



## Rapport De Stage

### FORENSIQUE - Investigation numérique de fonds d'archives culturelles

*Classification automatique*

*mars - juillet 2025*

**AISSAOUI Mohammed Salah**

**GIGUET Emmanuel  
Tanguy Gernot**

Projet soutenu par le ministère de la Culture

# Table des matières

<b>1</b>	<b>Remerciements</b>	<b>4</b>
<b>2</b>	<b>Introduction</b>	<b>5</b>
<b>3</b>	<b>Environnement de stage</b>	<b>6</b>
3.1	Laboratoire GREYC . . . . .	6
3.2	Équipe SAFE . . . . .	6
3.3	Environnement de travail . . . . .	6
<b>4</b>	<b>État de l'art</b>	<b>7</b>
4.1	Introduction à la classification automatique . . . . .	7
4.2	Analyse des besoins . . . . .	8
4.3	Analyse du jeu de données . . . . .	8
4.4	Synthèse . . . . .	9
<b>5</b>	<b>Travail réalisé</b>	<b>10</b>
5.1	Traitement du texte . . . . .	10
5.1.1	Classification globale . . . . .	10
5.1.2	Dataset . . . . .	11
5.1.3	Machine Learning . . . . .	12
5.1.4	Fine-tuning (Apprentissage par transfert) . . . . .	14
5.1.5	Synthèse et comparaison des approches . . . . .	17
5.2	Traitement d'images . . . . .	19
5.2.1	Objectif et typologie . . . . .	19
5.2.2	Jeu de données . . . . .	19
5.2.3	Préparation des données . . . . .	19
5.2.4	Stratégie d'apprentissage (transfer learning) . . . . .	20
5.2.5	Expérimentation 1 — Binaire (Photo vs Autre) . . . . .	20
5.2.6	Expérimentation 2 — Multiclasse (4 classes) avec ViT . . . . .	22
5.2.7	Discussion et choix final . . . . .	23
5.3	Classifieur IANEC . . . . .	23
<b>6</b>	<b>Approche expérimentale</b>	<b>25</b>
6.1	Clustering (recherche par similarité) . . . . .	25
6.2	RAG — Recherche augmentée par génération . . . . .	26
6.3	Classifieur IANEC 2 . . . . .	28
<b>7</b>	<b>Conclusion</b>	<b>30</b>

## Table des figures

1	Classification globale. . . . .	11
2	Répartition du dataset par catégorie . . . . .	12
3	Schéma du pipeline complet . . . . .	13
4	Schéma du modèle BART et principe du fine-tuning. . . . .	14
5	Comparaison fine-tuning partiel vs complet. . . . .	15
6	Résultats du fine-tuning partiel. . . . .	16
7	Résultats du fine-tuning complet. . . . .	16
8	Répartition accord/désaccord pour les documents $\geq 200$ mots. . . . .	17
9	Répartition accord/désaccord . . . . .	18
10	Pipeline prétraitement . . . . .	19
11	Courbes ROC — ViT vs ResNet . . . . .	20
12	ResNet-18 — perte et accuracy . . . . .	21
13	ViT — perte et accuracy . . . . .	21
14	Courbe ROC multiclasse (ViT) . . . . .	22
15	ViT — perte et accuracy (multiclasse). . . . .	23
16	Aperçu global du <b>classifieur IANEC</b> sous Streamlit . . . . .	24
17	Clustering et recherche par similarité . . . . .	26
18	Prototype RAG : recherche sémantique locale (MiniLM) et génération contrainte par le contexte (Mistral). . . . .	27
19	Aperçu du <b>classifieur IANEC 2</b> . . . . .	29

# 1 Remerciements

Je tiens à remercier M. Emmanuel Giguet et M. Gernot Tanguy pour leur encadrement et leur soutien tout au long de ce stage.

Je remercie également le laboratoire GREYC et l'équipe SAFE pour leur accueil et les moyens mis à ma disposition.

Enfin, un grand merci à mes collègues travaillant sur le même sujet, pour leur esprit d'équipe et leurs échanges enrichissants .

## 2 Introduction

L'Institut Mémoires de l'édition contemporaine (IMEC) est une institution dédiée à la collecte, la conservation et la valorisation des archives liées au monde de l'édition, de la littérature, de la pensée et des arts contemporains. Véritable gardien de la mémoire éditoriale et intellectuelle française, l'IMEC rassemble des fonds d'archives d'écrivains, d'éditeurs, de revues et d'intellectuels majeurs du XXe et du XXIe siècle. Ces archives comprennent une grande diversité de documents : manuscrits, correspondances, tapuscrits, coupures de presse, enregistrements sonores ou encore fichiers numériques.

Avec l'évolution des pratiques d'écriture et de communication, une part croissante de ces archives est désormais nativement numérique. Face à cette transformation, l'IMEC s'est engagé dans une réflexion sur les outils et méthodes à développer pour assurer la conservation, la structuration et l'accessibilité de ces nouveaux types de documents. C'est dans ce cadre qu'est né le projet IANEC (Investigation Archivistique Numérique sur les Écrits Contemporains), un projet financé visant à expérimenter des approches innovantes pour l'analyse et la gestion des archives numériques.

L'un des axes du projet IANEC consiste à explorer les potentialités de l'intelligence artificielle et des outils de traitement automatique pour aider à la classification et à l'indexation des documents numériques. En effet, un des défis majeurs rencontrés par les archivistes est le tri des fichiers textuels massivement présents sur les disques durs des auteurs ou institutions. Ce volume conséquent de données rend le classement manuel laborieux, chronophage et peu scalable. Une solution de classification automatique permettrait non seulement de gagner en efficacité, mais aussi de mieux structurer les fonds d'archives numériques, en facilitant la recherche, l'exploitation scientifique et la valorisation patrimoniale de ces documents.

Ce stage s'inscrit donc dans ce contexte d'expérimentation et d'innovation, avec pour objectif de contribuer au développement de méthodes de classification automatique adaptées aux besoins de l'archivistique contemporaine.

## **3 Environnement de stage**

### **3.1 Laboratoire GREYC**

Le GREYC (Groupe de Recherche en Informatique, Image et Instrumentation de Caen) est un laboratoire de recherche en sciences du numérique, créé en 1995. Il est associé au CNRS, à l'Université de Caen Normandie (UNICAEN) et à l'École Nationale Supérieure d'Ingénieurs de Caen (ENSICAEN). Il regroupe des chercheurs et enseignants-chercheurs en informatique et en électronique, répartis sur plusieurs sites en Normandie, dont Caen, Alençon, Cherbourg, Lisieux, Saint-Lô et Vire.

Avec plus de 180 membres, le GREYC est structuré en plusieurs équipes de recherche abordant des thématiques variées : modélisation des algorithmes, sécurité informatique, intelligence artificielle, traitement de l'image et du texte, microélectronique, capteurs, etc. Le laboratoire se distingue par une forte dynamique de collaboration interdisciplinaire, à la fois académique et industrielle, au niveau régional, national et international.

Son activité s'inscrit dans une démarche de recherche fondamentale et appliquée, combinant production scientifique, développement de logiciels, et valorisation économique (brevets, entreprises issues de la recherche, projets collaboratifs). Le GREYC participe activement à de nombreux pôles de compétitivité et réseaux régionaux, dont le Réseau d'Intérêt Normand "Normandie Digitale".

### **3.2 Équipe SAFE**

Durant mon stage, j'ai intégré l'équipe SAFE (Sécurité, Architecture, Forensique, biométrie) . Cette équipe concentre ses travaux sur les questions de sécurité numérique, en s'appuyant sur trois axes principaux : Biométrie : conception et évaluation de systèmes biométriques, avec un accent sur la protection des données sensibles .

Architecture et modèles de sécurité : incluant la sécurité des réseaux, la cryptographie .

Science de l'investigation numérique (forensique) : développement d'outils de traitement automatique de la langue, de plateformes d'analyse et de méthodes de protection de la vie privée.

L'équipe mène également des projets collaboratifs avec des partenaires académiques et industriels, en France comme à l'international. Elle conçoit et développe des logiciels, des plateformes et des applications orientées vers la sécurisation des systèmes et des données, dans des contextes variés.

### **3.3 Environnement de travail**

Pendant toute la durée de mon stage, j'ai eu l'opportunité de travailler à temps plein et en présentiel dans les locaux du laboratoire GREYC, situés sur le campus 2 de l'Université de Caen. Cet environnement de travail m'a permis de bénéficier de conditions optimales : un espace dédié, un accès aux ressources matérielles nécessaires ainsi qu'aux infrastructures de calcul du laboratoire. Ces moyens ont été amplement suffisants pour mener à bien les missions qui m'ont été confiées.

