

Rapport de projet

Projet CleverContracts

Extraction des obligations et pénalités contractuelles



KLEE
GROUP



CentraleSupélec

Encadrants clients :

Justine GUEGAN justine.quegan@kleegroup.com

Thomas BOSSUAT thomas.bossuat@kleegroup.com

Encadrant école :

Celine HUDELOT celine.hudelot@centralesupelec.fr

Membres du groupe :

Idriss MORTADI idriss.mortadi@student-cs.fr

Rebecca BAYSSARI rebecca.bayssari@student-cs.fr

Quentin LEMBOULAS quentin.lemboulas@student-cs.fr

14 avril 2025

| | |
|--|----|
| 1. Contexte et enjeux..... | 3 |
| 2. État de l'art et recherches | 3 |
| 3. Données utilisées | 4 |
| 4. Schéma d'ensemble et description des différents livrables | 5 |
| 5. Analyse des points clés et justifications..... | 6 |
| 5.1 L'importance de l'Anonymisation..... | 6 |
| 5.2 L'utilisation de l'IA Générative | 7 |
| 5.3 Le Choix et Déploiement d'un Modèle de Langage Local | 7 |
| 5.4 Notre Approche Produit..... | 7 |
| 6. Grille quantifiée des tâches et réalisations | 9 |
| 7. Eléments de mise en œuvre du projet..... | 12 |
| 7.1 Outils utilisés pour la collaboration et l'exécution | 12 |
| 7.2 Interactions avec le client..... | 12 |
| 7.3 Répartition des tâches | 12 |
| 8. Résultats obtenus et analyse..... | 12 |
| 8.1 Rédaction de prompts pour LLMs | 12 |
| 8.1.1 Modèle Mistral AI Large | 13 |
| 8.1.2 Modèle Small LLMs Gemma3 version 27B paramètres..... | 15 |
| 8.2 Anonymisation des contrats | 18 |
| 8.3 Vérification automatique des obligations extraites | 26 |
| 8.4 Catégorisation et tri des obligations extraites..... | 26 |
| 8.5 Visualisation et interface utilisateur | 27 |
| 8.6 Limites..... | 31 |
| 9. Conclusion..... | 32 |
| References..... | 32 |

1.Contexte et enjeux

Klee Group conçoit, développe et implémente des solutions digitales sur mesure. L'entreprise répond à de nombreux appels d'offres et rédige des contrats pour ses clients. Ces documents, pouvant atteindre plusieurs dizaines à centaines de pages, contiennent des obligations contractuelles nécessaires à identifier. Les pénalités liées à ces obligations peuvent mettre en péril la rentabilité d'une mission. Il est donc nécessaire de les détecter au mieux pour y répondre. Aujourd'hui, les clauses sont identifiées par une personne du service juridique. Cette vérification est longue et sujette à l'erreur humaine.

En plus, plusieurs contrats doivent être gérés simultanément, et il faut catégoriser les obligations importantes et les mettre en évidence pour l'équipe juridique. Pour chaque contrat et pour chaque obligation qu'ils contiennent, on souhaite savoir quand l'obligation prend effet, qui est impliqué et quelles sont les pénalités associées.

La problématique est d'alléger le travail des équipes juridiques et de leur permettre de gagner du temps et de l'efficacité dans le traitement des contrats.

Pour répondre à cette problématique, notre projet vise à **étudier la faisabilité technique et à réaliser un MVP d'identification et de catégorisation automatique des obligations et des pénalités contractuelles grâce à l'intelligence artificielle générative** basée sur les modèles de langage larges (connus aussi en anglais sous le nom de Large Language Models ou LLM).

2.État de l'art et recherches

L'exploration approfondie de la littérature scientifique met en lumière un écosystème dynamique et en constante évolution, bien que principalement centré sur le traitement des contrats rédigés en langue anglaise. Au sein de ce domaine, plusieurs travaux de recherche se distinguent par leur importance et leur impact significatif. Parmi ces contributions notables, on trouve des études axées sur la classification hiérarchique des obligations contractuelles [\[1\]](#), ainsi que des datasets de référence tels que ContractNLI [\[2\]](#) et CUAD [\[3\]](#). Ces avancées ont permis de réaliser des progrès considérables dans le domaine de l'inférence naturelle appliquée aux documents contractuels et dans l'extraction d'informations essentielles pour la révision des contrats.

Singh et al. (2024) [\[1\]](#) ont introduit une méthode novatrice de classification hiérarchique multi-label, fondée sur une décomposition des données, qui permet d'organiser de manière efficace les obligations contractuelles. Leur approche, en se concentrant sur des documents contractuels complexes, a démontré des améliorations substantielles en termes de performance par rapport à des modèles de langage préexistants tels que

BERT et RoBERTa. Cette innovation ouvre la voie à une gestion plus précise et plus structurée des obligations au sein des contrats.

Le dataset **CUAD**, introduit par Hendrycks et al. (2021) [3], offre une ressource précieuse pour évaluer les modèles dans un contexte juridique exigeant. Avec plus de 13 000 annotations réalisées par des experts, ce benchmark a permis d'identifier des pistes d'amélioration pour les modèles existants, tout en soulignant leur potentiel.

ContractNLI, développé par Koreeda et Manning (2021) [2], se distingue par une approche innovante d'inférence naturelle appliquée aux contrats. Ce dataset, composé de 607 contrats annotés, adresse des défis tels que la gestion des négations et des exceptions, tout en fournissant une base robuste pour évaluer de nouveaux modèles d'inférence contractuelle.

Malgré ces avancées, les spécificités des contrats rédigés en français restent peu explorées, révélant un manque crucial de solutions adaptées. Notre projet vise à combler cette lacune en adaptant des méthodologies au contexte français et en développant des datasets annotés dédiés. Cette démarche ambitieuse positionne notre initiative comme pionnière dans l'analyse contractuelle multilingue.

3. Données utilisées

Klee Group nous a fourni un ensemble de contrats publics, servant de base à nos expérimentations. Ces documents proviennent majoritairement de procédures de marchés publics, et sont donc légalement accessibles. Leur utilisation permet de développer, tester et valider notre pipeline sans risque juridique ni contrainte de confidentialité.

Les documents utilisés sont de différents types :

- **RC (Règlement de Consultation)** : il s'agit du document qui encadre les modalités de la procédure de passation du marché. Il définit les règles de candidature, les critères de sélection, les délais, ainsi que les documents à fournir.
- **CCAP (Cahier des Clauses Administratives Particulières)** : ce document précise les conditions administratives spécifiques au marché, comme les modalités de paiement, les pénalités applicables, les délais d'exécution ou encore les clauses de résiliation.
- **CCTP (Cahier des Clauses Techniques Particulières)** : il définit les spécifications techniques des prestations attendues. On y trouve les exigences fonctionnelles, les livrables attendus, les normes à respecter, etc.

La diversité de ces documents nous a permis de travailler sur différents styles rédactionnels, structures contractuelles et types d'obligations. Cela a été particulièrement utile pour tester l'adaptabilité de nos modèles d'extraction et de classification.

Néanmoins, étant des documents publics, ces contrats contiennent peu d'informations réellement sensibles. Cela limite en partie l'évaluation fine des performances des modèles d'anonymisation. Pour contourner cette contrainte, nous avons enrichi notre base de tests avec des exemples générés artificiellement et des données externes annotées.

Ces jeux de données ont ainsi constitué une base solide et réaliste pour le développement et l'évaluation de notre solution.

4. Schéma d'ensemble et description des différents livrables

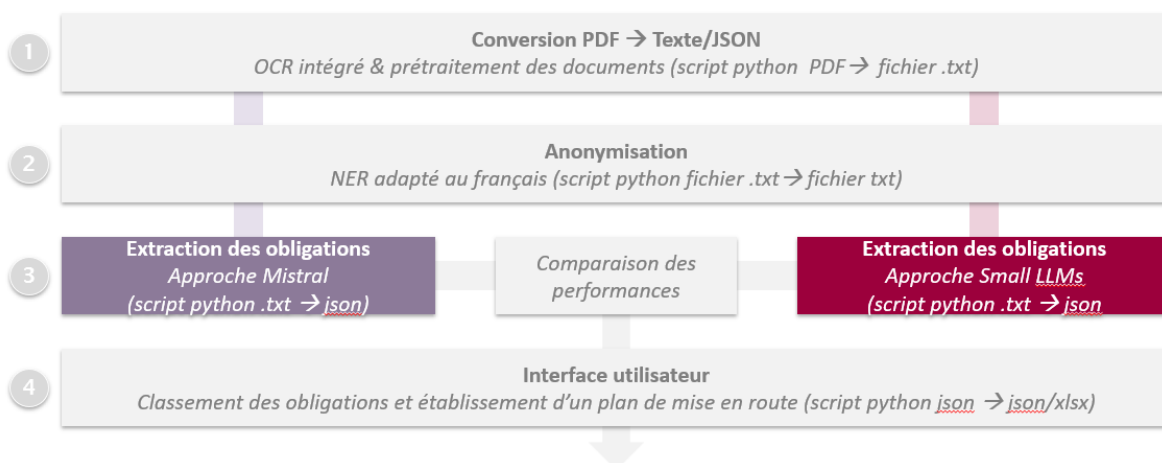


Figure 1 : Schéma d'ensemble de pipeline de traitement envisagé

Notre pipeline de traitement vise, à partir d'un contrat au format PDF, à extraire des obligations et les classer en vue d'obtenir un plan d'action, au format JSON ou xlsx. Les livrables de chaque étape sont des scripts python.

La première étape (1) est un pipeline de prétraitement pour convertir les documents PDF en texte exploitable. Cette phase intègre un système OCR pour les documents numérisés.

Ensuite, nous avons recours à une approche NLP avec une méthode de reconnaissance des entités nommées (NER) (2). L'objectif de cette étape est d'anonymiser les contrats pour pouvoir les traiter par un LLM par la suite sans divulguer des informations confidentielles.

