

Recueil du Symposium de 2024 de Statistique Canada : Le futur des statistiques officielles

Améliorer la saisie automatique des reçus d'achat fournis dans le cadre de l'Enquête sur les dépenses des ménages à l'aide de techniques d'apprentissage automatique

par Joanne Yoon et Oladayo Ogunnoiki

Date de diffusion : le 8 septembre 2025



Statistique
Canada

Statistics
Canada

Canada

Améliorer la saisie automatique des reçus d'achats fournis dans le cadre de l'Enquête sur les dépenses des ménages à l'aide de techniques d'apprentissage automatique

Joanne Yoon* et Oladayo Ogunnoiki*¹²

Résumé

L'Enquête sur les dépenses des ménages (EDM) de Statistique Canada recueille des journaux papier et des reçus d'achats comme source de données sur les dépenses des ménages. Un algorithme de saisie automatique a été créé pour l'EDM de 2023 afin de réduire le travail manuel des commis à la statistique qui consiste à extraire les renseignements importants de reçus numérisés de chaînes de magasins populaires. L'algorithme utilisait l'outil de reconnaissance optique de caractères (ROC) Tesseract pour extraire des caractères de texte à partir d'images de reçus et identifiait des entités de magasin et de produit en utilisant des expressions régulières, également appelées des « regex ». Cette étude visait à améliorer l'algorithme actuel de saisie automatique en mettant à l'essai des méthodes de ROC et d'apprentissage automatique plus avancées. Par conséquent, PaddleOCR, une trousse d'outils de ROC à accès libre, a été choisi comme nouveau moteur de ROC par défaut en raison de son rendement général en matière de reconnaissance exacte de textes, notamment au niveau des chiffres, sur des reçus de qualités diverses. De plus, les classificateurs d'entités basés sur des machines à vecteurs de support ont été entraînés sur les enregistrements historiques de l'EDM et les modèles de regex existants. En utilisant des classificateurs pour classer les différents éléments présents sur les reçus au lieu de s'appuyer uniquement sur des modèles de regex, le processus de reconnaissance des produits et des magasins est meilleur. On prévoit que ce nouvel algorithme sera utilisé dans l'EDM de 2025 afin d'améliorer la qualité de la saisie automatique et de réduire le fardeau manuel associé à la saisie des variables de reçus.

Mots-clés : ROC; apprentissage automatique; automatisation.

1. Introduction

L'Enquête sur les dépenses des ménages (EDM) est une enquête canadienne qui recueille des renseignements sur les habitudes de dépenses des Canadiens. Elle examine les dépenses des ménages en matière d'alimentation, d'habillement, de logement, de transport, de soins de santé et autres. Les répondants répondent à un questionnaire et tiennent un journal de leurs dépenses quotidiennes. Lorsqu'ils remplissent le journal, les répondants peuvent choisir d'écrire chaque article acheté dans un journal papier ou de mettre leurs reçus d'achats dans la pochette de leur journal. Statistique Canada recueille ainsi entre 20 000 et 30 000 reçus d'achats par cycle d'enquête. Les reçus papier sont numérisés manuellement à l'interne.

Pour réduire le travail manuel d'extraction des renseignements importants des reçus numérisés, l'équipe chargée de la méthodologie de l'EDM a mis au point un algorithme de saisie automatique des reçus de magasins populaires dont l'image est de bonne qualité. Pour chaque reçu, les détails, tels que le nom du magasin, la date de la transaction et le mode de paiement, ainsi que les variables de produit, notamment le nom du produit, le prix, l'escompte et le dépôt, sont automatiquement saisis. L'algorithme a été utilisé en production pour le cycle de l'EDM de 2023. Cet algorithme extrait le texte des reçus numérisés à l'aide de la reconnaissance optique de caractères (ROC) et reproduit les

*Ces deux auteurs ont contribué à parts égales aux présents travaux.