## 4 État de l'art

### 4.1 Introduction à la classification automatique

La classification automatique consiste à attribuer une catégorie ou une étiquette à un fichier, un document ou une donnée, en fonction de ses caractéristiques. Elle est particulièrement utile lorsqu'il s'agit de traiter un grand volume de fichiers textuels ou multimédias, comme c'est le cas dans les archives numériques. En automatisant ce processus, on facilite la gestion, l'organisation et la recherche d'informations pertinentes dans de vastes corpus documentaires.

Plusieurs approches ont été explorées dans le cadre de ce projet, dont les principales sont les suivantes :

- **Machine Learning**

Le machine learning supervisé repose sur la création d'un jeu de données étiqueté, où chaque exemple est associé à une catégorie cible. Un modèle apprend ensuite à généraliser à partir de ces exemples pour prédire la classe de nouveaux fichiers. Cette approche nécessite un certain travail en amont pour annoter les données, mais elle permet d'obtenir des modèles performants et adaptés aux cas d'usage spécifiques.

- **Transfert Learning**

Le transfert learning permet d'exploiter les connaissances déjà acquises par un modèle préentraîné sur une large base de données, pour l'adapter à une nouvelle tâche ou un nouveau domaine. Cela est particulièrement utile lorsqu'on ne dispose pas d'un grand volume de données annotées.

Deux stratégies principales sont utilisées dans ce cadre :

- Finetuning complet : l'ensemble des couches du modèle de base (ou "backbone") est réentraîné sur le nouveau jeu de données.
- Finetuning partiel : seules certaines couches, généralement les couches supérieures (ou la "tête"), sont adaptées à la nouvelle tâche, ce qui permet de réduire le coût de calcul et de limiter le surapprentissage.

- **Modèles LLM (Large Language Models)**

Les LLMs, ou modèles de langage de grande taille, sont des réseaux de neurones entraînés sur d'immenses corpus textuels, comportant plusieurs milliards de paramètres. Ces modèles, comme GPT, peuvent comprendre et générer du langage naturel, analyser le contenu d'un document, proposer des résumés ou extraire des informations contextuelles pertinentes. Ils offrent de nouvelles perspectives en matière de classification, notamment via des interactions directes avec l'utilisateur ou des chaînes de traitement hybrides.

- **Autres approches explorées**

Au cours des phases d'analyse et d'expérimentation, d'autres techniques ont également été mobilisées :

- RAG (Retrieval-Augmented Generation) : une méthode combinant recherche d'information et génération de texte pour enrichir le raisonnement d'un LLM.
- Clustering : méthode non supervisée visant à regrouper les documents similaires sans utiliser d'étiquettes prédéfinies, utile pour explorer la structure d'un corpus avant annotation ou modélisation.

## 4.2 Analyse des besoins

Dans la phase exploratoire du projet, plusieurs besoins fondamentaux ont été identifiés, en lien direct avec les missions de l'IMEC et la nature des corpus traités :

### **Caractérisation des fichiers numériques**

L'objectif est de détecter automatiquement des types de contenus spécifiques à la nature des archives :

- correspondance (lettres, emails) .
- documents liés au travail d'écriture ou à la création d'œuvres.

### **Extraction de métadonnées descriptives**

Pour améliorer l'identification et la recherche des documents :

- détection de l'auteur présumé via recoupement d'informations .
- extraction ou inférence de dates à partir du contenu ou des métadonnées système .
- génération de résumés automatiques pour une première lecture rapide du contenu.

### **Identification d'échanges privés ou non professionnels**

**Visualisation filtrable des données** pour faciliter la prise de décision.

Pour répondre à ces besoins, nous avons structuré notre approche en deux étapes principales :

- une classification globale des fichiers selon leur type (document, image, audio, vidéo, fichier système...) .
- puis, un affinage ciblé par des modèles de classification spécifiques, selon le type de contenu (notamment pour les documents textuels et les images).

## 4.3 Analyse du jeu de données

Le jeu de données à traiter est hétérogène, issu de disques durs personnels ou institutionnels, et contient des fichiers de natures très diverses (documents bureautiques, images, vidéos, fichiers systèmes, etc.).

Une première étape de prétraitement a consisté à identifier et filtrer les fichiers exploitables pour l'analyse (documents lisibles, images claires, absence de corruption. . .).

Après plusieurs phases de test et d'itération, une typologie simplifiée a été adoptée pour la classification des documents textuels :

- Administratif : factures, formulaires, documents professionnels ou personnels liés à des démarches.
- Activité d'écriture : brouillons, manuscrits, publications, textes littéraires .
- Correspondance : lettres, échanges de courriels, messages personnels.

Concernant les images, la classification a été basée sur leur contenu visuel, selon les catégories suivantes :

- présence ou non de personnes .
- représentation d'un document (scanné, manuscrit) .
- icône, logo ou image symbolique .
- image hors contexte ou sans signification claire .

Cette structuration a permis de guider les phases d'annotation, de modélisation et d'évaluation tout au long du projet.

## 4.4 Synthèse

La phase d'état de l'art et d'analyse des besoins a permis de définir une approche claire et structurée de la classification automatique dans un contexte archivistique. En combinant des techniques classiques de machine learning avec des méthodes plus avancées telles que le transfert learning ou les LLM, cette approche permet d'établir une méthodologie robuste et adaptée aux spécificités des archives contemporaines. Cette synthèse servira de base pour les étapes suivantes du projet, notamment la mise en œuvre et l'évaluation des modèles de classification des documents dans la suite de ce travail.

## 5 Travail réalisé

### 5.1 Traitement du texte

Le traitement automatique des documents textuels a constitué une partie centrale du travail. Deux approches principales ont été explorées pour construire des modèles de classification capables de distinguer différents types de contenus archivistiques : l'apprentissage automatique traditionnel (machine learning) et l'apprentissage par transfert (fine-tuning de modèles préentraînés).

#### 5.1.1 Classification globale

Avant la classification fine par type de contenu, nous avons effectué une **classification globale** afin de séparer chaque fichier selon sa nature. Cette étape sert à :

- filtrer les éléments non textuels ou non exploitables .
- router chaque fichier vers le traitement adapté .
- organiser le corpus dans une arborescence claire pour les étapes suivantes.

#### Méthode

1. Extraction des métadonnées (`parserd.get("metadata", {})`) .
2. Détection du type MIME (combinaison `file / magic`) pour une identification robuste.
3. Attribution d'une catégorie et rangement dans un dossier dédié.

**Catégories retenues** Conformément à la figure de référence, les catégories utilisées sont :

- Images
- Documents
- Vidéos
- Audios
- pptx
- Web-files (HTML/CSS/JS)
- Boîte mail (EML/EMLX)
- Autres-file-sys (PLIST/INI/TMP...)
- JSON-files
- `Img_disque` (images disque)
- log-files
- Archives (ZIP/TAR/...)
- Autres (catch-all)

# Classification Globale

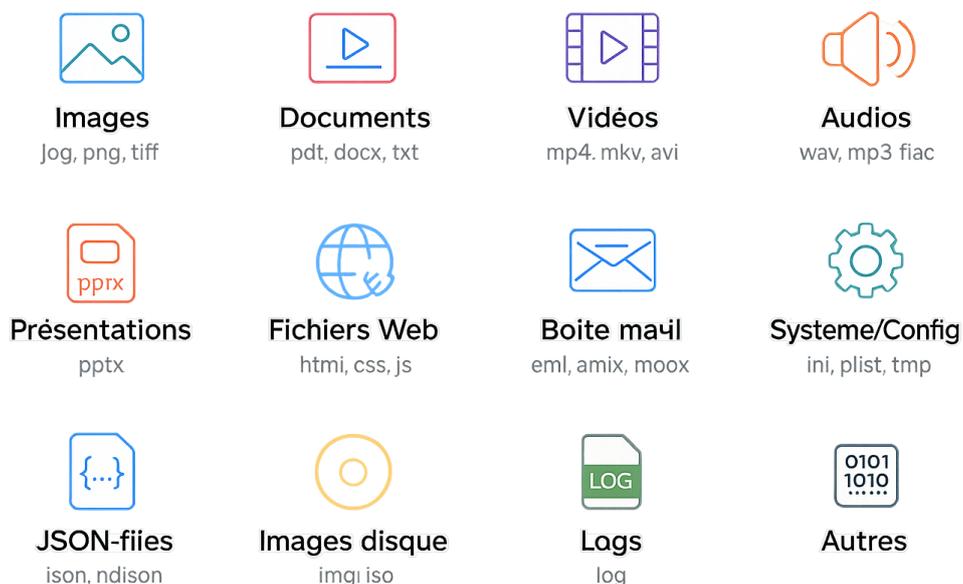


FIGURE 1 – Classification globale.

## 5.1.2 Dataset

Pour entraîner et évaluer les modèles de classification, nous avons constitué un jeu de données équilibré, composé de **200 exemples par catégorie**. Les trois catégories retenues sont : **Correspondance**, **Administratif** et **Activité d'écriture**.