Pour l'extraction des obligations, nous expérimentons deux approches indépendantes afin de comparer leur efficacité :

- **Nous testons l'approche LLM à l'aide d'une clé API sur le modèle Mistral (3a)**, sélectionné pour ses performances en français et sa compatibilité avec les exigences de souveraineté des données.
- **Nous testons aussi une approche basée sur un modèle de langage de petite taille (3b)** (small LLM), pour explorer une alternative légère et rapide.

Enfin afin de restituer les obligations dans un format clair et utile pour l'utilisateur, nous réalisons une classification des obligations (4) selon leur type, criticité et chronologie.

5. Analyse des points clés et justifications

5.1 L'importance de l'Anonymisation

L'utilisation d'une API pour traiter les contrats signifie que les données sont envoyées à une entité tierce. Or les réglementations en vigueur pourraient mener à une pénalisation lourde en cas de violation de la confidentialité des données des partenaires. Afin de garantir que les informations sensibles ne soient pas exposées, l'anonymisation des données est donc indispensable notamment pour les contrats privés. Pour garantir cela, notre outil vise à anonymiser les données suivantes :

- Noms de personnes (PER)
- Organisations (ORG)
- Localisations (LOC)

Dans ce projet, les deux principaux modèles que nous avons utilisés pour réaliser la reconnaissance d'entités nommées (NER) sont **Flair** et **spaCy**. Voici les raisons ayant motivé notre choix:

- **Flair**: Modèle basé sur des embeddings contextuels au niveau caractère, ce qui améliore la reconnaissance fine des entités dans des contextes variés.
- **Spacy**: Modèle transition-based, performant pour la reconnaissance rapide et précise d'entités nommées, facilement adaptable à partir de jeux de données spécifiques, ce qui le rend particulièrement adapté pour le traitement des contrats en français

5.2 L'utilisation de l'IA Générative

Les Large Language Models (LLM) sont des modèles capables de comprendre et de générer du texte de manière autonome, ce qui les rend adaptés à l'analyse et à la gestion des contrats. Le but du projet pour le client est de vérifier la possible application de ces modèles au traitement de contrats juridiques. Nous avons ensuite le choix pour le modèle à utiliser. Nous avons choisi le modèle mistral AI Large pour les raisons suivantes :

- Performances, notamment en français
- Large fenêtre contextuelle (128k tokens)
- Modèle français, encouragé pour le traitement de contrats publics

5.3 Le Choix et Déploiement d'un Modèle de Langage Local

Pour répondre aux exigences de confidentialité et de sécurité, notre projet expérimente également l'utilisation de modèles de langage hébergés localement ou sur un cloud privé. Cette approche permet d'éviter les processus d'anonymisation complexes et de conserver la souveraineté sur les données.

En gardant le contrôle total sur les infrastructures de traitement, nous pouvons garantir que les données sensibles ne quittent jamais un environnement sécurisé, répondant ainsi aux préoccupations de gouvernance des données de nos clients.

5.4 Notre Approche Produit

Notre projet se déploie en deux phases distinctes. Nous commençons par le traitement des contrats publics, ce qui nous permet de tester et d'affiner l'utilisation des LLM via des clés API. Cette première étape est essentielle pour valider notre technologie et préparer les infrastructures nécessaires.

Ensuite, nous étendons notre solution aux contrats privés, qui nécessitent des mesures de sécurité plus rigoureuses. Cette stratégie progressive nous permet de répondre aux exigences spécifiques de chaque type de contrat tout en assurant une transition fluide vers des environnements plus sécurisés et contrôlés.

Au-delà des aspects techniques, le projet intègre une approche centrée sur l'utilisateur final. Notre objectif est de fournir une solution accessible et intuitive, répondant aux besoins réels des professionnels juridiques. Pour ce faire, plusieurs besoins spécifiques ont été identifiés :

- **Besoins en Classification des Obligations :** Il est essentiel de classer les obligations contractuelles de manière précise et structurée. Cela permet de

catégoriser chaque obligation selon sa nature et son importance, facilitant ainsi la compréhension des engagements et la gestion des priorités. Une classification claire aide les professionnels à naviguer efficacement dans les contrats complexes.

- **Identification des Clauses Importantes** : Les contrats contiennent souvent des clauses cruciales qui peuvent influencer de manière significative les droits et les responsabilités des parties. Notre solution doit être capable d'identifier automatiquement ces clauses importantes, telles que les clauses de résiliation, de confidentialité, ou de propriété intellectuelle, afin de garantir qu'elles sont correctement prises en compte.
- **Gestion des Délais des Obligations** : La gestion des délais est un aspect crucial de la conformité contractuelle. Les professionnels juridiques doivent pouvoir suivre les échéances associées à chaque obligation pour s'assurer que toutes les parties respectent leurs engagements. Une solution efficace doit permettre de visualiser les délais, d'envoyer des rappels automatiques, et de générer des rapports sur l'état d'avancement des obligations.
- **Suivi des Livrables** : Les contrats définissent souvent des livrables spécifiques qui doivent être fournis. Il est donc nécessaire de disposer d'un outil permettant de suivre l'état de ces livrables, de vérifier leur conformité aux spécifications contractuelles, et de documenter leur réception. Cela aide à garantir que toutes les parties respectent leurs engagements et facilite la résolution des litiges potentiels.
- **Cadre de Référence et Conformité** : Enfin, il est important de fournir un cadre de référence clair pour l'ensemble des processus contractuels. Cela inclut la définition de normes et de procédures pour la rédaction, la révision, et la gestion des contrats. Un tel cadre permet de s'assurer que toutes les actions entreprises sont conformes aux exigences légales et réglementaires, tout en favorisant une approche cohérente et structurée de la gestion contractuelle.

6. Grille quantifiée des tâches et réalisations

| Tâche | Sous-tâche | Chiffrage (%) (total 300 %) | Degré d'avancement | Critères de validation |
|--|---|--------------------------------------|-----------------------|---|
| Extraction des PDFs par OCR | | 10% | Achevé | Manuel : Vérification des obligations |
| Chunkage des contrats | | | | |
| | Approche naïve : page par page avec superposition | 10% | Achevé | Manuel: Vérification des chunks et de leur cohérence |
| | Approche avancée : extraction de sous parties par expression régulière | 10% | En cours | Manuel: Vérification des chunks et de leur cohérence Robustesse sur divers contrats encore à établir |
| Annotation de contrats | | | | |
| | Annotation RC_2_MAINS | 5% | Achevé | |
| | Annotation CCAP_2_MAINS | 5% | Achevé | |
| Extraction des obligations : Approche Mistral | | | | |
| | Codage de l'extraction par API, choix et paramétrage du modèle Mistral | 5% | Achevé | Bon fonctionnement de l'envoi et de la réception des requêtes au LLM |
| | Rédaction du prompt : Etablissement de classification des obligations et mise en place de best-practices de prompt | 25% | Achevé | Validation par vérification des obligations extraites |
| | Vérification des obligations extraites | 35% | Achevé | Manuel : parcours de toutes les obligations extraites et comparaison avec les labels établis |
| | Vérification automatique des obligations : extraction de sous liste par cosine similarity | 10% | Achevé | Restitution de la 5-liste des obligations les plus proches par cosine similarity |

| | | | | |
|--|--|-----|----------|---|
| | Vérification automatique des obligations : Comparaison par LLM | 15% | En cours | Restitution d'une liste comparative des obligations. Manuel : comparaison avec la liste réelle - fiabilité faible |
| Implementation d'API LLM local | | | | |
| | Mise en place serveur Ollama | 10% | Achevé | Bon fonctionnement des requêtes LLM |
| Extraction des obligations : LLM local | | | | |
| | Rédaction du prompt: Définition d'une "obligation contractuelle" et insertion des exemples concrets dans le prompt | 15% | Achevé | Test sur des échantillons du contrat |
| | Résumer le contrat par LLM | 10% | Achevé | Manuel : Vérification en lisant le contrat et le résumé généré |
| | Validation de chaque obligation extraite: Demander au LLM de vérifier si les obligations extraites respectent la définition donnée en s'appuyant sur le contexte du contrat via le résumé. | 35% | Achevé | Manuel : Demander au LLM de fournir les justifications. Utiliser des logs de débogage pour examiner les réponses du LLM |
| Approche produit | | | | |
| | Identification des besoins des équipes juridiques | 2% | En cours | Retours des équipes juridiques de Klee |
| | Définition des informations pertinentes pour chaque obligation (énoncé, responsable, critérié, période) | 3% | Achevé | Retours des équipes juridiques de Klee |
| | Interface utilisateur streamlit | 3% | En cours | Validation par le client |
| Nettoyage des textes contractuels | Prétraitement des contrats (regex) | 5% | Achevé | Vérification manuelle sur plusieurs contrats que les éléments indésirables sont bien supprimés. |

| | | | | |
|--|---|-----|--------|--|
| Préparation de l'évaluation des modèles NER | Extraction manuelle des entités pertinentes pour l'anonymisation dans chaque contrat | 10% | Achevé | |
| Évaluation des modèles NER | Test de plusieurs modèles de NER pour l'extraction des entités nommées (Flair, spaCy, RoBERTa, Camembert...) | 20% | Achevé | Validation manuelle par double lecture des contrats pour s'assurer que toutes les entités sensibles (ORG, LOC, PER) ont été correctement identifiées et annotées, en cohérence avec le format attendu pour l'évaluation. |
| Fine-tuning des modèles NER | Entraînement de Flair et spaCy sur des jeux de données externes (AdminSet-NER, WikiNER) + génération de données d'entraînement depuis les contrats. | 25% | Achevé | Amélioration des scores de précision/recall sur jeux de test ; visualisation des entités extraites avec amélioration visible. |
| Méthodes de fusion et évaluation comparative | Développement de scripts de fusion entre modèles | 20% | Achevé | Résultats d'évaluation stockés dans fichiers JSON, graphes comparatifs générés (précision / rappel / F1) |
| Fonction d'anonymisation | Implémentation de la fonction d'anonymisation (remplacement par ## ou par type [ORG], [PER], etc.) | 10% | Achevé | Texte anonymisé lisible avec entités correctement remplacées ; test sur plusieurs formats (masquage et substitution). |
| Analyse des performances globales | Calcul de la moyenne de la précision, du rappel et du F1-score de tous les modèles sur les 4 contrats | 2% | Achevé | Moyennes calculées et utilisées dans la conclusion pour identifier les approches optimales |

7.Éléments de mise en œuvre du projet

7.1 Outils utilisés pour la collaboration et l'exécution

Les scripts du projet ont été réalisés sur python, pour sa facilité d'utilisation et les connaissances que chacun des membres du groupe avaient sur le langage. Le code était partagé à l'aide d'un gitlab privé de Klee Group.

