Changement de nomenclature statistique à l'ère du Machine Learning: réentraînement basé sur les LLM (Large Language Models) pour la révision de la NAF

Nathan Randriamanana (*), Thomas Faria (**)

(*) Insee, Direction des statistiques d'entreprises
(**) Insee, Direction de la méthodologie et de la coordination statistique et internationale

nathan.randriamanana@insee.fr thomas.faria@insee.fr

Mots-clés : Classification de texte, codification automatique, prompt, annotation, révision de nomenclature

Domaines : Codification automatique, science des données, analyse du langage naturel

Résumé

La **Nomenclature d'activités française (NAF)** a connu une révision en décembre 2023. Dès lors, le passage de la NAF rév. 2 à la NAF 2025 impacte la codification automatique de l'Activité Principale Exercée (APE) des entreprises, à partir de sa description littérale. Cette codification représente un enjeu opérationnel majeur pour l'Insee, en particulier pour le répertoire *Sirene* (Système national d'identification et du répertoire des entreprises et de leurs établissements) qui est le garant de l'APE pour les autres administrations. Au quotidien, le code APE se retrouve notamment dans les déclarations sociales et fiscales, les bulletins de paie, les tickets de caisse et bientôt dans la facturation numérique. Au-delà de classer les entreprises par secteur d'activité dans le but de produire des statistiques et de réaliser des analyses, pour un objectif d'information économique et sociale, son impact dépasse largement la sphère de l'Insee. Le code APE a des impacts transversaux sur l'ensemble de l'écosystème économique, ses partenaires et ses clients. Sa révision se propage en cascade aux systèmes d'information des partenaires ou clients. Pour Sirene, la question se pose de savoir comment construire un modèle probabiliste capable d'attribuer le code APE en nouvelle nomenclature, malgré l'absence naturelle de données d'apprentissage, à partir du libellé décrivant l'APE et sa valeur en ancienne nomenclature.

L'article présente, premièrement, le contexte de la récente refonte de Sirene 3 à **Sirene 4**. Pour traiter les formalités d'entreprises en flux, l'application a intégré la **mise en production** d'un modèle de *machine learning* pour la codification automatique de l'APE, via la librairie *fastText*, depuis novembre 2022, pour remplacer l'ancien système expert Sicore: un choix technologique essentiel pour gérer le volume, la fréquence et la variété des descriptions textuelles soumises par les entreprises via le portail du Guichet Unique des Entreprises.

La seconde partie de l'article présente les enjeux d'annotation et le travail préparatoire des experts pour constituer la **vérité terrain** (ou *ground truth*). Jusqu'ici l'entraînement du modèle se repose sur une base d'entraînement constituée d'exemples historiques. Le changement de nomenclature rend caduque la base d'entraînement dès lors qu'il existe des codes NAF **multivoques**, c'est-à-dire qu'il existe plus d'une seule correspondance entre un code de l'ancienne version de la nomenclature et la nouvelle. Ainsi, il n'est pas possible de recoder tous les codes de la base par une bijection. A priori, en l'absence totale de données, une intervention manuelle de la part d'experts de la codification de l'APE est donc nécessaire pour **compléter la base d'entraînement**. Une vaste campagne d'annotation a donc eu lieu au second semestre de 2024 pour **obtenir des données en nouvelle nomenclature**.

Toutefois, la campagne d'annotation pose un problème d'échelle critique puisque la capacité d'annotation des experts est limitée à environ 30 000 libellés, alors que le volume à traiter excède le million d'observations. Les stratégies métiers, telles que la requalification de codes à correspondance multivoque en biunivoque et la limitation aux multivoques, ne suffisent pas à réduire significativement cet écart. Or, pour maintenir un taux de codification

satisfaisant en production et ainsi alléger la charge de reprise gestionnaire, notre modèle fastText requiert un corpus d'entraînement varié et massif (des millions d'observations). Compte tenu des contraintes de charge, il est **impossible voire contre-productif de mobiliser la seule force des experts pour constituer l'exhaustivité** de la base d'apprentissage requise.

Face à ce volume critique de données manquantes, la troisième partie de l'article présente une méthodologie innovante afin de reconstituer un jeu de données massives en NAF 2025 en particulier sur les cas multivoques en s'appuyant sur le travail et le mode opératoire des experts pour entraîner un modèle de codification automatique performant en production. La méthodologie de génération de texte présentée s'articulera sur des stratégies de génération augmentée s'appuyant sur l'usage de **Large Language Models (LLM)**, architecture actuellement la plus courante pour l'IA générative textuelle. La quatrième partie est consacrée aux résultats du réentrainement, la qualification et les enjeux de surveillance du prochain modèle en production en NAF 2025.

Ce sujet de réentrainement de modèles probabilistes suite à un changement de nomenclature peut intéresser plusieurs instituts nationaux de statistiques (INS), particulièrement pour les équipes concernées par le passage à la NACE rev 2.1. Outre la thématique de l'APE, les stratégies présentées ne sont pas restreintes à une nomenclature ou un modèle de codification automatique particuliers, sous réserve de disposer de notes explicatives de la nomenclature en question et de capacité d'évaluation.

Abstract

The overhaul of the French Nomenclature of Activities (NAF 2025) creates a critical training challenge for the automatic main activity code (APE) coding model used by the French administrative register Sirene. Business creators in France use a one-stop shop to complete their administrative procedures. In particular, they describe their main activity in writing. This activity is then coded in the NAF by INSEE using an automatic coding model based on machine learning. While the model handles the coding task, its performance requires a massive training corpus (millions of observations). The transition invalidates historical data for ambiguous cases, and manual expert annotation capacity is severely limited and unsustainable for generating the required volume. To overcome this initial data deficit crucial to model retraining, the presented work introduces an innovative methodology. The solution leverages Augmented Generation Techniques based on Large Language Models (LLMs) to generate the full NAF 2025 training base. This approach models expert operational logic, providing a scalable, reliable, and replicable solution for any National Statistical Institute facing a nomenclature change.

Introduction

De plus en plus de modèles de *machine learning* sont voués à être mis en production à l'Insee. Les domaines d'application sont multiples: intégration de nouvelles sources de données, correction d'anomalies-imputation (data editing), classification-codification automatique. En particulier, la classification automatique, dans une nomenclature prédéfinie, concerne de nombreux projets passés en production. C'est le cas de la codification de l'activité de l'entreprise en APE dans Sirene[1], à l'occasion de la bascule progressive à Sirene 4 à partir de mi-novembre 2022. À titre d'exemples, la codification à partir de libellés concerne aussi la classification des produits (en nomenclature *COICOP* et *NA2008*), des dépenses des ménages déclarées dans les carnets de l'enquête *Budget de famille* et de la *PCS 2020* dans les enquêtes annuelles de recensement. La spécificité du codage de l'APE est son utilisation quotidienne en flux du modèle dûe à la fréquence du traitement des formalités en temps quasi-réel.

Par ailleurs, l'Insee a des engagements d'évaluation et validation des données produites aux différentes étapes de la production. Ce principe d'exactitude et de fiabilité est introduit par le *code de bonnes pratiques de la statistique européenne* dans les termes suivants: **les données collectées, les données intégrées, les résultats intermédiaires et les productions statistiques sont régulièrement évalués et validés**. Cela s'applique évidemment aux modèles de *machine learning* dans le processus de production. Par conséquent, ces modèles ont besoin de maintenance dans le temps.

De surcroît, Sirene est un répertoire administratif et en tant que répertoire de référence, des méthodes rigoureuses et éprouvées doivent *assurer la qualité de la codification* des variables. Ainsi, le traitement de l'APE repose sur une approche algorithmique, probabiliste mais aussi sur une reprise des cas complexes par les gestionnaires. Pour la tenue d'un répertoire, il est indispensable que :

- des procédures performantes de codification soient mises en œuvre
- les procédures automatiques soient contrôlées par des expertises individuelles
- le responsable du répertoire exerce une veille sur l'état de l'art en matière de techniques de codification pour améliorer son efficacité.

La qualité des modèles de *machine learning* impacte directement la qualité de la codification. Il y a donc un enjeu de qualité de ces modèles. En effet, un modèle d'apprentissage, supervisé ici, est entraîné pour résoudre une tâche à partir de *données de référence*. Or, les données réelles peuvent dévier de ces données de référence dans le temps pouvant ainsi mener à une perte de performance.

C'est ce qu'on appelle une *dérive des données* et notamment une *dérive conceptuelle* lorsque la relation se dégrade entre les entrées (X) et la cible (Y). Pour maintenir voire améliorer la qualité de codification et donc du répertoire, le réentrainement du modèle devient ainsi nécessaire. À plus forte raison, pour un répertoire de référence, une **surveillance** par des expertises individuelles doivent assurer la qualité de la codification. Qui plus est, un répertoire est un *système vivant*: de nouvelles activités apparaissent, d'autres disparaissent, et les descriptifs textuels peuvent évoluer au fil des années. Tout particulièrement, les libellés d'activités sont envoyés à Sirene par le portail du *Guichet Unique des Entreprises* (GUE) de l' *Institut National de la Propriété Industrielle* (INPI), plateforme sur laquelle les déclarants renseignent les informations de leur entreprise dont leur activité principale, ce qui implique que toute modification du formulaire ou de son interprétation est susceptible de provoquer une dérive de données.

Un cas extrême de dérive conceptuelle, courant dans le cas de la statistique publique, se produit ici: un changement de nomenclature, celui de la NAF. Il s'agit d'une rupture totale et exogène de la définition même des modalités de la variable cible, rendant les codes non bijectifs immédiatement obsolètes. Une simple application de la table de correspondance sur le jeu d'apprentissage ne saurait

[1] Du *data editing* est aussi employé pour corriger la base d'apprentissage dans une moindre mesure.

suffire dans la mesure où des sous-classes de la nomenclature, et donc des activités, ne seraient pas représentées. Par conséquent, ces données du jeu d'apprentissage sont incomplètes pour entraîner un modèle performant en terme de précision et d'automatisation.

Pour coder des millions de données en nouvelles nomenclature, il faut une expertise métier et un volume d'annotation conséquent. L'expert APE dispose de la connaissance métier mais ne peut traiter l'exhaustivité manuellement. Comment passer à l'échelle et tirer profit du travail et du savoir-faire des experts ?

Après avoir présenté le contexte de traitement en flux dans lequel s'inscrit le modèle de machine learning actuellement en production et son évolution, la seconde partie de cet article s'articulera autour de la problématique de la correspondance entre la NAF rev 2 (ou NAF 2008) et la NAF 2025 pour la conversion du jeu d'apprentissage, y compris la campagne d'annotation débouchant sur une vérité terrain et l'explicitation du mode opératoire des experts sur cet exercice. Puis, la troisième partie décrira la génération de données d'apprentissage par LLM calibrées sur la vérité terrain, en s'appuyant sur la table de correspondance théorique et les notes explicatives de la nomenclature pour augmenter le contexte. Ensuite, la quatrième partie présentera la construction du modèle d'apprentissage supervisé adapté en nouvelle nomenclature, ses performances et l'approche pour le corriger et le qualifier pour un passage en production. Enfin, des pistes prometteuses d'améliorations seront illustrées, ainsi que des perspectives de cas d'usage au sein de répertoires de référence de la statistique publique intégrant des systèmes de codification.

1. La codification APE dans Sirene 4, une nouvelle application pour traiter les déclarations d'entreprises en continu

Cette partie donne le contexte ayant permis et conditionnant la mise en production actuelle et à venir du modèle d'apprentissage automatique. Il s'agit de la genèse de notre base d'apprentissage et de notre modèle. Il est important pour être en mesure de saisir les conséquences transversales du changement de nomenclature et du rôle d'un nouveau modèle dans ce nouveau contexte.

1.1. Le Guichet Unique des Entreprises

Dans le cadre de la loi *PACTE (n° 2019-486 du 22 mai 2019)*, la mise en place d'un *guichet unique* pour remplacer *les centres des formalités des entreprises (CFE)* est décrite en ces termes: **à partir du 1er janvier 2023, l'ensemble des formalités d'entreprise doit être déposé en ligne auprès d'un organisme unique** et ce, afin de dématérialiser les démarches ainsi que d'avoir un unique point d'entrée pour effectuer les différentes formalités de création, de modification et de cessation d'activité directement en ligne sur le guichet

Par conséquent, le *guichet unique des entreprises* est le principal flux d'alimentation des déclarations du répertoire Sirene qui les reçoit par envoi du portail du guichet unique. L'envoi des liasses se fait en temps quasi-réels dans la mesure où les informations sont envoyées à Sirene dès que le déclarant a terminé sa démarche sur le portail. Le déclarant peut être un chef d'entreprise, un salarié-délégataire ou un mandataire.

Parmi les informations, est déclarée la description textuelle de l'activité principale de l'entreprise **ou libellé d'activité**. Avant la mise en place du guichet, un gestionnaire accompagnait le déclarant dans la manière de rédiger ce libellé d'activité. Le libellé était donc méticuleusement formaté et harmonisé, de sorte que l'attribution d'un code d'activité[2] par le système expert Sicore s'en trouvait grandement facilitée. Pour connaître les limites de Sicore, il est recommandé de lire l'article de la dernière session des JMS à ce sujet bien exposées par T. Leroy [1].

Dorénavant, le déclarant n'est plus encadré par un gestionnaire ; la saisie s'effectue en totale autonomie via un champ libre non structuré sur le portail en ligne. Si l'absence de contraintes est un avantage, elle expose au risque d'erreurs de saisie notamment les fautes syntaxiques et à une incompréhension de ce qui doit être saisi en tant qu'activité principale, ce qui exige un système de codification plus élaboré pour s'adapter à ce choc.

C'est dans ce contexte que Sirene et ses gestionnaires ont dû être en mesure de gérer un flux constant de libellés d'activité à codifier. Pour cela, ils ont mis en place une méthodologie adaptée pour gérer à la fois le volume, la fréquence et la variété des descriptions saisies, qui se comptent par millions annuellement, avec notamment des libellés pouvant être déroutants.

1.2. Sirene 4: une nouvelle refonte face aux demandes de modernisation

Sirene est un répertoire historique de l'Insee, dont l'application de gestion porte le même nom. Avant d'aborder la réponse qu'apporte Sirene 4 face à ce contexte, rappelons brièvement la raison d'être et le rôle de ce répertoire de référence, qui sont encore d'actualité, particulièrement dans ce contexte.

[2] code APE ou code NAF

1.2.1. Le répertoire Sirene

Sirene *« Système Informatique pour le Répertoire des Entreprises et de leurs Établissements »* est un répertoire historique de l'Insee créé (par le décret n°73-314 du 14 mars 1973) dont la gestion a été confiée à l'Insee.

Ce répertoire de référence enregistre l'état civil de toutes les entreprises et de leurs établissements et gère leur cycle complet: de leur naissance à leur cessation. C'est un processus vivant qui continue à traiter les différents évènements du cycle de vie de l'entreprise.

Sirene est notamment garant juridique du SIREN, du SIRET[3] et du code APE. Ce rôle demande une disponibilité de l'application de gestion et de ses gestionnaires pour traiter des formalités au quotidien.

En tant que répertoire de référence, les exigences de qualité du code APE, la disponibilité de l'application et la nécessité d'un traitement régulier des formalités sont inhérentes à sa mission historique. Ce rôle exige toujours la mobilisation constante des gestionnaires.

1.2.2. La bascule de Sirene 3 à Sirene 4: une rupture et une évolution

Historiquement, particulièrement dans Sirene 3, ce processus s'effectuait par le traitement de formalités en batch (ou en lots), avec des interventions groupées à intervalles réguliers.

Ceci exige de l'Insee qu'elle soit capable de répondre rapidement. Le traitement en flux continu impose une **gestion unitaire et très réactive des dossiers d'entreprises**, avec la flexibilité nécessaire pour les débloquer. C'est une performance que le traitement par lots (batch) peut difficilement garantir, ce dernier nécessitant d'attendre la prochaine transmission.

C'est précisément en ce sens que Sirene 4 marque un véritable changement de paradigme, permettant de quitter définitivement le monde des batches. La modernisation du répertoire s'appuie sur une architecture en micro-services qui rend possible la gestion continue des formalités.

Au cœur de cette architecture, chaque module est autonome et orchestré par un workflow structuré qui appelle les services métier. L'architecture permet à un service d'appeler des API internes ou externes à différents endroits de l'application selon les besoins. Cela offre la souplesse d'utiliser le langage le plus adapté pour des cas d'usage spécifiques, par exemple en intégrant un modèle de codification exposé via une API Python. Cette organisation assure un couplage faible et facilite l'évolution du système ainsi que l'intégration de nouvelles fonctionnalités.

Le workflow séquence de manière souple les différentes étapes du traitement d'une formalité, incluant notamment son analyse métier et la codification, assurant la flexibilité nécessaire à la gestion unitaire des dossiers. En somme, l'automatisation des tâches répétitives soulage le circuit et permet aux équipes de se concentrer sur les cas complexes à forte valeur ajoutée. Cette réorganisation des tâches pose de nouveaux défis organisationnels et structurels.

Par ailleurs, l'outil Sicore ne convenait plus face au **format non-structuré des libellés provenant du Guichet Unique**. Son taux de codage automatique n'atteignait que 30%, impliquant près de 70% de reprise manuelle et surchargeant considérablement les équipes.

De plus, le fonctionnement du système expert Sicore était fortement adhérent à l'ancienne architecture de l'application de gestion du répertoire (Sirene 3). La nouvelle architecture en micro-services de Sirene 4 a permis de découpler cet outil, rendant le système de classification moins dépendant de l'application principale. Il a ainsi été possible de le remplacer par un modèle d'apprentissage supervisé, dès le début du projet, tirant pleinement parti de la modularité de Sirene 4.

[3] respectivement les identifiants uniques des entreprises et leurs établissement

C'est dans cette logique de modularité et de performance qu'il est pertinent d'expliquer comment s'articule un modèle dans le nouveau système de codification.

1.3. Le système de codification dans Sirene 4

Face à cette concentration de l'expertise humaine sur les cas complexes, la performance du système de codification lui-même devient un enjeu majeur. Il est donc essentiel d'en détailler le fonctionnement. La codification APE s'appuie sur deux piliers:

1. La codification automatique
2. La reprise gestionnaire

1.3.1. Le traitement automatique des libellés

Le système de codification de Sirene 4 gère les flux hétérogènes en combinant le libellé d'activité brut et des variables annexes dont catégorielles comme la catégorie juridique (dont certaines sont spécifiques aux administrations), le domaine et le sous-domaine d'activité d'une association.

Le processus est séquentiel et vise à garantir la méthode la plus fiable en premier :

1. **codification déterministe** : Tentée en premier lieu, elle s'appuie uniquement sur des variables catégorielles annexes pour les cas simples (ex: associations, institutions publiques). La tentative de codage est effectuée séquentiellement : d'abord à partir de la catégorie juridique, puis, si elle échoue, une tentative est appliquée sur la nature d'activité, et ainsi de suite sur les autres variables jusqu'au passage à la méthode suivante.
2. **rejet automatique**: En cas d'échec du déterministe, un filtre rejette les déclarations invalides pour soumission au modèle de classification. Ce rejet concerne les libellés vides de sens (exemple : « idem ») et sur l'utilisation d'expressions régulières pour détecter les saisies invalides, comme la simple mention d'un code APE dans le champ. Ces cas activent le retour de la formalité au portail du guichet unique pour demander au déclarant des précisions.

   Note : Ce filtre peut évoluer avec le changement de la nomenclature d'activités française (NAF), car certains libellés pourraient devenir incodables dû à des activités devenus caduques dans la nomenclature. L'extension du rejet à d'autres libellés incodables est une question en cours d'étude pour soulager la charge de travail.
3. **codification probabiliste**: concerne principalement l'usage d'un classifieur probabiliste, déclenchée si les étapes précédentes échouent, elle cible les libellés non structurés en exploitant le texte libre et les variables annexes pour proposer un code APE avec un niveau de confiance.

Le schéma ci-dessous illustre le séquencement des étapes.

La modularité de l'architecture Sirene 4 (découplage par micro-services) permet de remplacer la codification probabiliste sans impacter les étapes précédentes. Cela a facilité sa migration rapide vers un appel d'API Python déployée sur Kube (la plateforme des nouvelles applications), favorisant son évolution. À terme, l'API entière intégrant toutes les étapes de codification peut également être migrée sur Kube, en bénéficiant de cette faible adhérence.