¹Joanne Yoon, Statistique Canada, 170, promenade Tunney's Pasture, Ottawa (Ontario), Canada, K1A 0T6 (Joanne.Yoon@statcan.gc.ca); Oladayo Ogunnoiki, Statistique Canada, 170, promenade Tunney's Pasture, Ottawa (Ontario), Canada, K1A 0T6 (Oladayo.Ogunnoiki@statcan.gc.ca)

²Les opinions exprimées dans les documents de recherche sont celles des auteurs et ne reflètent pas nécessairement les opinions de Statistique Canada ou du gouvernement fédéral.

renseignements pertinents en format structuré. Les données produites sont validées manuellement et toute erreur est corrigée.

Afin d'accroître la portée et le rendement de l'algorithme de saisie automatique des reçus de l'EDM, plusieurs améliorations ont été apportées à l'algorithme existant, tel qu'il est décrit dans l'article. Plus précisément, le projet visait à améliorer la qualité des renseignements extraits en explorant davantage de moteurs de ROC, en intégrant un classificateur automatique d'entités et en peaufinant les étapes d'analyse syntaxique et de nettoyage des données. Selon les évaluations qui ont été menées, le nouveau moteur de ROC, le correcteur orthographique et la méthode améliorée de reconnaissance des entités ont permis d'améliorer l'exactitude des variables saisies automatiquement.

2. Méthodes examinées permettant d'améliorer la saisie automatique des reçus

L'exécution en cascade de la saisie automatique était constituée des étapes suivantes, comme le montre la figure 2-1 :

Reconnaissance optique de caractères (ROC) : Les textes des reçus numérisés sont convertis en textes lisibles par ordinateur à l'aide de la ROC. Le processus renvoie à la fois le texte et l'indice de confiance pour le texte numérisé. L'indice de confiance peut donc être utilisé pour évaluer la qualité du reçu numérisé. La section 2.1 décrit les moteurs de ROC qui ont récemment été explorés.

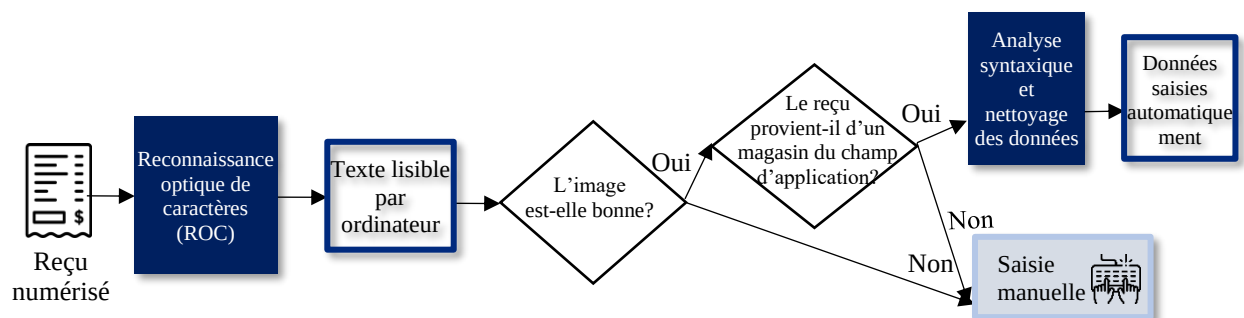
Évaluation de la qualité de l'image : La qualité de l'image est évaluée en vérifiant que l'indice de confiance de la ROC dépasse un seuil prédéterminé, établi au moyen d'essais préalables. Si la qualité est jugée insuffisante, le reçu numérisé est envoyé à des fins de saisie manuelle des données. Il n'entre pas dans le champ d'application du processus de saisie automatique, car un faible indice de confiance indique que la ROC pourrait ne pas bien interpréter les données.

Validation des champs d'application des magasins : En raison des écarts de présentation des reçus d'un magasin à l'autre, l'algorithme se spécialise dans le traitement des reçus de certains magasins populaires dont les reçus sont plus longs. Ces reçus sont plus complexes et plus longs à saisir manuellement. Si la qualité de l'image est acceptable, l'algorithme vérifie si le reçu provient d'un magasin qu'il connaît avant de l'analyser. Les reçus provenant de magasins ne faisant pas partie de ce champ d'application ne se prêtent pas à un traitement automatisé; ils sont donc saisis manuellement.

