuelles des images détectées sur chaque page, concaténées dans une seul colonne, comme le montre la Figure 2.19.

La partie surlignée en gris indique le texte extrait et celle surlignée en bleu indique la partie contenant la descriptions des images.

The screenshot shows the 'Workshop GEN AI Tailor' interface. On the left, there's a sidebar with 'Back home', 'Workshop GEN AI Tailor' (549 objects), 'Overview' (selected), 'Properties' (8), 'Security', 'Datasources' (Index failed), 'Capabilities', and 'Materializations'. The main area shows a document page with the following content:

Internal Documentation Page 15/39 FAU-I-LSG-5251 – Issue 10 – 11/22  
Property of FORVIA Faurecia DLP – Data Loss Protected document  
The English master file is the official version of FCP documents. Any printed version of this document shall be compared with the latest revision published in the FCP Documentation Library to ensure correct application.  
9.2. Basic Requirements  
Fire resistant noncombustible materials must be used to construct the room.  
Plasterboard/gypsum plates are not supported. Plastic materials, including those of fire-retardant composition, must not be used because they can produce large quantities of smoke in case of a fire.  
All open or rough surfaces must be sealed / painted using plaster to prevent dust or chemically active particles being distributed because of the constant air flow.  
The use of foam to seal openings is forbidden.  
(truncated text)  
"Yes, there is a photograph located towards the bottom right of the page."  
The photograph depicts a set of double doors, likely illustrating the type of fire-rated and airtight doors mentioned in the text. These doors appear sturdy and possibly made of metal, consistent with the requirements outlined in section 9.3 titled "Doors." The doors in the image are designed to meet the minimum dimensions specified for datacenter room doors, with solid panels and potentially a locking mechanism visible, indicating protection against burglary as described. There are no visible handles on the outside, which aligns with the requirement for a fixed doorknob outside and a door handle inside IT system room doors for security reasons.  
(truncated text)

FIGURE 2.19 – Colonnes obtenues pour chaque page (Texte + Description des images)

L'ensemble de ces données est ensuite encodé pour créer l'**objet final** (*Object Type*) intitulé Workshop GenAI Tailor, contenant ainsi, non seulement une colonne totalisant les informations obtenues sur chaque page (texte et description des images) dans la colonne nommée *data\_per\_page* mais aussi leurs représentations vectorielles obtenues grâce à l'étape de l'encodage (*Embeddings*) dans une colonne nommée *embedded\_data\_per\_page* (dont le titre est encadré en vert dans la Figure 2.19.)

**Un Object Type** est analogue dans l'ontologie de Palantir Foundry (La plateforme avec laquelle travaille l'entreprise Forvia afin de réaliser sa transformation digitale) à un ensemble de données, tandis que la définition d'un objet est analogue à celle d'une ligne dans l'ensemble de données.

### 2.2.3 Evaluation des Grands Modèles de Langages (*LLMs*)

Concernant l'évaluation et le *benchmarking* des grands modèles de langage (*LLMs*), nous avons commencé par définir les tâches sur lesquels on va évaluer les performances des (*LLMs*) considérés dans le cadre de ce projet, qui sont **GPT-4-32k** et **GPT-4o**.

Fixer ces tâches était une étape cruciale et très importante, car c'est la base sur laquelle se posent toutes les étapes suivantes du *Benchmarking*.

Pour nous, l'extraction d'informations, la reconnaissance d'entités nommées (*Named Entity Recognition*) et la réponse aux questions étaient les tâches les plus ciblées, car elles constituaient des éléments clés pour pouvoir réaliser la génération augmentée par récupération (*Retrieved Augmented Generation*).

Nous avons d'abord élaboré un ensemble de données contenant de 1 à 5 questions par document, couvrant 20 % des fichiers à évaluer.

Ensuite, nous avons généré des réponses à ces questions avec **un effort humain**, assisté par **Copilot** [Microsoft, 2024a].

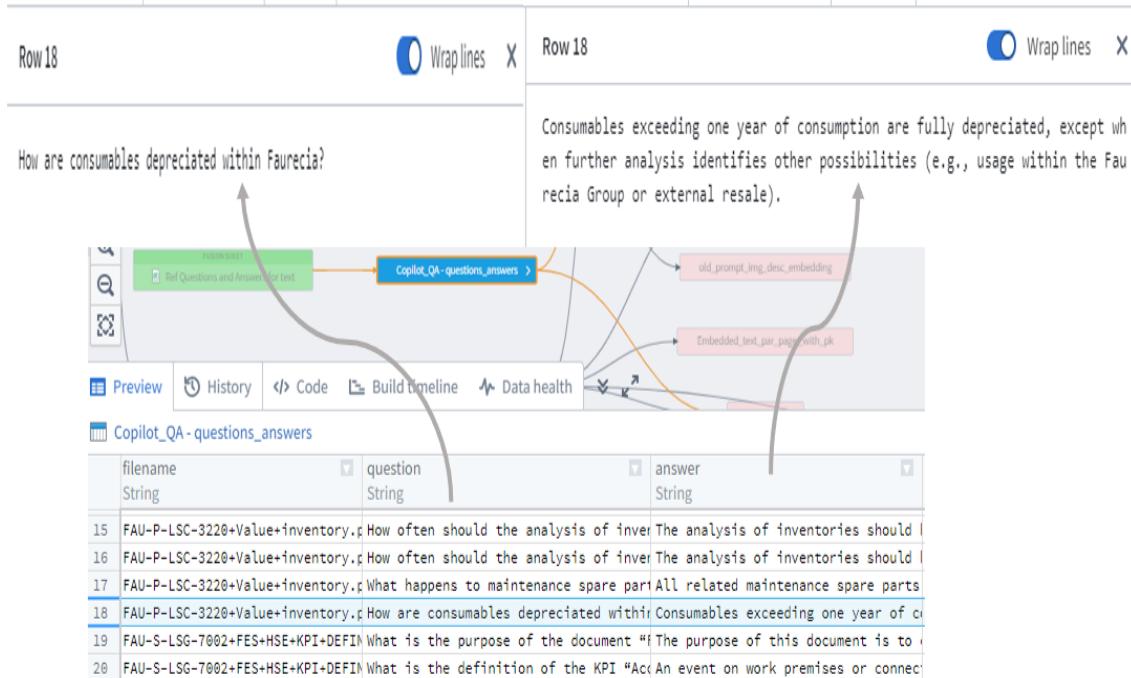


FIGURE 2.20 – Ensemble de données contenant les questions et les réponses de références

L'ensemble de données contenant les questions et les réponses est indiqué dans la Figure 2.20. Ces réponses ont été utilisées comme base ou référence, à laquelle on avait comparé les réponses générées par **GPT-4-32k** et **GPT-4o** sur les mêmes questions. Cette comparaison s'est basée sur des indicateurs clés de performance, qu'on va aborder dans la section suivante.

Les réponses à chaque question étaient formulées et synthétisées par les modèles de langage tout en appliquant la génération augmentée par récupération (*RAG*).

En effet, pour chaque question, nous avons spécifié le texte où chercher les informations nécessaires à la génération de la réponse (dans le prompt).

La Figure 2.21, ci-dessous schématise le processus de la génération augmentée par récupération.

L'indication du texte de référence a été réalisée grâce à une indexation prédéfinie via la **clé primaire** créée pour chaque texte extrait par page (comme mentionné précédemment).

Le texte, la clé primaire et les questions étaient tous présents dans le **même ensemble de données**, comme indiqué dans la Figure 2.22 étant donné que l'API du modèle ne permettait pas de recherche extérieure dans une source externe.

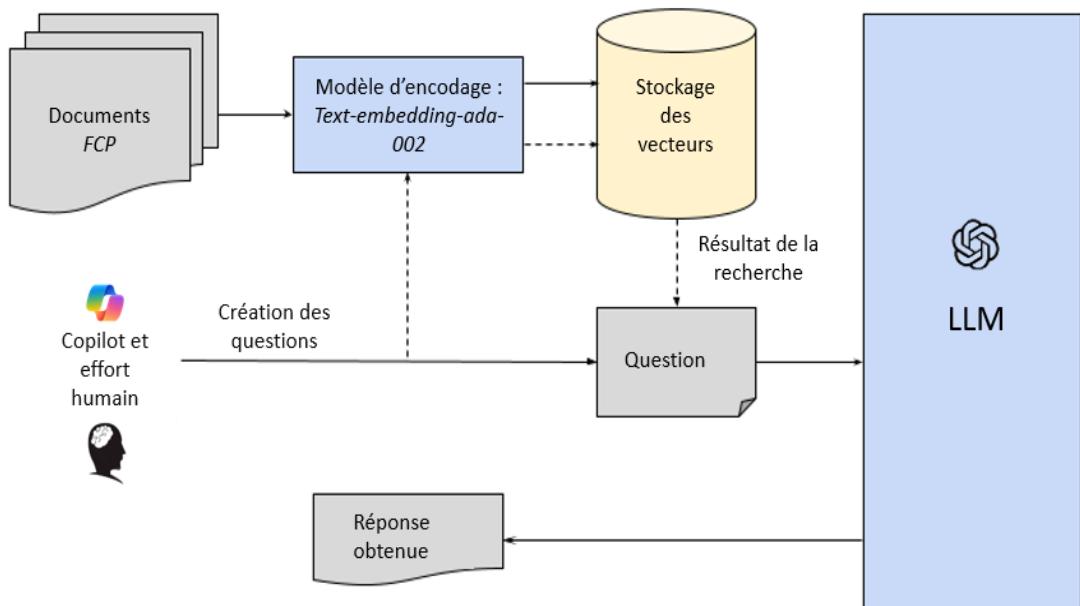


FIGURE 2.21 – Génération augmentée par récupération (*RAG*)

question	primary_key	join_exploded_text	filename
What is the primary	FAU-C-LSG-2400+Code+Of+Ethi	11 • FAU - C - LSG - 2400 - Issu FAU-C-LSG-2400+Code+Of+Ethics	
What is the maximum	FAU-C-LSG-2400+Code+Of+Ethi	19 • FAU - C - LSG - 2400 - Issu FAU-C-LSG-2400+Code+Of+Ethics	
How should employees	FAU-C-LSG-2400+Code+Of+Ethi	21 • FAU - C - LSG - 2400 - Issu FAU-C-LSG-2400+Code+Of+Ethics	
How does Faurecia ha	FAU-C-LSG-2400+Code+Of+Ethi	24 • FAU - C - LSG - 2400 - Issu FAU-C-LSG-2400+Code+Of+Ethics	
What is the primary	FAU-C-LSG-2400+Code+Of+Ethi	24 • FAU - C - LSG - 2400 - Issu FAU-C-LSG-2400+Code+Of+Ethics	