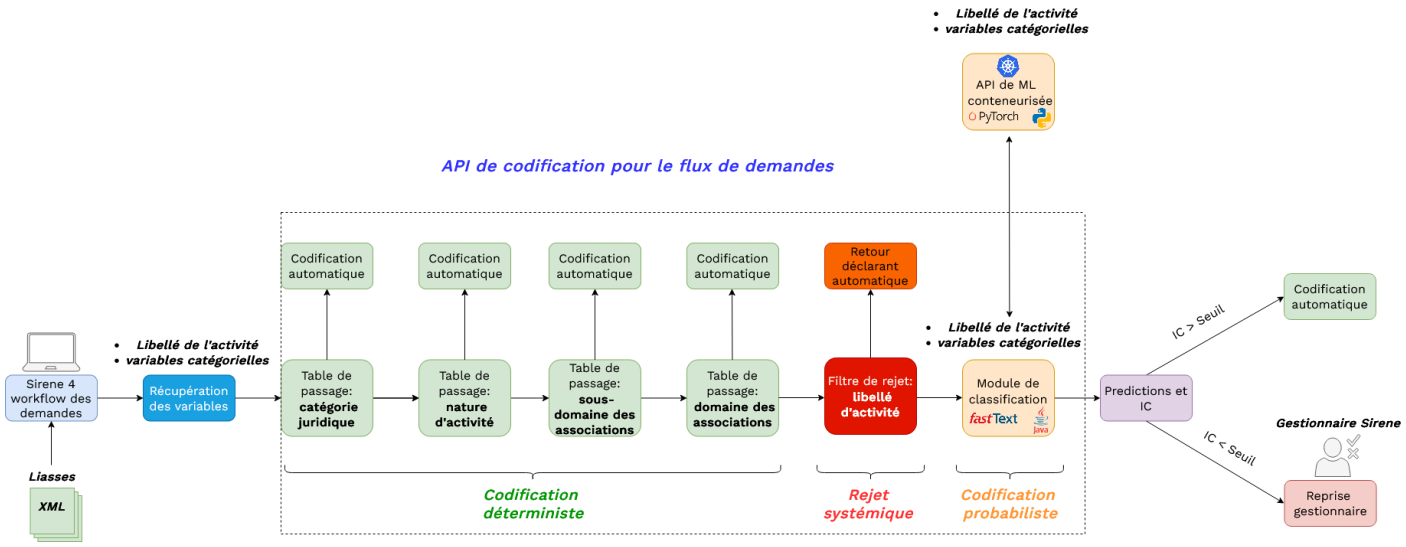


Figure 1. – Détail de la codification de l'APE dans Sirene 4

En résumé, le processus de codification automatique est le suivant : le système tente d'abord une codification déterministe. En cas d'échec, il passe à la codification probabiliste. Si le résultat probabiliste n'atteint pas un critère de *seuil*, le système conclut à un échec de codification automatique et renvoie le dossier en reprise manuelle.

1.3.2. Critère de passage en reprise

Le mécanisme pour valider ou rejeter la codification probabiliste s'appuie sur un *indicateur de Confiance* (IC). L'objectif est simple : séparer les bonnes codifications des mauvaises ou incertaines afin de garantir la qualité des données.

1.3.2.1. L'Indicateur de Confiance (IC)

L'Indicateur de Confiance est une mesure de la confiance accordée à la prédiction du modèle. Il se calcule comme la différence entre la probabilité de la meilleure prédiction (p1) et celle de la deuxième meilleure prédiction (p2):

$$IC = p_1 - p_2,$$

où p_i est la probabilité de la i-ème prédiction, les probabilités étant triées par ordre décroissant.

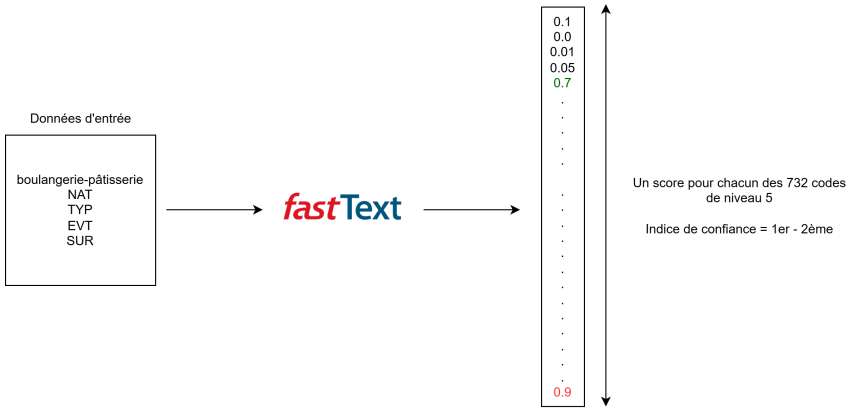


Figure 2. – Illustration du calcul de l'indice de confiance par un modèle fastText

Le calcul se fait par la différence des deux premières probabilités. Ce calcul est valable pour tout modèle de classification supervisée qui fournit des probabilités en sortie

1.3.2.2. Le Critère de décision

Le critère de décision est un choix binaire, basé sur un *seuil* prédéfini :

$$\text{Ind}_{\text{Automatique}} = \mathbf{1}_{\{IC \geq \text{ seuil}\}} = \begin{cases} 1 \text{ si } \ IC \geq \text{seuil}, \\ 0 \ \text{sinon.} \end{cases}$$

Le risque majeur de ne pas fixer de *seuil* serait d'accepter des codifications complètement aléatoires et incertaines, ce qui nuirait gravement à la qualité du répertoire. Le *seuil* théorique recommandé se situe entre 0.65 et 0.7 pour le modèle *fastText*[4]. Par précaution, Sirene 4 avait initié un *seuil* de 0,8 qui a depuis été abaissé plus proche du *seuil* recommandé pour soulager la charge.

1.3.2.3. Arbitrage entre la qualité et l'automatisation

Le choix du *seuil* de l'Indicateur de Confiance (IC) est une décision stratégique qui établit l'équilibre entre la productivité et la qualité. Baisser le *seuil* allège la charge des gestionnaires mais augmente le risque d'erreur, tandis que l'augmenter garantit la qualité mais surcharge les équipes. L'objectif est de maximiser l'automatisation tout en maintenant le taux d'erreur sous contrôle.

La distribution des indicateurs de confiance permet d'objectiver cet arbitrage. Le graphique ci-dessous illustre comment, selon le *seuil* choisi, on parvient à séparer plus ou moins fortement les distributions des bonnes et des mauvaises codifications.

⚠ Le *seuil* : Une décision méthodologique et métier

Le *seuil* n'est pas un simple paramètre opérationnel, mais un élément méthodologique sensible. Toute modification doit être rigoureuse : un abaissement mal contrôlé risquerait de polluer la base d'apprentissage du modèle, mettant en péril la performance à moyen-long terme.

En définitive, la détermination du *seuil* est un arbitrage répondant aux besoins du métier. Pour une prise de décision quantitative fine, un tableau de bord (dashboard) peut être fourni au *product owner* ou responsable de l'application, permettant de suivre l'impact du *seuil* sur les métriques clés. Ce besoin de surveillance constante fera l'objet de la dernière sous-partie.

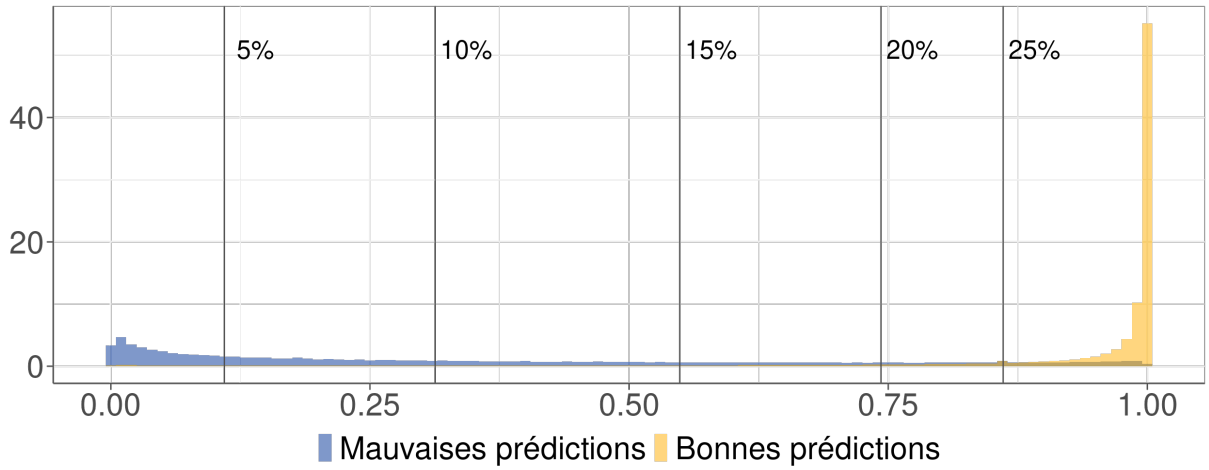


Figure 3. – Distributions des bonnes et mauvaises codifications selon IC

[4]Le position du seuil dépend du modèle d'apprentissage. Si une dérive des données réduit la performance du modèle, le seuil devient insuffisant. Face à la dégradation, la solution est le réentraînement du modèle. L'augmentation temporaire du seuil est une alternative qui, cependant, n'est pas toujours suffisante en cas de forte baisse de performance.

Approche méthodologique partagée par d'autres projets de codification automatique à l'Insee. Sur un échantillon de test. L'axe vertical représente la fréquence des codifications pour chaque niveau d'IC. Les lignes verticales indiquent, pour différents seuils d'IC, la part de codifications devant être reprises manuellement. Par exemple, avec un seuil de 0,75, environ 20 % des codifications nécessitent une reprise, soit 80 % peuvent être automatisées.

Afin de soulager la charge des gestionnaires et d'augmenter l'automatisation, le seuil de l'indicateur de confiance, initialement fixé à 0,8 par prudence et supérieur à la recommandation théorique de 0,65-0,7, a été abaissé pour se rapprocher de la valeur théorique.

1.3.3. La reprise gestionnaire des cas incertains

Bien que la codification automatique soit la plus prépondérante[5], le rôle du gestionnaire est crucial pour traiter les cas incertains et garantir la qualité.

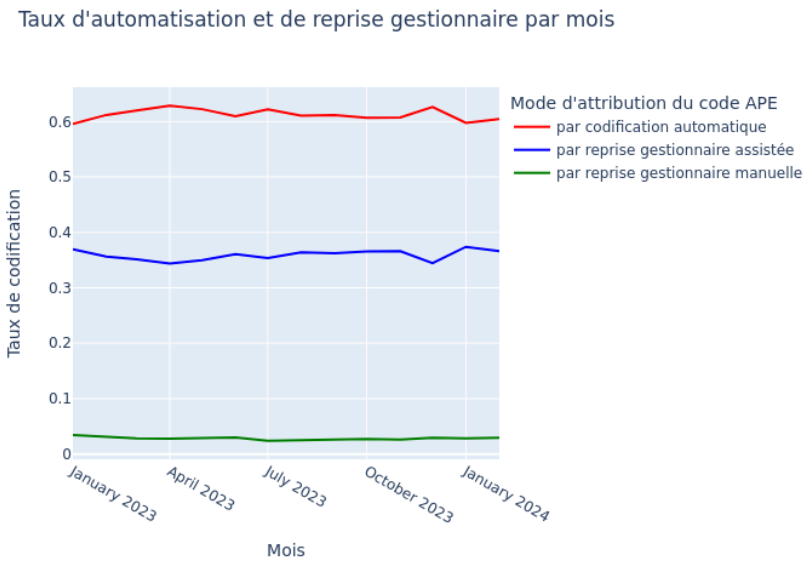


Figure 4. – Représentation temporelle de la part des différents types de codification

Est représenté ici, une année complète de codification APE

1.3.3.1. Aide à la codification et ergonomie

Lorsque le système envoie un dossier en reprise, le modèle est intégré pour aider à la codification :

- Il propose jusqu'aux cinq codes les plus probables et n'affiche pas ceux ayant un Indicateur de Confiance (IC) trop bas, afin d'assurer l'ergonomie et d'éviter des propositions aléatoires.
- Si le gestionnaire n'est pas satisfait ou si aucun code n'est proposé, il peut relancer des propositions en modifiant ou en saisissant un nouveau libellé, ou en recherchant directement dans la nomenclature via un moteur de recherche.

[5] de l'ordre de 2 millions de codifications l'année

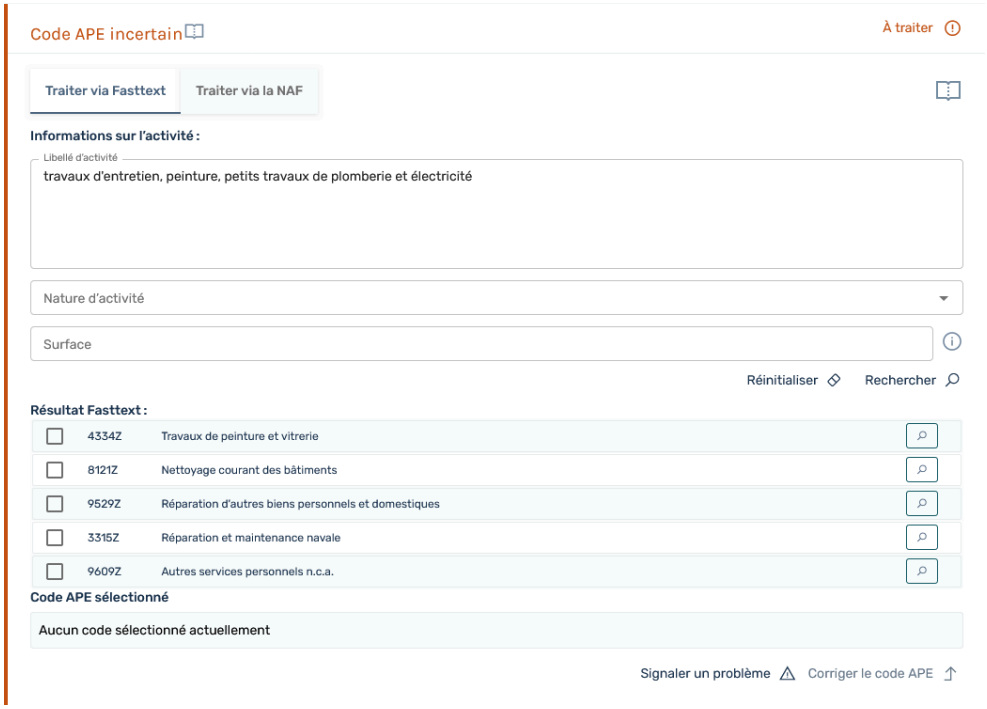


Figure 5. – Exemple de poste de reprise de la codification dans Sirene 4

Le modèle hésite entre plusieurs codes possibles. La proposition des échos permet au gestionnaire de coder plus vite, ce qui lui fait gagner du temps de recherche.

1.3.3.2. Signalement et retour au déclarant

Le gestionnaire peut rejeter manuellement la demande avec le bouton **Signaler un problème**. Cette action produit un retour immédiat au Guichet Unique pour demander au déclarant des précisions, selon les indications fournies par le gestionnaire.

L'efficacité opérationnelle repose sur la capacité du modèle à gérer la variété, la fréquence et le volume des libellés dans le flux continu. Ces contraintes de production sont fortes et dépassent la seule précision statistique.

1.3.4. Choisir un bon modèle d'apprentissage supervisé

Pourquoi ne pas déployer dès maintenant en production l'un des modèles de dernière génération, tels que les LLM, afin de tirer parti des avancées les plus récentes ?

Un modèle adéquat pour Sirene 4 ne se résume pas à atteindre l'état de l'art, aussi appelé State of the Art ou SOTA. Son intégration en production nécessite de trouver un équilibre entre performance, valeur métier et pragmatisme. Il ne s'agit pas de viser la sophistication maximale, mais plutôt d'opter pour une solution utile, maîtrisable et durable, développant itérativement le degré de maturité technique et fonctionnelle[6] nécessaire avant d'introduire des modèles plus avancés.

Par ailleurs, l'adoption en environnement de production engage plusieurs facteurs critiques, cités à titre d'exemple :

[6] notamment en terme de LLMops, plus généralement MLops et la progression métier en terme d'organisation des remontées des gestionnaires et de capacité d'évaluation.

- **contraintes d'architecture et de coût** : le modèle doit être compatible avec une architecture open source et maîtriser le coût d'entraînement et d'inférence, notamment la compatibilité des ressources si une évolutivité verticale est requise.
- **taux d'automatisation** : pour limiter la charge de travail.
- **facteurs de pérennité** : le choix d'une librairie standard est privilégié pour faciliter la maintenance du code source et minimiser la vulnérabilité de la librairie sur le long terme.
- **stabilité**: la robustesse du modèle face aux données bruitées ou aux légères dérives des données et précision.
- **précision**: l'exactitude des classifications.
- **temps d'inférence**: la rapidité de la prédiction, essentielle pour le traitement en flux continu.
- **explicabilité** : la capacité à justifier le choix du modèle, facilitant la confiance et le débogage.

L'architecture micro-services de Sirene 4 a permis d'itérer rapidement sur les choix de modélisation :

1. **première implémentation (FastText)** : la première version a été déployée en s'appuyant sur la librairie FastText. Ce choix open source était adapté pour une mise en production rapide, offrant un bon compromis entre la vitesse et la précision, avec un faible coût de ressources sur CPU. Toutefois, cette librairie limite l'évolution du modèle de codification en termes de maintenance, prise en main et de performances.
2. **seconde implémentation (PyTorch via torchTextClassifiers)** : La transition vers PyTorch est stratégique car elle permet de répondre plus finement aux besoins métiers. L'implémentation est désormais basée sur PyTorch via la librairie torchTextClassifiers, maintenue par le SSP Lab. Utiliser ce cadriciel (ou *framework*) open source offre une plus grande souplesse et une mainmise complète sur l'architecture du modèle. Cela devient crucial pour répondre aux spécificités du métier, notamment pour permettre l'**explicabilité** des prédictions ou optimiser les différentes briques du modèle voire en ajouter, ce que le FastText standard ne permettait pas.

1.3.4.1. Implémentation avec la librairie *fastText* de Meta

La première implémentation de la codification probabiliste a utilisé la librairie fastText, un outil open source de FAIR (Facebook's AI Research), choisi pour sa rapidité et son faible coût en ressources sur CPU.

1.3.4.1.1. Une méthodologie mutualisée à l'Insee

L'approche méthodologique adoptée pour Sirene est la même que celle utilisée et mutualisée pour d'autres projets de codification automatique à l'Insee.

Pour les détails méthodologiques, il est recommandé de se référer à l'article JMS T. S. Théo Leroy Lucas Malherbe [2] de la session dernière sur l'application des techniques de machine learning pour coder les professions en PCS 2020. Cette méthodologie est employée aux données Sirene; en particulier, la concaténation des variables catégorielles au libellé pour prendre en compte ces variables.

1.3.4.1.2. architecture et enjeux techniques

Un papier de FAIR présente et décrit l'architecture de fastText P. B. T. M. Armand Joulin Edouard Grave [3]. En somme, fastText utilise une architecture simple mais efficace :

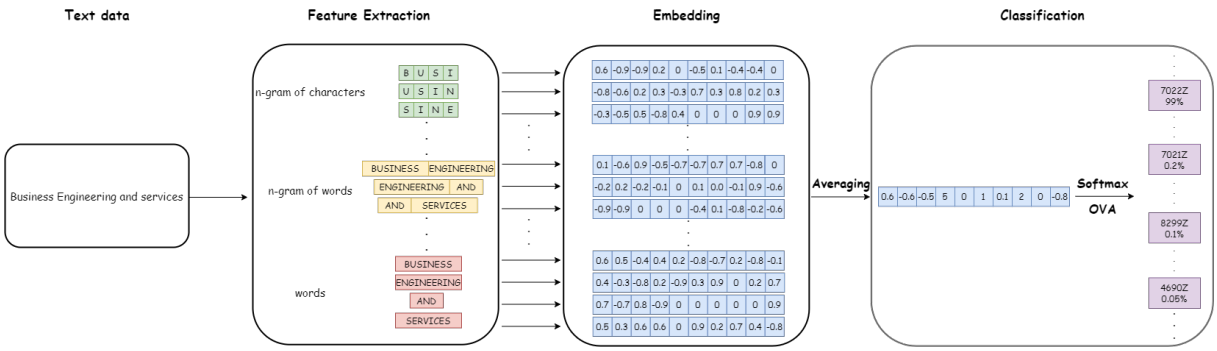


Figure 6. – Architecture de fastText

Architecture simplifiée avec plongement lexical et moyennage des plongements (ou des vectorisations), classification des vecteurs, puis conversion des scores en probabilité.

De manière classique, un modèle attribue un score brut qui est converti en probabilité. Pour interpréter ces scores comme des probabilités, la fonction *softmax* s'applique:

$$\text{Softmax}\,(x_i) = \frac{\exp x_i}{\sum_{j=0}^{9} \exp x_j}.$$

Cette opération convertit les scores en une distribution de probabilité, dont la somme vaut 1, et permet de sélectionner la classe associée à la probabilité la plus élevée comme prédiction finale. La fonction *softmax* est particulièrement utile car elle est infiniment dérivable, ce qui garantit la dérivabilité de tout le modèle f_θ par rapport à ses paramètres θ. Cela rend possible l'utilisation de la descente de gradient pour ajuster les poids du réseau au cours de l'apprentissage.

L'algorithme d'entraînement retenu correspond à un schéma one-versus-all. Dans ce schéma, un classifieur binaire indépendant est entraîné pour chaque classe. Cette variation est particulièrement utile lorsqu'on souhaite pouvoir prédire plusieurs classes pour un même libellé.

La librairie FastText, surcouche optimisée en C++, offre une exécution très rapide et efficace. Elle présente néanmoins plusieurs limites techniques et méthodologiques. Les fichiers binaires résultants (~1,5 Go) nécessitent un stockage adapté et une bonne connexion réseau, car un transfert classique peut corrompre le fichier[7]. Méthodologiquement, l'utilisateur ne peut pas modifier l'architecture du modèle : il faut concaténer toutes les variables dans le libellé, ce qui fonctionne mais limite la flexibilité pour traiter différemment certains types de données. Ces contraintes montrent que, malgré sa rapidité, l'usage de cette librairie impacte la capacité d'innovation et d'adaptation pour des besoins futurs.