Analyse syntaxique et nettoyage : En ce qui concerne les reçus faisant partie du champ d'application, le texte est analysé sur le plan syntaxique, nettoyé et structuré pour être saisi avec exactitude dans la base de données de l'EDM. Un classificateur d'entités à apprentissage automatique (décrit dans la section 2.2) et un correcteur orthographique (décrit dans la section 2.3) ont été ajoutés à cette étape pour améliorer le processus d'extraction des données.

Figure 2-1

Représentation visuelle de l'exécution en cascade de la saisie automatique



2.1 Moteurs de ROC

L'algorithme initial de saisie automatique utilisait l'outil de ROC Tesseract, mais d'autres moteurs de ROC, dont EasyOCR, PaddleOCR, KerasOCR, TROCR et Donut Transformer OCR, ont fait l'objet d'une étude pour améliorer la reconnaissance du texte. Les moteurs ont été évalués par rapport à toutes les variables de reçus, y compris les noms de magasin, les dates et les articles. La vitesse n'était pas d'une importance capitale, mais le moteur devait être compatible avec le système actuel et pouvoir être exécuté sur celui-ci.

L'outil de ROC Tesseract (Smith, 2007) était le moteur de référence pour cette étude. Tesseract a été développé par Hewlett-Packard (HP) dans les années 1980 en tant que logiciel propriétaire. Il est devenu libre en 2005, et Google l'a maintenu et mis à jour. Il s'agit d'un moteur de ROC largement utilisé, car il reconnaît et extrait le texte des images avec précision et relativement rapidement, tout en prenant en charge de nombreuses options personnalisées. L'une de celles-ci concerne le mode de segmentation des pages (Page Segmentation Mode [PSM]), qui définit la manière dont le moteur de ROC divise et traite une disposition de texte. L'algorithme existant du projet utilisait PSM 6, qui suppose un seul bloc de texte uniforme, puisque les reçus structurent les textes en blocs uniformes. Les essais effectués avec d'autres configurations de PSM ont montré que PSM 6 offrait le meilleur rendement, aucune amélioration significative n'ayant été observée dans d'autres configurations. Malgré sa fiabilité générale, Tesseract n'a pas réussi à reconnaître les noms de magasin de certaines marques, ce qui suggère une gestion limitée de certaines tailles et de certains styles de polices.

On a mis à l'essai le moteur EasyOCR (Jaided, s.d.) en raison de sa simplicité, de sa mise en œuvre en code source ouvert et de son large éventail de langues prises en charge. Il a toutefois éprouvé des difficultés à reconnaître les entêtes de texte stylisés ou de petite taille de certaines marques de magasins. Cet inconvénient a réduit sa capacité à satisfaire aux exigences du projet.

PaddleOCR (PaddlePaddle, s.d.) est un système de ROC léger et pratique, à code source ouvert, basé sur la technologie PaddlePaddle (une abréviation de **Parallel distributed deep learning**), une plateforme d'apprentissage profond à code source ouvert développée par Baidu Research. Il a affiché un rendement impressionnant, mais les images devaient être prétraitées pour optimiser l'exactitude. Elles devaient notamment être recadrées pour réduire les espaces blancs et ajuster leur hauteur. Même si cette dépendance à l'égard du traitement préalable des images a ajouté de la complexité, PaddleOCR a produit des résultats cohérents et précis, ce qui en a fait le choix par défaut pour le projet.

TROCR (Microsoft Research, s.d.) est un modèle de texte qui utilise une architecture de transformeur pour comprendre les images et générer des textes au niveau des mots. Comme le modèle s'appuyait sur un algorithme de détection du texte avant l'étape de reconnaissance, Tesseract a été utilisé conjointement à TROCR. Bien que TROCR se soit avéré très précis, il était nettement plus lent que les autres moteurs et nécessitait une accélération graphique (GPU) à des fins pratiques. Les délais de traitement étaient excessifs et d'autres algorithmes de détection du texte devaient être étudiés en vue d'améliorations futures.