Les calculs de l'utilisation du small LLMs étaient effectués sur une machine virtuelle hébergée par OVH et fournie par Klee Group. La clé Mistral API était aussi fournie par Klee Group.

7.2 Interactions avec le client

Une réunion hebdomadaire a été réalisée avec Klee Group pour vérifier l'avancement du projet et la cohérence avec leurs objectifs.

7.3 Répartition des tâches

Trois canaux de travail se sont rapidement dégagés dans l'exécution du projet :

- Anonymisation : Rebecca
- Approche Mistral : Quentin
- Approche small LLM : Idriss

En plus de ses canaux principaux, les tâches périphériques (détaillées dans la grille des tâches) ont été réparties en fonction de la charge de travail du moment de chacun.

8.Résultats obtenus et analyse

8.1 Rédaction de prompts pour LLMs

Le prompting est encore un domaine d'exploration et de recherche. Nous avons cependant cherché l'état de l'art des meilleures pratiques, afin d'obtenir l'extraction d'obligations la plus pertinente possible. Afin de mesurer la performance d'un prompt nous avons annoté deux contrats, en extrayant manuellement ce que nous considérons comme des obligations contractuelles. Les obligations annotées de ces contrats seront dénommées **labels** dans la suite du rapport. Les obligations extraites par le LLM pour chaque prompt sont comparés à la liste des labels pour en déduire une précision et un rappel comme métrique de mesure.

Remarque : Pour l'usage de notre client, la métrique la plus importante est **le rappel** car rater une obligation contractuelle peut coûter cher au client. Le rappel est calculé par le ratio suivant :

$$\text{Rappel} = \frac{\text{Nombre de labels extraits par le modèle}}{\text{Nombre de labels totaux}}$$

La métrique de précision a pour intérêt la facilité d'utilisation, car l'utilisateur doit parcourir moins d'extraits qui ne sont pas des obligations. Elle est calculée par le ratio suivant :

$$\text{Précision} = \frac{\text{Nombre de labels extraits par le modèle}}{\text{Nombre d'obligation totales extraites}}$$

8.1.1 Modèle Mistral AI Large

Mistral AI Large est un modèle avec une grande fenêtre de contexte possible. Notre approche d'itération sur le prompt n'était donc pas ici de réduire au maximum le prompt, mais plutôt d'inclure tous les éléments importants. Voici la structure finale de notre prompt :

- Séparation des tâches et du contenu entre le rôle « system » et « user »
- Dans le prompt system :
 - o Contextualisation (rôle du LLM pour ce prompt)
 - o Définition de ce qu'est et n'est pas une obligation :
 - Explication du rôle de chaque acteur présent dans le contrat
 - Sous-liste d'obligations pouvant être extraites par le modèle
 - o Demande de classification des obligations entre trois catégories (livrable, opération, cadre)
 - o Demande d'extraction des pénalités associées si possible
- Dans le prompt user :
 - o Exemple de contrat, et des réponses attendues pour ce contrat
 - o Contextualisation avec le titre de l'extrait à étudier
 - o Extrait du contrat
- Le format de sortie est un JSON, qui est meilleur pour standardiser la sortie et pouvoir réaliser les traitements suivants du pipeline

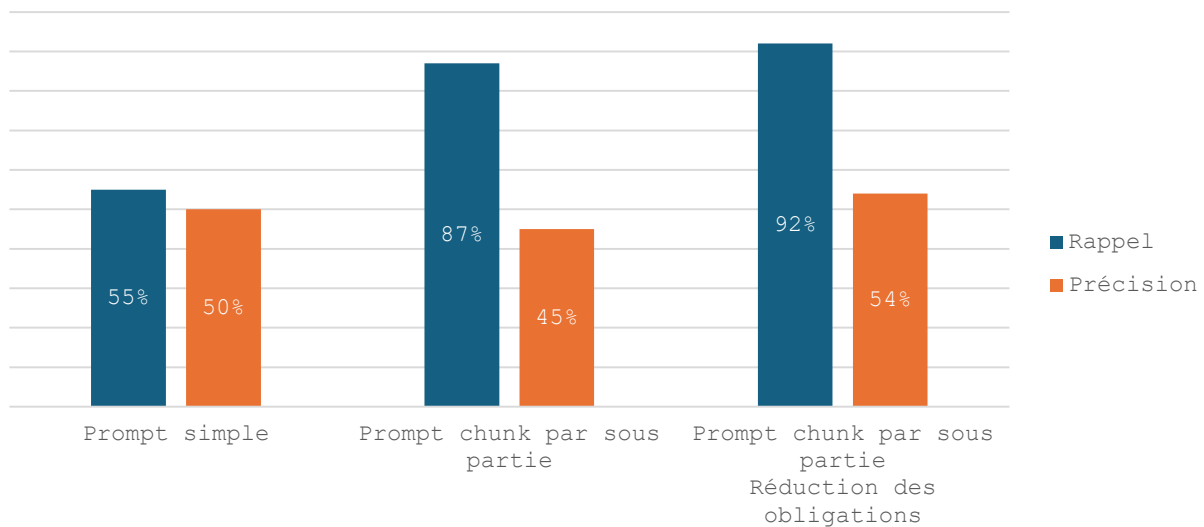
La liste des textes qui ne sont pas des obligations a été définie en itérant sur les prompts. En effet le modèle a tendance à être trop loquace, et retourner des parties du contrat qui ne sont pas des obligations. Nous avons alors parcouru les obligations extraites pour comprendre quel type de partie était retourné par le modèle sans être une obligation. Nous avons retenu cinq catégories principales qui ne sont pas des obligations :

1. Les informations générales sur le contrat
2. Les informations sur les parties prenantes
3. Les phrases courtes et déclaratives, ou titres
4. Les droits des entreprises attribuées et des acheteurs

5. Les informations sur les prix et les paiements

Les résultats obtenus en fonction des prompts sont les suivants :

Evolution du rappel et de la précision en fonction du prompt : Contrat RC (38 labels)



Evolution du rappel et de la précision en fonction du prompt : Contrat CCAP (89 labels)

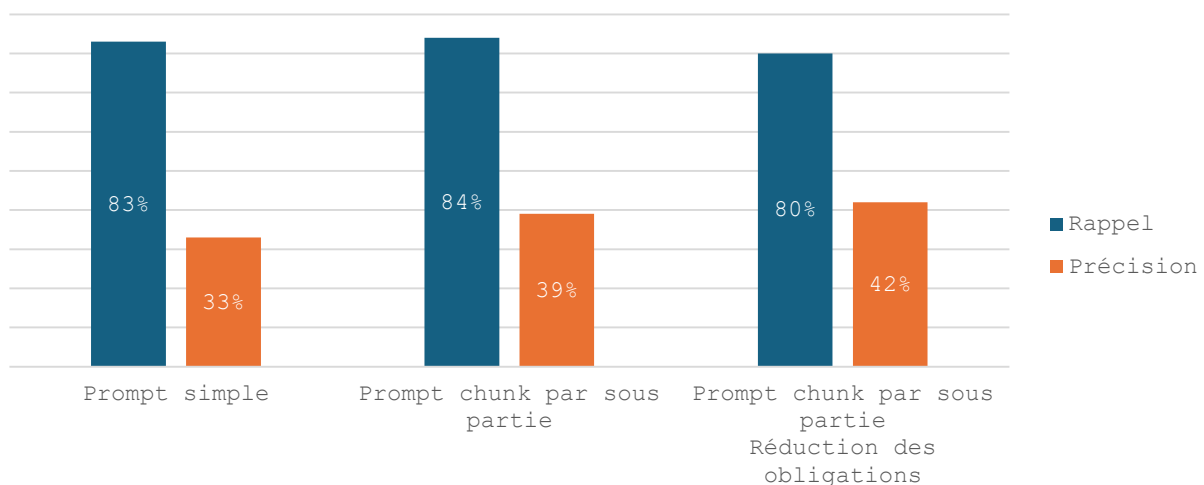


Figure 2 : Résultats du nombre d'obligations détectées en fonction du prompt d'entrée pour un contrat RC et un contrat CCAP

Les résultats obtenus pour chaque prompts nous indiquent plusieurs choses, nous rappelons que la métrique la plus importante est le **rappel**, suivi de la **précision** :

- Les résultats dépendent du contrat, et plus le nombre de labels est grand (ex CCAP) moins les différences entre les prompts sont notables
- Pour le contrat RC, le prompt qui exclut certains types d'obligations possède le meilleur rappel et devient même plus précis, ce qui montre l'importance de bien définir le cadre d'une obligation. Cela pourrait notamment faire l'objet d'une prochaine étape, en collaboration avec les équipes métiers.
- Pour le contrat CCAP nous préférons en revanche le prompt qui n'exclut pas certaines obligations, car il possède un meilleur rappel, bien que le prompt « exclusif » soit efficace (80%)

Pour conclure notre solution utilisant Mistral AI est capable d'obtenir un rappel d'environ 80 % sur les contrats les plus complexes. L'importance de bien définir les règles de ce qu'est une obligation a été démontrée, et pourrait faire l'objet d'une prochaine étape, en collaboration avec les équipes métiers.