1.3.4.2. Implémentation avec Pytorch

Le passage à une implémentation basée sur PyTorch a été une décision stratégique motivée par la nécessité de pérenniser le modèle et de répondre plus finement aux besoins métiers après la mise en lecture seule de la librairie FastText.

1.3.4.2.1. Justification du changement

La nouvelle implémentation a été conçue pour surmonter les limitations de FastText, à la fois technique et méthodologiques, tout en respectant les fortes contraintes de production :

[7] excepté en utilisant le transfert de fichiers par le protocole S3 où la corruption de fichier pour cause de transfert n'est pas possible. En cas d'échec de chargement, il y a absence du fichier complet.

- réponse aux besoins métiers : la transition est cruciale pour permettre des fonctionnalités spécifiques, comme l'explicabilité des prédictions ou l'optimisation des différentes briques du modèle, ce que le FastText standard ne permettait pas.
- performance opérationnelle : Le modèle est conçu pour satisfaire aux contraintes de production strictes, notamment le maintien d'un temps d'inférence acceptable (inférieur à 20 ms) et l'atteinte d'un taux d'automatisation élevé.
- optimisation, coût et architecture : le nouveau modèle est conçu pour être plus léger, facilitant le transfert rapide et répondant aux exigences de coût. Il prépare également le système à une inférence GPU pour une évolutivité verticale future, même si cette ressource n'est pas encore prévue en production.
- maintenabilité et gouvernance: Face à l'archivage de fastText de Meta depuis le 19 mars 2024, le développement d'une librairie maison est essentiel pour assurer la maintenabilité et la gouvernance de l'outil, sécuriser les corrections de bugs et la compatibilité, et positionner l'Insee dans les outils open source de classification supervisée.
- capacité d'innovation : Le choix de PyTorch offre une mainmise complète sur l'architecture comme changer le tokenizer ou ajouter des couches et permet d'intégrer des avancées qui rendront le modèle plus performant que FastText[8].

1.3.4.2.2. Implémentation et architecture

L'implémentation est désormais basée sur PyTorch via la librairie torchTextClassifiers, un développement open source maintenu par le SSP Lab.

Pour stabiliser l'adaptation et l'organisation métier en production, la première itération sous PyTorch est restée volontairement proche de l'architecture FastText. Une différence majeure réside toutefois dans le traitement des variables catégorielles :

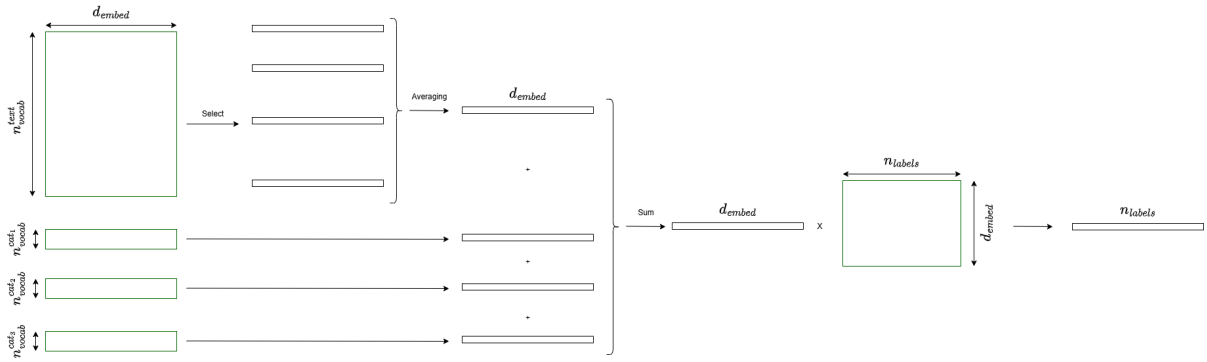


Figure 7. – Exemple d'architecture possible séparant les variables catégorielles

Il s'agit de l'architecture par défaut. Au lieu d'être concaténé avec le libellé en entrée, chaque variable catégorielle possède sa propre représentation vectorielle.

Pour en savoir davantage sur cette librairie, il est conseillé de consulter l'article N. R. J. P. Meilame Tayebjee Cédric Couralet [4].

[8] qui était, lui-même, déjà meilleur que l'ancien système Sicore

1.3.4.3. Quelle librairie en production ?

Actuellement, la librairie *fastText* est toujours utilisée. Toutefois, le passage à la nouvelle librairie est programmé pour début d'année 2026, en même temps qu'une migration de la codification probabiliste sur Kube.

1.4. L'entraînement et la livraison du modèle

Mettre un modèle en production est un processus transversal qui dépasse la simple livraison du fichier de poids. Il nécessite une organisation rigoureuse et une coordination entre le Data Scientist (qui joue le rôle d'interface), l'équipe métier et l'équipe informatique, notamment en raison des différences de compétence (Python pour le modèle, Java pour l'application Sirene).

Ce processus est soumis à de nombreuses contraintes : gestion des vulnérabilités, tests de charge, validation fonctionnelle et métier, et performance technique. Le changement de nomenclature, par exemple, peut doubler le travail nécessaire dans le cycle de vie du modèle.

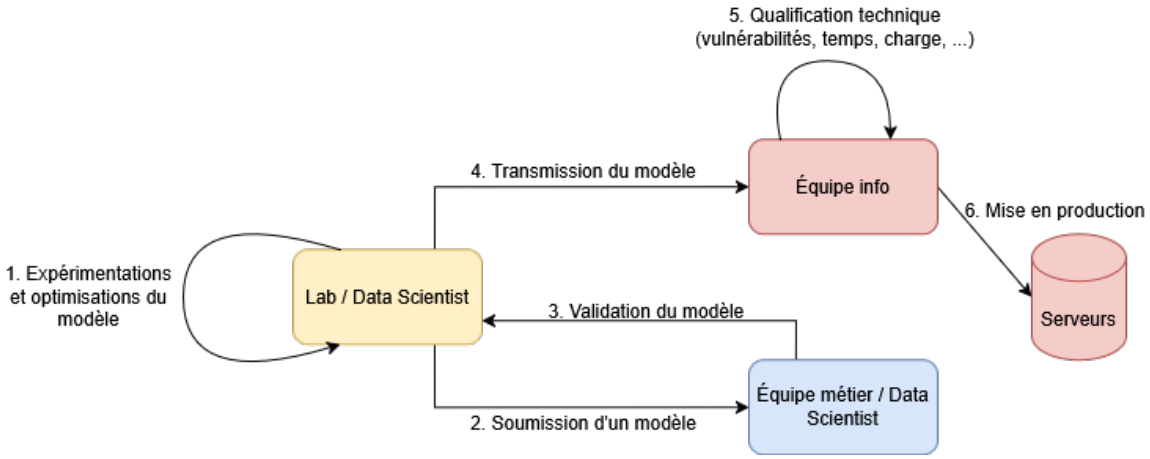


Figure 8. – Process simplifié de qualification du modèle

Le déploiement du modèle en production est du ressort de l'informatique. Toutefois, le Data Scientist peut contribuer à la construction de l'image et la fourniture de cas tests

Deux principaux environnements sont utilisées respectivement pour entraîner le modèle et pour l'inférence en production.

1.4.1. Le datalab comme environnement d'entraînement

Le caractère non-sensible des données de codification Sirene nous permet d'utiliser le SSP Cloud. Un hébergement en interne d'Onyxia est recommandé pour les données sensibles. Le datalab est l'environnement privilégié pour l'entraînement et l'expérimentation pour plusieurs raisons:

- un environnement reproductible : L'environnement d'entraînement repose sur le même type de cluster (Kubernetes) que l'environnement d'inférence en production. Cela rend l'inférence potentiellement reproductible du développement à la production (moyennant quelques contraintes internes).
- des ressources : le datalab donne accès à des ressources computationnelles conséquentes en termes de CPU mais aussi de GPU.
- gestion des modèles (model store) : les modèles sont gérés via un gestionnaire de modèles comme MLflow, le model store. Ce dernier permet de qualifier fonctionnellement les modèles, de gérer la persistance des versions et des artefacts du modèle, et de promouvoir les modèles qualifiés vers les environnements suivants.

- stockage : la plateforme bénéficie d'une grande capacité de stockage standard (type S3). Ce stockage est utilisé pour archiver tout ce qui touche au projet IA ou data science : les modèles, les versions stables, les données d'entraînement, et tous les intermédiaires d'expérimentation.

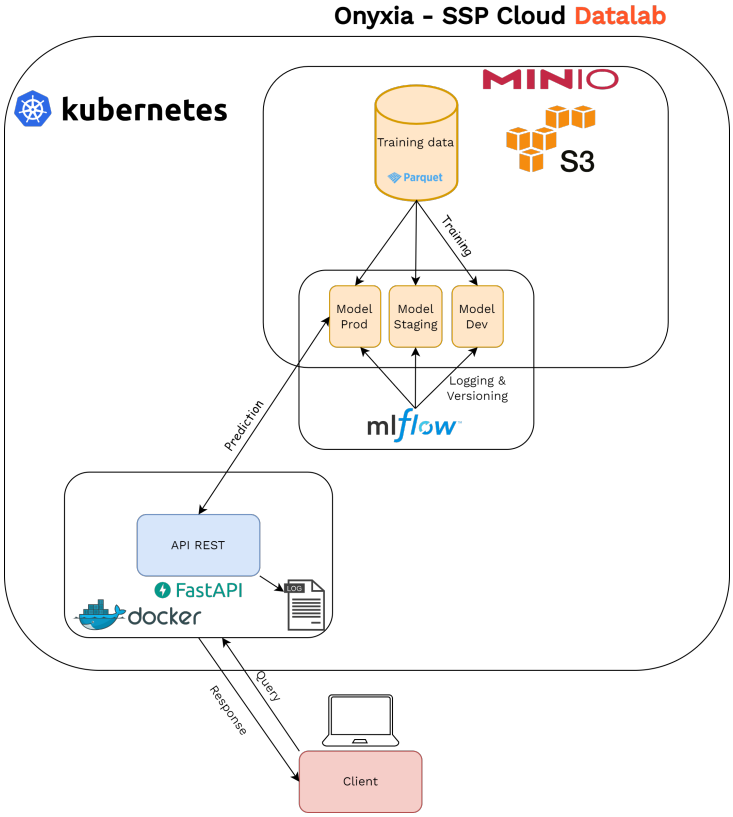


Figure 9. – Environnement et gestion de l'entraînement du modèle

1.4.2. Environnement d'inférence: qualification et mise en production

Une fois qualifié, le modèle est préparé pour l'inférence en production :

1. transfert à l'équipe informatique : Le modèle final est préalablement déposé dans le système de stockage standard (type S3) du cluster de développement interne; s'en suit ensuite, plusieurs étapes de promotion, avant sa mise en production.
2. déploiement en API REST : Le déploiement est réalisé en exposant le modèle sous la forme d'une API REST sur la plateforme Kube. L'application Sirene (en Java) peut alors appeler cette API (en Python) pour obtenir la prédiction.

Ce schéma volontairement simplifié illustre l'étape de livraison du modèle, déployé ensuite sous forme d'API par l'équipe informatique

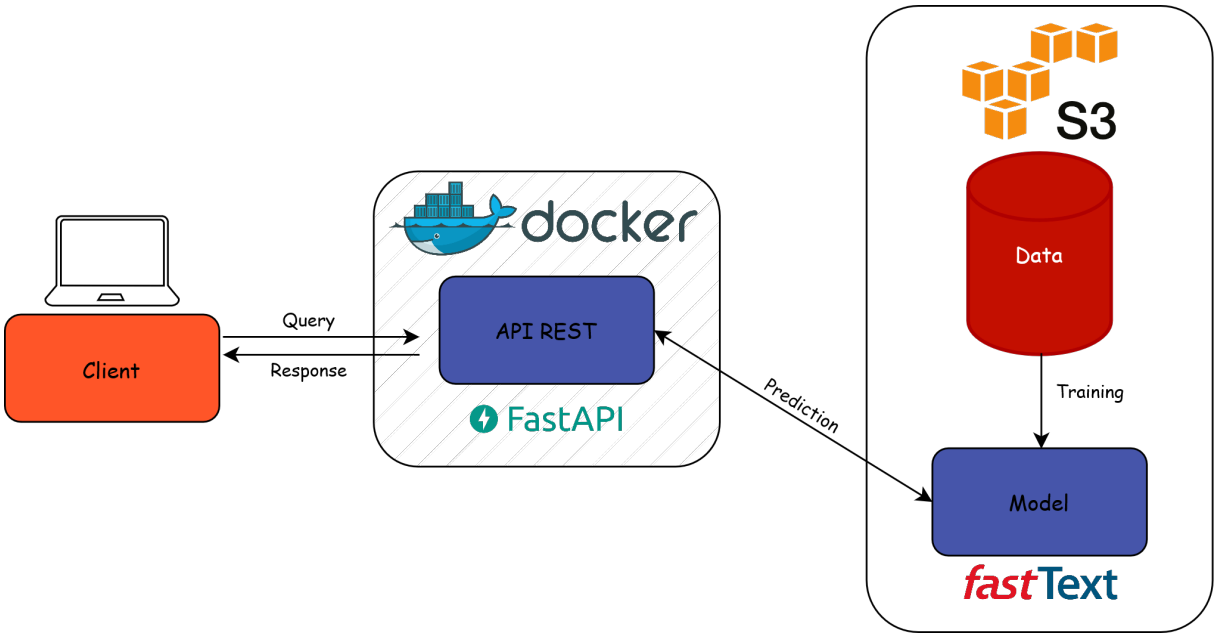


Figure 10. – Inférence du modèle en environnement de développement

1.5. Maintenance et surveillance du modèle

La pérennité et la fiabilité d'un modèle d'apprentissage supervisé en production, surtout en flux continu, exigent que la surveillance devienne une tâche routinière et permanente, et non pas une simple campagne d'annotation ponctuelle.

1.5.1. Une campagne d'annotation pour valider la démarche

En préparation du changement de Nomenclature d'Activités Française (NAF), une campagne d'annotation spécifique a été menée en NAF rév. 2 (la nomenclature actuellement utilisée), avec un double objectif précis.

Pour cela, la codification a été appliquée sur un échantillon représentatif de 10 000 annotations. Les données étaient échantillonnées aléatoirement sur une période glissante afin d'avoir les informations les plus fraîches possibles. Les objectifs étaient:

1. vérifier la qualité de codage global : Appliquer la codification en NAF rév. 2 sur un échantillon représentatif de 10000 annotations, effectuées par le *Pôle Qualité Sirene*, pour vérifier la qualité de codage globale. Un intérêt particulier a été porté aux cas traités en automatique pour en garantir la fiabilité. Les données étaient échantillonnées aléatoirement sur période glissante pour avoir les données les plus fraîches possibles.
2. préparation du changement : utiliser ces données de bonne qualité pour préparer la future campagne d'annotation en NAF 2025 sur des données réelles et variées provenant du Guichet Unique.

Cette expérimentation a été fructueuse, permettant la validation globale de la codification en NAF rév. 2 et le développement d'outils qui sont la base du futur processus de surveillance routinière.

1.5.2. Outils de surveillance et pilotage métier

Le suivi des performances est crucial pour détecter la dérive ou aider à ajuster les paramètres comme le seuil de l'indicateur de confiance IC en continu.

Ci-dessous, quelques extraits d'un tableau de bord pour éclairer la décision.

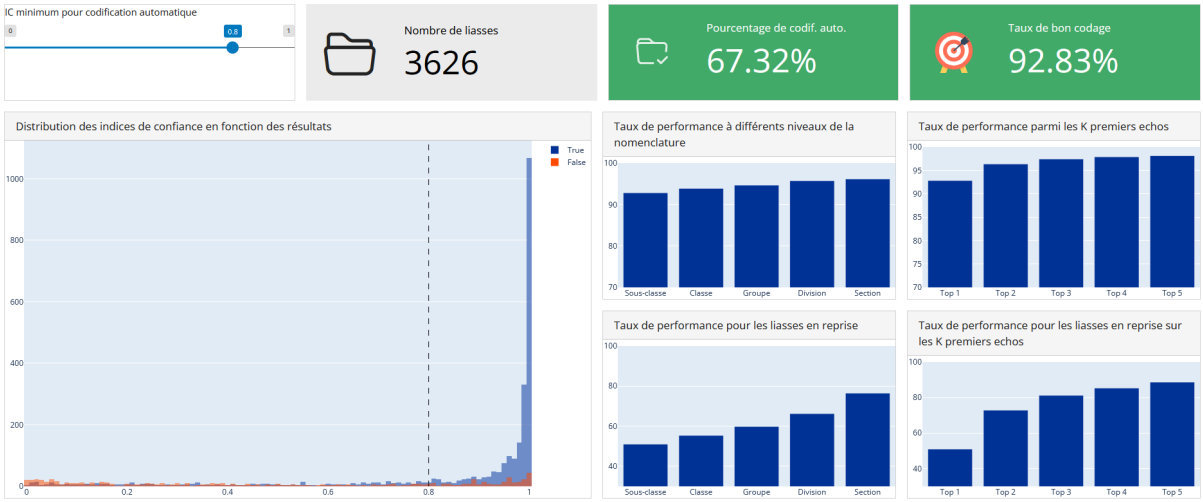


Figure 11. – Monitoring du modèle sur inférence d'un échantillon de surveillance représentatif

L'hypothèse sous-jacente est que les annotations humaines représentent la vérité terrain[9]

Figure 12. – Comparaison des taux d'automatisation des activités par période

Une visualisation utile pour surveiller l'évolution de la confiance du modèle par activité à différents niveaux de la nomenclature

[9]La vérité terrain ou *ground truth* désigne les faits ou les données réelles et vérifiées qui servent de référence absolue pour entraîner et évaluer la performance d'un modèle d'apprentissage supervisé.

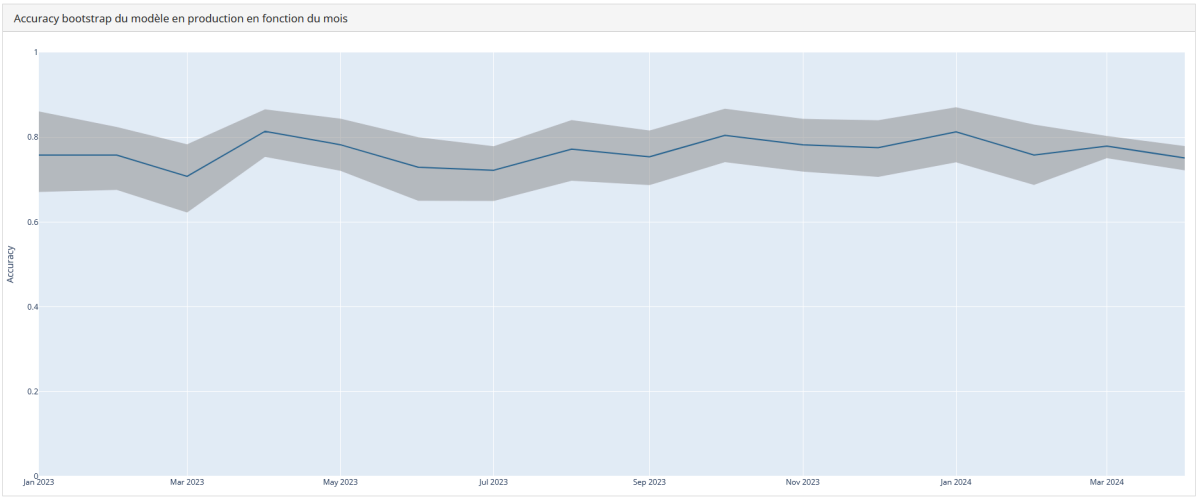


Figure 13. – Surveillance des performances au cours de l'année 2024

Le couloir représente l'intervalle de confiance[10]: plus il y a d'annotations, plus il est resserré

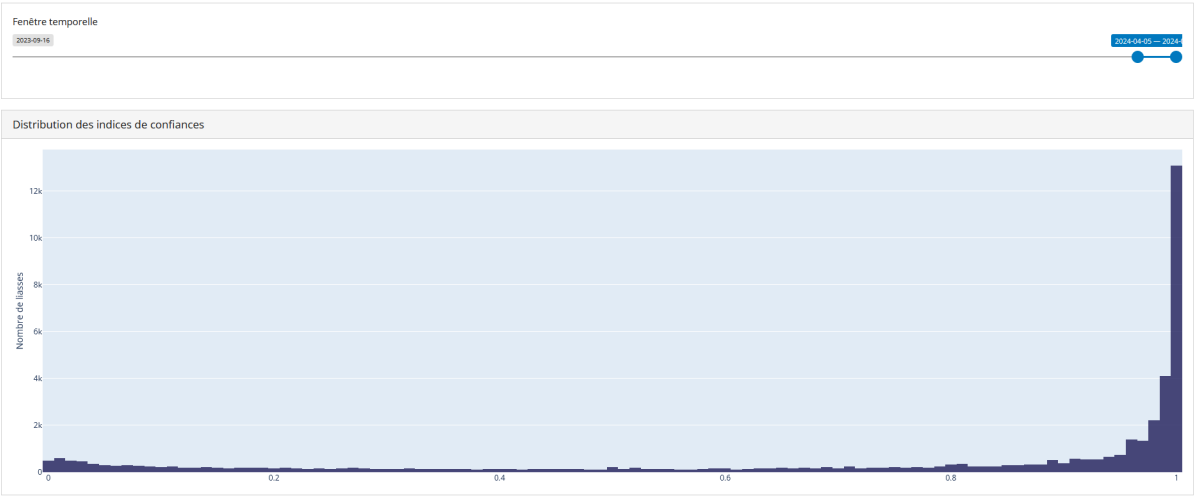


Figure 14. – Surveillance de la distribution des IC sur une période donnée

Un échantillon de test permettrait aussi d'afficher la séparation entre bon et mauvais codage comme dans la Figure 3. Ainsi, il serait possible de surveiller la pertinence du seuil appliqué.

