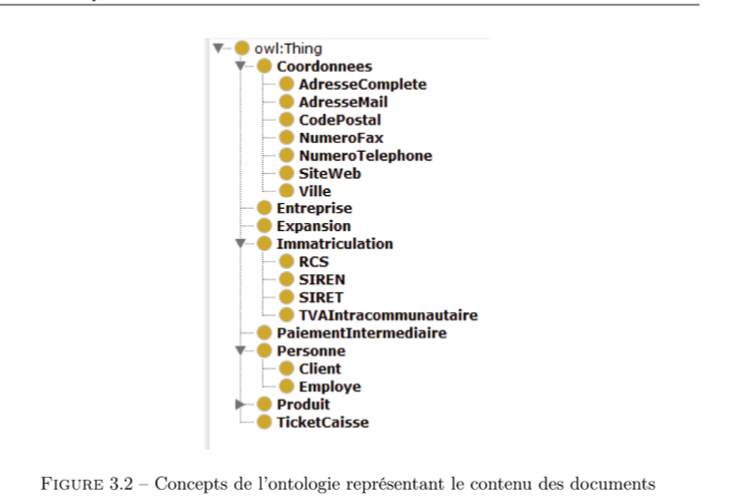
Lecture article ticket de caisse

Cette tache de détection de fraude au ticket de caisse peut se faire avec des techniques de traitement automatique du langage naturel (TALN), une approche basée sur la régression en tirant partit d’un modèle de langage pré-entrainé afin de représenter le contenu textuel, ils ont enrichi la représentation avec des entités et des relations basée sur une ontologie spécifique au domaine.

****

La contrefaçon est la production d’un document authentique par imitation et le faux est l’altération d’un ou plusieurs éléments sur un document authentique. L’un des principaux défis de la détection de la fraude de documents est le manque de données annotées librement disponibles.

La plupart des recherches en matière de fraude ce concentre principalement sur l’analyse d’image de document notamment grâce à la vision par ordinateur, effectivement l’image d’un document peut être modifiés sur le document numérique original ou dans sa version imprimée et numérisée tout cela a l’aide d’un logiciel d’édition d’image.

**Le modèle**

Le modèle de détection de fraude est pré-entrainé sur le modèle de CamemBERT qui est un modèle linguistique pré-entrainé de pointe pour le français lui-même basé sur le modèle RoBERTa.

Mettre des exemples de comment fonctionne CamemBERT.

**Les données**

Le dataset est composé de 998 images de ticket français et de leurs transcriptions associées, Ce jeu de données a été collecté pour fournir un corpus parallèle image/texte et un point de référence pour évaluer les méthodes de détection de fraude basées sur le texte. Les faux tickets sont le résultat d’ateliers de fraude, au cours desquels les participants ont reçu un ordinateur standard avec plusieurs logiciels d’édition d’images pour modifier manuellement les images et les transcriptions associées des tickets

**Situations réelles**

Des demandes de remboursement frauduleuses on pus être effectuées en modifiant le prix d’un article , son nom, le moyen de paiement, etc. La falsification peut également viser une extension indue de la garantie en modifiant la date.

D’autres falsifications peuvent impliquer l’entreprise émettrice dans le but de blanchir de l’argent.

**Pré-traitement et choix de l’entrée**

Ils ont expérimenté avec quatre types d’entrées :

**Texte :** le texte brut sans aucun pré-traitement

**Entités :** extraire les entités présentes sur la base d’une ontologie de ticket de caisse et les concatène avec un séparateur d’espace

**Texte + Entités :** enrichir le texte du ticket de caisse en introduisant des marqueurs spéciaux pour chaque type d’entité (tel que : entreprise, produit…) et remplace chaque entité dans le texte par son libellé entourer des marqueurs de son type d’entité.

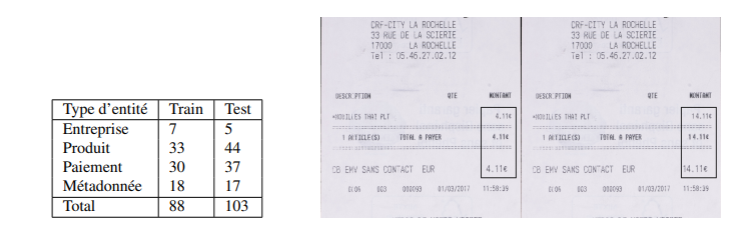
**Triplets :** basés sur la même ontologie, mais en extrayant également les relations sémantiques. (Relations sémantiques ?)

**Création de l’ontologie des tickets :**

Elle a été peuplée automatiquement avec des expressions régulières créées manuellement et basées sur les régularités des tickets de caisse. Par exemple, les produits et leurs prix ont été extraits des lignes du document terminées par le symbole « euro », ou l’utilisant comme séparateur décimal, à l’exclusion des lignes qui rapportent le total ou le paiement, pour ensuite crée automatiquement l’ontologie automatiquement grâce à la bibliothèque Owlready2.