### Répartition des sources par catégorie

- **Correspondance** : 125 exemples proviennent directement des fonds d'archives fournis par l'IMEC, le reste ayant été collecté depuis des sources externes.
- **Administratif** : 60 exemples issus des fonds d'archives, complétés par des documents provenant de ressources publiques.
- **Activité d'écriture** : les 200 exemples proviennent exclusivement des fonds d'archives fournis.

Les exemples externes ont été principalement trouvés sur des plateformes ouvertes telles que *Kaggle* et *GitHub*, en veillant à sélectionner des contenus représentatifs et conformes aux typologies définies lors de l'analyse des besoins.

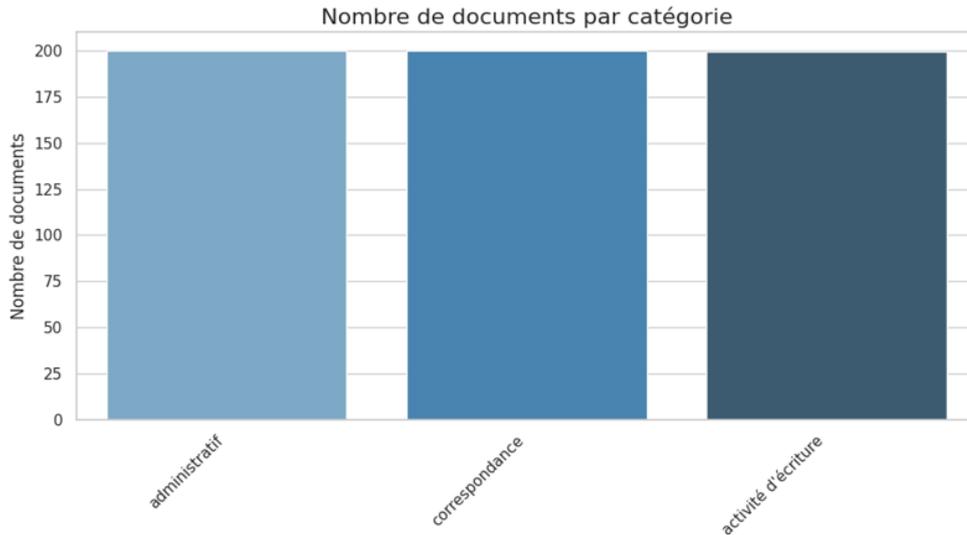


FIGURE 2 – Répartition du dataset par catégorie .

### 5.1.3 Machine Learning

La première approche a consisté à mettre en place un pipeline complet de *machine learning* supervisé pour classer les documents textuels dans trois catégories : **Administratif**, **Activité d'écriture** et **Correspondance**.

**a) Construction des features** Pour chaque document, un ensemble riche de caractéristiques a été généré, combinant données textuelles et variables numériques.

#### Caractéristiques textuelles :

- texte : contenu brut extrait .
- texte\_nettoyé : texte après nettoyage (suppression de la ponctuation inutile, des caractères spéciaux, etc.) .
- texte\_normalisé : minusculation, lemmatisation, normalisation des formes verbales .
- pos\_tags : séquence d'étiquettes grammaticales (Part-Of-Speech).

#### Caractéristiques numériques :

- Variables de fréquences/ratios issues de la NER et de champs lexicaux thématiques, p. ex. pct\_aux (verbes auxiliaires), pct\_je (pronom « je »), pct\_personne, pct\_lieu, pct\_orga, pct\_date, pct\_mois, pct\_heure, etc.
- Champs lexicaux : administratif (pct\_admin, pct\_rh, pct\_compta ...), affectif (pct\_affectif, pct\_fam), médical (pct\_medical, pct\_sante), etc.

**b) Listes de mots-clés** Des listes lexicales ont été construites pour détecter et généraliser certains contenus (ex. administratif). Elles servent à généraliser le texte via des tags et *mettre à jour* les variables numériques associées.

Listing 1 – Exemple de liste de mots-clés administratifs

```

1 mots_administratif = {
2     "dossier administratif", "formulaire", "cerfa", "attestation"
3     ,
4     "avis d'imposition", "notification officielle", "registre",
5     "permis de conduire", "document officiel", "dossier",
6     # ...
7 }

```

**c) Pipeline d'apprentissage** Trois représentations sont combinées et concaténées en un seul vecteur de caractéristiques :

1. Texte normalisé (TF-IDF) .
2. POS-tags (vectorisation catégorielle) .
3. Valeurs numériques (features NER + champs lexicaux).

Pipeline complet

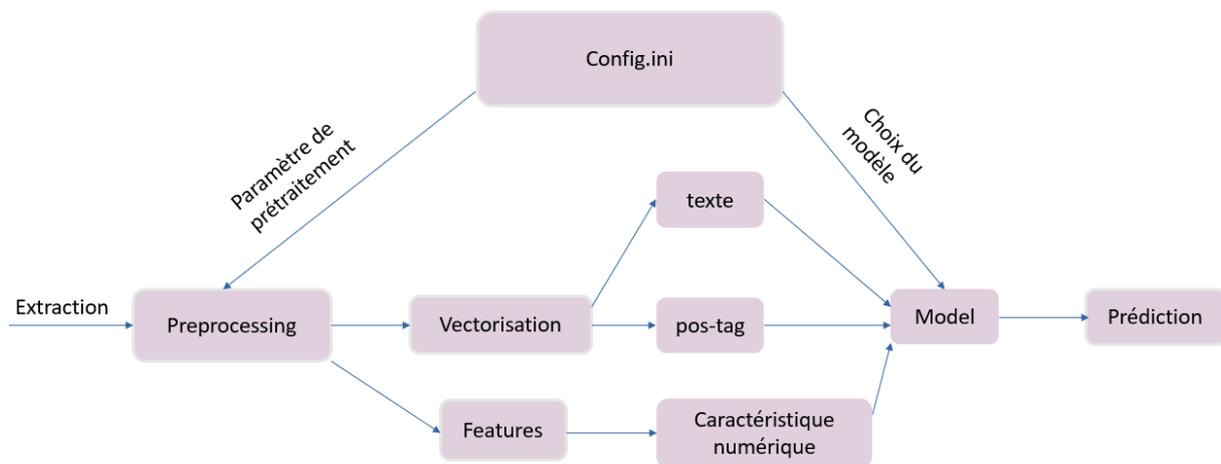


FIGURE 3 – Schéma du pipeline complet .

**d) Stratégie d'entraînement en deux étapes**

- **Phase d'évaluation** : séparation 80 % entraînement / 20 % test pour estimer précision, rappel, F1 et accuracy.
- **Phase finale** : réentraînement sur 100 % des données pour maximiser l'apprentissage (au prix de l'absence de métriques sur un jeu indépendant).

**e) Résultats** Plusieurs modèles ont été évalués : *SVM, Régression Logistique, Random Forest, XGBoost*. Les meilleurs scores (split 80/20) ont été obtenus avec *XGBoost*.

TABLE 1 – Comparaison des performances (split 80/20)

Modèle	Précision	Rappel	F1-score	Accuracy
SVM	0,88	0,88	0,88	0,88
Régression Logistique	0,88	0,89	0,88	0,88
Random Forest	0,93	0,93	0,93	0,93
<b>XGBoost</b>	<b>0,97</b>	<b>0,97</b>	<b>0,97</b>	<b>0,97</b>

#### 5.1.4 Fine-tuning (Apprentissage par transfert)

Après la mise en place d'une approche de *machine learning* classique, nous avons cherché à exploiter les avantages de l'apprentissage par transfert afin d'améliorer les performances sans repartir de zéro. Cette méthode consiste à réutiliser un modèle de langage préentraîné sur de larges corpus textuels et à l'adapter à notre tâche spécifique.

**a) Modèle choisi** Nous avons retenu **BART-LARGE-MNLI**, un modèle Transformer développé par Facebook et préentraîné sur la tâche *MNLI (Multi-Genre Natural Language Inference)*. BART se distingue par sa structure hybride :

- **Encodeur bidirectionnel** (comme BERT), permettant une analyse contextuelle fine du texte .
- **Décodeur auto-régressif** (comme GPT), optimisé pour la génération et la reformulation.

En étant entraîné à détecter si une phrase implique, contredit ou est neutre vis-à-vis d'une autre, BART-large-MNLI développe une compréhension riche du sens et du contexte, ce qui le rend particulièrement adapté à notre classification en trois catégories :

- **Administratif** : factures, contrats, formulaires, déclarations, etc. .
- **Correspondance** : lettres, e-mails, messages avec formules de politesse et signatures
- **Activité d'écriture** : manuscrits, articles, brouillons, notes.

Ces représentations préentraînées permettent de reconnaître les structures formelles d'un document administratif, les tournures propres à la correspondance, ou le style libre des textes littéraires, avant même l'adaptation aux données spécifiques de notre projet.

