Rapport de fin de Sprint

Sprint numéro : 4

Introduction

Comme convenu avec le client, le quatrième Sprint est un Sprint spécial qui a la possibilité d'être rallongé d'une à deux semaines, en plus des deux semaines accordé.

Nous avons extrait :

- L'affiliation de(s) auteur(s),
- L'introduction,
- Le corps,
- La discussion,
- La conclusion.

L'extraction de ces informations est correcte pour tous les articles du Corpus.

D'autres part, nous avons modifié la forme de sortie du fichier au format XML et créer un menu textuel.

Toutes ces modifications, additionnées aux ajouts des Sprint précédents, nous ont poussé à faire une restructuration du code afin qu'il reste performant.

Problèmes rencontrés

Initialement, nous avons été confrontés à des morceaux de code mal positionnés, nécessitant leur réorganisation pour éviter les doublons et assurer leur bon fonctionnement. L'un des principaux obstacles a été la manipulation du buffer de sortie, qui se remplissait à chaque itération pour chaque PDF sélectionné. En redirigeant la sortie vers un buffer fictif pendant l'affichage du menu, puis en la rétablissant pour l'analyse des PDF, nous avons réussi à résoudre ce problème, réduisant ainsi le temps d'exécution.

Notre dernier problème a été autour de l'analyse du fichier « Das_Martins.pdf » où un simple espace pouvait totalement compromettre son analyse.

Optimisations et Performances

Avec les modifications substantielles apportées au code, nous avons réussi à réduire le temps d'exécution de 6 secondes à une plage comprise entre 1 et 1.3 secondes pour l'ensemble du corpus de PDF. Cette amélioration significative est dû au remplacement de tous les accents problématiques dans les PDF avant leur analyse. C'était une opération particulièrement gourmande en ressources.

Afin de tester les performances sur un plus large Corpus, nous avons appliqué notre programme sur 92 fichiers « .pdf ». Le traitement de ces 92 fichiers s'est terminé au bout de 7,7 secondes.

Pour finir sur l'optimisation, nous avons simplement divisé notre programme principal en plusieurs fichiers Python pour une meilleure lisibilité du code.

Stratégie d'Analyse

Pour optimiser le processus d'analyse, notre équipe a développé un algorithme permettant de localiser les mots-clés dans les documents PDF, facilitant ainsi la découpe du texte en paragraphes et la récupération du contenu essentiel. Cette approche a permis de réduire la taille de chaque fonction et d'améliorer la lisibilité du code.

Tests et Fiabilité

Pour garantir la fiabilité du programme, l'équipe a mis en place des tests vérifiant la précision de l'analyse et la récupération correcte des sections, des adresses électroniques, des affiliations, etc. En utilisant une méthode de comparaison basée sur la distance de Levenshtein, nous avons pu détecter les incohérences et les corriger efficacement. Notre Corpus nous a beaucoup aidé à ajuster le « parser » et notamment le fichier « shattered.pdf ».

Gestion des Erreurs

Une amélioration notable a été apportée à la gestion des erreurs. En isolant chaque analyse dans son propre processus, nous avons assuré la robustesse du programme, évitant ainsi les arrêts intempestifs en cas d'erreur lors du traitement d'un PDF spécifique.

Conclusion

En conclusion, ce sprint a permis d'optimiser significativement le processus d'analyse de fichiers PDF, avec des améliorations notables en termes de performances et de fiabilité. Les ajustements apportés au code, la mise en place de tests rigoureux et la gestion efficace des erreurs ont contribué au succès de ce Sprint. Nous sommes désormais en mesure de traiter efficacement un large corpus de documents PDF, tout en assurant la précision et la robustesse de notre programme. Il est à noter que la coopération et l'échange de connaissances au sein de l'équipe ont été essentiels pour surmonter les défis rencontrés et garantir le succès de ce Sprint.