L'enjeu fondamental est d'intégrer la surveillance dans le flux de travail habituel (tâche routinière). Celle-ci repose sur l'annotation journalière d'un échantillon représentatif du flux de production. L'équipe devra reprendre cette surveillance avec la NAF 2025 dès son déploiement pour piloter le nouveau modèle, garantissant une qualité constante des données du répertoire.

Le changement de nomenclature (NAF 2025) n'implique pas qu'une simple mise à jour du modèle de *machine learning*. Il a des conséquences transversales qui impactent profondément plusieurs maillons de la chaîne de production de l'application de gestion Sirene 4.

[10]En particulier, il s'agit d'un intervalle de confiance bootstrap

2. Comment convertir notre base d'apprentissage en NAF 2025 ? campagne d'annotation et vérité terrain

Cette partie abordera les conséquences de la révision de la nomenclature NAF, tant au niveau de sa structure que sur le volume de travail qu'elle représente pour les données d'entraînement. En particulier, les enjeux soulevés par l'exercice d'annotation en NAF 2025 y seront détaillés. Cette expérience permettra d'alimenter la matière première nécessaire à la méthodologie innovante qui sera présentée dans la partie suivante. Enfin, cette section dédiée à la tâche humaine permettra de mettre ultérieurement en perspective l'exercice manuel par rapport à l'utilisation des Large Language Models (LLM) dans la partie suivante.

2.1. Les conséquences du changement de nomenclature

2.1.1. la table de correspondance théorique

Comme documenté par le rapport du groupe de travail du CNIS, la révision de la NAF pour 2025 implique d'importantes modifications structurelles, notamment des regroupements de codes mais aussi de nombreuses scissions. Pour constater le détail de ces évolutions, il convient de se référer à l'Annexe 10 - Bilan détaillé de l'instruction, classe par classe du rapport S. D. Clotilde MASSON [5].

Table 1. – Quelques éléments de vocabulaire sur la structure de la NAF

Niveau de la Nomenclature	Terme (Intitulé)	Nombre de Positions
Niveau 1	Section	1 (Lettre)
Niveau 2	Division	2 (Chiffres)
Niveau 3	Groupe	3 (Chiffres)
Niveau 4	Classe	4 (Chiffres)
Niveau 5	Sous-classe	5 (Chiffres + Lettre)

Tout au long de cet article, nous emploierons les termes univoques et multivoques pour caractériser ces changements, selon les définitions détaillées ci-dessous

2.1.1.1. Univoques

Les correspondances **univoques** (ou bijections) désignent les cas où un code de l'ancienne nomenclature (NAF rev 2 ou NAF 2008) correspond à un seul code dans la nouvelle nomenclature (NAF 2025). Ces cas ne présentent pas de difficulté majeure pour la conversion automatique des données d'apprentissage.

Figure 15. – Exemple de correspondance univoque 1:1

Cette figure illustre une correspondance de type 1:1 entre un code de l'ancienne NAF et un code unique de la NAF 2025.

Table 2. – Effectif des codes univoques ou bijections. Cette table indique qu'une large majorité des codes, 75%, sont univoques.

Niveau classe (niveau 4)	Niveau sous-classe (niveau 5)
573	551

2.1.1.2. Multivoques

Les cas multivoques représentent les situations où l'expertise humaine est indispensable, car ils introduisent une ambiguïté dans la conversion. Deux cas de figures se présentent mais l'attention se portera sur un cas particulier de correspondance.

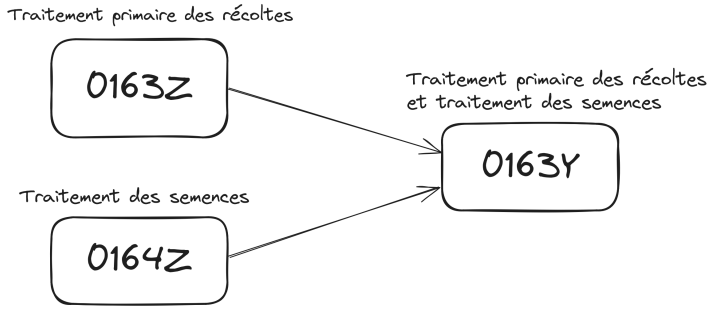


Figure 16. – Exemple de correspondance N:1 ou surjection

Cette figure illustre une surjection (N:1) où plusieurs anciens codes sont regroupés dans un seul nouveau code

Les surjections (correspondances N:1) ne posent pas de problème fonctionnel majeur dans le système de codification, car pour chaque ancien code considéré individuellement, il n'y a qu'un choix possible dans la nouvelle nomenclature. Elles peuvent donc être traitées de manière univoque dans le sens de la conversion.

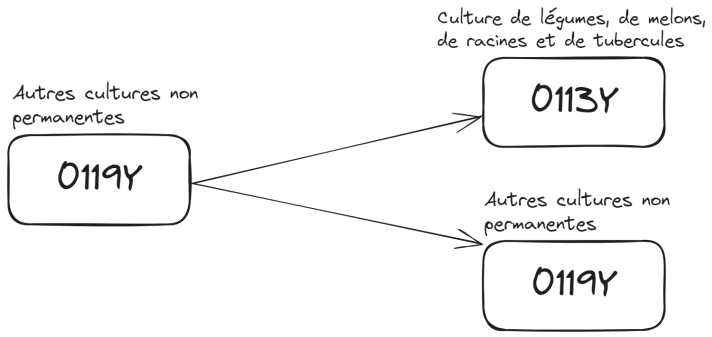


Figure 17. – Exemple de correspondance multivoque 1:N

Cette figure illustre un éclatement (1:N) où un ancien code correspond à plusieurs codes possibles de la NAF 2025.

Les **multivoques désigneront**, tout au long de cet article, principalement les correspondances 1:N (éclatements), où N représente le nombre de possibilités dans la nouvelle nomenclature. Ces cas, qui concernent 25 % des codes, sont ceux qui requièrent une annotation manuelle pour déterminer la sous-

Table 3. – Distribution des codes multivoques pour la NAF

(a) Niveau classe: 4 positions

1-to-N	Occurrences
2	101
3	28
4	15
5	3
6	5
7	1
8	1
17	1
25	1
29	1
30	2

(b) Niveau sous-classe: 5 positions

1-to-N	Occurrences
2	109
3	30
4	24
5	6
6	4
8	1
9	2
21	1
27	1
36	1
38	2

classe correcte en NAF 2025 associé au libellé dans le jeu d'apprentissage, dont la conversion sera approfondie dans la section qui suit.

Table 4. – Représentation des multivoques dans la table de correspondance

Part de multivoques	Part d'univoques
25 %	75 %

2.1.2. Le jeu d'apprentissage

Le jeu d'apprentissage est fixé sur les données de l'année 2024 pour servir de base de travail à l'ensemble de l'article.

Afin de rationner les ressources d'annotation, une première étape consiste à séparer les codes univoques (1:1), qui peuvent être facilement convertis en NAF 2025 via la table de correspondance théorique. La problématique se restreint alors à ne traiter que les codes multivoques. La question est de savoir quel volume de données ces derniers représentent.

2.1.2.1. Un volume de multivoques toutefois plus conséquent

Bien que les codes multivoques ne représentent que 25% des codes dans la nomenclature elle-même (comme établi dans la section précédente), leur fréquence d'apparition dans les données d'apprentissage est importante. Ils constituent en effet 52% du volume total des données d'entraînement à convertir en NAF 2025. Cette proportion révèle l'ampleur du travail d'annotation nécessaire pour obtenir une vérité terrain fiable.

Table 5. – Représentation des multivoques dans le jeu d'apprentissage

Part de multivoques	Part d'univoques
52 %	48 %

Malgré un impact sur seulement 25% des codes de la NAF, ces cas multivoques représentent l'essentiel du volume des données d'entraînement à convertir en NAF 2025.

2.1.2.2. Une approche métier insuffisante pour réduire le nombre

Une stratégie métier a été mise en place pour réduire ce volume. Elle consiste en la requalification de certains codes multivoques en correspondances biunivoques sur la base du principe de réalité métier et des informations contextuelles disponibles dans la déclaration.

Cette table de passage métier permet d'identifier et de résoudre les « faux multivoques »[11]. Prenons l'exemple du code 6820A (« Location de logements ») : bien qu'il puisse théoriquement correspondre à plusieurs nouveaux codes NAF, l'analyse manuelle montre qu'il est recodé en 6820G dans la grande majorité des cas multivoques (16% des cas ambigus). Le postulat métier est alors retenu que cette situation est univoque, car ces locations sont majoritairement de longue durée. De même, les activités de production d'électricité des personnes physiques (3511Z) sont approximées comme de l'électricité de type renouvelable (3512Y), en supposant qu'il s'agit principalement de la production par panneaux solaires.

Table 6. – Requalification de certaines correspondances en biunivoque dans le jeu :

Type d'activité principale concerné	Code initial	Code final	Proportion	Parti pris métier
Location de logements	6820A	6820G	16%	Très souvent de longue durée
Location de biens immobiliers	6820B	6820H	10 %	
Production d'électricité	3511Z	3512Y	3 %	Type renouvelable, si personne physique
Restauration traditionnelle	5610A	5611G	2 %	Service principalement à table
Réparation de voitures	4520A	9531G	1,5 %	Quasi-absence d'intermédiation
Commerce d'alimentation générale	4711B	4711G	0,5 %	Rarement intermédiation ou vente en ligne
Construction de maisons individuelles	4120A	4100G	0,5 %	Exclure la restauration de monuments

Grâce à cette approche métier, 32,5 %, soit près d'un tiers, des codes multivoques initialement identifiés ont pu être corrigés ou reclassifiés dans la base d'apprentissage. Cependant, cette méthode n'est applicable qu'à moins de 50 % du volume total des cas multivoques. Étant donné que la base d'apprentissage complète (univoques et multivoques) atteint deux millions d'observations, et qu'un nombre important de cas ambigus subsiste même après cette première reclassification, il resterait environ 500 000 multivoques à convertir. Ce volume résiduel est trop important pour être traité manuellement, car il nécessiterait des moyens humains importants que l'équipe n'a pas à disposition. Cela démontre l'inefficacité de cette méthode pour une conversion à grande échelle et souligne la nécessité de mettre en place un outil de codage automatique pour traiter ces cas de manière exhaustive.

2.2. La campagne d'annotation: à l'origine de nos matériaux

En l'absence totale de données étiquetées en NAF 2025, une campagne d'annotation a été menée pour constituer la première vérité terrain exploitable. Ce travail était indispensable pour plusieurs

[11] Certains codes dont le 6820B et le 4520A ont, au final, été corrigé lors de la mise à jour de la table de correspondance officielle, confirmant l'hypothèse de départ de reclassification.

raisons : il permettait non seulement de faire vivre la nouvelle nomenclature en conditions réelles et de préparer les futurs gestionnaires, mais aussi d'obtenir un retour concret sur sa structure. L'objectif était également d'affiner la table de correspondance théorique, de soulever les cas de classement récurrents qui demandent une décision métier pour arbitrage, et d'assurer une mobilisation progressive des équipes.

2.2.1. Déroulé

La campagne d'annotation a été gérée dynamiquement grâce à une plateforme de labellisation[12], mise à disposition par l'équipe métier. L'approche méthodologique était simple : les experts réalisaient une annotation simple pour chaque libellé. La plateforme garantissait nativement l'accès concurrent aux données, s'assurant qu'un libellé n'était vu et annoté qu'une seule fois par un expert donné. Les participants pouvaient aussi, pour chaque tâche, attribuer un commentaire et une note de 0 à 5 étoiles sur la difficulté ressentie[13]. Les consignes mettaient l'accent sur la qualité du codage plutôt que sur la quantité, avec la possibilité de déléguer un libellé inclassable dans la nomenclature au sein d'un « pot commun » pour qu'il soit reproposé à un autre expert annotateur. Lorsqu'un libellé ne décrivait pas clairement une activité ou était inclassable (par manque de précision, ambiguïté, etc.), l'annotateur pouvait appliquer un code **inclassable**. Le travail s'est appuyé sur les tables de correspondance et les notes explicatives disponibles. Initialement, une charge de travail d'environ une heure par jour était recommandée pour l'ensemble des participants.

Au niveau du calendrier, la campagne s'est déroulée sur environ 5 mois, après une phase de formation à la NAF 2025 et un bêta-test. Elle a mobilisé près de 25 experts du réseau APE. La campagne a été rythmé par des réunions mensuelles avec la *Division Nomenclatures économiques* (DNE) et la *Division Répertoire interadministratif Sirene* (RIAS), toutes deux rattachées au *Département Répertoires, infrastructures et statistiques structurelles* (DRISS), assurant un retour d'expérience régulier avec les experts de la nomenclature et des administrateurs de Sirene. Ces échanges étaient précieux pour la montée en compétence des annotateurs, la résolution des cas de classement les plus complexes et la gestion collective des cas à trancher. Ces opportunités ont permis d'anticiper certaines consignes de la NAF 2025 et de profiter de l'expertise de la DNE, favorisant un travail stimulant.

L'organisation quotidienne a varié selon les sites, par demi-journée ou selon convenance des agents. Malgré l'intérêt du travail et les échanges de pratiques, la charge prévue d'une heure par jour est devenue difficile à maintenir vers la fin de la campagne en raison de missions annexes.

2.2.2. Stratégie d'échantillonnage pour l'annotation

Pour garantir la plus grande diversité possible, les libellés soumis à l'annotation ont été échantillonnés de manière distincte au sens strict, ce qui permet d'accepter les variations incluant du bruit ou des coquilles. L'échantillonnage était également stratifié afin d'assurer le codage des libellés les plus fréquents. Cette approche permet de prioriser les cas courants, tout en évitant de traiter les activités très rares qui seraient, de toute façon, destinées à de l'expertise gestionnaire.

Au final, cette approche a permis de constituer un jeu de données très particulier, ne concernant que les multivoques les plus représentatifs de la distribution, par la stratification, tout en assurant la plus grande variété possible sur un échantillon limité.

[12] il s'agit de la version open source de Label Studio

[13] la note la plus élevée indiquant la plus grande difficulté

2.2.3. Bilan et apports de la campagne

Au niveau national, la campagne a généré un volume important de données et de retours qualitatifs. Le bilan est le suivant:

- Total annoté : 30 915 libellés.
- Tâches inclassables : 1 830 libellés, appliqués lorsque l'activité n'était pas claire ou manquait de précisions.
- Tâches déléguées : 3 553 libellés, mis de côté pour être reproposés à un autre expert.

2.2.3.1. Documents de référence et vérité terrain

Le principal apport de cette campagne a été la stabilisation précoce des ressources de référence nécessaires à la nouvelle nomenclature. Grâce aux remontées des experts, la structure de la NAF 2025, la table de correspondance théorique et les notes explicatives ont pu être affinées. Surtout, cet exercice a permis de constituer la première vérité terrain exploitable, un matériau précieux préparant l'ensemble de la méthodologie suivante.

2.2.3.2. Mode opératoire des experts

La campagne a permis de mettre au point une feuille de route face aux cas ambigus. Pour guider l'annotation, un arbre de décision a été formalisé, décrivant la séquence logique suivie par les experts pour attribuer un code NAF 2025 à un libellé donné. Ce mode opératoire est illustré ci-dessous.

Cet arbre de décision inspirera fortement la méthodologie décrite dans la partie suivante

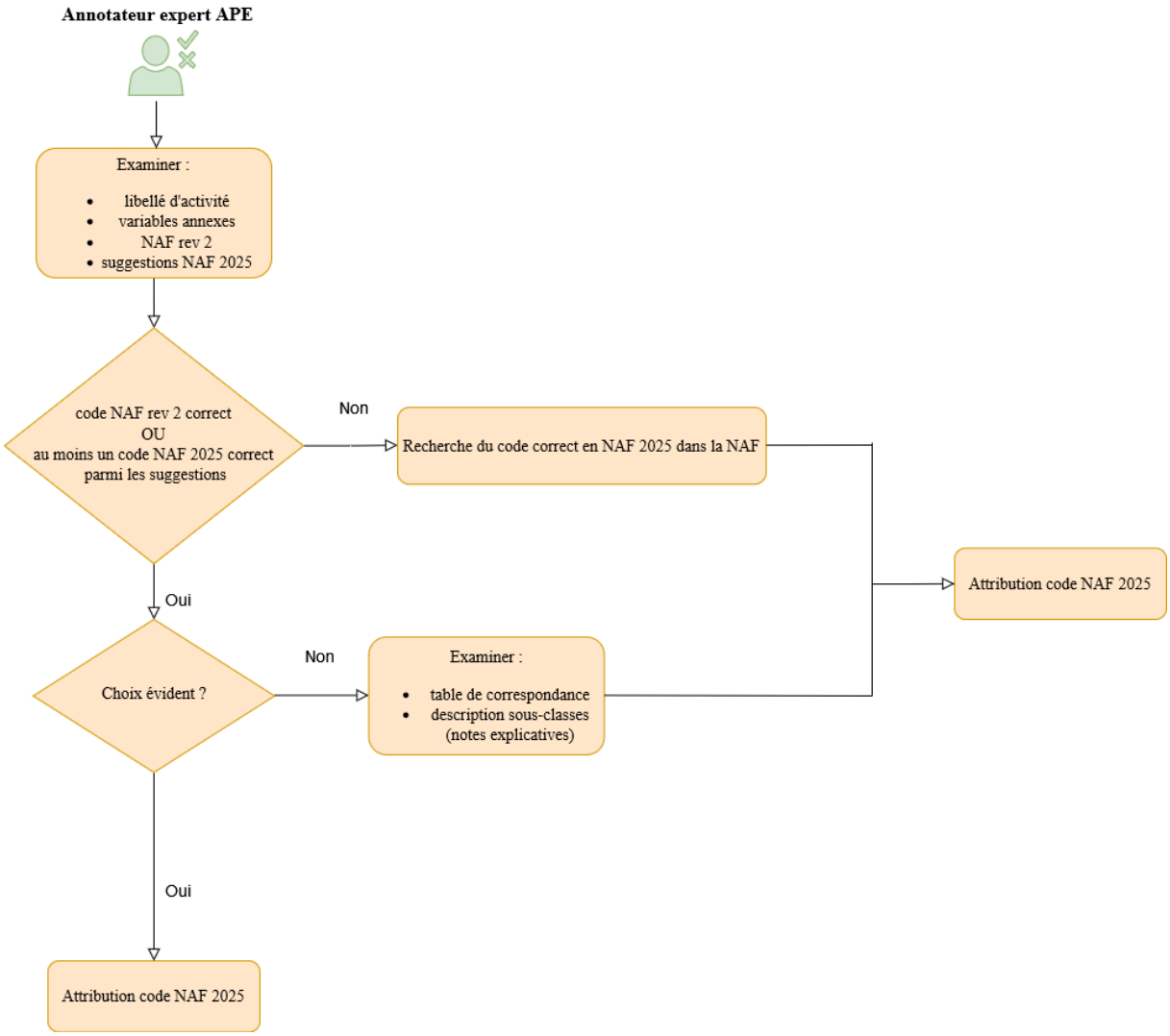


Figure 18. – Arbre de décision pour coder en NAF 2025

2.2.3.3. Evaluation de la difficulté de la campagne

Afin de collecter des retours sur l'effort cognitif, la plateforme de labellisation a permis aux participants d'attribuer une note de difficulté à chaque tâche selon une échelle simple de cinq modalités. Bien que cet indicateur ne présente pas une rigueur absolue, il constitue la seule mesure de difficulté disponible pour l'évaluation de la campagne.

Les modalités de la note de difficulté sont les suivantes:

- 0 étoile: aucune difficulté ou rien à signaler
- 1 étoile: plutôt facile
- 2 étoiles: facile
- 3 étoiles: moyenne
- 4 étoiles: difficile
- 5 étoiles: très difficile

Bilan global

- 0 étoile: 28,85 %
- 1 étoile: 56,47 %
- 2 étoiles: 6,17 %
- 3 étoiles: 6,48%
- 4 étoiles: 1,19 %
- 5 étoiles: 0,1 %

L'exercice d'annotation est jugé de difficulté globalement modéré par les experts, avec seulement quelques cas jugés très complexes. Une analyse est nécessaire pour déterminer si ces cas difficiles correspondent nécessairement à des libellés issus des multivoques présentant le plus grand nombre de possibilités.