8.1.2 Modèle Small LLMs Gemma3 version 27B paramètres

Généralement, les modèles locaux sont moins performants que les grands modèles disponibles auprès des fournisseurs (Mistral, OpenAI, etc.). De plus, l'application des LLMs à ce domaine spécifique est rare, et l'on ne voit que les grands modèles. Ainsi, dans notre conception de l'application, nous avons privilégié un cadre agnostique, permettant ainsi d'évaluer plusieurs modèles et de s'adapter à n'importe quel nouveau modèle qui sort.

Pour notre projet, nous avons utilisé [Ollama](#), un fournisseur de modèles de langage open source, qui permet de créer une API locale pour servir les modèles de langage localement. Ollama simplifie le processus de gestion des modèles de langage en local, offrant une solution flexible et sécurisée pour les développeurs et les entreprises.

Les modèles testés sont :

- **Gemma 3:27B** : modèle multimodal de 27 milliards de paramètres développé par Google, capable de traiter des entrées textuelles et visuelles, avec une fenêtre de contexte de 128 000 tokens, supportant plus de 140 langues.
- **Mistral Small 3.1** : modèle de 24 milliards de paramètres conçu pour des tâches d'IA générative nécessitant une faible latence, performant dans le suivi d'instructions et l'assistance conversationnelle.
- **Mistral NeMo** : modèle de 12 milliards de paramètres développé en collaboration avec NVIDIA, offrant une fenêtre de contexte de 128 000 tokens, avec des performances de pointe en raisonnement, connaissances générales et précision en codage.

- **Llama 3.1:8B** : modèle de 8 milliards de paramètres de Meta, optimisé pour les dialogues multilingues, performant en résumé de texte, classification et traduction, adapté aux environnements avec des ressources computationnelles limitées.

Tous les modèles ont été testés sur l'infrastructure proposée par Klee sur le fournisseur de cloud OVH. Il s'agit d'une machine équipée d'un GPU Nvidia L4 avec 24 Go de VRAM.

Le traitement des contrats par les petits LLM suivent la même démarche mentionnée pour [Modèle Mistral AI Large](#).

Après examen manuel des résultats de chaque modèle, Gemma 3 s'est révélé être le meilleur choix, probablement en raison de sa taille, étant le plus grand parmi les quatre. Gemma 3 a démontré une bonne capacité à suivre les instructions données, ainsi qu'une bonne compréhension de la définition d'une obligation contractuelle.

Objectifs:

- Extraire les obligations du titulaire EXACTEMENT comme elles sont écrites dans le texte
- Extraire les pénalités associées (si présentes) EXACTEMENT comme elles sont écrites dans le texte

Consignes:

- CRITIQUE: Copiez le texte EXACTEMENT tel qu'il apparaît dans le document, sans AUCUNE reformulation
- N'ajoutez PAS vos propres mots ou interprétations
- N'utilisez PAS de synonymes ou de paraphrases
- Extrayez uniquement les informations claires, précises et pertinentes
- Ignorez les informations redondantes, générales ou hors sujet
- Concentrez-vous sur les obligations concrètes et leurs pénalités

Format de réponse (JSON):

```
{
  "Enoncé": "Texte EXACTEMENT comme il apparaît dans le document, sans aucune modification",
  "Partie": "Nom de la section analysée",
  "Catégorie": "LIVRABLE" ou "CADRE" ou "OPERATIONS",
  "Pénalité": "Texte de la pénalité associée EXACTEMENT comme écrit" ou "" si aucune
}
```

Définitions:

- CADRE: Les règles qui entourent le contrat et les relations entre le titulaire et l'acheteur (ex: durée, résiliation, sécurité, confidentialité, propriété intellectuelle)
- OPERATIONS: Qu'est ce qui doit être fait concrètement (ex: livraison, installation, maintenance)
- LIVRABLE: Les résultats attendus (ex: rapports, documents, produits)

IMPORTANT: Votre tâche principale est d'extraire le texte EXACTEMENT comme il est écrit. Toute reformulation ou paraphrase est une erreur critique.

De plus, les petits modèles de langage (small LLM) rencontrent également le même problème mentionné précédemment : ils renvoient de nombreuses clauses non pertinentes et très générales, telles que « le titulaire doit respecter le droit français ». Pour pallier ce problème, nous avons mis en place une deuxième étape de validation.

Nous demandons au LLM de revérifier la pertinence de l'obligation. Voici le prompt utilisé à cet effet :

Vous êtes un expert juridique chargé de valider les obligations extraites d'un contrat. Votre tâche est de vérifier si chaque obligation extraite correspond au texte du document, sans aucune reformulation (Des différences mineures comme la casse ou la ponctuation ne sont pas des erreurs)

Consignes:

- Pour chaque obligation fournie, déterminez si le texte correspond à ce qui est présent dans le contrat.
- Indiquez "is_valid": true si l'obligation est valide, ou "is_valid": false sinon.
- En cas de non-validité, fournissez une brève explication dans "reason".
- Respectez le format JSON comme indiqué ci-dessous sans ajouter de texte supplémentaire.

Définition d'une obligation valide:

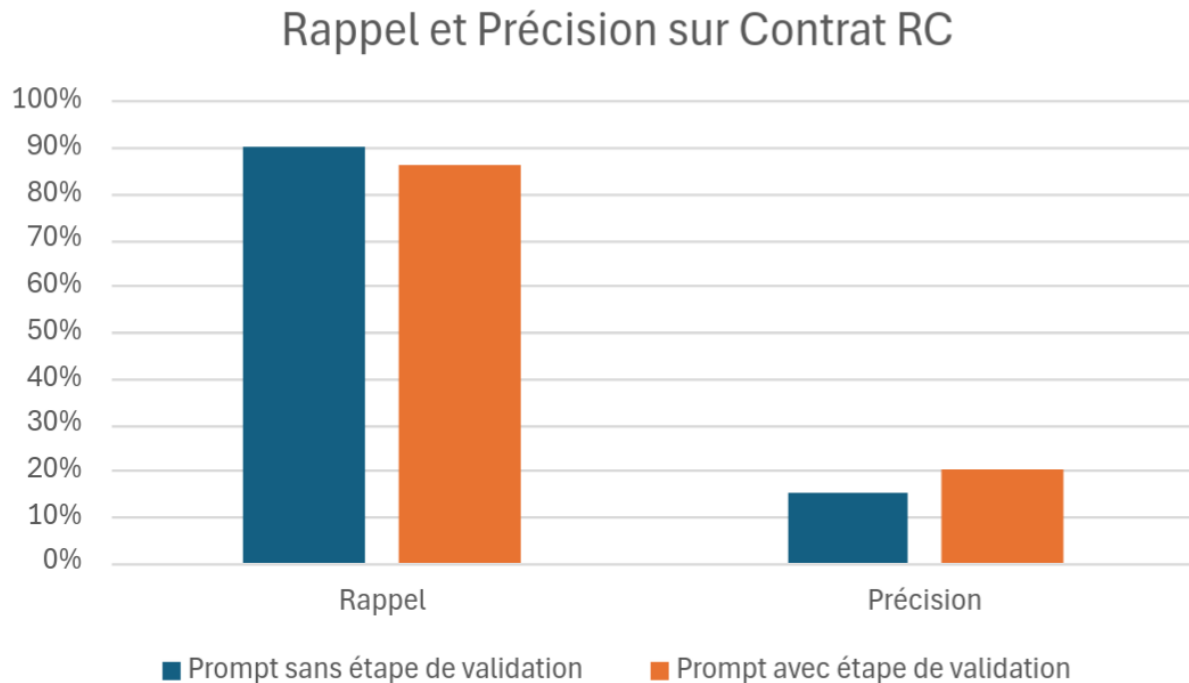
- L'obligation doit être pertinente pour le titulaire du marché (ce qu'il doit faire ou ne pas faire).
- Pas de généralités ni de formulations vagues qui ne constituent pas des obligations précises.
- Ne pas extraire des phrases qui décrivent uniquement des informations générales, des contextes, ou des actions concernant l'acheteur uniquement.
- Les obligations doivent concerner spécifiquement des actions, des livrables, ou des cadres que le titulaire doit respecter.

Format de réponse:

```
{
  "validated_obligations": [
    {
      "obligation_id": 0,
      "is_valid": true,
      "reason": ""
    },
    {
      "obligation_id": 1,
      "is_valid": false,
      "reason": "Raison de la non-validité"
    },
    ...
  ]
}
```

Cela a permis de réduire le nombre d'obligations de 265 (dont 184 faux positifs) à 186 (dont 116 faux positifs).

Les résultats finaux sur un contrat :



À noter que le contrat utilisé pour tester le modèle n'est pas le même que le précédent, en raison de contraintes de temps. Le contrat RC, plus court, a été utilisé à la place car l'étape de validation ajoute beaucoup de temps.

8.2 Anonymisation des contrats

8.2.1 Prétraitement des contrats

Notre approche de prétraitement repose sur l'utilisation de regex (`re.sub`) afin de nettoyer efficacement les textes contractuels. Elle comprend la suppression des tables des matières avec points et numéros, des lignes contenant uniquement des points, des erreurs de format comme "Articl e", ainsi que des symboles non pertinents tels que • ou √. Nous éliminons également les caractères problématiques, normalisons les espaces et les lignes vides, et supprimons tous les caractères non imprimables en dehors d'une liste de caractères autorisés.