Deux autres moteurs de ROC ont été étudiés : KerasOCR (Morales, s.d.) est une mise en œuvre par Keras d'un réseau neuronal récurrent convolutif pour la reconnaissance de texte. Donut Transformer OCR (NAVER AI, s.d.) est un transformeur de compréhension de documents (**document understanding transformer**) qui utilise un modèle de transformeur de bout en bout sans ROC. Leur rendement initial sur des échantillons de reçus n'était toutefois pas prometteur. Donut Transformer OCR n'offrait pas la souplesse nécessaire pour renvoyer des résultats spécifiques en fonction du format de son ensemble de données d'entraînement. Ces résultats illustrent l'importance de sélectionner des moteurs qui correspondent aux exigences spécifiques du projet.

Après des essais approfondis, PaddleOCR a été choisi comme moteur de ROC par défaut pour ce projet, en raison de son exactitude constante avec des images de reçus traités au préalable. Tesseract avec PSM 6 et d'autres moteurs sont pris en charge comme solutions de ROC de rechange. Les utilisateurs peuvent choisir une autre option de ROC pour leurs cas d'utilisation spécifiques.

2.2 Correcteur orthographique

Lorsque la ROC reconnaît mal un caractère dans un mot en raison de problèmes tels qu'une mauvaise qualité d'image ou des lignes verticales sans encre, un correcteur orthographique le corrige en se basant sur l'information à proximité. Un correcteur orthographique en python basé sur SymSpell a été entraîné à l'aide d'un vocabulaire spécifique au domaine compilé à partir de la base de données de l'EDM et de mots-clés liés aux entités des reçus. Si un mot provenant de la ROC ne figurait pas dans son vocabulaire, il le remplaçait par un mot de sa liste de vocabulaire qui nécessitait le moins de changements de lettres. Il s'appuyait également sur des mots à proximité pour corriger les fautes d'orthographe, mais lorsque plusieurs candidats étaient identifiés, le mot le plus fréquent dans la base de données de l'EDM était sélectionné.

2.3 Classificateur d'entités

Les regex sont des séquences de caractères utilisées pour définir des modèles de recherche permettant de faire correspondre du texte dans un texte d'entrée. À l'origine, les expressions régulières servaient à identifier des entités, telles qu'un mode de paiement, dans une ligne de texte. Cette méthode fonctionne bien dans la pratique, mais les classificateurs d'apprentissage automatique ont été envisagés comme option plus robuste, capable de cerner des modèles dans les données afin de traiter des structures complexes ou ambiguës. Néanmoins, les expressions régulières ont été conservées pour identifier les dates, puisque les reçus comportaient des formats standard de dates qui étaient extraites avec plus d'exactitude à l'aide de regex.

Pour catégoriser toutes les entités pertinentes, deux classificateurs différents ont été mis au point : un classificateur au niveau du reçu et un classificateur au niveau du produit. Un classificateur d'entités au niveau du reçu a d'abord été entraîné pour classer chaque ligne de texte d'un reçu en une entité. Les entités comprenaient les éléments suivants : Département, Transaction incomplète, Mode de paiement, Touche de démarrage, Touche d'arrêt, Nom du magasin, Dépôt, Escompte, Écofrais, Prix par quantité, Nom du produit, Subvention, Nul et Autre. Les éléments Touche de démarrage et Touche d'arrêt sont des lignes qui marquent le début et la fin de la liste des produits achetés. Les lignes situées entre les éléments Touche de démarrage et Touche d'arrêt ont été classées à nouveau à l'aide d'un classificateur au niveau du produit qui ne prédit que les entités du produit : Nom du produit, Dépôt, Escompte, Écofrais, Prix par quantité, Subvention, Nul et Autre.