FIGURE 2.22 – Question avec son texte de référence

Cependant, il existe des modèles et des plateformes qui permettent des recherches externes pour enrichir les réponses générées. Par exemple, Azure Cognitive Services propose des fonctionnalités avancées de recherche à travers son API Bing Search. Cette API permet aux modèles de langage de chercher des informations en temps réel sur le web pour fournir des réponses plus complètes et précises.

De même, des plateformes comme OpenAI avec leur API **GPT-4** permettent l'intégration de *plugins* de recherche qui peuvent effectuer des requêtes externes. Ces capacités ne sont malheureusement pas disponibles sur la plateforme Foundry de Palantir.

### Evaluation sur des questions basées sur les images et Application de la *RAG*

En suivant un processus similaire à celui mentionné dans la section précédente, nous avons effectué le *benchmarking* ou l'évaluation des grands modèles de langage dans le traitement des images. Cependant, cette évaluation des images s'est avérée plus complexe et chronophage que celle du texte.

En effet, les questions créées sur les images n'ont pas bénéficié de l'aide de Copilot, ce dernier étant incapable de traiter des schémas complexes et étroitement liés au contexte spécifique des données de FORVIA, comme par exemple le diagramme indiqué dans la Figure 2.23.

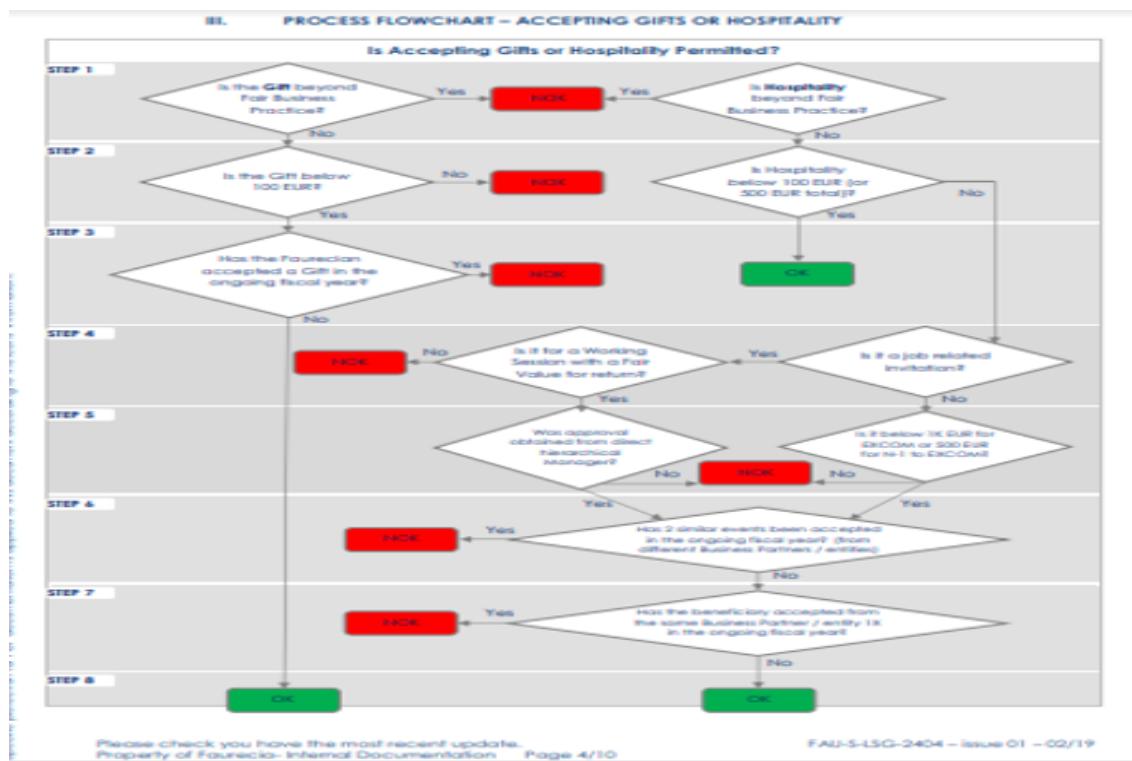


FIGURE 2.23 – Exemple de diagramme dont Copilot (IA) n'est pas capable d'interpréter

Cependant, nous avons pu créer un ensemble de données contenant une cinquantaine de questions, comme indiqué dans la Figure 2.24 nous permettant de réaliser un *Benchmarking*, vu que les questions couvrent 20 % de la totalité des documents PDFs considérés dans ce projet.

Nous avons ensuite demandé au Grand Modèle de Langage (*LLM*) de répondre aux questions, cette

fois en ayant recours à la description des images, qui est elle même contextualisée grâce à la technique de l'ingénierie de co-suggestion (*co-prompt engineering*).

The screenshot shows a Foundry workspace interface. At the top, there is a pipeline diagram with nodes: 'final\_img\_question\_answered' (yellow), 'img\_benchmark' (yellow), 'answers\_about\_image\_embedded' (orange), and 'img\_c' (yellow). Below the pipeline is a table titled 'answers\_about\_image\_embedded'. The table has columns: question (String), answer (String), pk (String), final\_img\_descri... (String), answers\_img\_llm (String), ref\_answer\_embed... (Float), and embedded\_answer... (Array[Float]). The table contains four rows of data. Below the table, three callout boxes provide annotations:

- Row 7:** What is the central red area labeled as? (Annotation: The central red area is labeled as "COOL LAN 3.")
- Row 7:** What is the central red area labeled as? (Annotation: Yes, the page contains visual elements. The image in question appears to be a series of technical diagrams or plans, located throughout the document page. These diagrams are likely meant to detail specifications for a workspace setup, including measurements and placements of technical equipment related to FORVIA Faurecia.)
- Row 7:** What is the central red area labeled as? (Annotation: In the top part of the image, there's a horizontal diagram detailing the arrangement of a LAN cabinet and cable trays within a specified space, which is outlined with dimensions such as "2,8 m inside without LAN cabinet" and "3,6m inside." It shows the orientation of crash barriers and the location of the LAN Cabinet I regarding these barriers. The diagram uses colors and labels to highlight features such as the "COOL LAN 3" area and includes measurements and symbols to denote specific components like the front side, service side, and the directions in which the LAN cabinet can be moved.)

FIGURE 2.24 – Question en rapport avec les images avec la description fournie au modèle

**Fixations des approches adaptées pour le calcul des (KPIs)** L'indicateur de performance clé considéré dans ce projet était la similarité sémantique (*semantic similarity*).

Nous cherchions à déterminer si les modèles de langage intégrés dans Foundry indiqués dans la Figure 2.25, étaient capables de comprendre le texte de référence proposé via la *RAG*.

Pour ce faire, nous comparons leurs réponses à celles générées humainement avec l'aide d'une IA, ces dernières étant considérées comme référence.

Les modèles **GPT-3.5**, **GPT-3.5-16k**, **GPT-4** et **GPT-4 Turbo** ont été déjà le sujet d'autres projets, notre tâche était d'évaluer seulement les modèles nouvellement intégrés dans Foundry à savoir **GPT-4-32k** et **GPT-4o**, comme indiqué dans la Figure 2.26

**Model Catalog**

**Completion**

Completion models are pre-trained language models that generate text completions based on a given input prompt. They can understand and generate human-like text, which can be used for tasks like summarization, question-answering, and semantic search.

Model Type	Count
Completion	6
Embedding	1
Vision	1

Model Creator	Count
OpenAI	8

**GPT-3.5**  
A cheaper and faster alternative to GPT-4, with a 4k-token context window. Consider for high-volume use-cases, where cost is a concern. Supports natural...

**GPT-3.5 (16k)**  
A cheaper and faster alternative to GPT-4, with a 16k-token context window. Consider for high-volume use-cases, where cost is a concern. Supports natural...

**GPT-4**  
Recommended for general use in AIP Logic. Has an 8k-token context window. Supports natural language prompting in multiple languages, as well as tool use a...

**GPT-4 Turbo**  
Recommended for general use in AIP Logic. Has a 128k-token context window, and lower cost, compared to GPT-4. Supports natural language prompting in multip...

**GPT-4 (32k)**  
Recommended for use in AIP Logic where GPT-4's 8k-token context window is insufficient, e.g. when retrieving and working with long text passages. Suppo...

**GPT-4o**  
GPT-4o is OpenAI's most advanced flagship model. It is multimodal (text and vision), has 128K tokens context window, and is cheaper and faster than GPT-4 Turbo.

FIGURE 2.25 – Catalogue des Modèles de langages disponibles sur Foundry

**Completion**

Completion models are pre-trained language models that generate text completions based on a given input prompt. They can understand and generate human-like text, which can be used for tasks like summarization, question-answering, and semantic search.