8.2.2 Modèles NER explorés pour l'anonymisation

Plusieurs modèles de reconnaissance d'entités nommées (NER) ont été testés pour identifier automatiquement les entités sensibles dans les contrats. Pour assurer une uniformité dans le traitement du texte, tous les modèles ont suivi le même processus : le texte est initialement segmenté en phrases, avant d'être assemblé en

blocs de deux phrases (chunks), ce qui permet de rester dans les limites de tokens tout en préservant le contexte immédiat. Suite à l'extraction des entités, un mécanisme de regroupement d'entités permet de reconstituer les entités complètes en combinant les tokens associés. Cette fonctionnalité ne garde que certains types d'entités autorisées (tels que ORG, PER et LOC), et regroupe les entités de même type ou débutant par un indicateur de continuation (I-). Elle nettoie également les mots en supprimant les artefacts de tokenisation (comme les caractères `_`) et fusionne les mots successifs appartenant à la même entité pour restituer une entité complète, avec ses positions de début et de fin dans le texte d'origine. Les modèles testés alors sont :

- **Jean-Baptiste/camembert-ner** : basé sur l'architecture CamemBERT, ce modèle est pré-entraîné sur le corpus OSCAR et fine-tuné sur WikiNER-fr, qui combine tokenisation, encodage contextuel et classification pour identifier les entités dans un texte.
- **Flair** : utilise des embeddings contextuels au niveau caractère, combinés à un réseau BiLSTM et un CRF, ce qui lui permet de capturer finement le contexte et d'effectuer des prédictions structurées, grâce à sa classe `SequenceTagger` propre au framework Flair
- **XLNet-Roberta (fine-tuné sur WikiNER-fr)** : un modèle multilingue de grande taille (ayant 24 couches et à peu près 560 millions de paramètres), utilisant la tokenisation en sous-mots, adapté à la reconnaissance d'entités dans différents contextes linguistiques, dont le français.
- **spaCy (modèle français)** : un modèle léger et rapide, reposant sur une architecture transitionnelle efficace pour la détection d'entités dans des documents longs, comme les contrats juridiques. Son pipeline est composé des étapes suivantes : tokenisation, extraction des vecteurs de contexte (*tok2vec*), reconnaissance des entités (*ner*) puis génération du document annoté (*doc*).

8.2.3 Adaptation des modèles par fine-tuning

La majorité des expérimentations de fine-tuning ont été réalisées sur Flair et spaCy, les deux modèles principaux retenus pour l'anonymisation des contrats. Bien que les jeux de données utilisés ne soient pas directement issus du domaine juridique, différentes approches ont été mises en œuvre afin d'améliorer les performances des modèles sur des textes rédigés en français :

Flair a été fine-tuné selon trois approches différentes :

1. A partir de deux paragraphes extraits d'un des quatre contrats testés, puis appliqué à l'ensemble du corpus

2. En réutilisant ces mêmes paragraphes, enrichis avec des exemples générés par ChatGPT pour augmenter le jeu d'entraînement ;
3. A l'aide du dataset AdminSet-NER [4], composé de plus de 50 millions de phrases ou fragments extraits de documents administratifs français. Ce jeu comprend 583 documents pour l'entraînement, 146 pour la validation et 85 pour le test, et contient des annotations sur trois types d'entités : organisations (2 246 mentions), personnes (1 856) et localisations (883).

Dans ces trois expérimentations, les données d'entraînement ont d'abord été converties au format BIO. Ensuite, le modèle pré entraîné flair/ner-french a été chargé, puis ré entraîné à l'aide de la classe ModelTrainer, en utilisant ce corpus annoté pour adapter le modèle aux nouvelles données.

RoBERTa, quant à lui, a également été fine-tuné en utilisant le même extrait de contrat que pour Flair, préalablement converti au format BIO. Le texte a été tokenisé, les étiquettes alignées avec les sous-mots, puis le modèle pré entraîné a été à nouveau entraîné à l'aide d'un Trainer, après une division du jeu en données d'entraînement (80 %) et de validation (20 %). Le modèle ajusté a ensuite été appliqué à l'ensemble des contrats pour prédire les entités.

spaCy a été entraîné selon deux approches complémentaires :

1. À partir du dataset **AdminSet-NER**, en construisant manuellement les objets `Doc` contenant les entités annotées, puis en convertissant l'ensemble au format `.spacy` pour un entraînement **from scratch** du modèle
2. En combinant les datasets AdminSet-NER et WikiNER-fr [5] (multilingue annoté automatiquement à partir de Wikipédia), regroupés sous un même format (DocBin) avec un mélange aléatoire des documents (90 % entraînement, 10 % validation), pour effectuer un fine-tuning de la base pré entraînée `fr_core_news_sm`, avec un pipeline configuré pour utiliser des vecteurs pré entraînés.

8.2.4 Fusion Multi-modèles pour la Reconnaissance d'entités

Pour améliorer la reconnaissance d'entités nommées (NER), nous avons exploré deux méthodes d'Ensemble pour combiner les résultats de plusieurs modèles NER :

1. Exact Match Ensemble

Cette méthode repose sur la correspondance exacte du texte de l'entité. Deux variantes sont utilisées :

- Merge Entities : Pour regrouper des entités distinctes par leur texte, on construit un dictionnaire (entity_dict). Si une entité avec le même texte existe déjà dans le dictionnaire, on compare leur type : si les types sont identiques, on met à jour l'entrée avec la nouvelle entité ; sinon, on conserve l'ancienne.
- Get Common Entities : on extrait les entités communes aux deux modèles en comparant leurs textes (en minuscules) et leurs types.

2. Fuzzy Match Ensemble

Cette méthode est plus flexible : elle cherche les entités présentes dans les deux modèles avec le même type et une similarité de texte $\geq 90\%$ (fuzzy string matching). Une variante ignore même le type, ne comparant que la similarité textuelle.

La majorité des combinaisons testées concernent des paires de modèles Flair et SpaCy finetunés sur les différents jeux de données, ou Flair et Roberta finetunés sur des textes contractuels.

8.2.5 Méthodes d'évaluation

Nous avons d'abord extrait manuellement les entités à anonymiser à partir de quatre contrats distincts : CCAP_2MAINS_BPM031594_vF.txt, CCTP_2MAINS_BPM031594_vF.txt, REGLEMENT DE LA CONSULTATION.txt et CCAP MARCHÉ DATAVAL.txt, dans le but de créer un ensemble de données de référence pour l'évaluation des modèles. Ces entités ont ensuite été sauvegardées dans un fichier JSON pour être utilisées dans les différentes étapes de comparaison.

La première méthode d'évaluation consiste à effectuer une comparaison stricte et positionnelle des entités anticipées par chaque modèle avec les annotations de référence, en suivant le format BIO. Cette approche fait appel à la bibliothèque seqeval pour déterminer les métriques standards (précision, rappel, F1-score), en partant du principe que seules les entités correctement positionnées sont considérées comme valides.

Toutefois, cette approche comporte quelques contraintes, notamment lorsque la position subit un léger décalage ou que des variations minimales apparaissent dans la formulation textuelle de l'entité. Cela risque de minimiser la qualité réelle des prédictions.

C'est pourquoi une seconde approche a été utilisée. Elle s'appuie sur une comparaison directe des entités extraites, sans tenir compte de leur position dans le texte. Une étape de déduplication est réalisée avant toute comparaison, permettant de garder uniquement un exemplaire par entité (couple texte/type), ce qui prévient les répétitions dues aux redondances dans le texte. L'évaluation se fait ensuite en deux étapes :

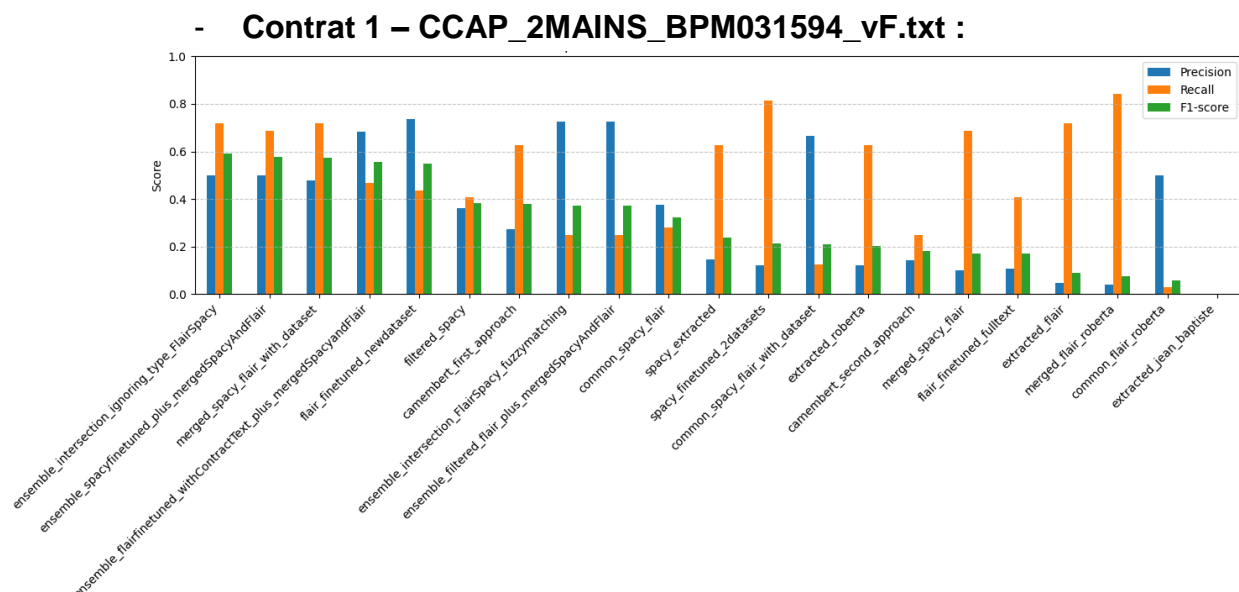
d'abord par correspondance exacte, puis par fuzzy matching (avec un seuil de similarité $\geq 85\%$) sur le texte, à condition que le type d'entité soit identique. Cette technique permet de mieux détecter les cas où les modèles prédisent des entités valables mais formulées légèrement différemment ou mal alignées.