2.2.3.4. Difficulté de l'exercice et nombre de multivoques: un lien pas si immédiat

Les figures ci-dessous illustrent la difficulté ressentie par les experts en fonction du nombre de correspondances multivoques possibles (N) en NAF 2025 pour un code NAF rev 2 donné.

L'analyse de ces résultats révèle que la difficulté ressentie n'est pas une fonction croissante simple du nombre de multivoques. La difficulté perçue par les répondants reste globalement stable pour les cas présentant entre 10 et 20 options. Une montée progressive de la charge cognitive n'est observée qu'à partir d'une vingtaine de correspondances possibles, et s'accentue nettement autour d'un seuil de 30 options au niveau classe et 35 options au niveau sous-classe.

Les activités jugées particulièrement difficiles ne sont pas uniquement associées au plus grand nombre de multivoques. Par exemple, le code 4399D (Autres travaux spécialisés de construction), avec seulement N=8 correspondances possibles, est également concerné par les pics de difficulté, tout autant que le 4799A (Vente à domicile, N=38) et le 4791B (Vente à distance sur catalogue spécialisée, N=36).

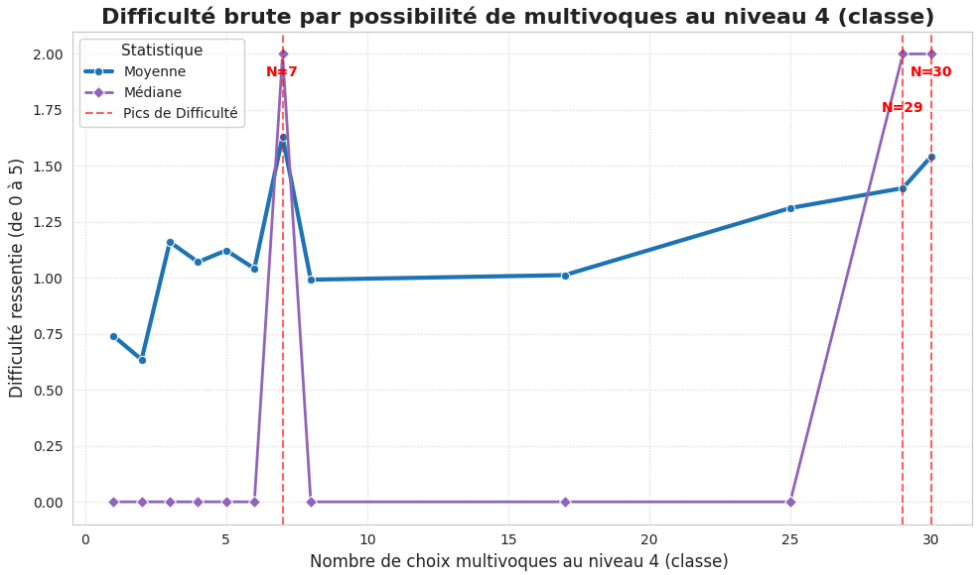
En particulier, **la difficulté de classement de la vente à domicile et à distance a une explication connue des experts métier**. Elle provient du fait que les libellés soumis ne précisent souvent pas le type de produits vendus, une information non requise par la NAF rev 2 mais essentielle pour le classement en NAF 2025. Les libellés sont donc parfaitement classables sous l'ancienne nomenclature, mais de fait inclassables en NAF 2025. Par conséquent, un nombre plus restreint de multivoques pour cette sous-classe n'aurait pas rendu la tâche plus facile.

Cette observation met en évidence que d'autres facteurs métier influencent la difficulté, au-delà du simple nombre de choix. L'ensemble de ces cas complexes ne représente au final que 2 077 tâches, soit moins de 7% de l'annotation totale.

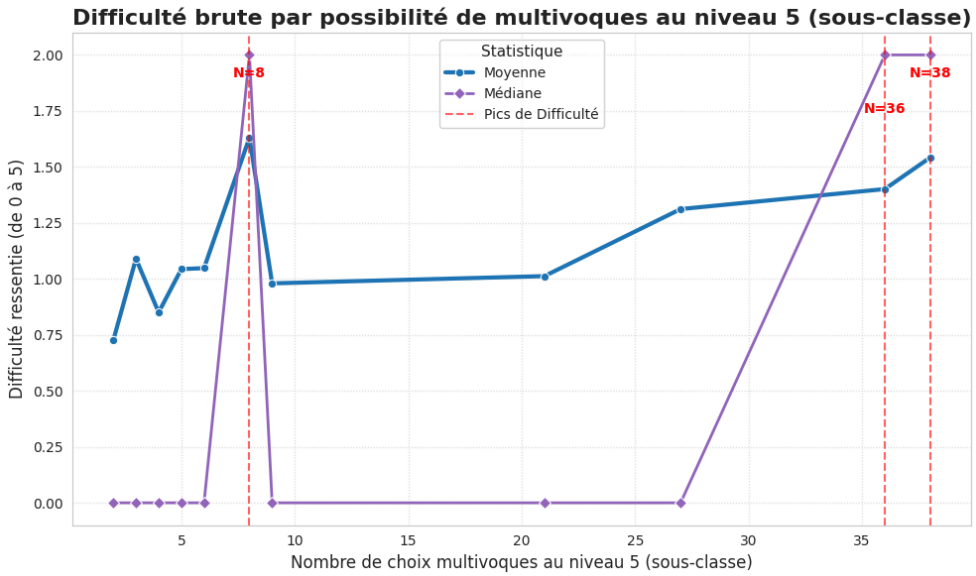
L'analyse de la proportion de cas jugés difficiles confirme cette observation : un nombre élevé de correspondances multivoques n'entraîne pas nécessairement une augmentation proportionnelle de la part de tâches difficiles. Le même phénomène est observé lorsque l'on analyse les cas déclarés inclassables. Ces résultats sont particulièrement intéressants à mettre en perspective avec la *Table 3 – Distribution des codes multivoques pour la NAF*, qui présente la rareté des cas avec un grand nombre d'options.

2.2.3.5. Concordance des annotateurs

Pour évaluer la difficulté de l'annotation et apprécier la solidité de la vérité terrain issue de la campagne, un exercice de contrôle a été réalisé sur un échantillon de **256 libellés**. Pour des raisons de moyens, cet exercice a été mené par **trois volontaires**, notés **A, B et C**, sur leur temps disponible. Ce contrôle qualité a reposé sur une triple annotation, réalisée en aveugle par trois volontaires distincts (désignés ici A, B et C) sur ce même échantillon. Cet exercice, contrairement à la campagne principale,



(a) Moyenne des avis bruts au niveau classe



(b) Moyenne des avis bruts au niveau sous-classe

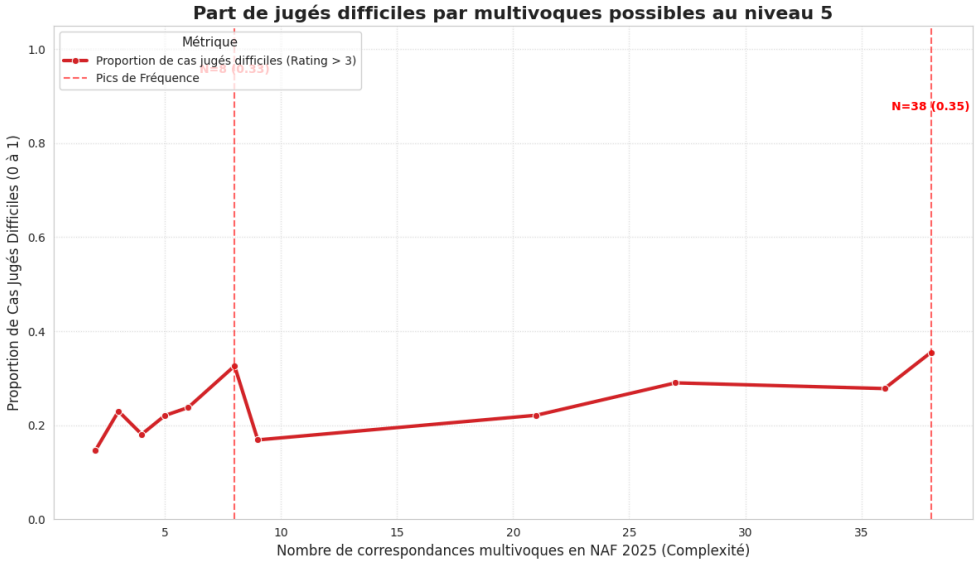
Figure 19. – Difficulté ressentie en fonction des multivoques de l'exercice d'annotation

n'autorisait pas les experts à passer la tâche en cas de doute, limitant le biais de sélection. Il ne s'agissait pas d'un audit exhaustif, mais d'un échantillon exploratoire destiné à éclairer la qualité de l'annotation humaine. L'objectif n'était pas d'obtenir un résultat parfait, mais de disposer d'indicateurs objectifs permettant de situer la variabilité naturelle du jugement humain sur ce type de tâche.

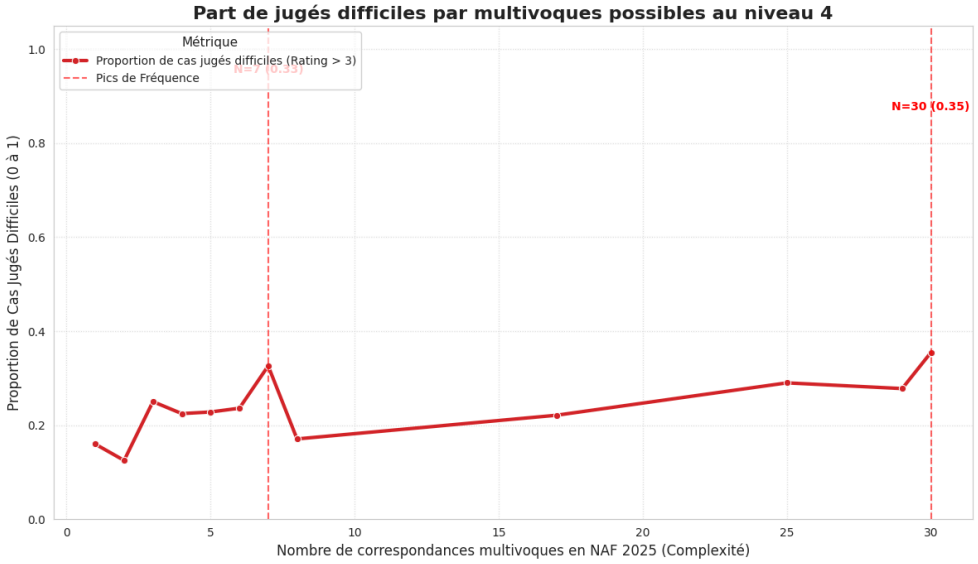
2.2.3.5.1. Deux niveaux de concordance

- **Concordance *absolue*** : les trois annotateurs choisissent exactement le même code.
- **Concordance *partielle*** : au moins deux annotateurs sur trois convergent vers le même code.

Sur cet échantillon, la **concordance absolue** entre les trois annotateurs s'élève à environ 62 %. Ce résultat, relativement modeste, est cohérent avec la nature interprétative du classement NAF, où



(a) Part de jugés difficiles au niveau classe



(b) Part de jugés difficiles au niveau sous-classe

Figure 20. – Proportion de cas jugés difficiles (avis supérieur à 3/5) en fonction des multivoques de l'exercice d'annotation

des nuances existent entre certains codes proches. Il est attendu que la concordance absolue soit sensiblement plus faible que la concordance partielle sur ce type de tâche.

Afin d'illustrer la variabilité entre annotateurs, le graphique ci-dessous montre la concordance par paire :

Malgré la faiblesse de la concordance pure, soit près de 62 %, l'analyse révèle qu'un accord majoritaire est atteint dans 96 % des cas, ce qui signifie qu'au moins deux annotateurs sur trois ont convergé vers le même code. Cela démontre que l'annotation est robuste dans sa grande majorité, même si les trois experts ne sont pas systématiquement parvenus à une concordance absolue.

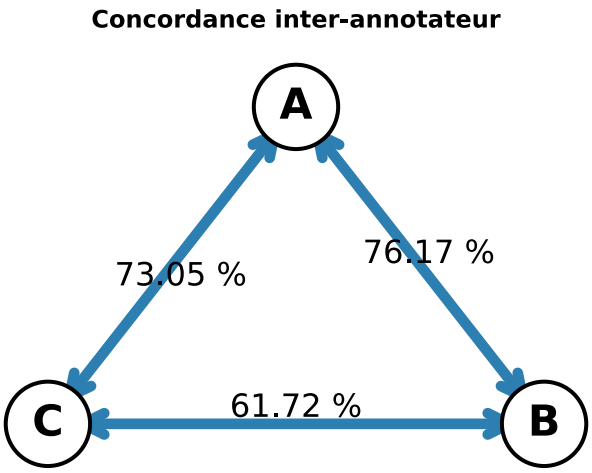


Figure 21. – Graphe de concordance entre les annotateurs A, B et C

Il semble que le taux de concordance pure ait été tiré vers le bas par la divergence prononcée entre deux annotateurs spécifiques, B et C, dont l'accord par paire n'atteint que 62 %. Cette faiblesse de la concordance partielle entre B et C limite l'accord simultané de l'ensemble du triplet.

L'analyse du schéma de concordance met en évidence une hétérogénéité dans les niveaux d'accord entre annotateurs. Deux annotateurs présentent un niveau de concordance sensiblement plus faible, ce qui laisse penser à une divergence d'interprétation ou de sensibilité aux cas plus ambigus. Ce point n'invalide pas l'exercice, mais rappelle que même entre experts motivés et formés, une part de subjectivité subsiste dans le codage NAF, en particulier au niveau 5.[14]

Cette observation est importante, car elle montre qu'un désalignement marqué entre deux personnes peut réduire la concordance absolue à trois, tout en laissant émerger un consensus majoritaire fiable. Autrement dit, l'absence d'unanimité ne signifie pas absence de cohérence : dans la majorité des cas, deux annotateurs convergent vers la même réponse, ce qui permet à un consensus d'émerger malgré des profils d'annotation différents.

La même analyse menée au **niveau 4** et au **niveau 5** donne les résultats suivants :

Pair	Niveau 4	Niveau 5
A – B	76.56%	73.05%
A – C	80.86%	76.17%
B – C	79.69%	61.72%

Malgré une concordance absolue limitée, la **concordance partielle est élevée**, ce qui signifie que le consensus majoritaire est très souvent atteint :

Concordance partielle (≥ 2 sur 3)

- Niveau 4 : **97.27 %**
- Niveau 5 : **96.09 %**

Ce résultat est important : il montre que l'annotation humaine reste **largement convergente**, même si les trois annotateurs ne choisissent pas toujours strictement le même code.

[14] Outre des erreurs manifestes, les divergences de codage peuvent être attribuées à plusieurs facteurs, notamment des erreurs de saisie, des différences d'interprétation, l'ambiguïté des cas de classement ou une asymétrie d'information causé par une différence d'expérience.

2.2.3.6. Mise en perspective avec la vérité terrain de la campagne

Sur les **256 cas annotés**, **97** correspondaient à des libellés issus de la campagne principale. Cela a permis d'estimer dans quelle mesure un vote majoritaire entre trois annotateurs serait aligné avec le **code officiel retenu lors de la campagne** :

Alignement avec le ground truth	Taux
Au niveau sous-classe (niv 5)	**79.38 %**
au niveau classe (niv 4)	**85.57 %**

La proximité de ces taux avec les niveaux de concordance observés montre que :

- la vérité terrain issue de l'annotation simple est **cohérente avec un vote majoritaire**,
- les divergences sont souvent **à un niveau de granularité fine** (concordance niveau 4 > niveau 5),
- un désaccord ne signifie pas nécessairement une erreur, mais peut refléter la subtilité du choix final.

Le rôle du ground truth de l'annotation

La vérité terrain (ou ground truth) représente l'ensemble des données de référence en NAF 2025. Dans le cadre de cette campagne, elle repose sur le jugement moyen des experts et doit être considérée comme suffisamment acceptée par le métier pour être qualifiée de donnée de référence. Elle est fondée sur la confiance accordée au travail collectif et au consensus majoritaire des annotateurs, en tant que première base de travail.

Quelle lecture pour la suite ?

Ces résultats suggèrent que la vérité terrain construite lors de la campagne est **suffisamment solide** pour servir de référence dans l'évaluation des modèles de la prochaine partie, tout en gardant à l'esprit qu'elle n'est pas exempte d'imperfections comme toute annotation humaine. Chercher une concordance de 100 % entre un modèle et cette vérité terrain n'est ni réaliste ni souhaitable. L'enjeu est plutôt d'identifier les modèles capables de s'aligner suffisamment bien avec cette référence pour garantir un niveau satisfaisant de qualité de codage.

La section suivante s'inscrit dans cette logique : évaluer le degré d'alignement des LLM comme annotateurs avec cette vérité terrain afin de sélectionner ceux qui offrent le meilleur compromis entre performance, robustesse et cohérence métier.

3. Comment générer une base d'apprentissage en nouvelle nomenclature sur la base de la vérité terrain ?

Cette partie présente une méthodologie innovante visant à reconstruire l'exhaustivité de la base d'apprentissage en tirant parti des LLM, et plus particulièrement des modèles compétitifs publiés en open source. L'approche s'appuie sur le mode opératoire des annotateurs, les notes explicatives de la DNE et l'inspiration de projets réussis dans d'autres INS, comme P. L. Iva Spakulova [6]. Les technologies cloud et la puissance de calcul du datalab SSP Cloud permettent de tester cette approche prometteuse et de l'adapter aux besoins métier spécifiques de la base Sirene, en intégrant le passage à la NAF 2025.

[15] Pour en savoir plus sur l'apport des technologies cloud à la statistique publique, il est recommandé de consulter l'article de l'édition JMS de cette année T. F. Frédéric Comte Romain Avouac [7].

L'utilisation de données non-sensibles évite les contraintes d'anonymisation, et l'auto-hébergement avec les technologies cloud[15] assure indépendance et reproductibilité sur des infrastructures similaires pour tous les utilisateurs d'Onyxia[16].

3.1. Méthodologie

L'objectif n'est pas de développer un classifieur basé sur un LLM, mais de produire une base annotée en NAF 2025 suffisamment large et qualitative pour entraîner ultérieurement un modèle de classification. Conçue comme une opération ponctuelle et sans contrainte de mise en production, l'expérimentation a bénéficié d'une grande liberté technique, permettant d'innover, d'itérer rapidement et de se concentrer sur les approches les plus pertinentes pour construire la base d'apprentissage NAF 2025.

Le principe consiste à utiliser les LLM comme des *annotateurs virtuels*, en reproduisant la démarche décrite précédemment. À partir des notes explicatives, le LLM attribue le code NAF 2025 le plus approprié parmi les choix proposés et, lorsqu'il ne peut statuer, classe le cas comme incodable. La base recodée correspond donc à l'ensemble des unités que le LLM parvient à annoter : une partie restera incodable et exclue, mais une proportion substantielle de la base multivoque est recodée, constituant ainsi une première version de la base d'apprentissage NAF 2025, destinée à l'entraînement d'un futur classifieur.

À partir de la base d'apprentissage initiale en NAF 2008, la démarche consiste d'abord à séparer les unités univoques des unités multivoques à l'aide de la table de correspondance. Les univoques sont ensuite directement recodées et constituent une première composante de la future base d'apprentissage. Les multivoques font, quant à elles, l'objet d'un traitement spécifique: après suppression des libellés dupliqués afin de réduire le nombre d'inférences, un LLM est mobilisé comme *annotateur virtuel*. Comme le modèle ne dispose pas de l'expertise métier des annotateurs, et encore moins en NAF 2025 qui est encore provisoire, son contexte est enrichi en lui fournissant, pour chaque observation, la liste des codes NAF 2025 envisageables accompagnés de leurs notes explicatives. Plusieurs approches d'augmentation du contexte seront présentées, notamment le RAG[17] et le CAG ; ce dernier sera privilégié, car il répond le mieux aux besoins métier tout en minimisant le coût d'inférence. Une phase d'évaluation permet ensuite d'identifier le LLM dont les annotations sont les plus alignées avec la vérité terrain issue de la campagne d'annotation manuelle. Enfin, les multivoques ainsi recodées sont réunies avec les univoques, afin de constituer la nouvelle base d'apprentissage en NAF 2025 destinée à entraîner le futur classifieur.

Le schéma illustre fastText comme modèle de classification, car il a servi aux premières itérations, mais la méthodologie est indépendante du modèle : une fois la base d'apprentissage NAF 2025 constituée, tout autre classifieur supervisé peut être utilisé, notamment ceux de la librairie torchTextClassifiers ou d'architectures plus récentes adaptées au contexte métier.

[16] ou plus largement de cluster Kubernetes

[17] Le RAG (Retrieval-Augmented Generation) décrit plus se montre très prometteur. Cependant, son intégration reste aujourd'hui confrontée à des défis techniques et métier (temps d'inférence, évaluation et efficacité du retrieval, performance globale avec la génération). Pour ces raisons, le RAG est encore en phase de recherche et de prototypage dans notre contexte.