**Extraction d’entités :**

Une entité extraite correspond à une instance d’une classe définie dans l’ontologie, qui définit sont type, ils ont annoté chaque entité fraudée, c’est-à-dire chaque entité ayant été modifier pendant la constitution du dataset (altération, ajout ou suppression), les modifications ne sont pas comptabilisées en elles-mêmes, seules les entités modifiées le sont : par exemple une date « 11/02/2023 » modifier en « 23/02/2024 » compte pour une entité modifiée, même si elle a subi deux modification graphique.



Le nombre d’entités modifiées est présenté dans le tableau de gauche et exemple de ticket de caisse modifié à droite.

La plupart des entités modifiées impliquent des montants d’argent (entités de produit et de paiement), même si ceux-ci ne sont pas toujours modifiés de manière cohérente, sur la figure de droite il y est présent trois fraudes le prix du produit, le prix total à payer et le montant payé qui ne comporte pas d’incohérence numérique.

**Expériences**

Premièrement il simule manuellement un vérificateur qui ne prend en compte que les incohérences simples telle qu’un écart entre le total et la somme des prix, entre le total et le total payé, ou entre la quantité, le prix unitaire et le prix du produit.

Ensuite il crée un classifieur de régression par machine à vecteur de support (SVM) avec des hyperparamètres par défaut comme notre modèle de base appliqué à la représentation fréquence des termes-fréquence inverse des documents (TF-IDF) des unigrammes et des bigrammes extraits des tickets en minuscules.

**Transformation d’un mot en TF-IDF :**

Les tickets de caisse sont analysés pour extraire des mots individuels (unigrammes) et des paires de mots (bigrammes), puis ces mots sont transformés en une forme spéciale appelée TF-IDF, qui aide à représenter leur importance dans les tickets. Cette représentation est effectuée en utilisant la fréquence des termes (TF) et la fréquence inverse du document (IDF).

Prenons un exemple simple pour illustrer la transformation d'un mot en TF-IDF.

Supposons que nous ayons un ensemble de documents, et nous voulons calculer le TF-IDF pour le mot "chat". Voici un exemple de quelques documents :

1. Document 1 : "Le chat est sur le tapis."
2. Document 2 : "Le chien et le chat jouent ensemble."
3. Document 3 : "Le chat dort paisiblement."

Maintenant, le calcul de TF-IDF se fait en deux étapes :

1. **TF (Term Frequency)** : C'est la fréquence du mot dans un document par rapport au nombre total de mots dans ce document.
   * Pour le document 1 : Le mot "chat" apparaît une fois sur un total de 5 mots. Donc, le TF pour "chat" dans le document 1 est 1/5.
   * Pour le document 2 : Le mot "chat" apparaît une fois sur un total de 6 mots. Donc, le TF pour "chat" dans le document 2 est 1/6.
   * Pour le document 3 : Le mot "chat" apparaît une fois sur un total de 4 mots. Donc, le TF pour "chat" dans le document 3 est 1/4.
2. **IDF (Inverse Document Frequency)** : C'est une mesure de l'importance du mot dans l'ensemble des documents. Cela permet de réduire le poids des mots très fréquents qui apparaissent dans de nombreux documents. Le calcul de IDF est effectué en prenant le logarithme de l'inverse de la fraction du nombre total de documents dans le corpus contenant le mot.
   * Le mot "chat" apparaît dans 3 documents sur un total de 3 documents. Donc, IDF pour "chat" est log(3/3) = log(1) = 0.

Maintenant, nous multiplions TF par IDF pour obtenir le score TF-IDF final pour le mot "chat" dans chaque document :

* Pour le document 1 : TF-IDF = (1/5) \* 0 = 0
* Pour le document 2 : TF-IDF = (1/6) \* 0 = 0
* Pour le document 3 : TF-IDF = (1/4) \* 0 = 0

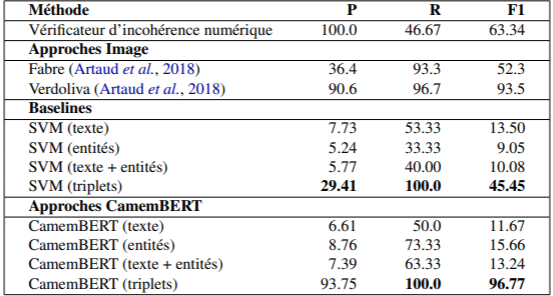
Dans cet exemple simplifié, le score TF-IDF pour le mot "chat" est 0 dans tous les documents, ce qui signifie que le mot "chat" n'est pas considéré comme très important dans cet ensemble de documents. Cependant, dans des ensembles de documents plus larges et plus variés, les scores TF-IDF peuvent varier pour refléter l'importance relative des mots dans chaque document.