À partir de ces deux méthodes, nous avons évalué l'ensemble des modèles sur chacun des quatre contrats. Nous avons alors calculé trois métriques classiques :

- **Précision** : proportion des entités prédites qui sont effectivement correctes ;
- **Rappel** : proportion des entités attendues (vérité terrain) qui ont bien été retrouvées ;
- **F1-score** : moyenne harmonique entre précision et rappel, permettant de mesurer l'équilibre entre les deux.

8.2.6 Analyse des résultats

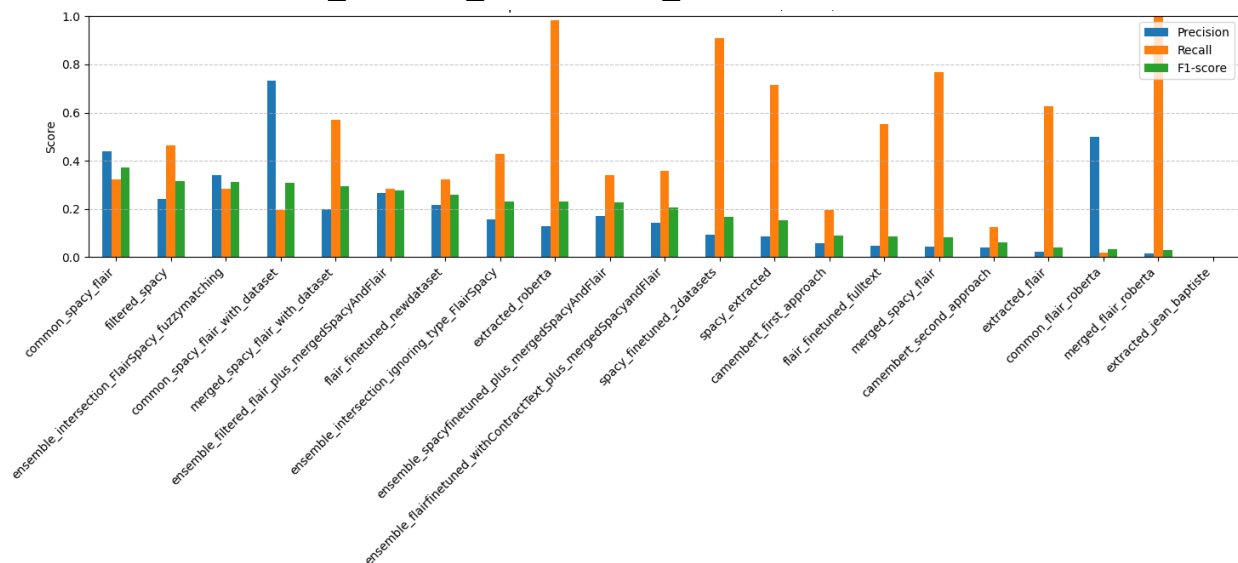
On observe globalement que les modèles obtenant une précision élevée ne sont pas toujours ceux présentant un rappel élevé, et inversement. Cela reflète une tendance fréquente dans la reconnaissance d'entités : certains modèles préfèrent extraire moins mais avec plus de justesse, tandis que d'autres capturent davantage d'entités au risque d'introduire plus d'erreurs. Voici, pour chaque contrat, les approches qui se distinguent :



La fusion entre Flair et RoBERTa a permis d'atteindre un rappel très élevé (84%), mais avec une faible précision. À l'inverse, le modèle Flair fine-tuné sur AdminSet-NER offer

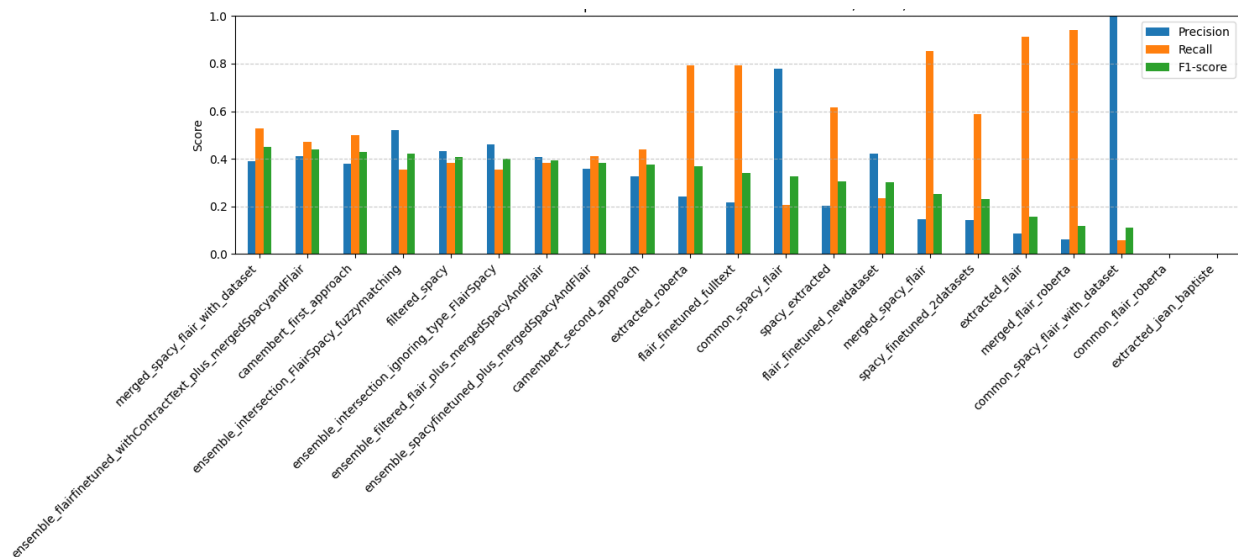
une bien meilleure précision (74%). Finalement, la meilleure performance globale qui offre un équilibre entre précision et recall a été obtenue par une approche d'ensemble combinant Flair (fine-tuné sur le dataset Adminset) avec spaCy (entraîné depuis zéro), croisée avec Flair seul, et utilisant le fuzzy matching sans contrainte de type. (59%)

- **CCTP_2MAINS_BPM031594_vF :**

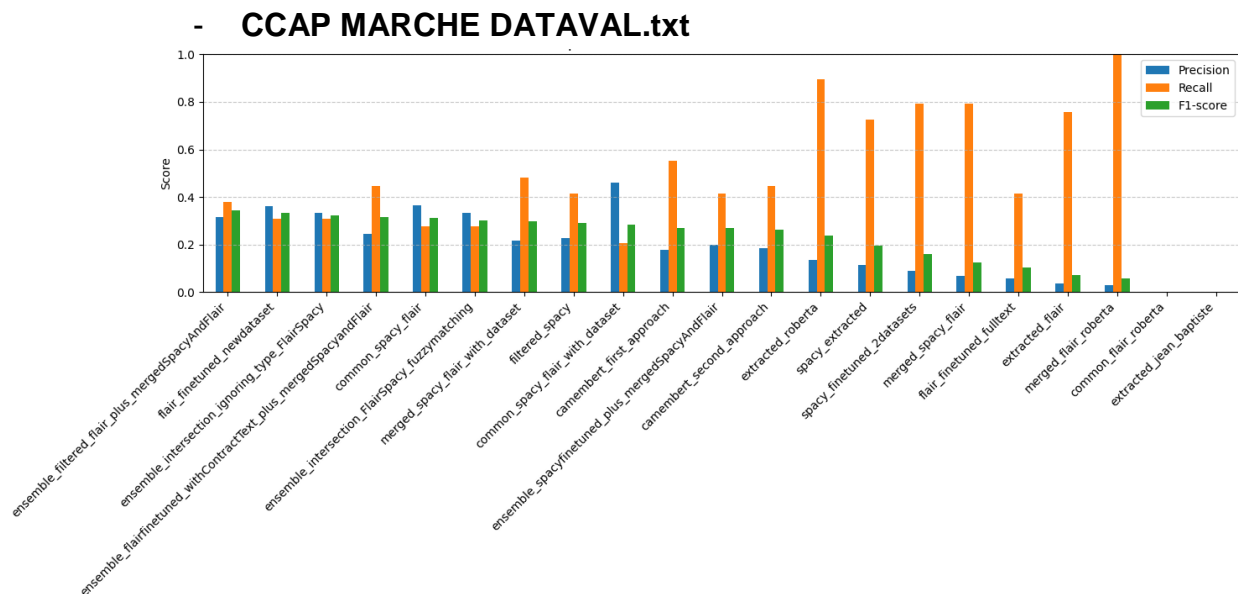


La fusion entre Flair et RoBERTa a permis d'obtenir un excellent rappel de 100%, capturant presque toutes les entités annotées, mais au prix d'un très grand nombre de faux positifs, ce qui a fortement réduit la précision. En revanche, l'approche qui combine les entités communes entre spaCy entraîné de scratch et Flair finetuné avec Adminset, a généré des résultats d'une grande précision (74%), bien que restreints en termes de couverture. Finalement, la meilleure performance globale a été atteinte par une approche d'ensemble combinant Flair et spaCy (38%)

- **REGLEMENT DE LA CONSULTATION.txt :**



Dans ce contrat, la fusion entre Flair et RoBERTa a de nouveau permis d'atteindre un très bon rappel (94 %). Cette fois, l'approche croisant Flair avec spaCy, enrichie d'un filtrage post-traitement, a obtenu la meilleure précision (78 %). Le modèle combinant les résultats de spaCy fine-tuné et ceux de Flair s'est révélé particulièrement performant pour équilibrer cette dynamique, affichant le meilleur F1-score global (45 %).