Model Type	Count
Completion	6
Embedding	1
Vision	1

Model Creator	Count
OpenAI	8

**GPT-3.5**  
A cheaper and faster alternative to GPT-4, with a 4k-token context window. Consider for high-volume use-cases, where cost is a concern. Supports natural...

**GPT-3.5 (16k)**  
A cheaper and faster alternative to GPT-4, with a 16k-token context window. Consider for high-volume use-cases, where cost is a concern. Supports natural...

**GPT-4**  
Recommended for general use in AIP Logic. Has an 8k-token context window. Supports natural language prompting in multiple languages, as well as tool use a...

**GPT-4 (32k)**  
Recommended for use in AIP Logic where GPT-4's 8k-token context window is insufficient, e.g. when retrieving and working with long text passages. Suppo...

**GPT-4o**  
GPT-4o is OpenAI's most advanced flagship model. It is multimodal (text and vision), has 128K tokens context window, and is cheaper and faster than GPT-4 Turbo.

FIGURE 2.26 – Modèles pris en considération dans ce projet

Cette évaluation a été effectuée en calculant la similarité sémantique entre les réponses de Copilot (considérées comme références) et celles fournies par ces deux *LLMs*.

Cette similarité sémantique a été mesurée selon deux approches distinctes : le calcul de la similarité cosinus et l'évaluation par un autre *LLM* qui est **GPT-3.5**.

1. **Calcul de la similarité cosinus** : Nous avons utilisé une méthode mathématique pour appliquer cette formule sur les encodages (*embeddings*) de chaque réponse. La formule de la similarité cosinus est donnée par :

$$\text{similarité cosinus} = \frac{A \cdot B}{\|A\| \|B\|} \quad (2.1)$$

$$= \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (2.2)$$

où  $A$  et  $B$  sont les vecteurs *d'embeddings* des deux réponses.

2. **Évaluation par un autre *LLM*** : Nous avons demandé à un autre modèle de langage (**GPT-3.5**) de fournir un score de similarité en comparant les deux réponses en texte brut, sans aucun prétraitement dans le but d'évaluer le *LLM* de manière complète.

Voici le prompt utilisé pour réaliser cette approche :

```
semantic_similarity_PROMPT =
Act as a specialist in semantic analysis.
You'll be presented with two different responses provided by two different LLMs for
a given question.
Your task is to measure the semantic similarity between the two responses .
Give a singular score between 0 and 1 to quantify this semantic relatedness ; 0 means
not close at all, and 1 means they almost have identical meaning.
Explain the rate you give briefly.
Follow this layout when giving the output :
"semantic_similarity: <value between 0 and 1>. Explanation: <what is the reasoning>".
```

L'adaptation de ces deux approches a été faite après une recherche dans la littérature qui a montré que **GPT-3.5** peut surpasser la similarité cosinus sur certains tests de référence de similarité sémantique. C'est le cas de l'étude SemScore [Maharana et al., 2022].

Cependant, d'autres études ont montré que la similarité cosinus pouvait être tout aussi efficace que **GPT-3.5** sur d'autres tests. Par exemple, une étude de [Wang et al., 2023] a comparé les performances de **GPT-3.5** et de la similarité cosinus sur diverses tâches et a constaté que les deux méthodes obtenaient des résultats globalement similaires.

En outre, la meilleure méthode pour mesurer la similarité sémantique dépendra des besoins spécifiques de la tâche et des ressources disponibles.

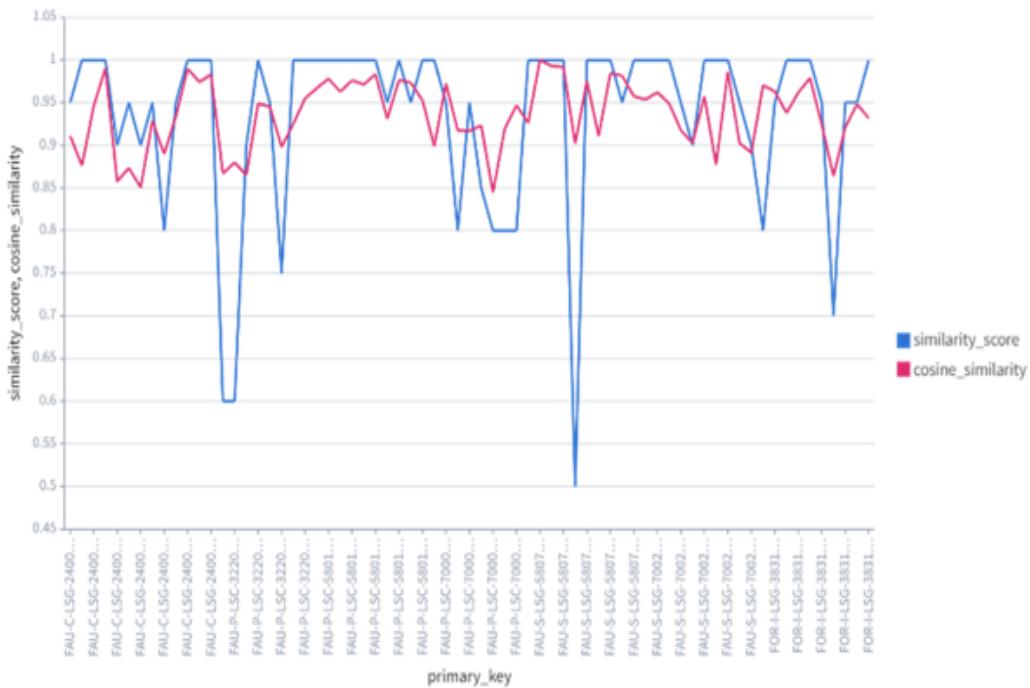
## 2.2.4 Calcul des KPIs

**Performances de GPT-4-32k** Les valeurs prises par les deux indicateurs clés de performances à savoir, la similarité cosinus (*cosine\_similarity*) et le score de similarité sémantique généré par **GPT-3.5**, mesurant la similarité sémantique entre les réponses de référence et celle générée par **GPT-4-32k**, pour une question donnée, sont illustrés dans la Figure 2.27 où chaque clé primaire (*primary\_key*), dans l'axe des abscisses est utilisée pour indexer les questions.

Nous avons obtenu des scores de similarité sémantique élevés, indiquant que les réponses de **GPT-4-32k** sont très proches des références.

Les deux scores varient généralement entre 0.8 et 1, à l'exception de quelques situations où un écart important entre les deux mesures a été observé.

Après investigation, nous avons constaté que ces questions étaient de type ouvert (*open-ended questions*), ce qui a provoqué des hallucinations chez le modèle de langage (**GPT-4-32k** dans ce cas). Ces hallucinations ont entraîné des réponses légèrement différentes sémantiquement des références, causant cet écart.



**Performances de GPT-4o** Lors de la mesure des mêmes indicateurs de performance clés, fournis par **GPT-4o** [open ai, 2024] cette fois-ci (qui est le modèle de langage le plus récent - sorti le 13 mai 2024 - et considéré comme le plus avancé de OpenAI), comme illustré par la Figure 2.28.

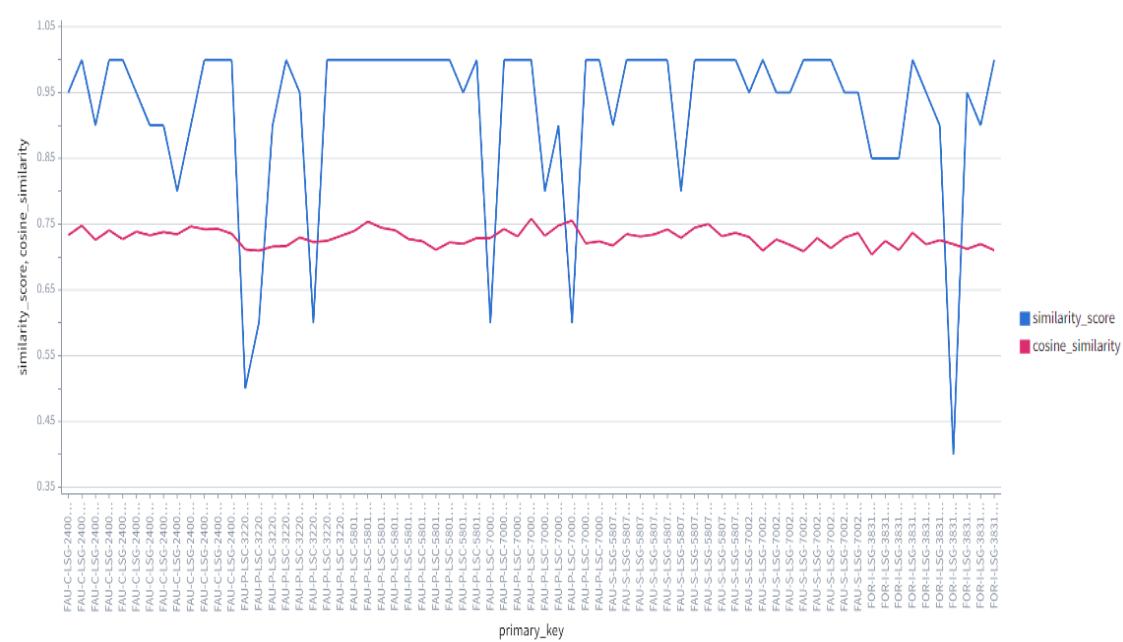


FIGURE 2.28 – Similarité cosinus (calculée) versus le score de similarité pour GPT-4o

Nous avons observé deux points principaux :

- La similarité cosinus entre les réponses de référence et celles fournies par **GPT-4o** est significativement inférieure à celle obtenue avec **GPT-4-32k**. En effet, la similarité cosinus de **GPT-4o** se situe entre 0,7 et 0,8, tandis que celle de **GPT-4-32k** varie entre 0,85 et 1.
  - L'écart entre la similarité cosinus, et le score de similarité (fourni par le Grand Modèle de Langage **GPT-3.5** utilisé comme juge) est beaucoup plus important pour **GPT-4o** que pour **GPT-4-32k**.