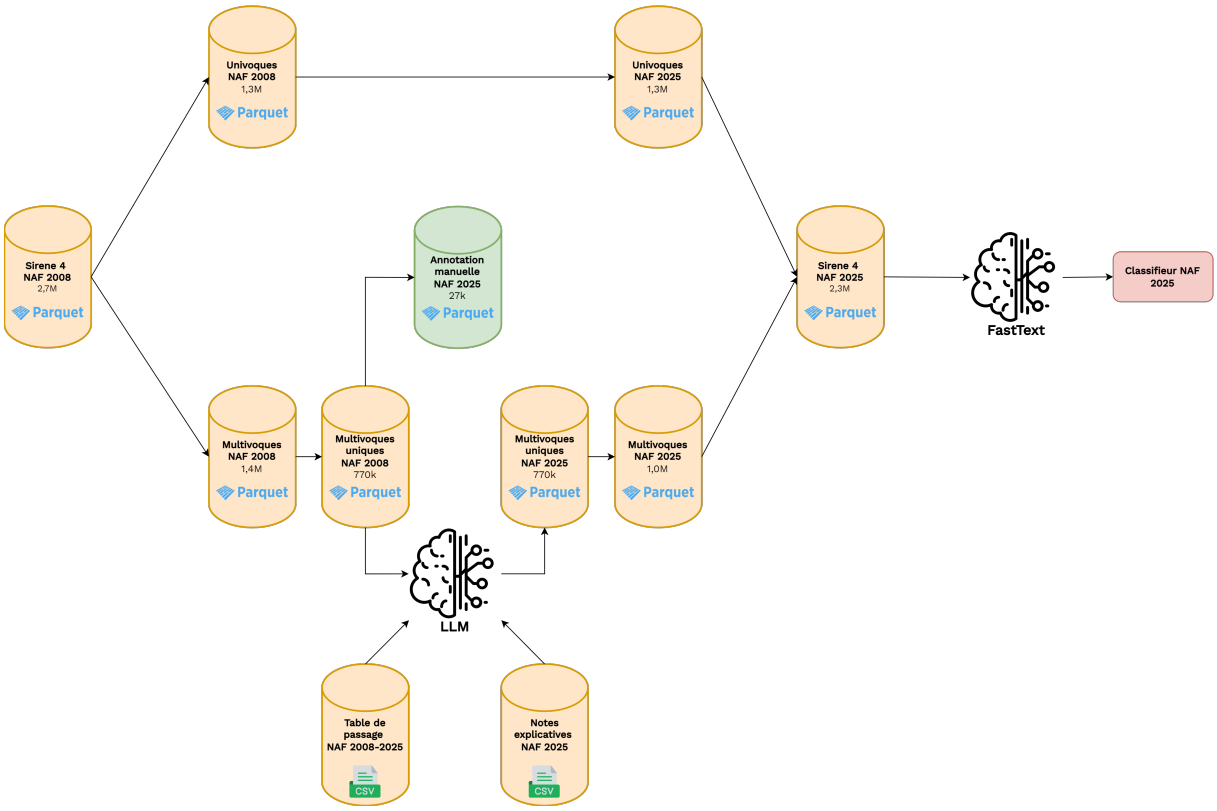


Figure 22. – Approche globale pour construire le modèle en NAF 2025

3.2. Le datalab SSP Cloud comme plateforme d'expérimentation

Le datalab *SSP Cloud* n'est pas destiné à héberger un LLM en production ; il s'agit d'un environnement expérimental et ponctuel qui permet de tester des modèles et des pipelines sans les contraintes d'une mise en service continue. Cette orientation expérimentale s'inscrit naturellement dans une démarche *open-source*, car elle favorise la transparence, la réutilisabilité du code et la maîtrise des coûts.

Développée à l'Insee, la plateforme SSP Cloud est un datalab à destination du *système statistique publique* (SSP) et au milieu académique. En outre, le SSP Cloud s'appuie sur un cluster *Kubernetes* qui constitue le socle d'une infrastructure robuste. Grâce à ce cluster, le déploiement automatisé d'applications potentiellement intensives en données devient possible, ce qui permet de simuler un véritable environnement de production tout en conservant la flexibilité d'un cadre de test.

Par ailleurs, le SSP Cloud a été construit selon les standards les plus récents des infrastructures de data-science. Concrètement, les services sont lancés sous forme de conteneurs *Docker* ; ainsi, chaque composant est empaqueté avec ses dépendances, ce qui assure une forte reproductibilité des déploiements. Cette approche implique toutefois une phase de développement légèrement plus coûteuse, car elle requiert la définition et la gestion des images Docker. Le datalab adopte également une philosophie *cloud-native* : il est organisé autour d'un ensemble modulaire de briques logicielles (code, données, configuration, environnement d'exécution). Cette séparation nette constitue un principe majeur des bonnes pratiques, car elle facilite la maintenance, le scaling et l'intégration de nouveaux outils.

Sur le plan de la souveraineté et de l'indépendance, toutes les technologies utilisées sont auto-hébergées et open-source. Même si les jeux de données manipulés ne sont pas sensibles, aucune offre propriétaire n'est employée, ce qui préserve la maîtrise totale du système. Il convient de préciser que,

conformément à la politique de la plateforme, seules des données non sensibles[18] (données publiques, jeux de données synthétiques, données agrégées anonymisées, etc.) peuvent être déposées ou traitées sur le datalab ; les données à caractère personnel ou confidentielles sont strictement exclues. En adoptant une approche frugale, nous sélectionnons des modèles open-source adaptés à nos besoins pour une période déterminée, réduisant ainsi les dépenses liées aux licences de solutions commerciales. L'auto-hébergement nous garantit en outre un contrôle complet sur nos données et un accès à des modèles affichant des performances comparables à celles des solutions propriétaires, spécifiquement pour notre cas d'usage.

Enfin, le SSP Cloud met à disposition des ressources matérielles précieuses. Chaque projet bénéficie d'un espace de calcul pouvant atteindre au moins 30 CPU, ainsi que d'un accès sur demande à des GPU NVIDIA H100 (PCI-e, NVLink), ressources rares mais essentielles pour les traitements IA massifs. Le stockage est assuré par un espace S3, offrant une capacité élevée et une grande disponibilité pour le versionnage et le partage des jeux de données.

En résumé, le datalab SSP Cloud combine normalisation, modularité, souveraineté et puissance de calcul, tout en imposant la restriction à des données non sensibles, ce qui en fait une plateforme idéale pour mener des expérimentations IA rigoureuses, maîtriser les coûts et respecter les exigences de la fonction publique.

3.2.1. Des services à disposition pour les traitements IA

Le SSP Cloud propose un catalogue de services variés, adaptés aux besoins des traitements IA. Certaines solutions sont particulièrement pertinentes pour les workflows de LLM et pour l'inférence sur de larges volumes de données. L'objectif est de présenter brièvement les concepts clés, les exemples disponibles dans le datalab et ceux qui seront exploités dans ce cadre.

3.2.1.1. Bases de données vectorielles

Les bases de données vectorielles permettent de stocker et d'interroger des représentations vectorielles de données, indispensables pour la recherche sémantique et l'analyse par similarité. Sur le datalab, plusieurs solutions sont proposées, pour en citer quelques-unes, Milvus, ChromaDB et Qdrant, cette dernière étant retenue pour les travaux présentés. Ce type de base peut notamment servir à conserver les représentations vectorielles de notes explicatives générées par un modèle de type transformer ou par un LLM spécialisé dans l'embedding, afin de faciliter des requêtes sémantiques pertinentes sur de larges corpus.

Au-delà du simple stockage, l'enjeu réside dans la capacité à interroger efficacement ces vecteurs. En effet, un dataframe se limiterait à conserver les embeddings et imposerait des comparaisons successives entre vecteurs, ce qui devient rapidement inefficace lorsque le volume de données augmente. À l'inverse, une base de donnée vectorielle comme *Qdrant* intègre des structures d'indexation dédiées permettant de retrouver rapidement les vecteurs les plus proches sur le plan sémantique. Cette capacité est déterminante pour interroger un nombre important d'embeddings de manière performante, par exemple pour identifier les notes explicatives les plus similaires au sein d'un corpus volumineux.

3.2.1.2. Serveurs d'inférence optimisés

Les serveurs d'inférence permettent l'exécution performante des modèles LLM. Plusieurs projets open source existent, dont *Llama.cpp* et *Ollama*, mais *vLLM* se distingue par son optimisation sur GPU et sa capacité à gérer l'inférence sur des millions de données. Cette solution est particulièrement

[18] Pour le traitement de données stratégiques, il est possible d'instancier Onyxia en interne pour avoir son propre datalab et manipuler des données sensibles à l'instar de *Nubo*, le cloud privé sécurisé de la DGFIP.

adaptée au traitement par batch, un domaine récent présentant des défis spécifiques, notamment en termes de vitesse et de scalabilité.

Les moteurs d'inférence classiques fonctionnent avec un *batching statique*, comparable à une chaîne de production traditionnelle : un lot de requêtes est traité, puis l'opération s'arrête avant de passer au lot suivant. Cette approche entraîne des latences, notamment lorsque de nouvelles requêtes arrivent alors qu'un batch est déjà en cours de traitement.

À l'inverse, *vLLM* adopte le principe du *continuous batching* : au lieu d'attendre que toutes les séquences d'un lot soient terminées, les requêtes finalisées sont immédiatement remplacées par de nouvelles, à chaque itération. Cela permet de maintenir les GPU constamment occupés, d'augmenter drastiquement le débit de traitement, de réduire la latence et d'améliorer l'utilisation des ressources. L'analogie serait celle d'un serveur de restaurant prenant les commandes de plusieurs tables en continu, plutôt que de multiplier les allers-retours pour chaque commande individuellement. Grâce à cette approche dynamique, vLLM est particulièrement adapté aux traitements portant sur des millions de données, en particulier lorsqu'une exécution en batch est requise.

3.2.1.3. Gestionnaire de prompts

Les gestionnaires de prompts permettent de versionner, tracer et évaluer les prompts afin d'optimiser l'interaction avec les LLM. Dans le datalab, la plateforme *Langfuse*, open source, est utilisée pour ces tâches. Dans le cadre de ce projet, il sera crucial de trouver les bonnes formulations de consignes pour les LLM afin d'assurer que le comportement attendu soit respecté, tout en conservant la traçabilité et la reproductibilité des expérimentations.

3.2.1.4. Gestionnaire de modèles

Les artefacts des modèles sont centralisés dans un model store, ici géré via *MLflow*, permettant de suivre les versions, les déploiements et la traçabilité des modèles utilisés dans les workflows. Cette gestion garantit un suivi précis des modèles employés pour les différents traitements IA et facilite la reproductibilité des résultats.

3.2.2. Les enjeux de l'inférence LLM par batch

L'inférence désigne le processus par lequel un modèle de langage (LLM) génère des prédictions ou des réponses à partir d'entrées données. Dans une inférence unitaire classique, chaque requête est traitée individuellement. Même dans ce cadre, plusieurs facteurs influencent déjà la performance et la fiabilité, tels que le modèle utilisé, la formulation du prompt, le serveur d'inférence ou les spécificités matérielles. Ces éléments déterminent le temps de réponse et peuvent provoquer des ralentissements ou des erreurs, mais restent généralement maîtrisables lorsque les volumes de données sont limités.

En revanche, lorsque l'inférence est effectuée par batch, des contraintes supplémentaires apparaissent. Le temps d'inférence n'est jamais linéaire et le traitement simultané de plusieurs éléments peut générer des variations importantes de performance. La vitesse globale est fortement impactée, le risque d'interruption augmente en cas d'erreur sur un élément, et la capacité de traitement devient limitée par l'infrastructure locale ou par d'éventuelles restrictions des services externes. Par ailleurs, le traitement de larges volumes de données entraîne des charges matérielles et financières significatives.

Afin de répondre à ces enjeux, une approche entièrement hors-ligne est adoptée, avec l'importation locale de toutes les technologies nécessaires. Cette méthode permet d'éliminer les ralentissements et interruptions liés aux flux réseau, de traiter de très grands volumes de données sans être limité par des quotas ou des restrictions d'API, et d'optimiser la parallélisation des tâches pour réduire le temps de traitement. Elle permet également de maîtriser pleinement les coûts et l'infrastructure.

Ainsi, l'inférence par batch devient plus rapide, fiable et scalable. À titre d'exemple, l'application de cette méthodologie sur une base de 2 millions de données nécessite jusqu'à 5 jours, bien que des optimisations aient permis de réduire ce temps de manière significative, ce qui souligne l'importance stratégique d'une telle organisation.

3.3. Détail des étapes et concepts clés appliqués

3.3.1. Étape 1: Préparer les données à disposition

La première étape consiste à rassembler et structurer l'ensemble des données nécessaires à l'annotation automatique par LLM. Les sources principales incluent les extractions préparées par le métier, issues du Système Sirene version 4, et destinées à être recodées en NAF 2025, constituant ainsi la base d'apprentissage initiale.

S'y ajoutent les ressources des experts de la nomenclature de la DNE, en particulier la table de correspondance théorique et les notes explicatives, qui apportent un contexte métier crucial pour traiter les cas multivoques ou ambigus. L'annotation manuelle effectuée lors de la campagne nationale fournit une première vérité terrain (ground truth), servant de référence pour évaluer la qualité des prédictions générées par les modèles.

Selon la stratégie choisie pour enrichir le contexte, les données sont organisées différemment. Avec la stratégie RAG (Retrieval-Augmented Generation), elles sont stockées dans une base de données vectorielle, permettant une recherche efficace de passages pertinents pour chaque libellé. Avec la stratégie CAG (Context-Augmented Generation), ces informations sont encapsulées sous forme d'objets en programmation orientée objet, chaque code et son descriptif devenant un attribut. Cette organisation facilite la manipulation et l'exploitation des informations lors de l'inférence, garantissant que le LLM dispose du contexte nécessaire pour produire des annotations cohérentes.

Les notions de RAG et CAG seront définies plus en détail ultérieurement, afin de préciser leur impact sur l'annotation automatique.

3.3.2. Étape 2 : Chargement de LLM

Un **LLM (Large Language Model)** repose sur l'architecture transformer, introduite dans l'article *Attention Is All You Need* N. S. e. a. Ashish Vaswani [8]. Cette architecture est centrée sur le mécanisme d'auto-attention, qui permet au modèle de pondérer l'importance relative de chaque élément d'une séquence lorsqu'il produit une sortie. Dans les LLM, cette structure est largement étendue : de très nombreux paramètres sont entraînés pour prédire la probabilité d'apparition d'un mot ou d'un token à partir de son contexte. Plusieurs modèles de type GPT existent, différenciés par leur taille, le volume de données d'entraînement et la finesse de leurs représentations internes.

Le modèle pré-entraîné dispose de représentations internes complexes qui codent de façon compressée des connaissances extraites du texte utilisé lors de son entraînement. Il ne s'agit pas d'une base de connaissances explicite ou consultable comme une base de données classique, mais d'une structure de paramètres qui capture des régularités statistiques et sémantiques. Cette compression de l'information permet au LLM de générer des réponses cohérentes, de compléter du texte ou d'attribuer des classes aux données, en mobilisant ces représentations implicites.

Cette perspective permet de comprendre pourquoi la taille du modèle et la richesse de ses données d'entraînement jouent un rôle central dans ses performances. La course actuelle à la puissance de calcul et à la multiplication des paramètres suit cette logique, comme l'illustre l'essai *The Bitter Lesson* R. Sutton [9].

Pour ce qui concerne les LLM à utiliser, la recherche se fait sur *Hugging Face*, la plateforme la plus répandue à l'heure actuelle pour accéder aux modèles de langage open source. Elle propose un large catalogue de modèles pré-entraînés et fournit des outils pratiques pour leur téléchargement et leur intégration dans différents environnements expérimentaux.

3.3.3. Étape 3: Paramétrer le LLM

3.3.3.1. Prompting

Un **prompt** correspond à la description textuelle de la tâche que le modèle doit accomplir. Il sert de guide au LLM pour orienter sa génération, fournir un format attendu et garantir que la réponse respecte les contraintes métier.

3.3.3.1.1. Donner une feuille de route au LLM: le prompt système, un prompt structurant

Le **prompt système** décrit globalement le rôle ou le mode opératoire du LLM avant qu'il traite les requêtes. Il agit comme un manuel d'instruction que le LLM « lit » avant de traiter toute requête. Il fixe le cadre général et la manière dont les réponses doivent être structurées.

Le prompt système appliqué au LLM pour labelliser en NAF 2025 est le suivant:

```
Tu es un expert de la Nomenclature statistique des Activités économiques dans la
Communauté Européenne (NACE). Tu es chargé de réaliser le changement de
nomenclature. Ta mission consiste à attribuer un code NACE 2025 à une entreprise,
en t'appuyant sur le descriptif de son activité et à partir d'une liste de codes
proposés (identifiée à partir de son code NACE 2008 existant). Voici les
instructions à suivre :
1. Analyse la description de l'activité principale de l'entreprise et le code NACE
2008 fourni par l'utilisateur.
2. À partir de la liste des codes NACE 2025 disponible, identifie la catégorie la
plus appropriée qui correspond à l'activité principale de l'entreprise.
3. Retourne le code NACE 2025 au format JSON comme spécifié par l'utilisateur. Si
la description de l'activité de l'entreprise n'est pas suffisamment précise pour
identifier un code NACE 2025 adéquat, retourne `null` dans le JSON.
4. Évalue la cohérence entre le code NACE 2008 fourni et la description de
l'activité de l'entreprise. Si le code NACE 2008 ne semble pas correspondre à
cette description, retourne `False` dans le champ `nace08_valid` du JSON. Note que
si tu arrives à classer la description de l'activité de l'entreprise dans un code
NACE 2025, le champ `nace08_valid` devrait `True`, sinon il y a incohérence.
5. Réponds seulement avec le JSON complété aucune autres information ne doit être
retourné.
```

3.3.3.1.2. Le prompt utilisateur, une requête contextuelle au LLM

Un **prompt utilisateur** correspond à la requête spécifique adressée à un LLM pour exécuter une tâche. Il fournit le contexte précis et structuré, ainsi que les indications nécessaires pour générer une réponse conforme aux attentes. Contrairement au prompt système, qui fixe les règles et le cadre global de réponse, le prompt utilisateur décrit la tâche concrète à traiter.

Le comportement du LLM dépend fortement de la formulation du prompt. En tant que modèle basé sur l'attention, le LLM pondère les éléments du texte et adapte sa réponse en fonction de la structure et des mots employés. Une variation dans la formulation peut entraîner des sorties différentes. Cela explique l'importance de versionner et d'optimiser les prompts, démarche connue sous le nom de

prompt engineering, afin d'obtenir des réponses fiables et reproductibles[19].

Un prompt spécifique est appliqué pour chaque observation comprenant :

- le **libellé** de l'activité principale de l'entreprise
- l'ancien **code NAF 2008 ou NAF rev 2** connu
- La **liste des codes possibles** issues de l'appariement avec leurs **notes explicatives**
- Une **instruction sur le format** de réponse attendu

Chaque prompt utilisateur constitue ainsi un ensemble d'informations structurées permettant au LLM de produire une sortie directement exploitable dans le travail d'annotation, tout en restant sensible à la formulation et à l'ordre des informations fournies.

3.3.4. Étape 4: Apporter le contexte au LLM

Les LLM pré-entraînés possèdent une *base de connaissance implicite*, issue de l'apprentissage massif sur de grandes quantités de textes, mais cette connaissance reste générale et insuffisante pour des tâches spécifiques, telles que le recodage précis des activités économiques selon la NAF 2025. Il est donc nécessaire d'enrichir le modèle avec un contexte externe pour orienter la génération et réduire les erreurs. Faute d'informations précises, le modèle risque d'*halluciner*, c'est-à-dire de générer des réponses plausibles mais factuellement erronées. Il est donc nécessaire d'enrichir le modèle avec un contexte externe pour orienter la génération et réduire les erreurs.

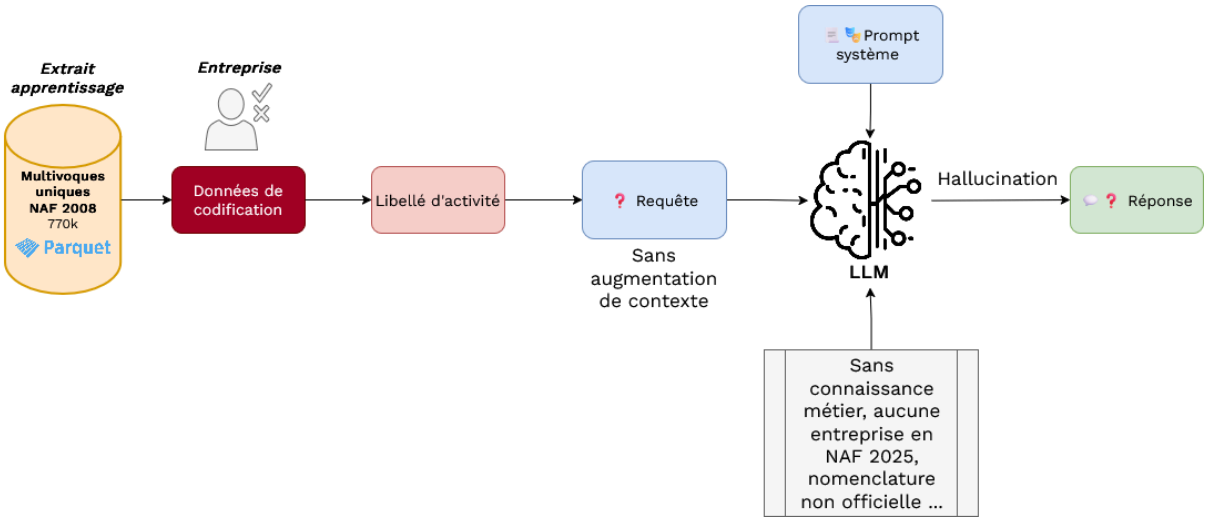


Figure 23. – Hallucination du LLM en l'absence de contexte métier

Deux stratégies principales permettent d'intégrer ce contexte :

3.3.4.1. RAG (Retrieval-Augmented Generation)

RAG combine un modèle génératif avec un module de recherche d'information. La requête est comparée à des documents ou embeddings existants via une mesure de similarité, et les documents les plus proches servent de contexte au LLM A. P. e. a. Patrick Lewis Ethan Perez [10]

Dans le contexte d'annotation pour la NAF 2025, la stratégie RAG consiste à rechercher, parmi un ensemble de documents ou d'embeddings, les descriptifs les plus similaires à l'activité de l'entreprise.