Le TF-IDF (Fréquence du terme-Fréquence inverse de document) a été utilisé pour transformer les textes en caractéristiques numériques qui servent d'entrée à un classificateur d'apprentissage automatique. Il s'agit d'une statistique numérique utilisée dans le traitement du langage naturel pour évaluer l'importance d'un terme contenu dans un document relativement à une collection de documents. Des termes de n-grammes de deux à cinq caractères ont été utilisés.

Divers algorithmes d'apprentissage supervisé ont été entraînés et comparés afin de cibler le modèle dont le rendement était le meilleur pour notre tâche. Un modèle d'arbre de décision a d'abord été mis à l'essai, car il s'agit d'un modèle interprétable qui divise récursivement les données en fonction des valeurs des caractéristiques, ce qui en fait un candidat de choix pour comprendre l'impact des différentes caractéristiques sur les résultats de la classification. Random Forest a permis d'améliorer le modèle d'arbre de décision en combinant plusieurs arbres entraînés à l'aide de différents sous-échantillons de données. Random Forest peut réduire le surajustement, un problème courant des arbres de décision individuels, afin d'améliorer la stabilité et la généralisation du modèle. Le renforcement du gradient dans XGBoost est une technique d'apprentissage automatique d'ensemble qui construit une série d'arbres de décision. Le nouvel arbre corrigeait les erreurs de l'arbre précédent en utilisant la descente de gradient pour minimiser la fonction de perte du modèle.

Le perceptron multicouche comporte plusieurs couches de nœuds interconnectés pour traiter les renseignements à l'aide d'une fonction d'activation. Les connexions pondérées entre les nœuds pouvaient apprendre des modèles complexes dans les données grâce à des algorithmes d'entraînement itératifs. Il s'agissait d'un choix valable lorsqu'il s'agit de traiter des ensembles de données complexes pour lesquels les modèles plus simples risquent de ne pas être à la hauteur.

Trois types de modèles de machines à vecteurs de support (SVM) ont été entraînés. Les SVM étaient des classificateurs discriminants qui trouvaient l'hyperplan optimal dans un espace à dimensions n , où n était le nombre de caractéristiques, qui maximise la marge entre les catégories. Une SVM avec un noyau linéaire était un modèle de base qui suppose une frontière de décision linéaire entre les caractéristiques. Une SVM linéaire avec descente stochastique de gradient (SGD) était une solution de rechange efficace pour les données à haute dimension. Elle estime le gradient de la perte pour chaque échantillon et le modèle est mis à jour avec une courbe de force à la baisse. Enfin, une SVM avec un noyau de fonction de base radiale (RBF) a été sélectionné pour saisir des relations complexes dans des données non linéairement séparables.

3. Résultats

3.1 Rendement des classificateurs d'entités

Pour entraîner et évaluer les classificateurs d'entités, des données étiquetées ont été recueillies auprès de plusieurs sources. La base de données de l'EDM comprenait 74 008 entrées de produits, 13 544 entrées de magasins et 498 entrées de départements. On a obtenu des données d'entraînement supplémentaires pour d'autres entités en exécutant le code de saisie automatique existant de l'EDM sur 1 469 reçus de l'EDM de 2023, dont on avait déjà vérifié la bonne analyse syntaxique. Cela a permis d'obtenir 20 118 lignes de données, toutes cruciales pour entraîner les modèles d'apprentissage automatique à reconnaître les différents formats utilisés par les différents magasins pour présenter des entités telles que les dépôts et les escomptes. Ce nombre inclut 12 538 lignes uniques qui ont été jugées non pertinentes pour l'EDM et qui appartiennent donc à la classe « Autres ». Les modèles d'expressions régulières existants et les données générées automatiquement à partir de ces modèles ont également été utiles pour saisir l'ensemble des formats d'entité identifiés par l'algorithme de saisie automatique.