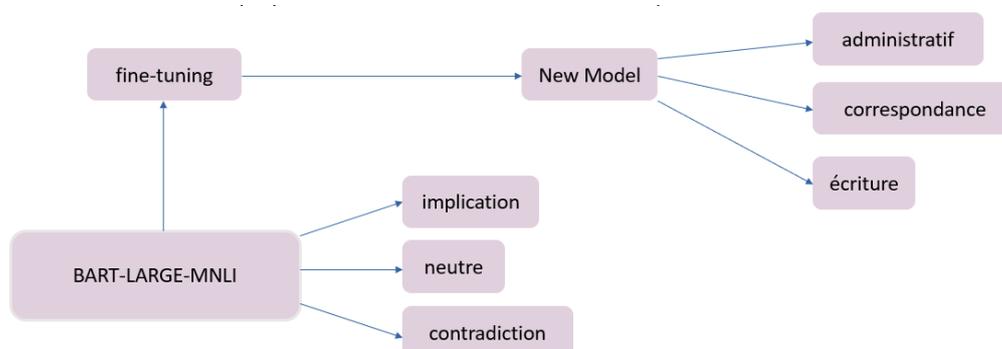


FIGURE 4 – Schéma du modèle BART et principe du fine-tuning.

**b) Méthodologie** Deux stratégies de fine-tuning ont été mises en œuvre :

- **Fine-tuning partiel** : ajout d'une tête de classification spécifique à notre tâche, seule cette tête est entraînée (les poids du modèle de base restent figés). *Avantage* : faible coût en calcul et rapidité d'entraînement.
- **Fine-tuning complet** : réentraînement de l'ensemble du modèle, y compris toutes les couches internes. *Avantage* : meilleure adaptation aux spécificités du corpus, au prix d'un temps de calcul plus important.

## Fine-tuning : complet vs partiel

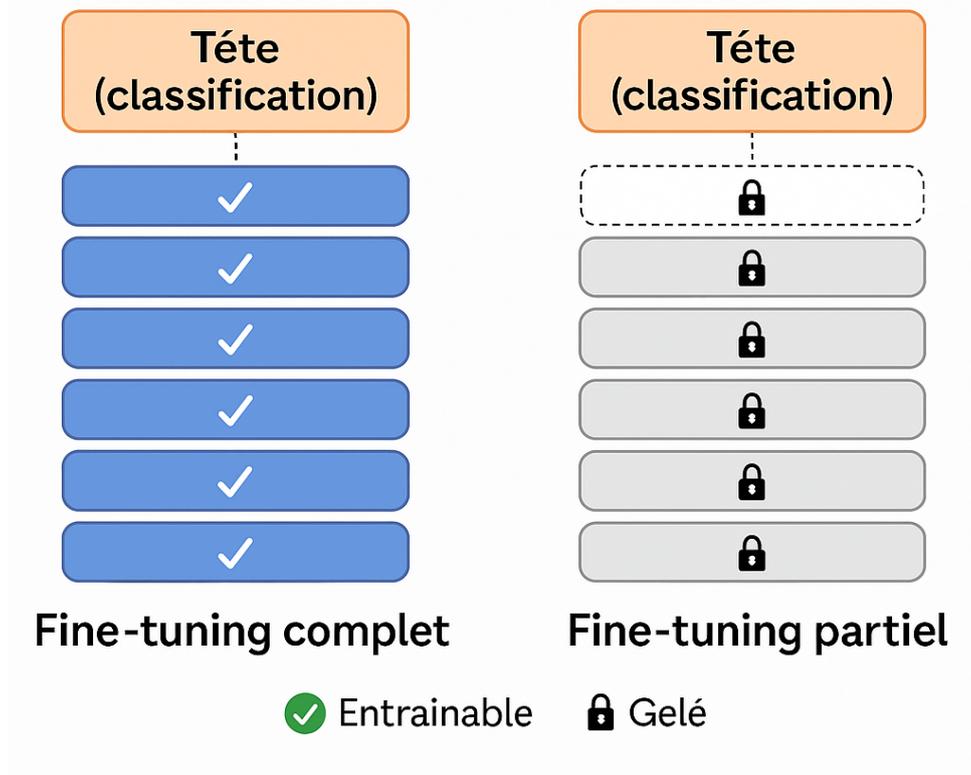


FIGURE 5 – Comparaison fine-tuning partiel vs complet.

### c) Préparation des données

- **Nettoyage** : conversion en texte brut (UTF-8), suppression des caractères indésirables et retours à la ligne superflus .
- **Mise au format CSV** : fichier contenant deux colonnes (texte, label) avec étiquetage uniforme (0 = administratif, 1 = correspondance, 2 = activité d'écriture) .
- **Mélange et séparation** : *shuffle* aléatoire (seed = 42), division 80 % entraînement / 20 % validation .
- **Tokenisation** :
  - Fine-tuning partiel : séquences jusqu'à 1024 jetons .
  - Fine-tuning complet : séquences limitées à 512 jetons.
- **Encapsulation PyTorch** : création d'une classe Dataset retournant les triplets (input\_ids, attention\_mask, label) pour le Trainer HuggingFace.

TABLE 2 – Performances comparatives du fine-tuning partiel et complet

Méthode	Loss	Accuracy	F1-macro
Fine-tuning partiel	0,66	72,5 %	72,0 %
<b>Fine-tuning complet</b>	<b>0,22</b>	<b>96,7 %</b>	<b>96,7 %</b>

**d) Résultats** Les performances obtenues montrent un avantage net pour le **fine-tuning complet**

#### Fine-tuning partiel

- **Loss:** 0,66
- **Accuracy:** 72,5 %
- **F1-macro:** 72 %

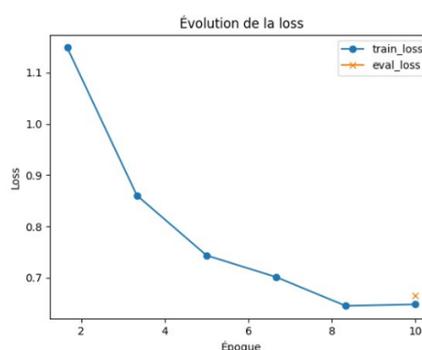


FIGURE 6 – Résultats du fine-tuning partiel.

Le fine-tuning complet apporte une amélioration de **+24 points** en accuracy et en F1-macro par rapport au fine-tuning partiel, confirmant l'intérêt de l'adaptation complète du modèle à notre corpus. Le fine-tuning partiel reste toutefois un bon compromis lorsque les ressources de calcul sont limitées.

#### Fine-tuning complet

- **Loss:** 0,22
- **Accuracy:** 96,7 %
- **F1-macro:** 96,7 %

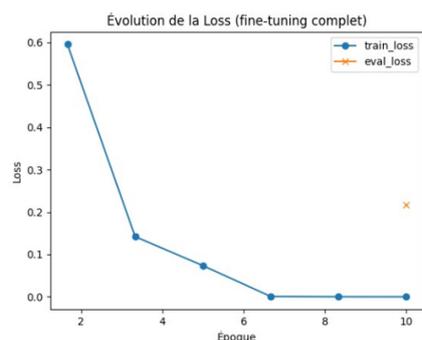


FIGURE 7 – Résultats du fine-tuning complet.

**e) Conclusion** L'apprentissage par transfert, et plus particulièrement le fine-tuning complet de BART-large-MNLI, s'est avéré particulièrement efficace pour notre classification en trois catégories. La compréhension contextuelle déjà acquise par le modèle, combinée à son adaptation à nos données spécifiques, permet d'obtenir un classifieur robuste, même avec un dataset relativement réduit.

### 5.1.5 Synthèse et comparaison des approches

Afin de disposer d'un indicateur commun d'évaluation, les modèles *machine learning* et *fine-tuning complet* ont été comparés sur l'un des fonds composé de documents réels filtrés. L'objectif n'était pas de déterminer quel modèle est le plus performant, mais d'identifier les zones d'accord et de désaccord afin de mieux comprendre leurs comportements respectifs.

**Résultats avec filtre  $\geq 200$  mots** Sur les documents contenant au moins 200 mots, les deux modèles sont d'accord dans **85,0 %** des cas, et divergent dans **15,0 %**.