Cet écart moyen de 0,25, bien que les deux scores soient cohérents dans leur direction, est intrigant et mérite une investigation approfondie.

En effet le score de similarité sémantique (*Semantic Similarity Score*) donné par le modèle **GPT-3.5**, est subjectif et basé sur l'évaluation contextuelle et sémantique. Il prend en compte non seulement la similarité lexicale, mais aussi la similarité contextuelle et conceptuelle.

Tandis que, la similarité cosinus (*Cosine Similarity*) est une mesure mathématique connue. Elle est basée sur une comparaison vectorielle des représentations des phrases, qui ne capte pas nécessairement les nuances contextuelles et sémantiques de manière aussi sophistiquée que **GPT-3.5**. Pour mieux comprendre la différence, prenons cet exemple hypothétique en considération :

1. Phrase 1 : "The cat is on the mat."
  2. Phrase 2 : "The feline rests on the rug."

Cosine Similarity pourrait donner un score élevé car les mots "cat" et "feline", "mat" et "rug" sont similaires dans un espace vectoriel. Tandis que le score de similarité sémantique (*Semantic Si-*

*milarity score*) donné par **GPT-3.5** pourrait donner un score plus nuancé avec une explication, par exemple :

"semantic\_similarity: 0.8. Explanation: Both sentences convey a similar idea of a cat resting on a surface, but use different terms and imply slightly different contexts.".

OpenAI, quant à elle, elle ne donne pas un accès direct aux utilisateurs pour voir la méthode adaptée pour générer ce score de similarité. Mais les approches mentionnées dans la littérature sont :

- Apprentissage supervisé : Le modèle est entraîné sur un ensemble de données de paires de phrases étiquetées avec des scores de similarité sémantique. Le modèle apprendrait alors à mapper les paires de phrases à des scores de similarité en fonction des modèles observés dans les données d'entraînement.
- Apprentissage par renforcement : Le modèle est entraîné à recevoir des récompenses pour générer des scores de similarité précis. Le modèle apprendrait par essais et erreurs à générer des scores qui correspondent mieux aux évaluations humaines de la similarité sémantique.
- Métriques basées sur l'attention : Le modèle utilise des mécanismes d'attention pour identifier les parties les plus pertinentes de chaque phrase lors du calcul du score de similarité. Cela permettrait au modèle de se concentrer sur les informations les plus importantes pour la similarité sémantique.
- Intégration de connaissances externes : Le modèle intégrer des connaissances externes, telles que des bases de connaissances ou des dictionnaires de synonymes, pour enrichir la représentation sémantique des phrases (Application de la génération augmentée par récupération (*RAG*)).

Pour conclure, certes, un score de similarité sémantique élevé, généré par un grand modèle de langage (**GPT-3.5** dans notre cas) est considéré comme un indicateur positif. Cependant, les modèles de langages sont sujets à des hallucinations, par conséquent le score de similarité (généré par **GPT-3.5**) soit être proche du score de la similarité cosinus pour assurer sa fiabilité.

Dans le cadre de notre évaluation (*benchmark*), la vérité absolue (*ground truth*) est fournie par un ensemble de données de référence préparées avec Copilot et validées humainement.

L'écart important entre la similarité cosinus et le score de similarité sémantique indique que les réponses de **GPT-4o** sont simplement plus verbeuses que celles générées par **GPT-4-32k**.

Par conséquent, vis-à-vis de notre évaluation (*benchmark*), il n'est pas possible de conclure que **GPT-4o** est meilleur, puisqu'on cherche des réponses concises.

Néanmoins, pour des questions nécessitant plus de détails et de contexte, on peut admettre que **GPT-4-32k** est probablement supérieur.

A ce stade il faudrait regarder s'il ne faut pas un ensemble de données de références, contenant des questions-réponses plus poussées.

On a donc décidé d'utiliser **GPT-4-32k** pour continuer notre étude.

**Aperçu plus généralisé sur les indicateurs de performance clés (KPIs)** Comme on l'a déjà mentionné dans la sous-section 2.2.4 précédente, le Modèle de Langage le plus adapté à notre projet et qui répond le plus à nos attentes est **GPT-4-32k**, c'est pour cette raison qu'on n'a fait des généralisations que sur ce modèle ( **GPT-4o** n'a pas été considéré). La Figure 2.29, présente les moyennes par document, des scores de similarité sémantique (*cosine similarity* et *similarity\_score*) calculés sur les réponses générées par **GPT-4-32k** aux questions posées sur le contenu textuel.

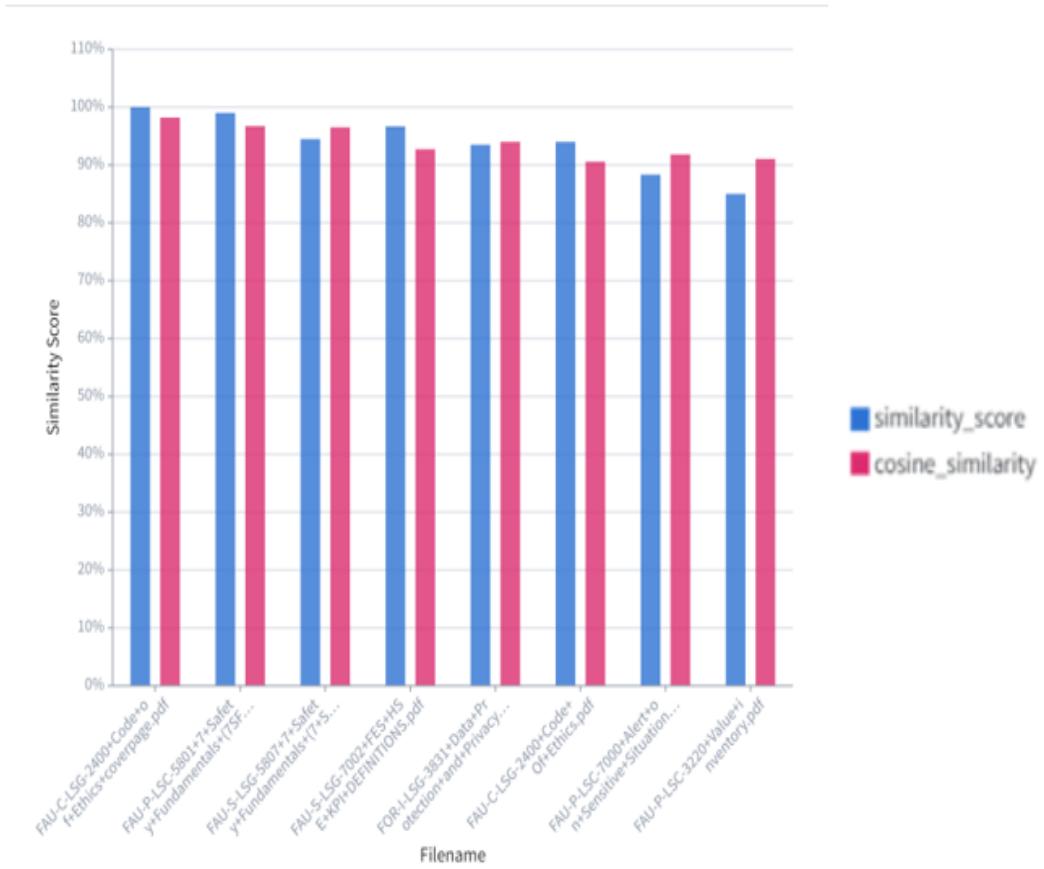


FIGURE 2.29 – Moyennes des scores de similarité obtenues par document

En outre, les moyennes de ces deux scores, calculées pour l'ensemble des documents, sont également très élevées, comme l'indique la Figure 2.30, qui d'ailleurs, représente une des pages de notre application créée sur le "Workshop" de Palantir Foundry [Palantir, 2024]. Cette page a été spécialement conçue, pour que l'utilisateur de l'application ait une idée sur les moyennes des (KPIs), obtenu lors du traitement du texte.

The screenshot shows a user interface for 'Forvia Core Procedures'. At the top, there's a navigation bar with icons for a profile, 'FCP AIP Chat', 'KPIs for text', and 'KPIs for images'. Below this is a table with columns: 'Question', 'Document', 'Cosine Similarity', 'Similarity Score', 'Reference Answer', and 'GPT Response'. The table contains 10 rows of data. A red box highlights the last row, which is a summary: 'Cosine Similarity' is 0.937, 'Similarity Score' is 0.936, and 'Difference' is 0.001. Another red box highlights a section on the right side of the table that reads: 'Moyennes des cosine similarity et Similarity\_score pour la totalité des documents'.