Pour évaluer la capacité du modèle à classer les entités malgré le déséquilibre des classes, chaque modèle de classification a été évalué à l'aide d'une méthode de validation croisée quadruple stratifiée. Le tableau 3.1-1 résume les différents modèles mis à l'essai et leur score F1 selon la macro pour les entités au niveau du reçu et au niveau du produit. Le score F1 selon la macro a été la mesure de rendement déterminante parce qu'elle pondère uniformément l'importance de chaque classe, ce qui signifie que même s'il y a très peu d'exemples d'une classe dans les données d'entraînement, la qualité des prédictions pour cette classe est tout aussi importante que pour une classe ayant beaucoup d'exemples.

Tableau 3.1-1
Score F1 des modèles de classification

Modèles mis à l'essai	Score F1 selon la macro du classificateur de reçus	Score F1 selon la macro du classificateur de produits
Arbre de décision	40,4 %	57,6 %
XGBoost	77,4 %	85,2 %
Perceptron multicouche	78,8 %	90,4 %
Forêt aléatoire	83,2 %	84,0 %
SVM linéaire avec entraînement au moyen de la descente par gradient stochastique	84,8 %	89,8 %
SVM avec noyau linéaire	85,5 %	90,4 %
SVM avec noyau de fonction de base radiale	87,4 %	91,4 %

Comme la SVM avec noyau de fonction de base radiale a obtenu les meilleurs résultats avec un score F1 selon la macro de 87,4 % pour les entités de reçus et de 91,4 % pour les entités de produit, c'est elle qui a été utilisée dans le projet.

3.2 Rendement de la saisie automatique en général

Le rendement des algorithmes existants et améliorés a été comparé à l'aide d'un ensemble d'essais de 3 620 reçus, ce qui correspond à deux mois de reçus numérisés dans le cadre de l'EDM. Les paramètres clés comprenaient le nombre total de reçus pouvant faire l'objet d'une saisie automatique et l'exactitude des variables de reçus et de produits. Les données sur les reçus du tableau 3.2-1 montrent une amélioration notable (de 41,3 % à 51,5 %) du pourcentage de reçus saisis automatiquement avec le nouvel algorithme. En raison des indices de confiance naturellement élevés de la ROC produits par PaddleOCR, comparativement à la variabilité observée avec Tesseract, aucun des reçus d'essai ne se trouvait en dessous du seuil de confiance pour être considéré comme hors du champ d'application en utilisant PaddleOCR. Par conséquent, les 8,0 % de reçus que Tesseract excluait en raison de la mauvaise qualité de la ROC n'étaient plus exclus par PaddleOCR et ont été traités. Par ailleurs, la ROC améliorée reconnaissait mieux les noms de magasins avec des polices fantaisistes, ce qui a permis de réduire de 1,9 % le nombre de magasins mal reconnus et donc de réduire le nombre de reçus hors du champ d'application.

Tableau 3.2-1
Pourcentage de reçus saisis automatiquement

Mesure	Algorithme existant	Nouvel algorithme
Nombre total de reçus	3 620	3 620
Hors du champ d'application – Confiance - ROC	291 (8,0 %)	0 (0 %)
Hors champ d'application - Magasin hors du champ d'application	1 833 (50,6 %)	1 754 (48,5 %)
Reçus dans le champ d'application	1 496 (41,3 %)	1 866 (51,5 %)

L'exactitude de variables clés, telles que le nom du magasin, la date et le mode de paiement (détaillées dans le tableau 3.2-2), a connu des améliorations mineures, mais constantes, ce qui a permis de maintenir un rendement élevé. L'exactitude du mode de paiement a été mesurée pour la première fois dans le cadre de cette expérience.

Tableau 3.2-2
Exactitude des variables de reçus

Mesure	Algorithme existant	Nouvel algorithme
Total - Reçus dans le champ d'application	1 496	1 866
Exactitude des noms des magasins	1 411 (94,3 %)	1 768 (94,7 %)
Exactitude des dates (jour et mois)	1 226 (84,6 %)	1 581 (84,7 %)
Exactitude des modes de paiement	Non calculée	1 628 (87,0 %)

La qualité de l'extraction des produits s'est également améliorée. Même si davantage de reçus, et donc davantage d'articles, se retrouvaient dans le champ d'application et devaient être saisis, le nombre d'articles manquants et d'articles supplémentaires a diminué. Dans l'ensemble, le nombre d'articles bien saisis a augmenté, de même que la similarité des textes des descriptions d'articles et l'exactitude des prix des produits. Des mesures d'exactitude de la saisie automatique de quatre variables supplémentaires, soit Nom du service, Escompte, Écofrais et Dépôt, ont été ajoutées pour cette expérience, tel que l'illustre le tableau 3.2-3.