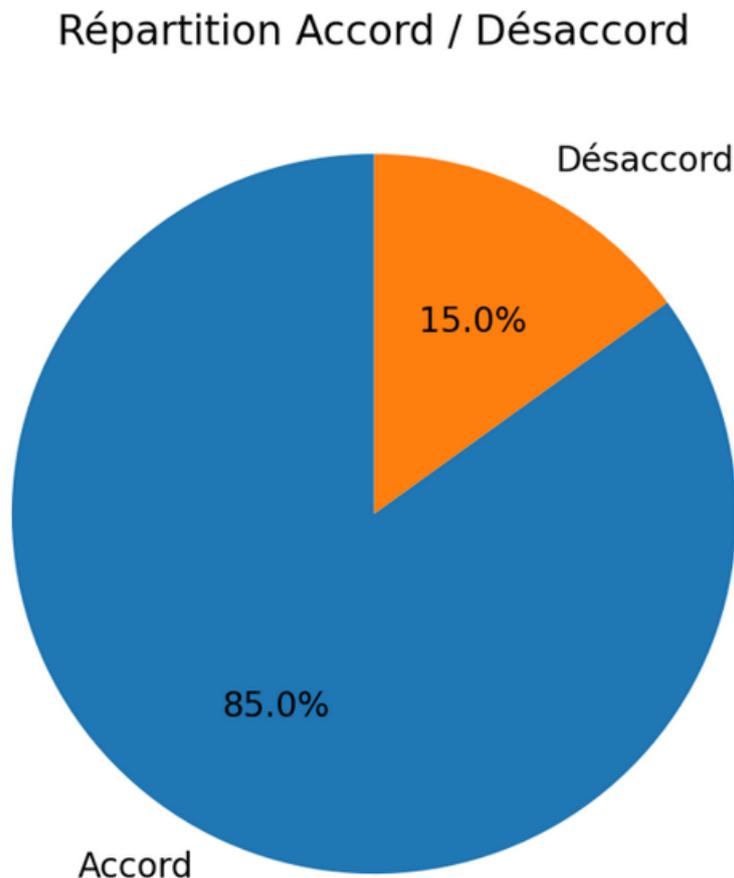


FIGURE 8 – Répartition accord/désaccord pour les documents  $\geq 200$  mots.

**Résultats globaux sur l'ensemble filtré** En considérant l'ensemble du corpus filtré (sans seuil de 200 mots), le taux d'accord global est de **79,6 %** :

- Total filtré : 1783 documents
- Accords : 1420 documents (**79,6 %**)
- Désaccords : 363 documents (**20,4 %**)

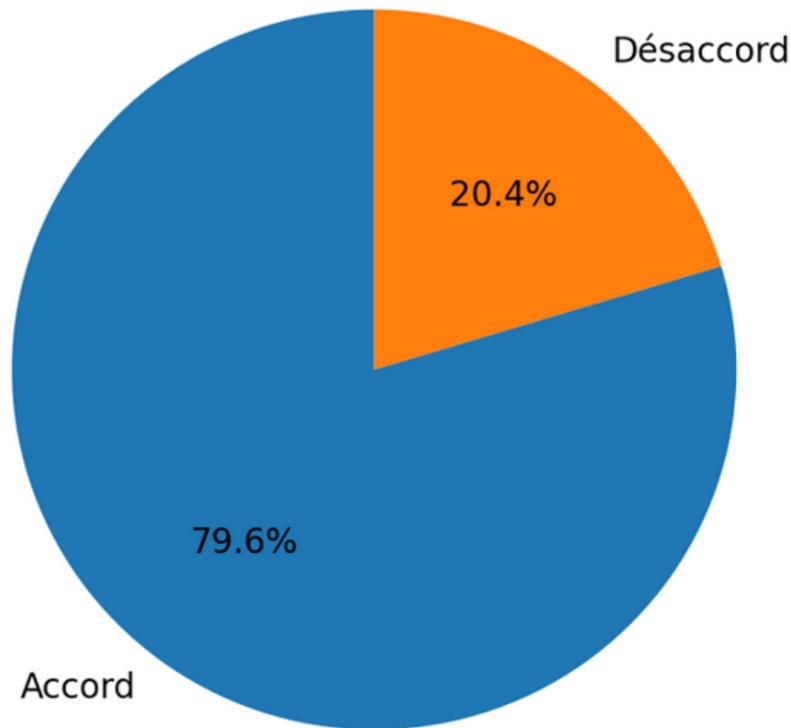


FIGURE 9 – Répartition accord/désaccord .

**Matrice de confusion croisée** La matrice ci-dessous compare directement les prédictions du *machine learning* (lignes) et du *fine-tuning complet* (colonnes) par catégorie :

TABLE 3 – Matrice de confusion croisée ML vs Fine-tuning complet

ML \ FT	Administratif	Correspondance	Écriture
Administratif	163	10	64
Correspondance	15	164	41
Écriture	183	50	1093

**Analyse des résultats Accords** : Les deux modèles convergent sur les cas les plus nets :

- **Administratif** : factures, contrats, formulaires bien structurés.
- **Correspondance** : lettres formelles, e-mails avec formules de politesse et signatures.
- **Écriture** : manuscrits, articles, brouillons littéraires.

**Désaccords** : Les divergences apparaissent sur les documents hybrides ou ambigus :

- Courriels professionnels au ton personnel.
- Brouillons contenant des mentions administratives ou légales.
- Documents mêlant narration et éléments administratifs.

Ces écarts reflètent les approches :

- **Machine learning** : très efficace sur les structures lexicales explicites, mais plus limité sur l'interprétation contextuelle.
- **Fine-tuning complet** : meilleure capture des nuances sémantiques, mais parfois influencé par des signaux faibles ou ambigus.

**Conclusion** Cette évaluation croisée montre :

- Une forte convergence ( $\approx 80\%$ ) entre les deux modèles.
- Des divergences concentrées sur les cas limites, nécessitant une vérification humaine.
- L'intérêt de considérer les deux approches comme **complémentaires** plutôt que concurrentes, afin de bénéficier à la fois de la précision lexicale du machine learning et de la compréhension contextuelle du fine-tuning.

## 5.2 Traitement d'images

### 5.2.1 Objectif et typologie

L'objectif est d'automatiser le tri des visuels afin de faciliter la gestion et la recherche dans les fonds. Après une première exploration des disques, nous avons retenu quatre classes couvrant les cas les plus fréquents :

- **Photo (person)** : image contenant au moins une personne .
- **Icon (icône/logo)** : icônes système, logos d'applications .
- **Document (scan/capture d'écran)** : scans ou écrans contenant majoritairement du texte .
- **Autre** : images résiduelles (paysages, immeubles, objets, etc.).

### 5.2.2 Jeu de données

Le dataset a été constitué pour chaque classe avec environ **4 000 images** d'entraînement et **500 images** de validation, soit un ensemble équilibré. Les images proviennent de sources variées : pour chaque classe, 200 images issues des fonds fournis, le reste provenant de sources externes (principalement *Kaggle*), afin de couvrir la diversité des styles et formats rencontrés en pratique.

### 5.2.3 Préparation des données

Pour chaque image, nous appliquons les *transforms* suivants avant l'entraînement :

- Resize carré .
- RandomHorizontalFlip (augmentation de données) .
- ToTensor .
- Normalize avec mean = [0.485, 0.456, 0.406] et std = [0.229, 0.224, 0.225] (compatibles ImageNet).

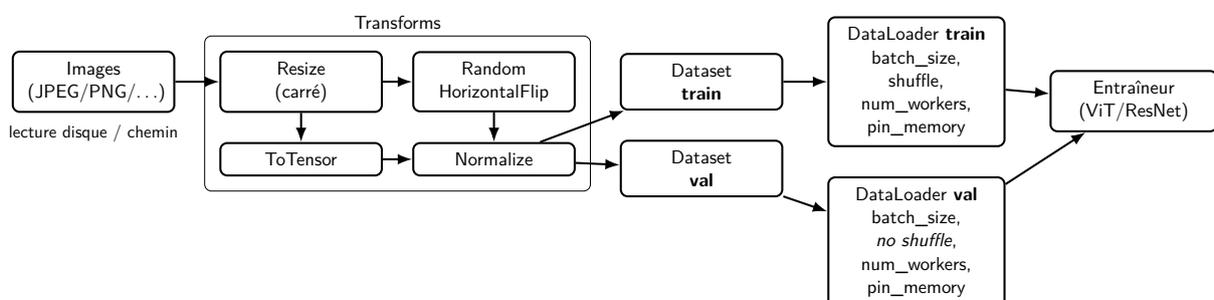


FIGURE 10 – Pipeline prétraitement .

## 5.2.4 Stratégie d'apprentissage (transfer learning)

Nous adoptons une démarche d'apprentissage par transfert, en distinguant :

- **Fine-tuning partiel** : backbone gelé, seule la tête de classification est entraînée;
- **Fine-tuning complet** : toutes les couches sont mises à jour.

Deux architectures pré-entraînées ImageNet ont été évaluées :

- **Vision Transformer (ViT, vit\_base\_patch16\_224)** : très performant et adapté aux datasets réduits, mais plus exigeant en GPU ;
- **ResNet-18** : plus léger/rapide, robuste sur les tâches classiques, mais souvent un cran en dessous de ViT sur les problèmes complexes.

## 5.2.5 Expérimentation 1 — Binaire (Photo vs Autre)

Pour sélectionner l'architecture la plus prometteuse, nous avons mené un fine-tuning complet en binaire (person vs no\_person).

- **ViT** : validation accuracy  $\approx 98\%$ , loss en décroissance régulière avec peu de fluctuations — bonne généralisation, pas d'overfitting observé.
- **ResNet-18** : moins performant mais stable.