Question	Document	Cosine Similarity	Similarity Score	Reference Answer	GPT Response
How often should the analysis of inventories be	FAU-P-LSC-3220+Value+inventor	0.945204644813	0.949999988079	The analysis of inventories should be	The analysis of inventories should be performed at
What are the measures for fire and chemical prevention	FAU-P-LSC-5801+7+Safety+Fund	0.952695206448	1	Evacuation plans must be clearly visible and	Measures for fire and chemical prevention in the
What is the main responsibility of everyone in	FAU-P-LSC-5801+7+Safety+Fund	0.966780100438	1	Everyone has the responsibility and	The main responsibility of everyone in the context of
When should an allowance for impairment in value be	FAU-P-LSC-3220+Value+inventor	0.948843552374	1	An allowance should be recorded when	An allowance for impairment in value shou
What is the purpose of the document "FES HSE KPI"	FAU-S-LSG-7002+FES+HSE+KPI+	0.962071071780	1	The purpose of this document is to clarify	The purpose of the document "FES HSE KPI"
What are the safety measures related to roof	FAU-P-LSC-5801+7+Safety+Fund	0.976300821496	1	Roof edges must be secured by collective	Safety measures related to roof access include:
Who is the owner of the Code of Ethics?	FAU-C-LSG-2400+Code+of+Ethic	0.974189380270	1	The owner of the Code of Ethics is	The owner of the Code of Ethics is Houda DAMON,
What are the training and validation requirements for	FAU-P-LSC-5801+7+Safety+Fund	0.971260684101	1	For all maintenance technicians that need	The training and validation requirements for those
What is the role of the Local Data Protection Officer	FOR-I-LSG-2021-DPO+Role+Obligations	0.931841774357	1	The LDPC supports	The Local Data Protection

Cosine Similarity	Similarity Score	Difference
0.937	0.936	0.001

FIGURE 2.30 – Moyennes des scores de similarité obtenues pour tous les documents

En ce qui concerne les images, les résultats sont légèrement inférieurs à ceux obtenus pour le texte, mais restent considérés comme bons ( par rapport aux normes trouvées dans la littérature). La Figure 2.31 montre les moyennes des deux scores de similarité sémantique à savoir la similarité cosinus et le score de similarité ( généré par GPT-3.5 pour comparer la proximité sémantique entre les réponses de GPT-4-32k et les réponses de références créées humainement), par document.

Les moyennes de ces deux scores, calculées sur l'ensemble des documents avec images, sont inférieures à celles des réponses générées pour des questions sur le contenu textuel. Cependant, elles restent satisfaisantes, en comparaison avec les normes mentionnées dans la littérature, comme le montre la Figure 2.32

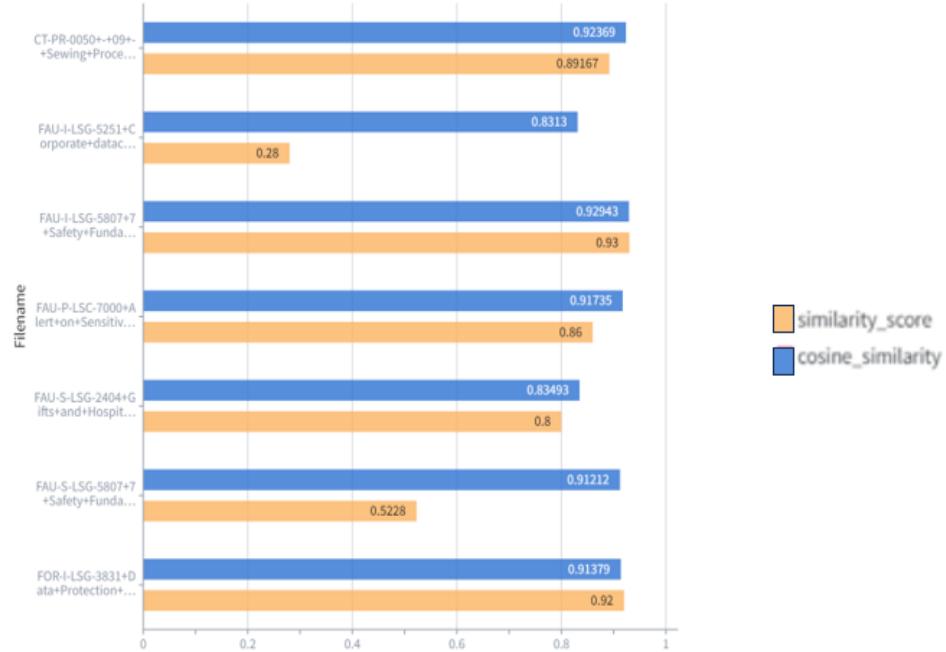


FIGURE 2.31 – Moyennes des scores de similarité obtenues par document pour les images

Forvia Core Procedures ☆					
	FCP AIP Chat	KPIs for text	KPIs for images		
Question	Document	Img Similarity Score	Img Cosine Similarity	Reference Answer	GPT Answer
What are the conditions for the different alert statuses in	FAU-P-LSC-7000+Alert+on+	1 91	0.956721501814 0.91	In RULE 2, if the target date is achieved with less than 30 days	The conditions for the different alert statuses in RULE 2 are as follows:
How is the "High Risk Operations" safety	FAU-S-LSG-5807+7+Safety+	0.200000002980 23224	0.89335328324 5793	The "High Risk Operations" safety fundamental is indicated by an heptagon	The "High Risk Operations" safety fundamental is indicated on a heptagon
What should be done after STOP WORK?	FAU-I-LSG-5807+7+Safety+	1 1227	0.932414886202 1227	After STOP WORK, you should notify your supervisor. Before doing	After STOP WORK, the worker should notify the supervisor and answer the
What is the third step in the DPIA process?	FOR-I-LSG-3831+Data+Prot	1 9126	0.993567687784 9126	The third step in the DPIA process is to "Execute DPIA".	The third step in the DPIA process is to "Execute DPIA".
What is the last step in accepting a hospitality?	FAU-S-LSG-2404+Gifts+and	0.600000023841 8579	0.781673669681 0352	Check whether the beneficiary had accepted 1X from the same	The last step in accepting a hospitality is according to the
What are the "7 Safety Fundamentals" depicted in	FAU-S-LSG-5807+7+Safety+	5596	0.713999986648 16	The 7 Safety Fundamentals are: Stop Work, Safe Traffic, Safe	The "7 Safety Fundamentals" depicted in the image are:
What is the step to follow when a Faurecian receives a	FAU-S-LSG-2404+Gifts+and	929	0.800000011920 8494	Check whether the Faurecian had accepted a gift in the ongoing fiscal	When a Faurecian receives a gift at less than 100 EUR, they should follow
What are the dimensions inside the crash barriers with	FAU-I-LSG-5251+Corporate	1 8104	0.923810759496 8104	With the LAN cabinet, it's 3.6m inside, and without it, it's 2.8m	The dimensions inside the crash barriers are 2.8m without the LAN cabinet and
Who is at the top of the	FOR-I-LSG-3831+Data+Prot	1	0.933565374124 1	The Management is at the top of the	The Management is at the top of the

**Moyennes des cosine similarity et Similarity\_score pour la totalité des documents**

Cosine Similarity 0.895	Similarity Score 0.748	Difference 0.148
----------------------------	---------------------------	---------------------

FIGURE 2.32 – Moyennes des scores de similarité obtenues pour tous les documents d'images

## 2.3 Crédit d'un *AIP Widget* dans le Workshop de Foundry

L'autre but de ce projet était de créer un *widget* de plateforme d'intelligence artificielle (*AIP Widget*) capable de réaliser diverses tâches (résumé, reconnaissance d'entités nommées, etc.).

La tâche la plus demandée par les utilisateurs, était de répondre à tout types de questions sur les documents *FCP* de FORVIA (que ce soit des questions posées sur le contenu texte ou images), en donnant le document sur lequel il s'est basé et la page considérée, dans la génération de sa réponse.

A l'issue des résultats de notre étude comparative menée précédemment, nous avons décidé d'utiliser **GPT-4-32k** dans notre *widget* de plateforme d'intelligence artificielle (*AIP Widget*). Voici un extrait de discussion entre un employé de FORVIA et l'agent conversationnel créé :

Employé : What is the primary focus of protecting Faurecia Group assets?

Agent conversationnel : The primary focus of protecting Faurecia Group assets is to ensure the security and integrity of the company's assets, including its information, intellectual property, and physical resources.

The source Document is : FAU-C-LSG-2400+Code+Of+Ethics.pdf

The page number is : 25

À la suite de la réponse de l'agent conversationnel, l'utilisateur peut vérifier le document et la page mentionnés par le modèle de langage pour valider la réponse grâce à la visionneuse de documents (*document viewer*). Les Figures 2.33 et 2.34 ci-dessous, montrent comment on peut utiliser cette visionneuse de documents mise en disposition dans notre application.