[19]Il est important de noter que la reproductibilité exacte d'une réponse n'est garantie que si le modèle est configuré de manière déterministe, c'est-à-dire avec une absence de randomisation (temperature=0) et des seeds fixes lorsque le LLM le permet. Dans la pratique, les modèles sont souvent configurés pour générer des réponses variées, ce qui peut produire des sorties différentes pour un même prompt. La discipline du prompt engineering consiste à concevoir et versionner les prompts pour maximiser la cohérence et la pertinence des réponses, tout en réduisant cette variabilité lorsqu'elle est préjudiciable au projet.

Ces éléments servent alors de contexte pour le modèle. Cette approche présente l'avantage de proposer des codes basés sur la similarité avec les descriptifs, indépendamment de l'ancien code attribué. Elle évite donc la propagation d'éventuelles erreurs historiques et permet de générer des correspondances nouvelles ou inattendues. En revanche, elle ne garantit pas la cohérence avec l'ancien code : si aucun des codes récupérés n'est pertinent, la tâche peut devenir incodable, ou produire des propositions incorrectes ou incohérentes.

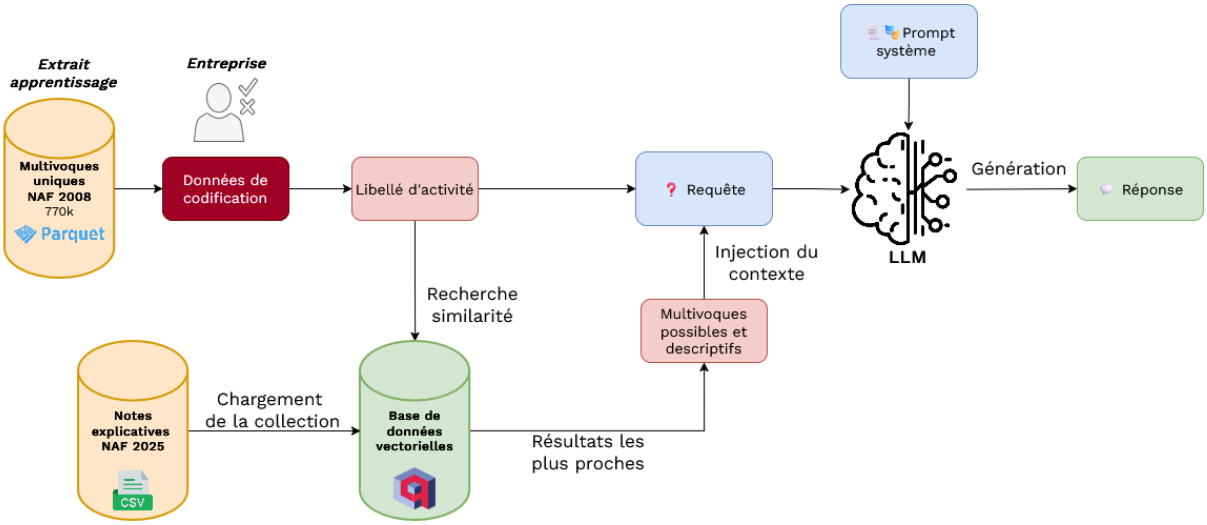


Figure 24. – Application du RAG comme stratégie pour la labellisation de masse

3.3.4.2. CAG (Context-Augmented Generation)

La stratégie CAG, quant à elle, consiste à fournir un contexte structuré et préexistant au LLM. Pour en savoir davantage, il est recommandé de le lire l'article de A. RANA [11]. Cette méthode tire parti des grandes fenêtres de contexte des LLM récents, qui permettent désormais d'inclure l'ensemble des informations nécessaires à la tâche. Cela présente un intérêt particulier, car le modèle peut s'affranchir des erreurs de retrieval lorsque le document complet est fourni comme contexte. Ce type d'intégration n'était pas possible avec les premiers modèles open source, dont la fenêtre de contexte limitée restreignait fortement la quantité d'informations exploitables.

Dans cette application, les codes NAF, leurs descriptifs et les notes explicatives correspondantes sont encapsulés sous forme d'objets en programmation orientée objet, chaque objet associant le code à son libellé et à ses notes explicatives. Cette approche permet de garantir que les codes proposés restent cohérents avec l'ancien code de l'entreprise. Lorsque les correspondances historiques sont fiables, comme cela a été vérifié lors de la campagne d'annotation, cette méthode offre une qualité élevée des propositions. Toutefois, si l'ancien code contient une erreur, celle-ci peut se propager dans la liste des codes candidats, augmentant les risques de mauvaise attribution ou de limitation dans le choix des codes disponibles.

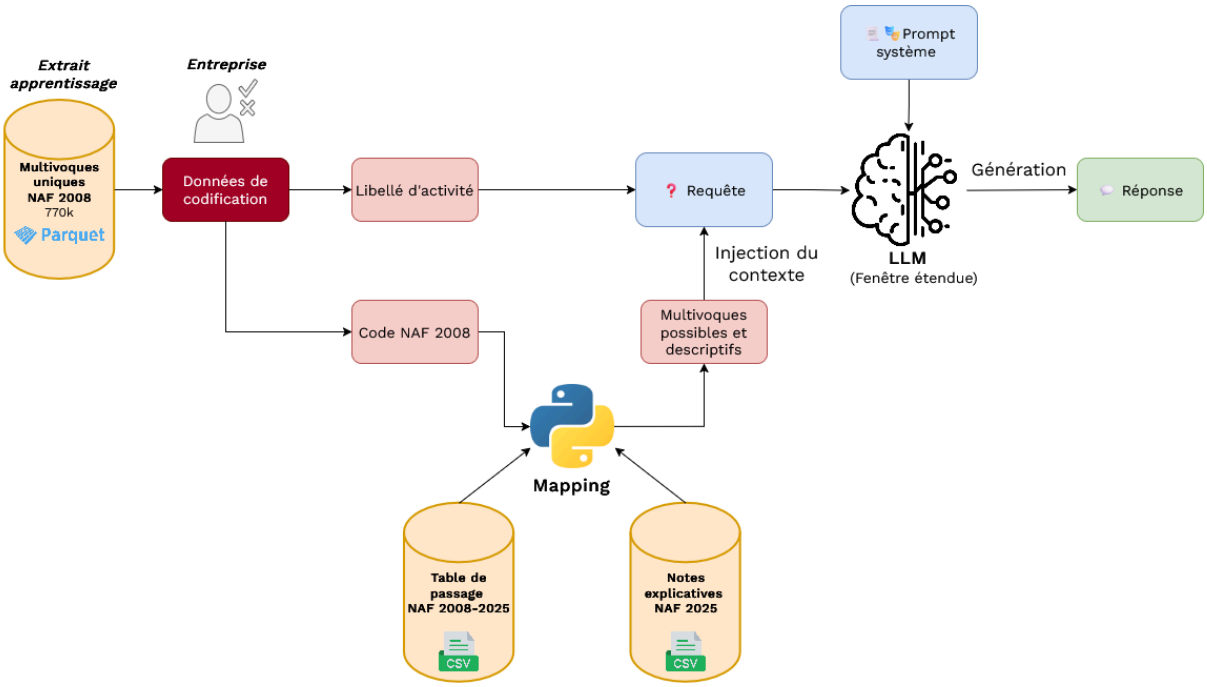


Figure 25. – Application du CAG comme stratégie pour la labellisation de masse

3.3.4.3. Quelle stratégie retenir pour la suite ?

3.3.5. Comparaison des stratégies de génération de contexte pour NAF 2025

Stratégie	Principe	Avantages	Limites	Pertinence pour NAF 2025
RAG	Recherche des descriptifs les plus similaires à la requête pour fournir un contexte au LLM	Permet de générer des codes indépendamment de l'ancien code, exploration de correspondances nouvelles	Cohérence avec l'ancien code non garantie, risque d'incodabilité si aucun code pertinent n'est trouvé	Utile si ancien code incertain ou pour explorer de nouvelles correspondances
CAG	Fourniture d'un contexte structuré préexistant au LLM, encapsulé sous forme d'objets	Assure la cohérence avec l'ancien code, haute qualité des propositions si l'ancien code est fiable	Propagation possible d'erreurs historiques, limitation par multivoques	Pertinent si l'ancien code est fiable, comme dans le cas de cette étude

Compte tenu des résultats obtenus lors de la campagne d'annotation et du faible risque de problèmes de cohérence évalué pour l'application métier, la stratégie CAG est retenue pour l'ensemble de l'étude et pour constituer la base du futur modèle en production. Cette approche garantit des propositions cohérentes et exploitables, notamment grâce à l'encapsulation des codes et descriptifs associés dans des objets en POO, facilitant leur manipulation et exploitation lors de l'inférence.

La piste RAG reste ouverte pour des expérimentations conjointes entre le SSP Lab, le Data Scientist et le métier, afin d'évaluer son intérêt dans des contextes où l'ancien code pourrait être incertain ou pour enrichir la recherche de correspondances multivoques. Elle permet alors d'explorer de nouvelles

associations de codes indépendamment de l'historique, mais avec un risque accru de propositions incohérentes ou incodables.

3.3.6. Étape 5: Générer les annotations

Cette étape concerne la génération des annotations par le LLM, avec une attention particulière sur le contrôle de sa sortie, la spécification des informations attendues et le formatage afin de garantir des résultats exploitables et cohérents.

Les LLM ont tendance à produire des réponses très **volubiles** et détaillées, ce qui peut compliquer leur exploitation directe pour des tâches structurées. Pour pallier ce phénomène, il est recommandé de construire une **réponse type** claire et concise, organisée selon un **format spécifique**. Dans de nombreux projets et selon la littérature scientifique, le format JSON est largement utilisé car il permet une manipulation simple et automatisée des données générées tout en assurant une validation facile.

```
{
    "codable": true,
    "nace_2008_valid": true,
    "nace2025": "0147J"
}
```

Une fois la réponse générée, un **parsing** est appliqué. Il consiste d'abord à **vérifier** que le format respecte le schéma attendu, puis à contrôler l'absence d'**hallucinations** ou d'informations incohérentes. Cette démarche de spécification et contrôle strict des sorties est largement documentée dans la littérature comme étant efficace pour fiabiliser les résultats des LLM. Des travaux récents, tels que l'approche StructuredRAG C. P. e. a. Connor Shorten [12], montrent d'ailleurs l'intérêt de combiner génération augmentée et contraintes de structure pour obtenir des sorties précises, normalisées et robustes.

3.3.7. Étape 6 : Évaluation des LLM

La sélection des Large Language Models (LLM) s'est concentrée sur les modèles les plus reconnus, considérés comme State-of-the-Art (SOTA) à un moment donné, ou **disponibles en open source**. Il est certain qu'au moment de la rédaction, d'autres LLM pourraient supplanter ceux présentés, et c'est une excellente nouvelle pour l'innovation. Cependant, ces modèles ont été choisis pour établir la base de référence (baseline) de cette première analyse. Cette approche était indispensable pour des raisons de faisabilité et d'optimisation des ressources, privilégiant l'étude approfondie de cet échantillon clé plutôt qu'une multiplication exhaustive des tests.

Les modèles retenus présentent des profils variés, permettant d'observer un éventail contrasté de comportements selon les ressources matérielles, les contraintes d'inférence et les exigences de performance.

3.3.7.1. Le défi de l'évaluation des LLM

L'évaluation des performances d'un LLM représente un véritable défi, en particulier lorsqu'il s'agit d'une tâche de classification fine dans une nomenclature. Contrairement à une classification classique, l'attribution d'un code NAF 2025 s'inscrit dans une nomenclature complexe où les frontières entre catégories peuvent être subtiles et où plusieurs choix peuvent paraître justifiables. Cette spécificité rend l'évaluation nettement moins triviale qu'il n'y paraît.

Afin de garantir une mesure fiable, les 30 000 annotations de la campagne en NAF 2025 ont été utilisé comme ground truth. Ce jeu de données constitue la référence permettant de comparer objectivement les prédictions générées par les modèles.

3.3.7.2. Les métriques d'évaluation

Trois métriques complémentaires ont été retenues pour évaluer les performances :

- la **précision totale** qui mesure la proportion de prédictions correctes sur l'ensemble des observations.
- la **précision sur les cas « codables »** qui évalue la justesse uniquement sur les observations pour lesquelles le modèle a estimé qu'un code pouvait être attribué. Cette métrique mesure la qualité lorsque le LLM prend effectivement une décision.
- la **précision « LLM »** qui isole les erreurs imputables au modèle lui-même en excluant les biais provenant de la stratégie de contexte ou du choix des candidats. Elle permet d'évaluer la performance intrinsèque du modèle génératif.

L'utilisation conjointe de ces trois métriques offre une vision plus nuancée de la performance, en distinguant la justesse globale, la fiabilité des décisions prises et la robustesse intrinsèque du LLM.

3.3.8. Étape 7 : Générer le jeu multivoques

3.3.8.1. Le principe

Une fois les prédictions produites par chaque modèle, l'objectif est de capitaliser sur la diversité des réponses afin de construire un jeu d'annotations multiples, plutôt que de retenir l'avis d'un seul modèle. L'approche consiste à considérer chaque LLM comme un annotateur individuel, à l'image d'un groupe d'experts humains fournissant chacun leur propre proposition de code. Cette logique rapproche l'usage des LLM des pratiques d'annotation traditionnelle en garantissant un regard croisé sur chaque observation.

L'idée sous-jacente repose sur un constat simple : plusieurs « cerveaux » valent mieux qu'un seul. En combinant les prédictions de différents LLM, il devient possible d'exploiter leurs forces respectives et, potentiellement, d'améliorer la qualité globale des annotations.

3.3.8.2. Les stratégies d'annotation

L'enjeu est alors de déterminer si la combinaison de plusieurs prédictions permet d'obtenir de meilleures performances qu'un LLM utilisé seul. Dans cette perspective, trois stratégies de consolidation ont été étudiées pour tirer parti de la complémentarité des modèles:

- **sélection en cascade**, consistant à définir un ordre de priorité entre annotateurs individuels et à retenir la première annotation jugée exploitable.
- **vote à la majorité**, où le code retenu est celui proposé par le plus grand nombre de modèles.
- **vote pondéré**, attribuant à chaque modèle un poids proportionnel à sa performance afin de privilégier les décisions des annotateurs réputés les plus fiables.

Ce jeu multivoques constitue ainsi une première base d'apprentissage en NAF 2025. La prochaine étape consiste à entraîner un modèle spécialisé sur cette nouvelle nomenclature et à évaluer ses performances en vue d'un passage en production, les bases univoques et multivoques étant concaténées pour former la nouvelle base d'apprentissage en NAF 2025.

4. Réentraînement en NAF 2025

À l'issue de la génération du jeu d'annotations en NAF 2025, cette partie aborde l'entraînement un premier modèle de classification sur cette nouvelle nomenclature, en tirant parti des multivoques produites par les LLM. Cette phase marque la transition entre l'expérimentation et la construction d'un modèle opérationnel utilisable par les équipes métier.

4.1. Réentraînement en NAF 2025

4.1.1. Un premier modèle NAF 2025 : un livrable obtenu avec succès

Les premiers résultats obtenus sont encourageants. Sur un stock initial de 2,7 millions d'unités légales de Sirene 4, environ 2,3 millions ont été reconstruites en NAF 2025, représentant la quasi-totalité des données exploitables. La distribution des données d'apprentissage est restée quasi-inchangée, ce qui garantit une continuité avec les modèles précédents.

Il est intéressant de noter que le taux d'inclassables est d'environ 6 % pour les experts, chiffre cohérent avec la campagne qualité menée en NAF 2008, alors qu'il atteint 15 % avec l'approche des annotateurs virtuels. Les modèles ont tendance à considérer davantage d'unités comme incodables, sans que cela n'affecte significativement la distribution ni l'ordre de grandeur du volume initial, qui reste proche de celui du stock de départ. Cet indicateur isolé ne permet pas de conclure de manière définitive si l'approche globale par LLM est plus prudente que celle des annotateurs humains, mais il constitue néanmoins un résultat intéressant à suivre.

Dans ce contexte, les annotations virtuelles en NAF 2025 ont servi comme vérité terrain pour l'évaluation du classifieur, en considérant que des erreurs peuvent subsister, à l'instar du ground truth en NAF rev 2 issu de la gestion courante. L'objectif principal reste d'assurer un codage cohérent et fiable à l'échelle globale, même si certaines erreurs ponctuelles ou un manque de précision au niveau 5 sont tolérables.

Plusieurs besoins d'amélioration ont été identifiés. Il s'agit principalement de corriger les cas problématiques détectés lors des premières évaluations et d'améliorer la cohérence globale entre les modèles NAF 2008 et NAF 2025. Pour ce faire, le modèle NAF 2008 actuellement en production est réentraîné et corrigé en parallèle, afin de garantir la compatibilité et l'alignement avec la nouvelle version.

Cette première version validée constitue donc une base exploitable, tout en restant perfectible pour intégrer les corrections nécessaires et optimiser la précision globale du classifieur en NAF 2025.

Un **premier modèle de prédiction NAF 2025** a été entraîné avec succès. Les résultats ci-dessous montrent une version initiale satisfaisante, atteignant un niveau de performance comparable au modèle actuellement en production (NAF 2008).

4.1.2. Résultats

L'évaluation des modèles s'appuie sur la méthodologie classique du machine learning, consistant à diviser les données en ensembles d'apprentissage, de validation et de test. L'ensemble d'apprentissage sert à entraîner le modèle, tandis que l'ensemble de validation permet d'ajuster les hyperparamètres. Enfin, l'ensemble de test n'est jamais utilisé pendant l'entraînement et constitue une référence objective pour mesurer la performance finale du modèle. Cette approche garantit que l'évaluation est indépendante de l'apprentissage et permet de détecter d'éventuels surapprentissages.