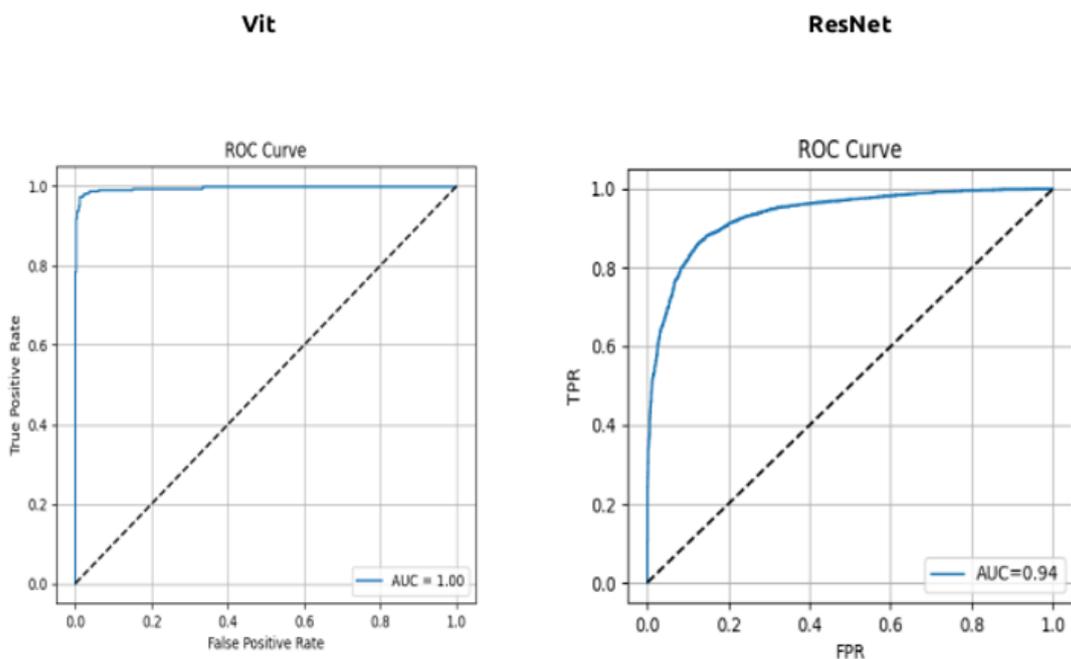


FIGURE 11 – Courbes ROC — ViT vs ResNet .

**ResNet :**

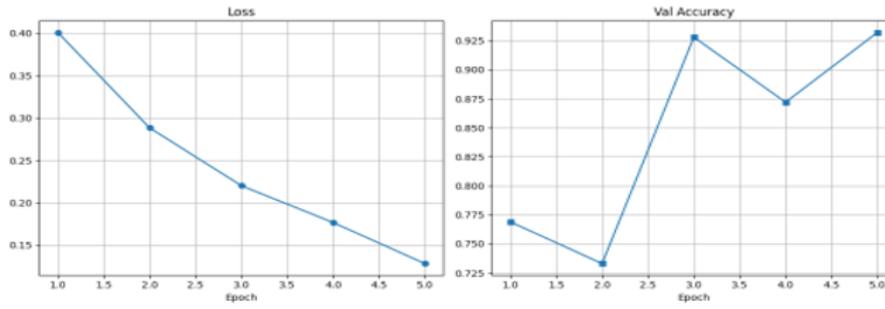


FIGURE 12 – ResNet-18 — perte et accuracy .

**Vit :**

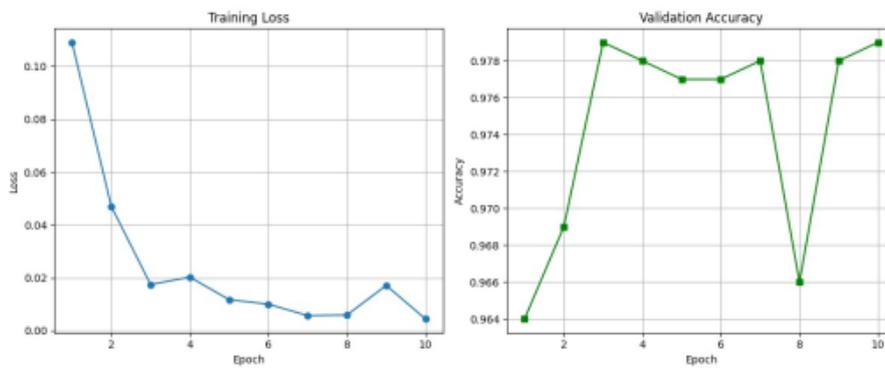


FIGURE 13 – ViT — perte et accuracy .

TABLE 4 – Rapport de classification binaire — ResNet-18.

Classe	Précision	Rappel	F1
no_person	0,80	0,93	0,86
person	0,92	0,76	0,83
Accuracy	0,85		
Macro avg	0,86	0,85	0,85
Weighted avg	0,86	0,85	0,85

TABLE 5 – Rapport de classification binaire — ViT.

Classe	Précision	Rappel	F1
no_person	0,97	0,98	0,98
person	0,98	0,97	0,98
Accuracy		0,98	
Macro avg	0,98	0,98	0,98
Weighted avg	0,98	0,98	0,98

### 5.2.6 Expérimentation 2 — Multiclasse (4 classes) avec ViT

Au vu des résultats précédents, nous poursuivons en multiclasse (4 classes) avec ViT. Résultats de validation :

- Accuracy globale = **0.99**
- Rapport de classification (val) :
  - **icon** : precision 1.00, recall 1.00, f1 1.00 (support = 500) .
  - **other** : precision 0.98, recall 0.98, f1 0.98 (500) .
  - **person** : precision 0.98, recall 0.98, f1 0.98 (500) .
  - **scan** : precision 1.00, recall 0.99, f1 1.00 (500) .
- macro avg / weighted avg = 0.99
- ROC multiclasse : AUC = 1.00 pour chaque classe.

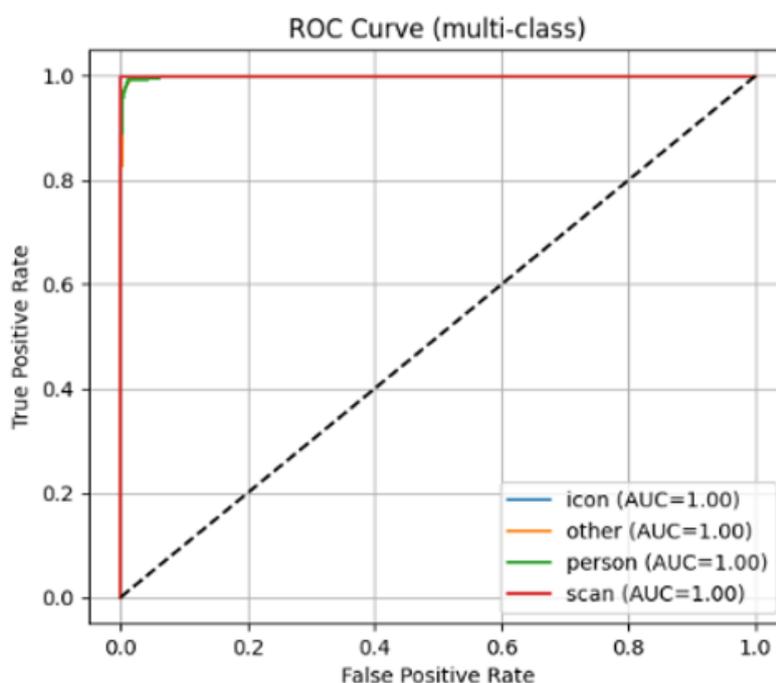


FIGURE 14 – Courbe ROC multiclasse (ViT) .

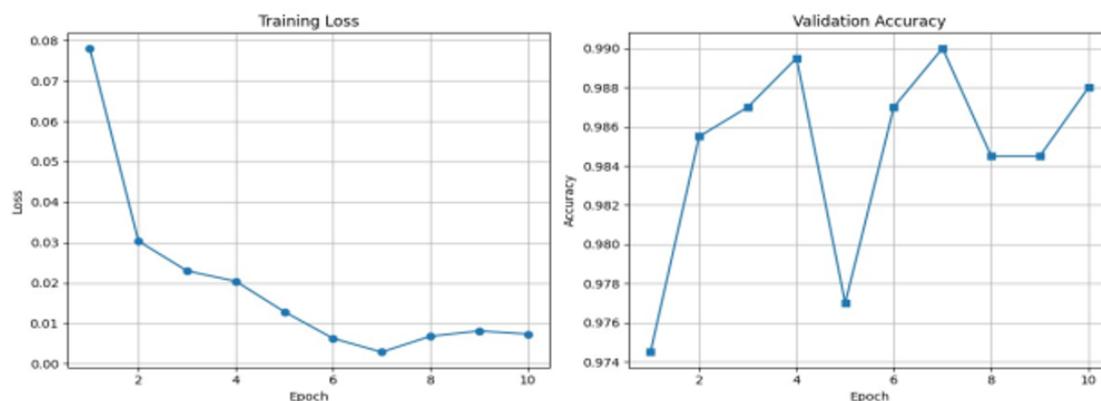


FIGURE 15 – ViT — perte et accuracy (multiclasse).