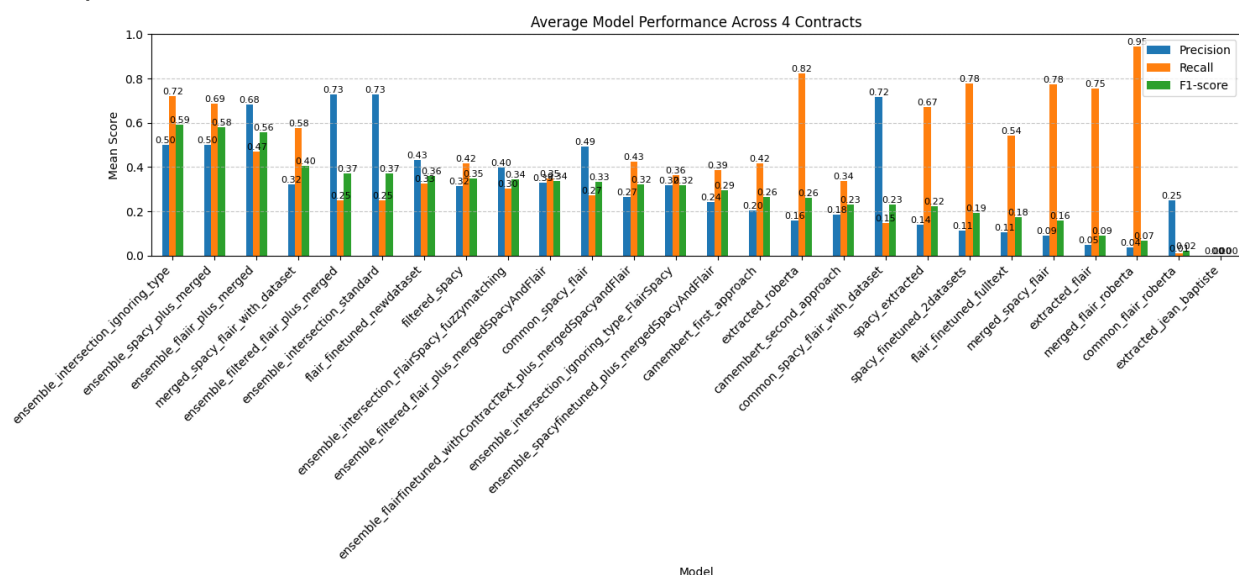


Dans ce dernier contrat le comportement observé demeure constant : l'approche fusionnée Flair-RoBERTa a assuré une couverture des entités complète, cependant avec une précision assez faible. Par ailleurs, l'intersection des entités communes entre spaCy et Flair fine-tunés (affinée avec un post-traitement) a généré les résultats les

plus précis (48%). L'intersection de Flair filtrée et de l'ensemble qui combine Flair et spaCy a le meilleur F1-score. (35%)

8.2.7 Conclusion sur l'efficacité des modèles

En moyenne sur les quatre contrats, les meilleures performances ont été obtenues par des approches d'ensemble, notamment celles utilisant le fuzzy matching sans contrainte de type appliqué à la combinaison de Flair (fine-tuné sur le dataset Adminset) et de spaCy (entraîné depuis zéro), croisée avec Flair seul. Les fusions directes entre spaCy et Flair, parfois enrichies par des modèles fine-tunés, se sont également révélées efficaces. Ces stratégies ont su maintenir un bon équilibre entre précision et rappel, se traduisant par les meilleurs scores F1 globaux, bien supérieurs à ceux des modèles individuels, souvent marqués par un fort déséquilibre entre ces deux métriques.



8.2.8 Fonction d'anonymisation

Après avoir identifié le modèle le plus fiable selon nos critères d'évaluation, nous avons appliqué une fonction d'anonymisation sur les textes. Celle-ci remplace les entités nommées détectées (personnes, organisations, lieux, etc.) à l'aide de deux stratégies possibles :

- Masquage : chaque entité est remplacée par une chaîne de caractères # de même longueur
- Substitution par type : chaque entité est remplacée par son type générique entre crochets, par exemple [ORG], [LOC].

Pour préserver les positions des caractères dans le texte, les entités sont traitées de la fin vers le début. Le mode d'anonymisation utilisé peut être sélectionné dynamiquement lors de l'exécution, en fonction du modèle retenu.

Exemple :

```
- personnes prises en charge dans les structures d'insertion par  
l'activité économique ([ORG]) mentionnée à  
l'article L. 5132 -4 du code du travail, cest -à-dire :  
mises à disposition par une association intermédiaire (AI) ou une  
entreprise de travail temporaire
```

8.3 Vérification automatique des obligations extraites

La vérification et comparaison des obligations avec les labels est chronophage. Aussi pour itérer et améliorer les prompts plus rapidement, nous avons souhaité mettre en place une vérification automatique des obligations extraites. L'objectif de cette fonctionnalité est de pouvoir rapidement obtenir les métriques d'évaluation d'un prompt (rappel et précision).

Notre approche se base sur deux méthodes :

- Le fuzzy matching, qui consiste à comparer les caractères de deux listes.
- La cosine similarity, qui embedde les labels et obligations du LLMs, avant de les comparer. Cette approche permet de comparer les obligations sémantiquement. Cela peut-être utile notamment si le LLM reformule les obligations, ce qui peut-être le cas pour un LLM plus puissant comme Mistral.
- Pour les deux méthodes, si le score de comparaison entre l'obligation et un label obtient une note plus élevée qu'une certaine frontière, alors l'obligation est considérée comme un label. Nous calculerons enfin les métriques de performance du prompt.

8.4 Catégorisation et tri des obligations extraites

Lors de l'extraction des obligations, nous procédons également à une analyse visant à en tirer de la valeur. Pour chaque obligation, nous effectuons une seconde analyse à l'aide d'un modèle de langage (LLM) afin de déterminer sa catégorie. Ces catégories sont définies en fonction des besoins métiers. Pour notre premier prototype, nous avons choisi de classer les obligations en trois catégories : Opération, Livrable, et Cadre. Cette classification est particulièrement utile pour les personnes juridiques, car elle permet de structurer et de clarifier les responsabilités et attentes contractuelles.

- **CADRE** : Les règles entourant le contrat et les relations entre le titulaire et l'acheteur (par exemple : durée, résiliation, sécurité, confidentialité, propriété intellectuelle).
- **OPÉRATIONS** : Les actions concrètes à réaliser (par exemple : livraison, installation, maintenance).
- **LIVRABLE** : Les résultats attendus (par exemple : rapports, documents, produits).

8.5 Visualisation et interface utilisateur

Une première interface utilisateur a été créée à l'aide de la bibliothèque Python Streamlit. Cette interface permet de charger un document au format TXT ou PDF, qui sera automatiquement analysé. Ensuite, le contenu du document est transmis à l'extracteur d'obligations, qui renvoie le texte de l'obligation, la section du contrat où elle a été trouvée, sa catégorie et la pénalité associée.

L'application se compose de trois sections principales, accessibles via un menu de navigation dans la barre latérale :

1. **Extraction des Obligations** : Cette page permet à l'utilisateur de télécharger un contrat, de le convertir du format PDF au format texte si nécessaire, puis d'extraire automatiquement les obligations contractuelles. Les obligations sont affichées en temps réel sous forme de cartes dans une grille à trois colonnes, facilitant la lecture et la compréhension rapide des engagements contractuels.

Sélectionnez une page

Extraction des Obligations

Extraction des Obligations

Analyse de Contrat

Métriques de l'extraction

☐ Utiliser l'API Mistral
☐ Valider les Obligations

CleverContracts - Application d'Analyse de Contrat

Choisissez un fichier

Drag and drop file here

Limit 200MB per file • PDF, TXT

Browse files

Choisissez un fichier

Drag and drop file here

Limit 200MB per file • PDF, TXT

Browse files

RC 2MAINS BPM031594 vF2.txt

66.4KB

×

✓ Fichier 'RC 2MAINS BPM031594 vF2.txt' téléchargé avec succès !

Fichier texte téléchargé avec succès !

Affichage du texte converti :

Texte Converti

4.1 Procédure de passation 4

4.2 Allotissement 4

4.3 Forme et étendue de l'accord-cadre 6

4.4 Tranches 6

4.5 Durée du marché 6

4.6 Lieu d'exécution 6

4.7 Variantes 7

4.8 Prestations sup plémentaires éventuelles 7

4.9 Considérations sociales 7

4.10 Considérations environnementales 8

4.11 Traitement de données à caractère personnel 8

4.12 Secret des affaires 9

Article 5 - INFORMATION DES CANDIDATS 9

5.1 Contenu des documents de la consultation 9

Démarrage de l'extraction des obligations...

2. **Analyse de Contrat** : Cette section propose deux modes d'utilisation :
 - a. Traitement d'un nouveau contrat en utilisant les données extraites lors de l'étape précédente
 - b. Chargement d'une analyse existante stockée au format JSON

Sélectionnez une page

Analyse de Contrat

Modèle LLM pour l'Analyse

mistral-small

Utiliser l'API Mistral pour l'Analyse

Source de l'Analyse

Choisissez la source de l'analyse

☐ Traiter un Nouveau Contrat
 ☒ Charger une Analyse Existante

Sélectionnez une Analyse Existante

Choisissez un fichier

contract_analysis_20250402_15245...

Filtres

Criticité (1-10)

Période

avant la soumiss...

CleverContracts - Application d'Analyse de Contrat

Analyse de Contrat

Rechercher une Clause

Entrez un texte pour rechercher dans les clauses :

Analyse Détaillée Filtrée

Clause : Compléter et renvoyer le questionnaire relatif aux actions conduites en matière de diversité et d'égalité professionnelle entre les femmes et les hommes.

Criticité : 3

Responsable : L'un des suivants : soumissionnaire du contrat, juridique, opérationnel, autre

Période : pendant la durée d'effet

Clause : Renvoyer le questionnaire dans le mois suivant sa réception.

Criticité : 3

Responsable : L'un des suivants : soumissionnaire du contrat, juridique, opérationnel, autre

Période : pendant la durée d'effet

Clause : Le titulaire doit réaliser des prestations de développement et de maintenance d'applications des directions et délégations de l'administration centrale des Ministères Sociaux.