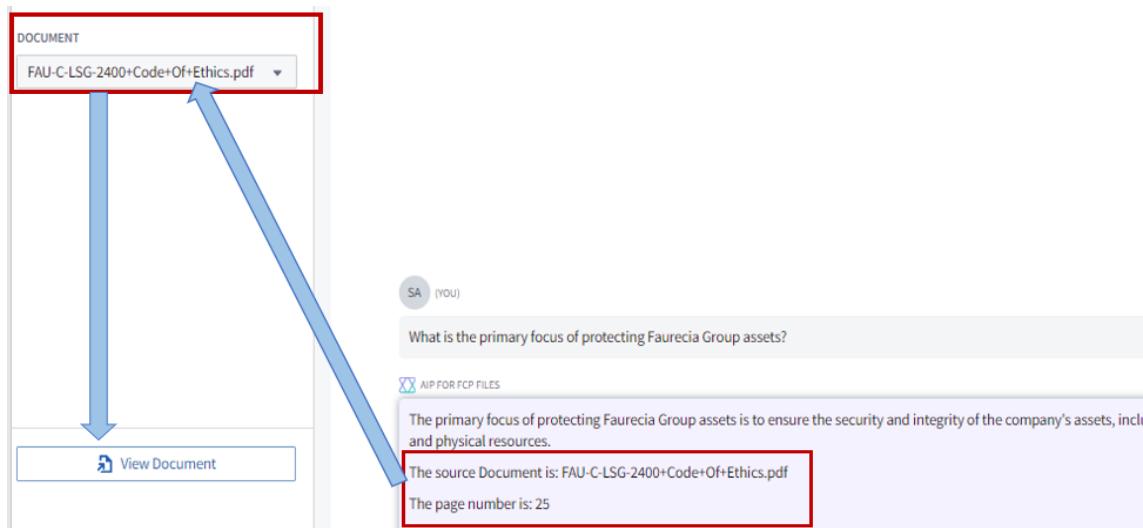


FIGURE 2.33 – Sélection du document proposé par l'agent conversationnel

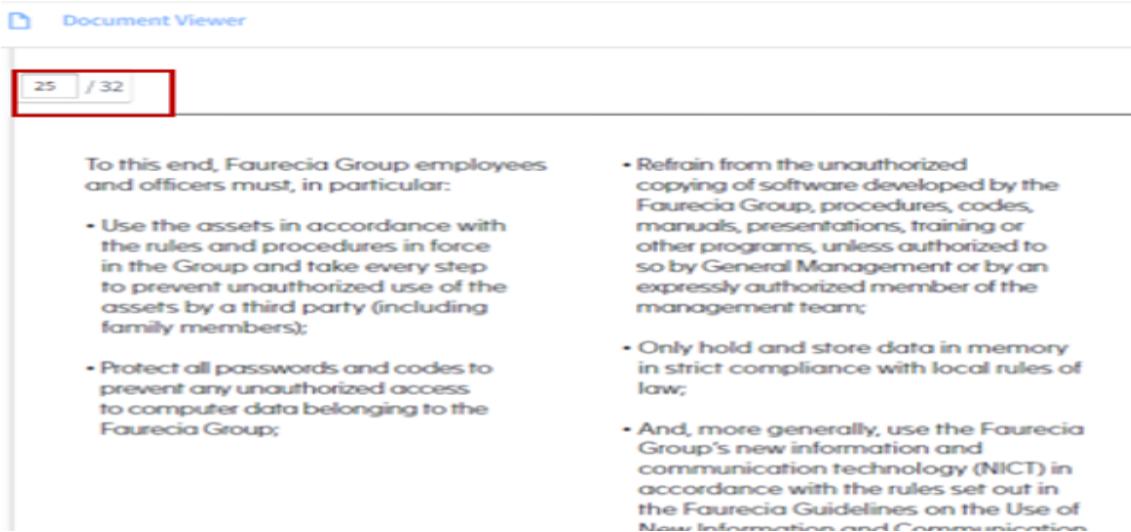


FIGURE 2.34 – Visionneuse de documents (*document viewer*)

Ce widget a été basée sur une fonction typescript permettant d'effectuer une recherche sémantique (*semantic\_search*) sur une chaîne de requête donnée grâce à une fonction *KNN*.

Voici un résumé de son fonctionnement :

1. **Création de l'encodage *Embeddings*** : Il crée un encodage (une représentation vectorielle) pour la requête d'entrée (principalement des questions) en utilisant le même modèle utilisé pour la création des encodages de notre ensemble de données, qui est *text-embedding-ada-002*.
2. **Recherche Sémantique** : Il effectue une **recherche sémantique** pour trouver les documents les plus pertinents, pour répondre à la requête d'entrée.  
Celà se fait en trouvant les voisins les plus proches de la représentation vectorielle de la requête, qui sont les documents de Forvia Core Procedures (*FCP*).
3. **Classement des Résultats** : Les résultats de la recherche sont classés par leur pertinence par rapport à la requête, et le nombre de résultats est limité aux **k Value instances** les plus pertinentes.

Cependant, on a vite remarqué qu'un K plus grand que 1 causait des problèmes d'hallucination pour le modèle de langage, vu qu'il prendra en considération deux documents ou plus, pour générer sa réponse. On a donc décidé de fixer  $K = 1$  dans la fonction *KNN*.

## 2.4 Conclusion

Dans cette section on a étudié le choix de l'encodage *text-embedding-ada-002* pour nos données, qui s'explique par deux raisons majeures.

Tout d'abord, cet algorithme s'est classé premier au *Massive Text Embedding Benchmark (MTEB)*, ce qui démontre sa capacité à capturer efficacement les relations sémantiques entre les segments de texte.

De plus, une étude plus approfondie, a montré sa capacité à détecter le contexte manufacturier d'une phrase même si elle contient des mots clés couramment utilisés dans d'autres contextes (Le terme

"Plant" par exemple, a deux sens très différents qui dépendent du contexte, soit il indique une plante dans un contexte de nature ou de jardinage, soit il indique une usine dans un contexte manufacturier).

On a aussi étudié l'architecture des grands modèles de langages (*LLMs*) qui est l'architecture du décodeur uniquement (*Decoder-only*) et sa composante clé (Masked Multi-Head Attention) qui lui a permis d'atteindre des performances meilleures que tous les modèles adaptant l'architecture du transformateur (avec ses deux parties encodeur et décodeur) en terme de génération de texte. Nous avons ensuite présenté les performances mesurées et les KPIs obtenus sur **GPT-4-32k** et **GPT-4o**. La Table 2.3 résume les mesures des indicateurs de performance clés.

Modèle	Cosine Sim Moyenne	Similarity Score Moyenne
<b>GPT-4-32k</b>	0.973	0.963
<b>GPT-4o</b>	0.73	0.928

TABLE 2.3 – Comparaison des scores de similarité

L'écart important entre la moyenne des similarités cosinus et les scores de similarités (généré par un Grand Modèle de Langage), pour le cas de **GPT-4o**, montre que ce dernier donne des réponses très verbeuses et détaillées. Certes il s'agit d'un avantage, mais ce n'est pas vraiment recherché dans le cadre de notre évaluation (*benchmark*) où on recherche des réponses concises contexte.

En prenant en considération tous les résultats obtenus, on a décidé que **GPT-4-32k** est le modèle qui répond le plus à nos besoins dans le cadre de ce projet. La Figure 2.35 montre une analyse plus approfondie sur les valeurs prises par les deux scores de similarité sémantiques, pour le cas de **GPT-4-32k**.