TABLE 6 – Rapport de classification multiclasse — ViT.

Classe	Précision	Rappel	F1
icon	1,00	1,00	1,00
other	0,98	0,98	0,98
person	0,98	0,98	0,98
scan	1,00	0,99	1,00
Accuracy		0,99	
Macro avg	0,99	0,99	0,99
Weighted avg	0,99	0,99	0,99

## 5.2.7 Discussion et choix final

Le **Vision Transformer** est retenu pour la suite, grâce à son meilleur compromis précision/généralisation observé en binaire puis confirmé en multiclasse.

## 5.3 Classifieur IANEC

Pour un premier livrable, nous avons livré un **classifieur IANEC** sous la forme d'une application *Streamlit* regroupant les trois modèles développés :

- (1) texte — *machine learning* classique,
- (2) texte — *fine-tuning* BART,
- (3) images — *fine-tuning* ViT (4 classes : person, icon, scan, other)

Ainsi qu'un **module de classification typée des fichiers**. L'interface permet la saisie directe ou le *glisser-déposer* de documents/images, affiche la prédiction et les scores, et propose l'export CSV. Le déploiement est immédiat grâce à un `requirements.txt` listant les dépendances et à un `Makefile` (`install/run/models`) assurant la reproductibilité ; l'application fonctionne **hors-ligne**, ce qui facilite son intégration dans l'environnement .



(a) Texte — *fine-tuning* BART



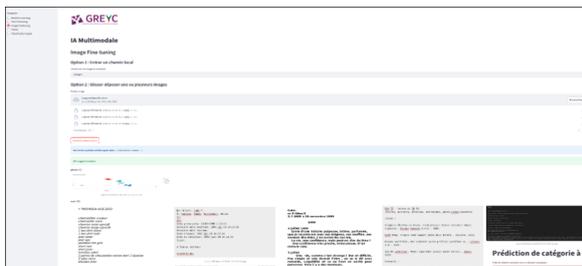
(b) Texte — ML classique



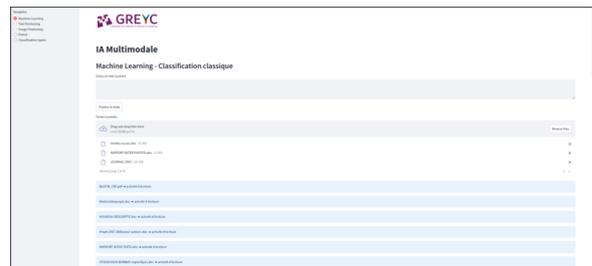
(c) Images — ViT



(d) Classification typée des fichiers



(e) Résultats image — ViT



(f) Résultats texte — ML

FIGURE 16 – Aperçu global du classifieur IANEC sous Streamlit .

## 6 Approche expérimentale

### 6.1 Clustering (recherche par similarité)

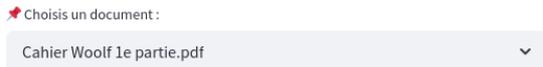
Pour compléter les classifieurs et faciliter la recherche de fichiers proches, nous avons mis en place une approche **non supervisée** de *clustering* et de recherche par similarité. Le principe est de convertir chaque document en un **vecteur numérique** puis de mesurer la proximité entre vecteurs .

**Encodage .** Nous utilisons *paraphrase-multilingual-MiniLM-L12* (Sentence-Transformers), un encodeur **multilingue** qui transforme chaque texte en vecteur dense capturant sa sémantique. Cette étape rend comparables des documents de langues et de longueurs différentes.

**Données et pipeline.** Sur un fonds (documents réels), nous :

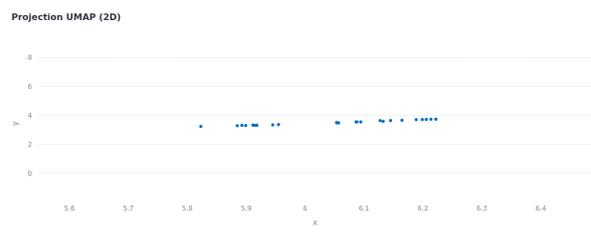
1. extrayons le texte , metadonnées (tika apache) .
2. générons une **description courte** (2–3 phrases) de chaque document via *Mistral LLM*.
3. calculons les *embeddings* pour tous les fichiers .
4. projetons les vecteurs en 2D avec **UMAP** pour visualiser le corpus (chaque point = un fichier) .
5. calculons la **similarité** et retournons, pour un fichier sélectionné, les **K** documents les plus proches (table + *heatmap*).

**Exemple .** . Lorsqu'on sélectionne par exemple *Cahier Woolf 1e partie.pdf*, la table des Top-K similaires remonte des parties du même cahier avec des scores élevés ( $> 0,8$ ), et la *heatmap* confirme la concentration locale des proximités. Cette approche s'est révélée efficace pour **suggérer des relations entre fichiers** et accélérer le repérage documentaire . elle sera réutilisé aussi comme module de recherche dans une architecture **RAG** .



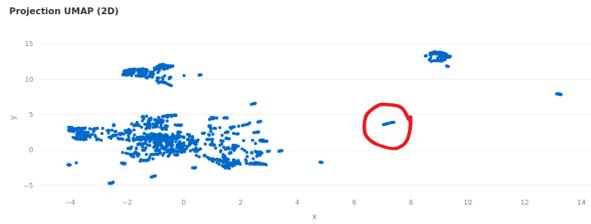
### Description

Ce document est un cahier contenant des photographies de membres de la famille Stephen et de leurs amis, ainsi que des informations de datations. Le cahier a été créé en 2009 avec QuarkXPress 6.5. Il semble avoir été assemblé à partir de plusieurs pages découpées et collées ensemble. L'objectif de ce document n'est pas clairement identifiable, mais il pourrait être utilisé pour des recherches historiques ou biographiques sur la famille Stephen et leurs relations.



(a) Sélection d'un document .

(b) Projection UMAP (2D) — zoom.



(c) Projection UMAP (2D).

(d) Interface complète .

FIGURE 17 – Clustering et recherche par similarité .

## 6.2 RAG — Recherche augmentée par génération

Pour permettre la recherche en langage naturel dans les fonds, nous avons mis en place un prototype **RAG** combinant une recherche sémantique locale et une génération de réponse appuyée par les extraits retrouvés.

### Pipeline.

1. **Pré-traitement** : extraction du texte , génération des descriptions .
2. **Embeddings** : encodage des documents avec **paraphrase-multilingual-MiniLM-L12** pour obtenir un vecteur par élément.
3. **Recherche** : pour une requête utilisateur, encodage de la requête avec le même modèle puis récupération des Top- $K$  éléments les plus proches . Un *slider* dans l'UI permet d'ajuster  $K$ .
4. **Génération** : les passages « contexte » sont fournis à *Mistral LLM* via un prompt imposant de « répondre uniquement à partir du contexte ». la réponse s'affiche avec les extraits cités.

**Exemples .** **Q1** : « y a-t-il un cahier de photographie de la famille stephan » **Q2** : « y a-t-il un cahier de photographie de la famille steeve » → le système remonte des documents décrivant des cahiers/photographies liés à la famille Stephen/Virginia Woolf dans les deux cas et synthétise une réponse confirmant la présence du cahier et ses caractéristiques. **Q3** : « qui est la famille stephan ? » → récupération de documents décrivant les liens familiaux et sociaux, puis génération d'un résumé contextualisé. **Q4** : « je cherche des correspondance de viviane » → le moteur remonte un document Word pertinent (titre, date, volume) nnnnnnnnnnnnnnnnnn n.

## Chatbot RAG sur Documents Locaux

Pose ta question et retrouve les informations pertinentes extraites de tes fichiers 📄

? Pose ta question ici

y a t-il un cahier de photographie de la famille steeve

Rechercher

(a) Accueil « Chatbot RAG » et champ de question.

? Pose ta question ici

qui est la famille stephan ?

Rechercher

### Contexte retrouvé

**Document 1015:** Ce fichier est un document Word créé par Jacques SÉDAT le 3 mai 2011 et modifié le 25 octobre 2011. Il s'agit d'une liste des membres de l'Association « Les Amis de Dominique et Jean-Toussaint Desanti ». Les noms des membres sont répertoriés avec le nom du comité, composé de personnes telles que Claude Alphandéry, Jean Bedel, Barbara Cassin, Noëlle Chatelet, Maurice Caveing, Simone Debout, Roger-Pol Droit, Geneviève Fraisse, Blandine Kriegel, Lau Kwok Ying, Patrice Lorau, Claude Mignot, Marie-José Mondzain, Pierre-François Moreau, Étienne Ollibé, Laurence Pancrazi, Dominique Pradelle et Jean-Michel Salanskis, ainsi que Michèle Sar. L'objectif de ce document est de fournir une liste des membres de l'association.