Tableau 3.2-3**Exactitude des variables de produits**

Mesure	Algorithme existant	Nouvel algorithme
Total - Reçus dans le champ d'application	12 350	14 755
Articles bien saisis	11 481 (93 %)	14 014 (95 %)
Articles manquants	869	741
Articles supplémentaires	3 107	1 747
Bon prix	8 834 (76,9 %)	12 171 (86,8 %)
Exactitude des noms des services	Non calculée	13 364 (95,4 %)
Exactitude des escomptes	Non calculée	13 148 (93,8 %)
Exactitude des écofrais	Non calculé	13 924 (99,4 %)
Exactitude des dépôts	Non calculée	13 874 (99,0 %)

Les nouveaux ajouts à l'algorithme ont permis de réduire les erreurs de reconnaissance, d'augmenter l'exactitude de la reconnaissance et d'améliorer l'efficacité de l'automatisation. Parmi les principales réalisations, on retrouve les suivantes :

- **Augmentation de la saisie automatique** : Le pourcentage de reçus saisis automatiques est passé de 41,3 % à 51,5 %.
- **Similitude entre les descriptions d'articles** : Augmentation de 93 % à 95 %.
- **Exactitude des prix** : Amélioration de 76,9 % à 86,8 %.
- **Nouvelles variables saisies** : Ajout des variables « Nom du service », « Écofrais » et « Escompte ». Selon les essais, ces variables sont bien saisies automatiquement et affichent en moyenne un taux d'exactitude de 96,2 %.

4. Conclusion

Cette exécution en cascade a permis d'améliorer l'exactitude de l'extraction du contenu des reçus en utilisant des méthodes de ROC avancées, des correcteurs orthographiques et des classificateurs d'entités. Ces améliorations ont également permis d'augmenter le nombre de reçus et de variables saisis automatiquement. L'algorithme amélioré sera mis en œuvre pour l'EDM de 2025 afin d'améliorer la qualité de la saisie automatique et de réduire l'effort manuel nécessaire pour saisir les variables de reçus.

Remerciements

Les auteurs souhaitent remercier Émilie Mayer pour sa contribution à ce projet. Ils remercient également Denis Malo, Kimberley Flak, Olfa Khazri et Johan Fernandes pour leur travail de révision ainsi que leurs commentaires constructifs.

Bibliographie

Jaied AI (2020), EasyOCR: Ready-to-use OCR with 40+ languages supported (v1.7.1) [Computer program], Extrait de <https://github.com/JaiedAI/EasyOCR>.

Microsoft Research (2021), TrOCR: Transformer-based Optical Character Recognition (v4.28.1) [Computer program], Extrait de <https://github.com/microsoft/unilm/tree/master/trocr>.

Morales, F. (2020), KerasOCR: A High-Level Deep Learning OCR Library (v0.8.5) [Computer program], Extrait de <https://github.com/faustomorales/keras-ocr>.

NAVER AI. (2021), Donut: Document Understanding Transformer (v1.0.9) [Computer program]. Extrait de <https://github.com/clovaai/donut>.

PaddlePaddle (2020), PaddleOCR: End-to-End OCR Solution (v2.6.0) [Computer program]. Extrait de <https://github.com/PaddlePaddle/PaddleOCR>

Smith, R. (2007), “An Overview of the Tesseract OCR Engine”, In ICDAR’07, *Proceedings of the Ninth International Conference on Document Analysis and Recognition*. Extrait de <https://github.com/tesseract-ocr/tesseract>