Criticité : 8

Responsable : Le titulaire

Période : pendant la durée d'effet

Source de l'Analyse

Choisissez la source de l'analyse

☐ Traiter un Nouveau Contrat
 ☒ Charger une Analyse Existante

Sélectionnez une Analyse Existante

Choisissez un fichier

contract_analysis_20250402_15245...

Filtres

Criticité (1-10)

Période

Choose an option

avant la soumission

pendant la durée d'effet

après la durée d'effet

Analyse de Contrat

Rechercher une Clause

Entrez un texte pour rechercher dans les clauses :

Lot 2

Analyse Détaillée Filtrée

Clause : L'ensemble des livrables cités dans le CCTP aux articles 5 (lot 1), 6 (lot 2), 7 (lot 3) et 8 (lot 4).

Criticité : 9

Responsable : Soumissionnaire du contrat, opérationnel

Période : pendant la durée d'effet

Source de l'Analyse

Choisissez la source de l'analyse

☐ Traiter un Nouveau Contrat
 ☒ Charger une Analyse Existante

Sélectionnez une Analyse Existante

Choisissez un fichier

contract_analysis_20250402_15245...

Filtres

Criticité (1-10)

Période

pendant la durée...

Responsable

opé

opérationnel

Analyse de Contrat

Rechercher une Clause

Entrez un texte pour rechercher dans les clauses :

Lot 2

Analyse Détaillée Filtrée

Clause : L'ensemble des livrables cités dans le CCTP aux articles 5 (lot 1), 6 (lot 2), 7 (lot 3) et 8 (lot 4).

Criticité : 9

Responsable : Soumissionnaire du contrat, opérationnel

Période : pendant la durée d'effet

L'analyse inclut un résumé du contrat et des informations détaillées sur chaque clause, notamment leur **criticité**, les **parties responsables** et la **période d'application**. Des filtres permettent de trier les clauses selon ces critères, et une barre de recherche facilite la localisation rapide d'obligations spécifiques.

3. **Métriques de l'extraction** : Cette page offre une représentation graphique et structurée des données contractuelles, facilitant l'interprétation des résultats. Elle présente :
 - a. Les métriques clés du contrat (nom, modèle utilisé, description)
 - b. Un résumé synthétique du contrat
 - c. La liste détaillée des obligations avec possibilité de les développer
 - d. Les pénalités associées aux obligations

29

- e. Des statistiques sur la précision de l'analyse (obligations détectées, correspondances)

Sélectionnez une page

Métriques de l'extraction

Sélectionnez un Fichier JSON d'Analyse

Choisissez un fichier

gemma3:27b_analysis_20250320_1...

Visualiseur d'Analyse de Contrat

Nom du Contrat

CCAP_2MAINS_BPM031594_vF.txt

Modèle Utilisé

gemma3:27b

Description de l'Analyse

Testing gemma3:27b on contract / workspace/clevercontracts/ Contrats/2MAINS/ CCAP_2MAINS_BPM031594_vF.txt

Métriques d'Évaluation

Obligations Détectées au Total

264

Clauses de Référence Totales

89

Correspondances Totales

40

Clauses de Référence Détectées

60

Pourcentage de Correspondances

13.64

Pourcentage Détecté

67.42

Obligations

Obligation 1

Énoncé : Compléter et renvoyer le questionnaire relatif aux actions conduites en matière de diversité et d'égalité professionnelle entre les femmes et les hommes.

Partie : Article 1 - PREAMBULE - CONTEXTE

Catégorie : LIVRABLE

Pénalité : Non spécifiée dans ce texte.

Obligation 2

Énoncé : Renvoyer le questionnaire dans le mois suivant sa réception.

Partie : Article 1 - PREAMBULE - CONTEXTE

Catégorie : LIVRABLE

Pénalité : Non spécifiée dans ce texte.

Obligation 3

Énoncé : Le titulaire doit réaliser des prestations de développement et de maintenance d'applications des directions et délégations de l'administration centrale des Ministères Sociaux.

Partie : Article 2 - OBJET DE L'ACCORD CADRE

30

8.6 Limites

8.6.1 Limites de la tâche d'anonymisation

- **Accès limité à des contrats publics** : les documents utilisés étaient des contrats publics qui, par nature, ne contenaient que très peu d'entités réellement sensibles à masquer, ce qui limite la portée pratique de l'évaluation.
- **Manque de modèles NER adaptés au domaine juridique en français** : la majorité des modèles NER disponibles sur Hugging Face sont optimisés pour l'anglais. Même s'il en existe quelques-uns pour le français, aucun n'est spécifiquement entraîné ou adapté au langage juridique.
- **Disponibilité très limitée de jeux de données annotés en français** : il existe très peu de datasets NER labellisés en langue française, ce qui rend difficile le fine-tuning de modèles adaptés à notre contexte.

8.6.2 Limites des LLMs

Les modèles de grande taille, malgré leur capacité à traiter un volume conséquent de texte, présentent néanmoins une difficulté importante pour limiter l'extraction de clauses génériques. En effet, même lorsqu'ils bénéficient d'une fenêtre contextuelle étendue, ces LLMs tendent à inclure dans leurs résultats des extraits trop généraux ou des passages qui ne correspondent pas précisément aux obligations contractuelles recherchées. Cette sensibilité aux formulations du prompt peut mener à des résultats imprécis, obligeant ainsi à élaborer des instructions toujours plus spécifiques pour écarter les clauses génériques non pertinentes.

Pour les modèles de petite taille, les limitations se font ressentir différemment. D'une part, ces modèles souffrent d'une capacité réduite à comprendre et contextualiser correctement l'ensemble du document, ce qui se traduit par une extraction plus fragmentée et parfois erronée. De plus, leur temps de calcul est fortement dépendant du hardware utilisé, engendrant des temps de traitement plus longs. Cette contrainte matérielle limite leur utilisation en production, surtout dans des environnements nécessitant une réponse rapide ou une analyse en temps réel.

Par ailleurs, la gestion de l'extraction basée uniquement sur l'élaboration du prompt s'est avérée particulièrement complexe pour l'ensemble des modèles, qu'ils soient petits ou grands. Cette méthode, bien que pratique pour tester diverses approches, montre ses limites face à la diversité et la complexité des formulations contractuelles. Une approche par fine tuning, qui permettrait d'adapter finement le modèle aux spécificités du domaine juridique, aurait été préférable afin de réduire les erreurs et d'améliorer la précision de l'extraction. Malheureusement, l'absence d'un jeu de données suffisamment riche et annoté pour ce domaine a empêché de déployer pleinement cette approche, laissant le prompt comme unique levier d'optimisation, avec ses inconvénients associés.

9. Conclusion

Ce projet s'est donné pour objectif de répondre à un besoin concret et critique du secteur juridique : l'automatisation de l'identification et de la gestion des obligations contractuelles. Pour répondre à ce besoin nous avons conçu un pipeline complet de traitement d'un contrat depuis son format brut jusqu'à l'extraction, la classification et la visualisation des obligations structurées, tout en garantissant la confidentialité des données sensibles.

L'utilisation de modèles de langage de grande taille comme Mistral AI Large était un des objectifs du projet. Nous avons ainsi réalisé un travail sur les prompts, la structuration des sorties, et une validation systématique des résultats. En parallèle, l'intégration de modèles locaux comme Gemma 3 via Ollama a permis d'explorer des alternatives souveraines, moins dépendantes de fournisseurs externes. Ce double axe, entre performance et souveraineté, nous a permis d'évaluer objectivement les compromis possibles selon les contraintes d'usage.

Un autre axe d'étude du projet a été l'anonymisation. Face aux enjeux de confidentialité, notamment dans le cadre du traitement de contrats privés, nous avons mis en œuvre une solution d'anonymisation fondée sur des modèles de NER optimisés. L'exploration de multiples modèles (Flair, spaCy, RoBERTa, CamemBERT), enrichie par des phases de fine-tuning et de fusion multi-modèles, a permis d'atteindre un équilibre entre rappel et précision, condition indispensable pour garantir la fiabilité du traitement en conditions réelles.

L'ensemble du système a été conçu selon une logique modulaire et orientée utilisateur. L'interface développée en Streamlit permet de charger et traiter les contrats de manière fluide, de visualiser les obligations extraites avec leurs métadonnées (catégorie, responsable, pénalité, période), et de naviguer facilement dans les résultats grâce à des outils de tri et de recherche.

Les résultats obtenus démontrent la possibilité de l'application de l'IA générative au domaine contractuel en français. La criticité de l'extraction de chaque obligation nécessite cependant d'améliorer encore les scores de rappels de ces outils.

References

1. Singh, A., Rose Anish, P., Verma, A., Venkatesan, S., V., L., & Ghaisas, S. (2024). A data decomposition-based hierarchical classification method for multi-label classification of contractual obligations for the purpose of their governance. *Scientific*

- Reports*, 14(1), 12755. <https://doi.org/10.1038/s41598-024-63648-x>. Accessed 2024-12-01.
2. Koreeda, Y., & Manning, C. (2021). ContractNLI: A Dataset for Document-level Natural Language Inference for Contracts. In M.-F. Moens, X. Huang, L. Specia, & S. W.-t. Yih (Eds.), *Findings of the Association for Computational Linguistics: EMNLP 2021* (pp. 1907–1919). Punta Cana, Dominican Republic: Association for Computational Linguistics. <https://doi.org/10.18653/v1/2021.findings-emnlp.164>. Accessed 2024-12-01.
 3. Hendrycks, D., Burns, C., Chen, A., & Ball, S. (2021). CUAD: An Expert-Annotated NLP Dataset for Legal Contract Review. arXiv. <https://doi.org/10.48550/arXiv.2103.06268>. Accessed 2024-12-01.
 4. <https://huggingface.co/datasets/ta1n-1s2n/Adminset-NER>
 5. https://huggingface.co/datasets/Jean-Baptiste/wikiner_fr