**Document 1192:** Ce document est un tableau intitulé « LEGENDES » qui représente des informations sur les relations familiales et sociales de la famille Stephen, notamment les liens entre les membres de la famille Stephen et d'autres personnalités importantes de leur époque, telles que Virginia Woolf, son mari Leonard et leurs amis. Le document a été créé par Viviane FORRESTER en 2008 et modifié en 2009. Il est composé de six pages, dont les informations sont regroupées par page. Chaque page contient des noms de personnes et des dates, ce qui suggère que ce document peut être un résumé ou une liste des membres de la famille Stephen et de leurs relations. L'objectif ou l'intention de ce document n'est pas clairement définie, mais il pourrait être destiné à des recherches ou à l'étude de la famille Stephen ou de leurs relations avec d'autres personnalités de leur époque.

**Document 1193:** C'est un document de type "leggendes" créé par Viviane FORRESTER en 2008 et modifié en 2009. Il s'agit d'une collection de légendes d'une famille, incluant des personnalités comme Virginia Woolf, Vanessa Bell, Adrian Stephen, et des membres de leur entourage. Le document est un document texte dans Microsoft Word et comprend 2 pages. L'objectif du document est de rassembler les différentes images de famille de la famille Stephen et leurs proches.

(c) Contexte retrouvé (Q2 :famille stephan).

? Pose ta question ici

y a t-il un cahier de photographie de la famille steeve

Rechercher

### Contexte retrouvé

**Document 1198:** Ce document est un tableau de photographies, intitulé "LEGENDES", qui présente des portraits de différentes personnes et des scènes de groupe, liés à la famille Stephen et aux amis de Virginia Woolf, écrivaine britannique. Il est créé par Viviane Forrester en 2008 et modifié pour la dernière fois en 2009. Le tableau de photographies est composé de 22 images, dont certaines sont datées, et est accompagné de quelques notes de texte qui identifient les personnes représentées et les dates des images. L'objectif de ce document est probablement de rassembler et d'organiser des photographies de la famille et des amis de Virginia Woolf pour une utilisation future.

**Document 18:** Ce document est un dossier de photos de Viviane Forrester, écrit en 2012 et modifié en 2013. Le document contient une liste de plusieurs images, chacune accompagnée d'une indication de lieu et de date. Les photos sont principalement des clichés de famille, mais il y a également des photographies prises lors de voyages ou de séances de travail. L'objectif de ce document est de rassembler et organiser ces photos en un seul endroit.

**Document 939:** Ce fichier est un cahier contenant des photographies de membres de la famille Stephen et de leur entourage, notamment Virginia Woolf, ainsi que des photographies de Leonard et T.S. Eliot. Le cahier a été créé en 2009 et a été produit par QuarkXPress 6.5. Le contenu du cahier est constitué de 4 pages. Le but du cahier semble être de retenir des moments mémorables de la vie de ces personnes.

(b) Contexte retrouvé (Q1 :famille steeve).

### Réponse du chatbot

Context: Document 1198: Ce document est un tableau de photographies, intitulé "LEGENDES", qui présente des portraits de différentes personnes et des scènes de groupe, liés à la famille Stephen et aux amis de Virginia Woolf, écrivaine britannique. Il est créé par Viviane Forrester en 2008 et modifié pour la dernière fois en 2009. Le tableau de photographies est composé de 22 images, dont certaines sont datées, et est accompagné de quelques notes de texte qui identifient les personnes représentées et les dates des images. L'objectif de ce document est probablement de rassembler et d'organiser des photographies de la famille et des amis de Virginia Woolf pour une utilisation future.

Document 18: Ce document est un dossier de photos de Viviane Forrester, écrit en 2012 et modifié en 2013. Le document contient une liste de plusieurs images, chacune accompagnée d'une indication de lieu et de date. Les photos sont principalement des clichés de famille, mais il y a également des photographies prises lors de voyages ou de séances de travail. L'objectif de ce document est de rassembler et organiser ces photos en un seul endroit.

Document 939: Ce fichier est un cahier contenant des photographies de membres de la famille Stephen et de leur entourage, notamment Virginia Woolf, ainsi que des photographies de Leonard et T.S. Eliot. Le cahier a été créé en 2009 et a été produit par QuarkXPress 6.5. Le contenu du cahier est constitué de 4 pages. Le but du cahier semble être de retenir des moments mémorables de la vie de ces personnes.

User: y a t-il un cahier de photographie de la famille steeve Assistant: Oui, il y a un cahier de photographie de la famille Stephen, intitulé "LEGENDES". Ce cahier a été créé par Viviane Forrester en 2008 et a été produit par QuarkXPress 6.5. Le cahier est composé de 4 pages et contient des photographies de membres de la famille Stephen et de leur entourage, notamment Virginia Woolf. Le but du cahier semble être de retenir des moments mémorables de la vie de ces personnes.

(d) Réponse du chatbot .

FIGURE 18 – Prototype RAG : recherche sémantique locale (MiniLM) et génération contrainte par le contexte (Mistral).

### Points clés & limites.

- **Multilingue** : l'encodeur gère plusieurs langues, utile pour des fonds hétérogènes.
- **Traçabilité** : affichage du **contexte** sous la réponse pour une vérification immédiate.
- **Limites** : dépend de la qualité d'extraction du texte ; risque de réponses peu sûres si le contexte est insuffisant.

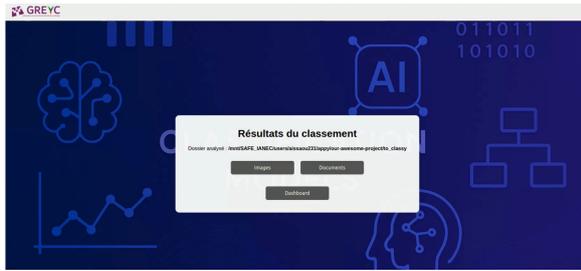
## 6.3 Classifieur IANEC 2

Pour aller plus loin, nous avons développé un prototype **classifieur IANEC 2**, intégrant les trois modèles précédents et **Pixtral OCR** pour la classification des images et l'extraction de texte manuscrit.

**Fonctionnement.** L'utilisateur importe un dossier contenant divers fichiers. L'interface propose trois entrées principales :

- **Images** : application du Vit fine-tuné et d'*Pixtral* pour une classification/description enrichie .
- **Documents** : conversion automatique en PDF et affichage en miniatures avec filtres de classification .
- **Dashboard** : statistiques globales (volumétrie, % OCR, accords/désaccords entre modèles, répartition par classes, ...).

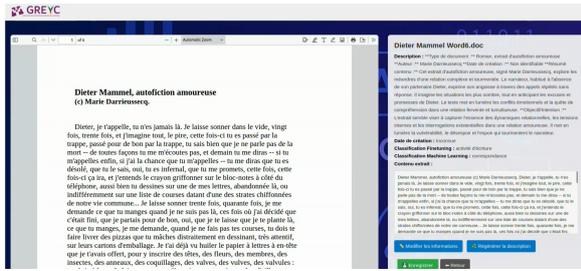
En cliquant sur un fichier (image ou document), une **page de détail** s'ouvre : résultats des modèles, *description générée* (Mistral ou Pixtral), champs éditables et **export CSV**.



(a) Écran « Résultats du classement »



(b) Galerie Images .



(c) Fiche détail : aperçu du fichier, description générée, résultats ML/FT.



(d) Dashboard

FIGURE 19 – Aperçu du classifieur IANEC 2.

## 7 Conclusion

Ce travail a posé les bases d'une **chaîne complète d'analyse d'archives numériques** : une **classification globale** des fichiers, des **modèles texte** (pipeline ML à caractéristiques + *fine-tuning* BART), un **classifieur d'images** (ViT, 4 classes), le tout **intégré** dans des prototypes utilisables (application *Streamlit* puis **IANEC 2**). Nous avons également exploré des voies expérimentales utiles à la recherche documentaire : **clustering** par *embeddings* (paraphrase-multilingual-MiniLM-L12) et **RAG** avec génération contrainte par le contexte.

Au-delà des résultats, ces développements offrent aux archivistes un **ensemble cohérent d'outils** pour trier, décrire et retrouver des contenus : import de dossiers complets, visualisation et filtres par classes, prédictions accompagnées de scores, export CSV, et tableaux de bord synthétiques.

### Bilan personnel

Je suis ravi d'avoir eu l'opportunité d'effectuer quatre mois de stage au laboratoire GREYC, où j'ai pu découvrir concrètement la *forensique* et l'investigation numérique tout en approfondissant l'intelligence artificielle. Cette expérience a été déterminante pour mes compétences techniques et ma compréhension des **enjeux archivistiques**.