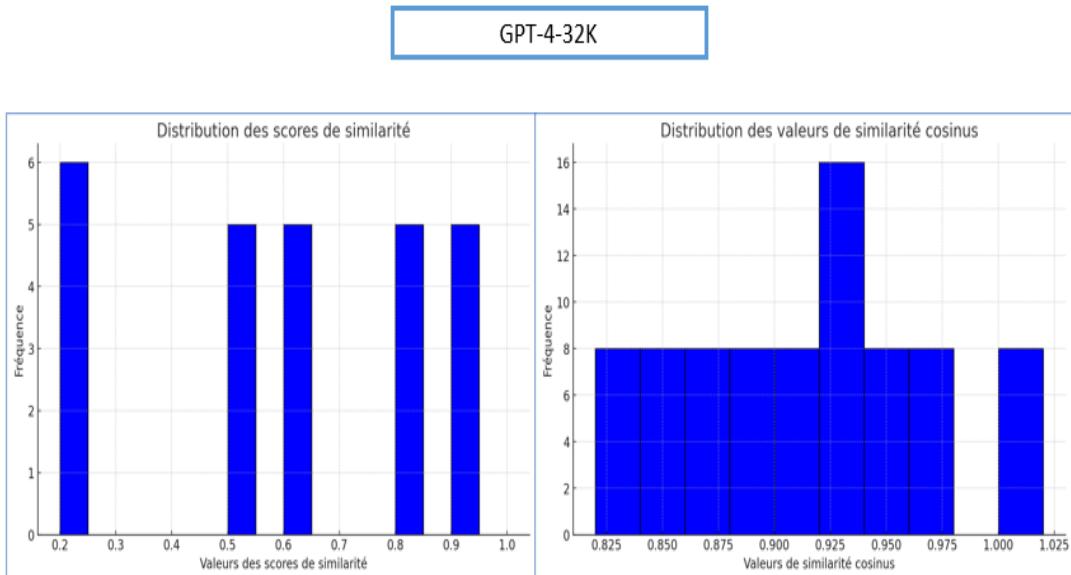


FIGURE 2.35 – Analyse des scores de similarité pour **GPT-4-32k**

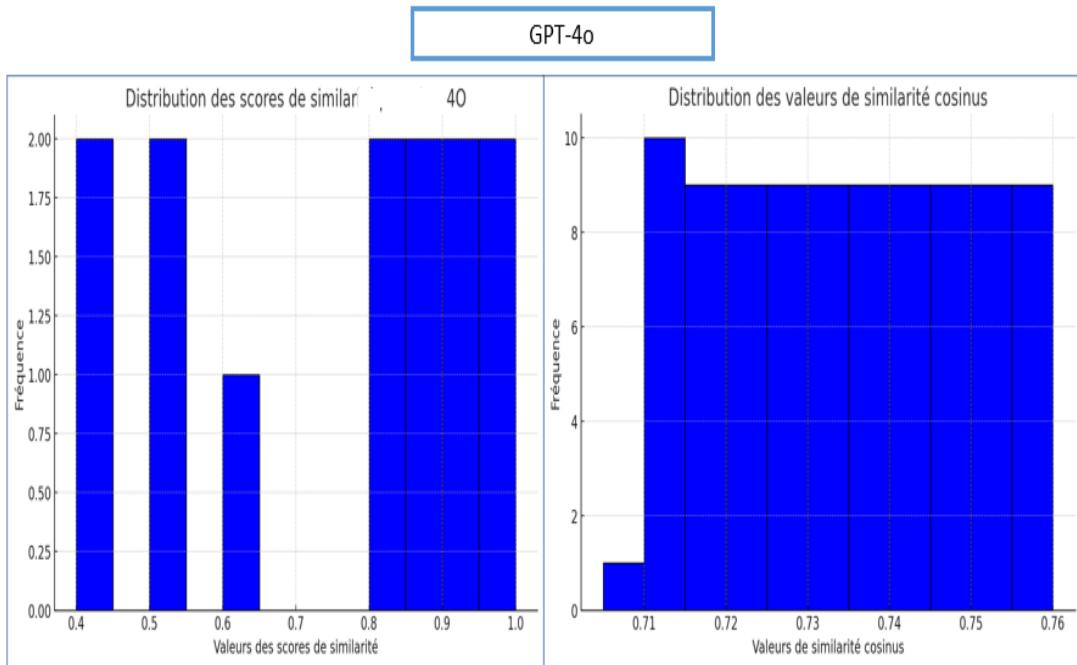


FIGURE 2.36 – Analyse des scores de similarité pour **GPT-4o**

Le graphique de gauche de la Figure 2.35 illustre la distribution des scores de similarité générés par un *LLM* (**GPT-3.5**). Le graphique de droite présente la distribution des valeurs de similarité cosinus. Ces analyses montrent la pertinence des réponses générées par **GPT-4-32k** mettant en valeur la performance du modèle, en termes de compréhension des questions posées, du texte de référence (introduit au modèle de langage à travers la génération augmentée par récupération *RAG*) et de génération de réponses contextuelles et sémantiquement pertinentes.

De la même manière, la Figure 2.36 montre une analyse plus approfondie sur les valeurs prises par les deux scores de similarité sémantiques, pour le cas de **GPT-4o**.

Bien qu'il n'existe pas encore de métriques, de modèles ou même de méthodes proposées par la littérature pour évaluer la performance de **GPT-4-Vision-Turbo**, nous avons constaté, grâce à une validation humaine, des performances convaincantes de ce modèle Multi-Modal en termes de génération de descriptions d'images. Cela a permis aux autres grands modèles de langage de disposer d'une base solide sur laquelle s'appuyer pour répondre aux questions basées sur les images.

D'ailleurs, comme on l'a déjà présenté dans cette section, les moyennes des mesures des indicateurs de performances clés considérés dans le traitement des images sont relativement bonnes : une moyenne de similarités cosinus égale à 0.895 ; une moyenne de scores de similarités (généré par **GPT-3.5**) pour juger les réponses de **GPT-4-32k** égale à 0.748.

# Conclusion et Perspectives

L'objectif initial de ce projet est de tirer parti de la documentation existante de l'entreprise FORVIA via des outils d'intelligence artificielle générative. Pour ce faire, nous avons trouvé la solution optimale pour adapter les grands modèles de langages intégrés dans Palantir Foundry [Palantir, 2024], qui sont les modèles (*Generated Pretrained Transformers* ou **GPT**) proposés par OpenAI, aux données de la société FORVIA sans qu'ils ne soient sujets à des problèmes de confidentialité.

Les modèles disponibles dans le cadre de ce projet, qui sont les modèles GPT, ne sont pas capables d'interroger directement des bases de données vectorielles externes à eux, même si ces dernières sont intégrées dans Palantir Foundry. Nous avons trouvé la solution de faire une jointure entre l'ensemble de données contenant les questions, et celui contenant les textes de références pré-traités. Par conséquent le modèle de langage (*LLM*) n'aura pas à rechercher des données en externes mais à parcourir les colonnes d'un même ensemble de données en allant vers celle contenant le texte de référence. Ensuite le repérage de la ligne en question se fait grâce à une indexation avec une clé primaire unique pour chaque texte, faite dans la phase de pré-traitement.

Nous avons commencé par créer un ensemble de données contenant les questions créées humainement (avec l'aide de Copilot pour les données textuelles seulement) et les réponses considérées comme référence. Après, nous avons comparé les réponses des *LLMs*, à savoir **GPT-4-32k** et **GPT-4o**. Cette comparaison a nécessité, en premier lieu, la fixation des indicateurs de performances clés KPIs. Les indicateurs qu'on a jugé comme les plus importants dans le cadre de ce projet sont la similarité cosinus et le score de similarité (donné lui-même par **GPT-3.5**), mesurant tous les deux la similarité sémantique entre les réponses des *LLMs* considérés et celles utilisées comme référence.

En second lieu, on a calculé ces indicateurs de performance clés. On a ainsi obtenu des résultats montrant la pertinence des *LLMs* testés dans le cadre de ce projet, non seulement dans leurs compréhension du texte qui leurs est indiqué par la génération augmentée par récupération (*RAG*) mais aussi des descriptions générées par **GPT-4-Vision-Turbo**.

Ces modèles ont pu générer des réponses sur les questions qui leur étaient posées, très proches sémantiquement des réponses de références. Ceci a été indiqué par des scores de similarité sémantique élevés. Pour **GPT-4-32k**, on a eu une moyenne de 0.937 pour la similarité cosinus mesurées entre les encodages, des réponses générées par ce modèle et de celles considérées comme référence, et une moyenne de 0.936 pour le score de similarité généré par **GPT-3.5**, en lui donnant comme entrée les deux réponses à comparer sémantiquement.

Idem pour **GPT-4o**, nous avons obtenu une moyenne de 0.928 pour le score de similarité généré par **GPT-3.5**, qui est donc un peu plus élevée que celle obtenue pour **GPT-4-32k**, mais avec une moyenne relativement faible de la similarité cosinus, mesurant 0.73.

L' écart important entre les deux moyennes (égal à 0,198) a montré que le modèle **GPT-4o** générait des réponses trop détaillées par rapport à nos réponses de références.

Vu qu'on cherchait des réponses plus concises, on a choisi de continuer l'étude sur les questions posées sur les descriptions des images en ayant recours uniquement à **GPT-4-32k**. Ce modèle s'est montré performant aussi en terme de génération de réponses posées sur les images en utilisant les descriptions qui lui sont fournies, avec une moyenne de similarité cosinus égale à 0.895 et une moyenne de score de similarité (généré par **GPT-3.5**) égal à 0.748.

Certes les réponses obtenues sur les questions posées sur les images sont moins pertinents que ceux obtenus suite à l'interrogation sur le texte, mais les scores obtenus pour le traitement des images sont considérés comme très bons dans la littérature. Cette reconnaissance témoigne aussi, de l'efficacité de **GPT-4-Vision-Turbo** dans le domaine de l'analyse d'images, car sans ses descriptions pertinentes et adaptées au contexte du document, les autres Modèles de Langages n'auront pas une bonne base à laquelle ils vont recourir pour répondre aux différentes questions.

Cette démarche a permis de confirmer le bon rapport qualité-prix des investissements menés par l'entreprise dans l'achat des APIs des modèles **GPT** de OpenAI, surtout pour **GPT-4-32k**, comme l'ont **attesté les techleads et les responsables du financement** de ce projet chez FORVIA.

Suite à ses performances qui dépassaient celles de **GPT-4o**, le modèle **GPT-4-32k** a été ensuite choisi pour créer une interface intuitive contenant un agent conversationnel capable de répondre aux questions des utilisateurs sur les documents des procédures de base de FORVIA (*Forvia Core Procedures (FCP)*) avec une précision accrue : fournissant le nom du document et le numéro de la page sur laquelle *GPT-4-32k* s'est basé pour générer sa réponse. L'utilisateur a la main pour valider les références par la suite grâce à une visionneuse de documents intégrées dans l'interface de l'application.

Nous avons réalisé une évaluation des performances des modèles disponibles dans la plateforme Palantir Foundry de FORVIA - à savoir **GPT-4-32k**, **GPT-4o** et **GPT-4-Vision-Turbo** - en appliquant la technique de la génération augmentée par récupération *RAG*.

Ce travail a été validé par M. David Luz l'expert scientifique au sein de FORVIA et aussi par M. Jesus Albors Gil, le responsable de la Fabrication en Flux et Manager de la Qualité chez FORVIA. Ainsi, nous avons démontré l'importance et l'intérêt d'investir dans des outils tels que les grands modèles de langage (*LLMs*), afin de favoriser et réussir la transformation digitale de FORVIA.

En perspective, des modèles open-source pourraient être intégrés dans Palantir Foundry, et comparés aux modèles payants existants. Ainsi, en appliquant un affinage (*Fine-tuning*) sur certains modèles open-source tels que LAMA et MISTRAL, une comparaison des performances pourrait être faite avec des modèles d'OpenAI.