4.1.2.1. Modèle *fasttext* en NAF 2025

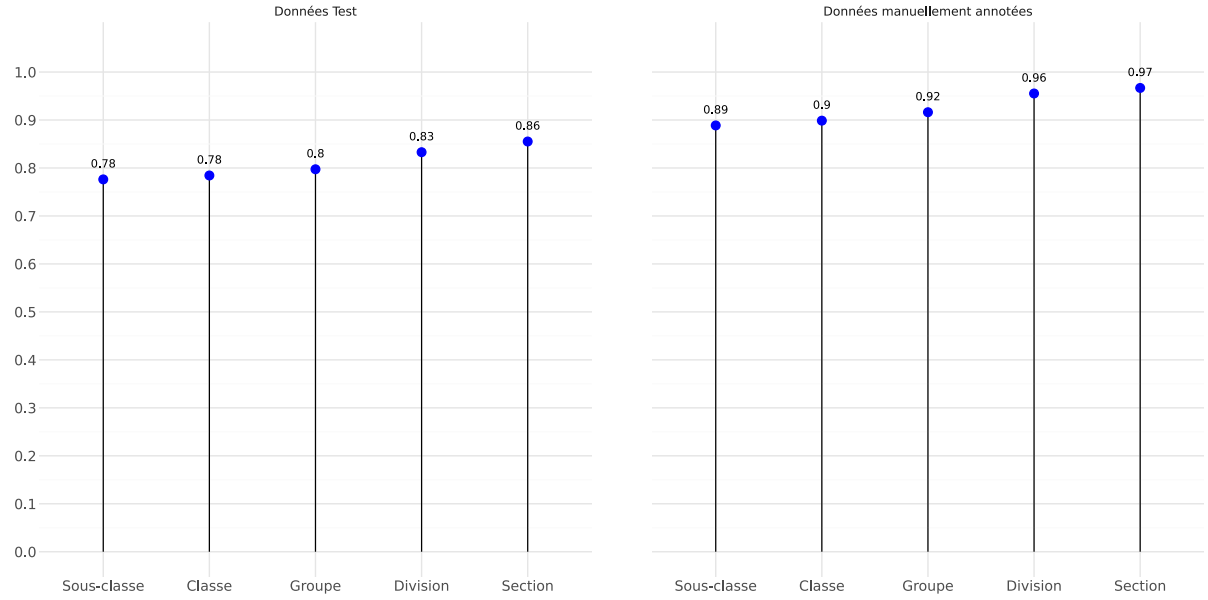


Figure 26. – Comparaison performances sur jeu de test classique et manuellement annoté

4.1.2.2. Modèle basé sur Torch naturellement meilleur

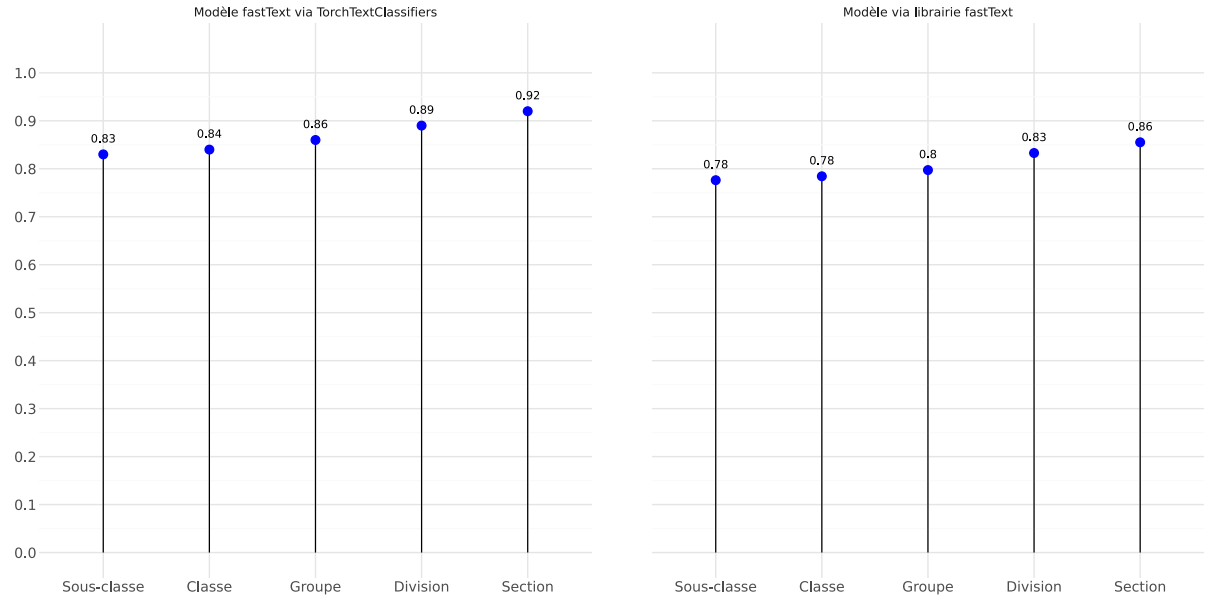


Figure 27. – Comparaison fastText vs torchTextClassifiers

L'entraînement sur la même base montre que les modèles Torch tirent avantage des variables annexes et atteignent de meilleures performances globales que fastText. Cela suggère que la gestion des variables supplémentaires, intégrée dans l'architecture Torch, améliore non seulement l'automatisation[20] mais aussi la précision. Toutefois, cette conclusion doit être interprétée avec prudence, car il s'agit d'une piste d'analyse et d'observation, et il est nécessaire de confirmer ces tendances sur d'autres ensembles de données pour valider scientifiquement l'impact exact des variables annexes.

[20] non présenté ici, l'automatisation est meilleure avec le modèle basé sur Torch

4.2. Opérations complémentaires avant passage en production

4.2.1. Data editing de la base d'apprentissage comme ground truth

Avant la mise en production du modèle, il est indispensable de qualifier et nettoyer la base d'apprentissage utilisée comme *ground truth*, en particulier pour les codes jugés sensibles par le métier. L'objectif est de corriger les erreurs historiques de codification afin d'améliorer la cohérence globale du jeu d'apprentissage.

Cette étape repose sur des corrections manuelles ou semi-automatisées, guidées par les remontées des experts métier. Au vu des résultats obtenus (cf. section performance), ces opérations n'ont pas dégradé les performances des modèles ; elles ont au contraire permis un gain significatif d'environ 13 points supplémentaires pour le modèle FastText et 5 points supplémentaires pour le modèle basé sur Torch, révélant une cohérence accrue de la base d'apprentissage.

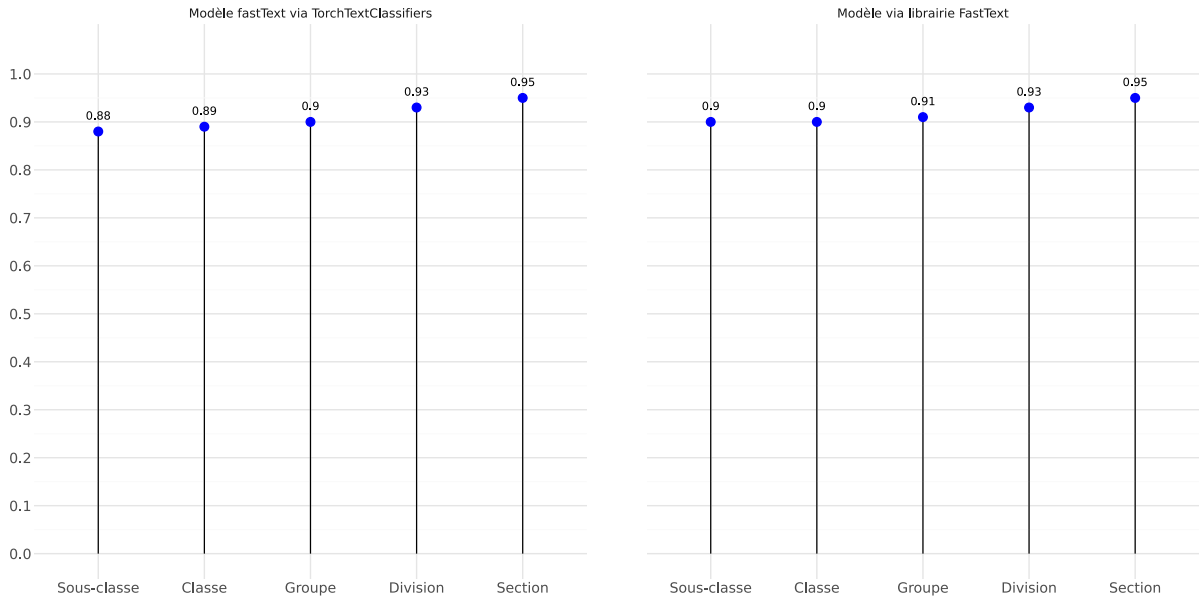


Figure 28. – Comparaison après nettoyage

4.2.1.1. Méthodologie de détection et de correction des libellés

Lorsqu'un libellé nécessite un recodage (ex. : si le libellé contient *LMNP*, alors le code doit être 6820G), la difficulté est de détecter automatiquement l'ensemble des variantes textuelles à corriger, présentes en stock et susceptibles d'apparaître en flux. Plusieurs méthodes complémentaires sont mobilisées, chacune présentant des avantages et limites :

Approche	Intérêt	Limites
Expressions régulières (regex)	Très fiable pour détecter des formes exactes ou des motifs spécifiques	Nécessite de lister les variantes ; peu généralisable
Fuzzy matching (distance de Levenshtein)	Excellente détection de fautes de frappe, pluriels, accords	Risque de faux positifs si deux termes proches n'ont pas le même sens
Similarité sémantique (ex. : all-MiniLM-L6-v2)	Détecte des variantes sémantiques non triviales, très complémentaire au fuzzy	Sensible au seuil choisi ; seuil trop bas ⟶ bruit, trop haut ⟶ manqués

Ces méthodes sont appliquées de manière combinée, afin de maximiser la détection des libellés à corriger sans introduire de bruit.

Concrètement, chaque méthode identifie un ensemble de lignes candidates au recodage ; le moteur de règles prend l'union des résultats lorsque cela est pertinent, pour couvrir un spectre de variantes le plus complet possible.

⚠️ Les seuils et combinaisons sont audités systématiquement : chaque changement est journalisé afin de contrôler et valider l'impact sur la base d'apprentissage, notamment sur la distribution statistique des codes.

4.2.1.2. Moteur de règles et data cleansing

L'ensemble de cette démarche constitue un moteur de règles dédié au data cleansing de la base d'apprentissage. Concrètement, les retours métier sont d'abord traduits en règles de recodage qui sont ensuite appliquées de manière systématique afin de corriger les libellés nécessitant une mise à jour, en particulier pour les codes jugés sensibles. Pour identifier les occurrences concernées, le moteur s'appuie sur les méthodes de matching textuel précédemment décrites, sélectionnées ou combinées selon la nature du libellé à traiter.

Cette approche permet d'élargir la détection des cas pertinents tout en conservant un haut niveau de précision, grâce à l'utilisation de seuils restrictifs et à une validation systématique par expertise. Chaque correction est enregistrée dans un journal afin d'en assurer la traçabilité et d'évaluer l'impact sur les données. L'objectif est ainsi d'améliorer la cohérence et la fiabilité de la base d'apprentissage, tout en préservant sa structure globale, afin de pouvoir surveiller la distribution des codes sans nécessairement la figer, puisque certaines évolutions peuvent être légitimes lorsque des erreurs historiques sont corrigées.

4.2.1.3. Idées ou pistes méthodologiques exploratoires

Dans une logique d'amélioration continue, plusieurs pistes pourraient être explorées pour renforcer encore l'efficacité du dispositif. Par exemple, des techniques de clustering sémantique appliquées aux embeddings (ex. all-MiniLM-L6-v2) permettraient de regrouper automatiquement les libellés textuels très proches. Cela offrirait un moyen de détecter, à large échelle, des variantes linguistiques conduisant à des codifications différentes alors qu'elles devraient relever du même code. Ce type d'analyse aiderait ainsi à identifier des incohérences structurelles dans l'historique de la codification.

De plus, une analyse ciblée permettrait d'identifier les libellés sensibles aux effets de bord (par exemple ceux partageant un vocabulaire proche mais renvoyant à des codes distincts). Ces libellés devraient alors être exclus des approches basées sur la distance de Levenshtein ou faire l'objet d'un traitement spécifique afin d'éviter des recodifications erronées par ressemblance textuelle.

Enfin, au-delà du nettoyage ponctuel, l'exploitation de métriques de cohérence intra-code (homogénéité sémantique des libellés regroupés dans un même code) et inter-codes (différenciation sémantique entre codes voisins) constituerait un levier précieux. Ces mesures permettraient non seulement de détecter des codes « fourre-tout » ou ambigus, mais aussi d'ajuster les consignes de codification pour renforcer la distinction entre codes proches, améliorer la précision sémantique du référentiel et, in fine, fiabiliser durablement la qualité du codage.

4.2.1.4. Point d'attention : un équilibre à préserver dans les opérations de data editing

Les opérations de data editing sont essentielles pour améliorer la qualité de la vérité terrain, mais elles comportent intrinsèquement des risques d'effets de bord. En corrigeant une anomalie locale, il est possible, par ricochet, de dégrader la qualité de codage d'autres segments de la base. Cette complexité tient au caractère multidimensionnel du codage, car l'amélioration d'un libellé isolé ne garantit pas la cohérence globale entre l'ensemble des codes, leurs frontières sémantiques et les volumes associés.

Ainsi, renforcer la qualité d'un code peut mécaniquement détériorer celle d'un autre si l'on n'adopte pas une vision systémique. Il est donc indispensable de réserver ces corrections aux cas présentant un enjeu métier réel, notamment lorsqu'elles concernent des codes sensibles ou fortement exposés dans la gestion opérationnelle.

Dans ce contexte, une démarche rigoureuse s'impose. Les ajustements doivent être testés avant leur déploiement à l'aide de cas tests représentatifs pour garantir qu'aucun biais supplémentaire n'est introduit. Parallèlement, le passage en production nécessite la mise en place d'une surveillance active et continue de la qualité du codage. Cette surveillance doit porter à la fois sur le stock existant et sur le flux des nouvelles codifications, afin d'identifier rapidement toute dérive, réapparition d'incohérences ou impact inattendu.

Cette approche de vigilance permanente s'inscrit dans les bonnes pratiques modernes de gouvernance des modèles, garantissant l'évolution de la vérité terrain tout en préservant la stabilité globale du système de codification.

4.2.2. Des besoins de surveillance en NAF 2025

Dans la perspective d'un déploiement opérationnel, il devient indispensable de disposer d'un dispositif de surveillance adapté au codage en flux. Se reposer uniquement sur un golden test ne permet pas d'assurer un suivi exhaustif et durable de la qualité des classifications en NAF 2025. Bien que ce test reste utile pour valider certains cas critiques, il ne couvre pas toutes les classes et ne permet pas de détecter d'éventuelles dérives globales dans un environnement en production.

Pour y remédier, il est nécessaire de définir des métriques de suivi plus globales et continues. Ces mesures doivent permettre de monitorer les performances du modèle sur l'ensemble du stock de données et de détecter les écarts significatifs, notamment pour les codes sensibles. Une telle approche offre la possibilité de garantir la fiabilité du système de codification, d'identifier les zones à risque, de prioriser les interventions humaines ou semi-automatisées et de soutenir une amélioration continue du modèle et de la vérité terrain associée.

5. Pistes d'amélioration et perspectives

Plusieurs pistes peuvent être explorées pour renforcer la qualité, l'efficacité et l'utilité de cette démarche. Un premier axe consiste à tester régulièrement des modèles de langage plus récents et à réévaluer la méthode sur des données plus actuelles que celles de la baseline, afin de mesurer les gains apportés par les nouvelles générations de LLM et de garantir l'adéquation aux pratiques et besoins métiers.

L'optimisation de l'inférence constitue un autre levier important. L'inférence par lots mobilise fortement les ressources de calcul, en particulier les GPU. Des améliorations sont possibles en optimisant le code, en affinant la gestion des lots et en choisissant des architectures logicielles et matérielles plus adaptées. Une approche plus sobre et alignée avec les principes du Green IT permettrait de réduire le coût énergétique des traitements et de favoriser des usages plus frugaux.

Le renforcement du RAG représente également une voie prometteuse. L'amélioration du retrieval, l'intégration de rerankers et une exploitation plus fine de la structure hiérarchique des nomenclatures pourraient permettre de mieux guider le modèle. Une évaluation plus précise permettrait de distinguer les erreurs liées à la recherche d'information de celles liées à la génération. Ces travaux pourraient être approfondis conjointement par le SSP Lab et les data scientists proches du métier pour identifier les cas où un RAG robuste compléterait efficacement le CAG.

Des perspectives existent aussi autour des approches agentiques. Un agent peut structurer sa réponse en plusieurs étapes de raisonnement, ce qui pourrait améliorer la qualité et l'interprétabilité du processus de codification. Les cas d'usage restent à explorer et pourraient dépasser le seul recodage du jeu d'apprentissage. Par exemple, une assistance interactive à la codification pour les gestionnaires pourrait être envisagée. Cette approche soulève néanmoins des défis d'inférence, notamment en matière de latence et de propagation d'erreurs. Une stratégie distincte peut consister à guider le codage par un cheminement progressif dans la nomenclature, de la section jusqu'au niveau sous-classe, afin d'améliorer la sélection du code final.

L'enrichissement du contexte par des sources internes ou externes constitue également un levier d'amélioration. La consultation de documents métiers supplémentaires ou de ressources issues de recherches Internet peut aider à clarifier certains termes présents dans les libellés, réduire les ambiguïtés et limiter les erreurs d'interprétation, notamment lorsque le vocabulaire est spécialisé ou polysémique. L'utilisation de nomenclatures complémentaires, comme la CPF (Classification des Produits Française), peut aussi apporter un contexte utile. La CPF classe l'économie à un niveau plus fin, au niveau des produits, biens et services. Elle peut s'avérer précieuse lorsque les libellés sont très précis et que plusieurs sous-classes NAF sont envisageables, car elle fournit des éléments concrets permettant de remonter au bon code en distinguant plus clairement la nature du produit ou du service associé. C'est d'ailleurs une pratique couramment utilisée par les experts et gestionnaires lorsqu'ils rencontrent ce type de difficulté, afin de conforter ou affiner le choix du code.

Cette expérimentation ouvre aussi des perspectives pour la mise en production et la maintenance des modèles. Les LLM peuvent soutenir les experts en facilitant le traitement de cas répétitifs ou longs à expertiser et en produisant des annotations utiles pour un meilleur suivi de la qualité, par exemple en surveillant la stabilité de la précision dans le temps afin de détecter des dérives. Toutefois, l'usage de LLM en production est limité par le coût, la latence et les besoins en GPU. En l'absence de moyens, il semble plus réaliste de les mobiliser de manière ponctuelle pour du suivi, de l'audit, de la préqualification de corpus ou des analyses ciblées. Les équipes métier peuvent contribuer à définir les cas d'usage où ces apports seraient les plus utiles. De leur côté, les équipes d'innovation informatique pourront étudier les solutions technologiques susceptibles d'intégrer ce type d'outils de manière plus sobre, fiable et maintenable.

Par ailleurs, si l'on devait imaginer ce type d'outils en production, plusieurs enjeux transverses devraient être anticipés. Sur le plan métier, il faudrait définir précisément les cas d'usage apportant une valeur ajoutée tangible. Sur le plan technique, les choix de modèles et d'architectures auraient un impact notable en termes de coûts et d'empreinte énergétique, renforçant la nécessité de privilégier des solutions plus frugales. L'intégration de LLM dans des chaînes opérationnelles introduirait également des défis de maintenance et de gouvernance, relevant de pratiques plus avancées que le MLOps actuel. Cela touche au domaine émergent du LLMops, incluant la gestion du contexte, des prompts, des versions de modèles et de la qualité de génération, et nécessiterait une montée en compétences et une collaboration renforcée entre experts métier, data science et innovation informatique.

Ces perspectives s'inscrivent dans la continuité des résultats obtenus et prolongent la conclusion en montrant que les LLM peuvent apporter de nouvelles capacités, tout en devant être utilisés avec discernement et en complément des expertises métier.

Enfin, les enseignements tirés dépassent le cadre de la NAF et de l'Insee. La démarche peut inspirer d'autres projets de codification ou d'usages de LLM dans d'autres instituts nationaux de statistique. Les méthodes testées pour le recodage, l'inférence par lots, le développement de bases d'apprentissage, l'usage raisonné du RAG et l'apport de l'agentique peuvent être adaptées à d'autres nomenclatures et mises en partage au sein de la communauté statistique internationale.

Bibliographie

[1] T. Leroy, « Quelques limites de l'algorithme implémenté dans l'outil Sicore », *JMS*, 2022, [En ligne]. Disponible sur: https://journees-methodologie-statistique.insee.net/quelques-limites-de-lalgorithme-implemente-dans-loutil-sicore/#article

[2] T. S. Théo Leroy Lucas Malherbe, « Application de techniques de machine learning pour coder les professions dans la nomenclature des professions et catégories socio-professionnelles 2020 », *JMS*, 2022, [En ligne]. Disponible sur: https://journees-methodologie-statistique.insee.net/application-de-techniques-de-machine-learning-pour-coder-les-professions-dans-la-nomenclature-des-professions-et-categories-socio-professionnelles-2020/#article

[3] P. B. T. M. Armand Joulin Edouard Grave, « Bag of tricks for efficient text classification », *arXiv preprint arXiv:1607.01759*, juill. 2016, [En ligne]. Disponible sur: http://arxiv.org/abs/1607.01759

[4] N. R. J. P. Meilame Tayebjee Cédric Couralet, « fastText, tout juste en production et déjà obsolète? Présentation du package torchTextClassifiers pour la modernisation de la codification automatique via l'exemple de l'APE », 2025, [En ligne]. Disponible sur: https://inseefrlab.github.io/codif-ape-prez/slides/jms-torchft/index.pdf

[5] S. D. Clotilde MASSON, « Révision de la Nomenclature d'Activités Française: ÉLABORATION DE LA NAF 2025 », déc. 2023. [En ligne]. Disponible sur: https://www.cnis.fr/wp-content/uploads/2024/02/rapport-cnis-163-avec-couverture-12-02-2024.pdf

[6] P. L. Iva Spakulova, « ClassifAI – Exploring the use of Large Language Models (LLMs) to assign free text to commonly used classifications », juill. 2024, [En ligne]. Disponible sur: https://datasciencecampus.ons.gov.uk/classifai-exploring-the-use-of-large-language-models-llms-to-assign-free-text-to-commonly-used-classifications/

[7] T. F. Frédéric Comte Romain Avouac, « Mettre les technologies cloud au service de la production statistique », 2025.

[8] N. S. e. a. Ashish Vaswani, « Attention Is All You Need », *arXiv preprint arXiv:1706.03762*, juin 2017, [En ligne]. Disponible sur: http://arxiv.org/abs/1706.03762

[9] R. Sutton, « The Bitter Lesson », mars 2019, [En ligne]. Disponible sur: http://www.incompleteideas.net/IncIdeas/BitterLesson.html

[10] A. P. e. a. Patrick Lewis Ethan Perez, « Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks », *arXiv preprint arXiv:2005.11401*, mai 2020, [En ligne]. Disponible sur: http://arxiv.org/abs/2005.11401

[11] A. RANA, « RAG (Retrieval-Augmented Generation) vs CAG (Context-Augmented Generation) ». [En ligne]. Disponible sur: https://medium.com/@anuragrana.anu/rag-retrieval-augmented-generation-vs-cag-context-augmented-generation-f789ea24a168

[12] C. P. e. a. Connor Shorten, « StructuredRAG: JSON Response Formatting with Large Language Models », *arXiv preprint arXiv:2408.11061*, août 2024, [En ligne]. Disponible sur: http://arxiv.org/abs/2408.11061