## Annexe A

## Conversion de documents

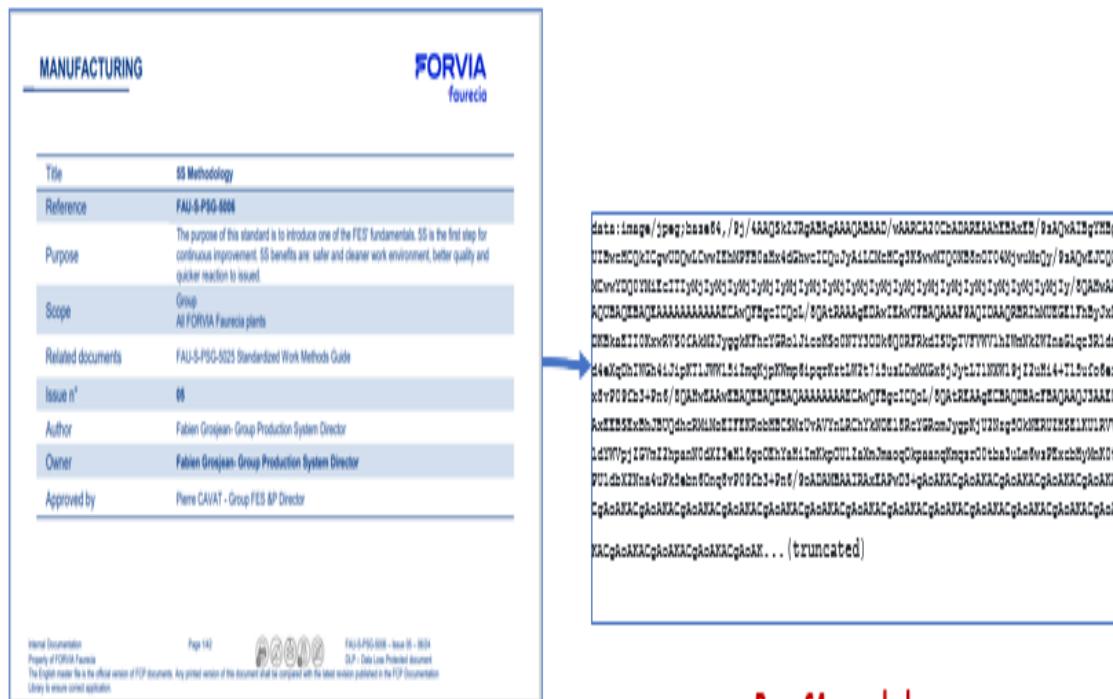


FIGURE A.1 – Conversion d'une page PDF en JPEG ensuite en *base64 encoded*.

# Bibliographie

- [Achiam et al., 2023] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. (2023). Gpt-4 technical report. *arXiv preprint arXiv :2303.08774*.
- [Alto, 2023] Alto, V. (2023). *Modern Generative AI with ChatGPT and OpenAI Models : Leverage the capabilities of OpenAI's LLM for productivity and innovation with GPT3 and GPT4*. Packt Publishing Ltd.
- [Bierman et al., 2014] Bierman, G., Abadi, M., and Torgersen, M. (2014). Understanding typescript. In *ECOOP 2014—Object-Oriented Programming : 28th European Conference, Uppsala, Sweden, July 28–August 1, 2014. Proceedings 28*, pages 257–281. Springer.
- [Booma, 2023] Booma, D. (2023). Building a GenAI-Powered Chatbot Using GPT-J : A Step-by-Step Guide. *Medium*. [Accessed 11-06-2024].
- [Cheng, 2024] Cheng, F. J. (2024). A Quick Overview of LLM Benchmarking — felixjcheng. <https://medium.com/@felixjcheng/>. [Accessed 04-04-2024].
- [Dey, 2024] Dey, R. (2024). Understanding Language Modeling : From N-grams to Transformer-based Neural Models. *Medium*. <https://medium.com/@roshmitadey/>.
- [Drabas and Lee, 2017] Drabas, T. and Lee, D. (2017). *Learning PySpark*. Packt Publishing Ltd.
- [Faurecia, 2024] Faurecia (2024). Digital transformation and data ontology. <https://fau-rus.faurecia.com/>. [Accessed 04-04-2024].
- [Hamad, 2024] Hamad, R. (2024). What is LSTM ? Introduction to Long Short-Term Memory. *Medium*. <https://medium.com/@rebeenhamad/>.
- [Han et al., 2021] Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., and Wang, Y. (2021). Transformer in transformer. *Advances in neural information processing systems*, 34 :15908–15919.
- [Hendrycks et al., 2021] Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. (2021). Measuring massive multitask language understanding.
- [Khemaïs, 2023] Khemaïs, J. (2023). Leveraging the Power of Large Language Models (LLMs) to Drive Progress in the Manufacturing Sector. *Medium*. [Accessed 04-04-2024].
- [Lusztrig, 2021] Lusztrig, G. (2021). Total positivity in symmetric spaces.
- [Maharana et al., 2022] Maharana, A., Lee, D.-H., Tulyakov, S., Bansal, M., Barbieri, F., and Fang, Y. (2022). Semscore : Evaluating llms with semantic similarity. *arXiv preprint arXiv :2204.01036*.
- [Maharana et al., 2024] Maharana, A., Lee, D.-H., Tulyakov, S., Bansal, M., Barbieri, F., and Fang, Y. (2024). Evaluating very long-term conversational memory of llm agents. *arXiv preprint arXiv :2402.17753*.
- [Malode, 2024] Malode, V. M. (2024). *Benchmarking public large language model*. PhD thesis, Technische Hochschule Ingolstadt.

- [Microsoft, 2024a] Microsoft (2024a). copilot. <https://learn.microsoft.com/en-us/copilot/microsoft-365/>.
- [Microsoft, 2024b] Microsoft (2024b). Microsoft power apps. <https://powerapps.microsoft.com/en-us/>.
- [OCR, 2024] OCR, T. (2024). Tesseract ocr. <https://tesseract-ocr.github.io/>.
- [open ai, 2024] open ai (2024). Hello gpt-4o. <https://openai.com/index/hello-gpt-4o/>. [Accessed 04-04-2024].
- [Or, 2020] Or, B. (2020). The exploding and vanishing gradients problem in time series. *Medium. Towards Data Science*.
- [Palantir, 2024] Palantir (2024). Palantir foundry. <https://www.palantir.com/platforms/foundry/>. [Accessed 05-04-2024].
- [Pchelina, 2023] Pchelina, M. (2023). Text embedding models : how to choose the right one — medium.com. <https://medium.com/mantisnlp/>. [Accessed 04-04-2024].
- [pdfplumber, 2019] pdfplumber (2019). pdfplumber 0.1.2. <https://pypi.org/project/pdfplumber/0.1.2/>.
- [Pramod, 2023] Pramod, O. (2023). Exploring N-Grams : The Building Blocks of Natural Language Understanding. <https://medium.com/@ompramod9921/>.
- [Smith et al., 2009] Smith, R., Antonova, D., and Lee, D.-S. (2009). Adapting the tesseract open source ocr engine for multilingual ocr. In *Proceedings of the international workshop on multilingual OCR*, pages 1–8.
- [Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Polosukhin, I., Kaiser, L., Polosukhin, I., and Kaiser, L. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 6000–6010.
- [Wang et al., 2023] Wang, H., Wu, W., Dou, Z., He, L., and Yang, L. (2023). Performance and exploration of chatgpt in medical examination, records and education in chinese : Pave the way for medical ai. *International Journal of Medical Informatics*, 177 :105173.



## Résumé

Le présent projet a pu réaliser deux volets. Le premier volet visait à améliorer l'exploitation de la documentation des procédures de base de FORVIA (*Forvia Core Procedures*) ou *FCP* en développant un agent conversationnel et plus précisément un module basé sur l'intelligence artificielle générative : (*AIP Widget*), capables de répondre à tout types de questions, posées par des utilisateurs en interne, concernant les documents *FCP*. Nous avons choisi des grands modèles de langage (*LLMs*) pour atteindre cet objectif. Parmi les modèles de langage envisagés, il y a **OpenAI GPT-4-32k**, **OpenAI GPT-4o** et finalement **OpenAI GPT 4-Vision-Turbo**, qui est un modèle Multi-Modal capable de traiter à la fois du texte et des images. Toutefois, ces modèles présentent des contraintes telles que des coûts élevés des appels de l'interface de programmation d'application. Par conséquent, on a eu recours à une évaluation (*Benchmarking*) dans le deuxième volet de ce projet, ce qui a permis de fixer et mesurer des indicateurs clés de performance recherchés par l'entreprise FORVIA, afin de répondre à la question : *L'investissement dans ce type de modèles de langage payants est-il vraiment nécessaire ou pas ?*

Cette analyse comparative a été précédée par l'application de la méthode de génération augmentée par récupération (*RAG*), afin d'adapter les *LLMs* aux données de l'entreprise (documentation, contrat, rapport financier, etc.). Ce projet a permis à FORVIA non seulement d'exploiter plus précisément la documentation de base, en évitant une analyse manuelle coûteuse, chronophage et sujette aux erreurs, mais également de disposer de mesures précises des indicateurs de performance clés. Ces mesures ont démontré l'efficacité des différents modèles intégrés dans sa plateforme (Palantir Foundry) pour les tâches ciblées.

**Mots clés**— Traitement Automatique de Langage, Intelligence Artificielle Générative, Génération Augmentée par Récupération, Analyse Comparative.

## Abstract

This project was able to achieve two main objectives. The first objective aimed to improve the utilization of the Forvia Core Procedures (*FCP*) documentation by developing a conversational agent, specifically a module based on generative artificial intelligence (*AIP Widget*), capable of answering all types of questions posed by internal users regarding the *FCP* documents. We selected large language models (*LLMs*) to achieve this goal. Among the considered language models were **OpenAI GPT-4-32k**, **OpenAI GPT-4o**, and finally **OpenAI GPT-4-Vision-Turbo**, which is a multi-modal model capable of handling both text and images.

However, these models present constraints such as high API call costs. Therefore, we conducted a benchmarking evaluation in the second part of this project, which allowed us to establish and measure key performance indicators sought by FORVIA, in order to answer the question : *Is the investment in these types of paid language models really necessary ?*

This comparative analysis was preceded by the application of the Retrieval-Augmented Generation (*RAG*) method to adapt the *LLMs* to the company's data (documentation, contracts, financial reports, etc.). This project enabled FORVIA not only to more precisely exploit the core documentation, avoiding costly, time-consuming, and error-prone manual analysis, but also to have precise measurements of key performance indicators. These measurements demonstrated the effectiveness of the different models integrated into its platform (Palantir Foundry) for the targeted tasks.

**Keywords**— Natural Language Processing (NLP), Generative AI, Retrieved Augmented Generation (*RAG*), Benchmarking.