Nous avons vu la manière mathématique d’aboutir à ce résultat mais il existe deux bibliothèques en python permettant ce dernier scikit-learn et NLTK (Natural Language Toolkit).

Il compare également les résultats à deux approches image déjà existantes, proposées dans la compétition Find it !. L’architecture Verdoliva est également basée sur un SVM et combine trois approche différentes : un module de détection de copier-coller, un module d’extraction (et de comparaison) de signature de caméra, et un module de détection de faux basée sur les caractéristiques locales de l’image, proposée a l’origine comme méthode de stéganalyse, et présente aussi les résultats de Fabre qui s’appuient sur un modèle pré-entrainé Resnet152 pour la classification.

**Résultat**

Le tableau ci-dessous présente les résultats de la classification binaire entres les classes Fraudé et Authentique. La classification étant très déséquilibrée, nous ne présentons que les résultats pour la classe Fraudé. Nous remarquons que les méthodes utilisant les Triplets comme entrée sont plus performantes que les autres, on pourra aussi remarquer que le recall est a 1 ce qui signifie que tous les tickets falsifiés sont retrouvés avec succès.



**Résumé**

- Le SVM est utilisé comme un type d'algorithme d'apprentissage automatique pour la classification des tickets de caisse en deux catégories : authentique ou frauduleux. Il est utilisé pour créer un classifieur basé sur les caractéristiques extraites des données des tickets.

- Le TALN est utilisé pour comprendre et extraire les informations textuelles des tickets de caisse. Dans ce cas, le TALN est utilisé pour traiter le texte brut des tickets de caisse, on peut donc déduire que le TALN travaillera sur les .txt de la BDD et que les données sur les qu’elle elle devra travailler et comprendre le sens sont sur les différents résultats des expressions régulières.

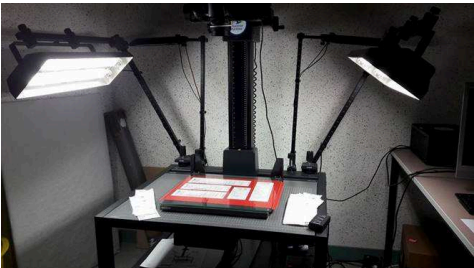
En combinant ces deux approches, le système peut utiliser à la fois les informations textuelles extraites des tickets de caisse et les caractéristiques des données pour détecter la fraude de manière efficace.

**Les critères de fraude à retenir :**

* Les incohérences simples telle qu’un écart entre le total et la somme des prix, entre le total et le total payé, ou entre la quantité, le prix unitaire et le prix du produit.
* Détection de copier-coller … suivre ce lien page 4 : chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://coria-taln-2023.sciencesconf.org/461873/document
* Modification du prix d’un article , son nom, le moyen de paiement. La falsification peut également viser une extension indue de la garantie en modifiant la date.
* Ce qui a été énumérée dans le résumé

Pour donner suite à cela je décortique une thèse qui est proposé à la fin de cette dernière de 202 pages qui illustre de bout en bout l’acheminement de leur projet.

Le projet se décompose donc en plusieurs partie le premier étai de constituer leur base de données en prenant en photo des tickets de caisse :



**OCR**

Ensuite après avoir pris en photo ils ont OCRisé ces tickets de caisse afin de passer de l’image au texte, pour ce faire ils ont utilisé l’OCR d’ABBYY FineReader Engine 11 Sample en utilisant différents paramètres qu’il leurs semblais intéressant, en voici la liste :

* EnableAggressiveTextExtraction (aeate) : détecte tout le texte dans l’image y compris le texte de mauvaise qualité ;
* DetectTextOnPictures (adtop) : détecte le texte inclus dans les images ;
* DontDetectTables (adt) : les tableaux ne sont pas détectés ;
* SingleLinePerCell (tslpc) : reconnait les tableaux avec une ligne de texte par cellule ;
* FastObjectsExtraction (aftda) : détecte tout le texte de l’image, y compris dans les images ;
* PermitModelAnalysis (apma) : analyse différentes mises en page et sélectionne la meilleure variante, ce qui peut améliorer la qualité de la reconnaissance ;
* TextTypes Receipt et Matrix (rtt) : le type de texte correspond au type de texte labellisé « ticket de caisse » ou « matrice » ;
* AltoDontWriteNondeskewedCoordinates (adwndc) : les coordonnées des caractères, mots et blocs sont celles de l’image utilisée pour la reconnaissance, qui peut donc avoir été pivotée.

Voir ici recherche sur OCR ABBYY: C:\Users\pierrontl\OneDrive - GIE SIMA\Documents\GitHub\Fraude\code\_Tom\Recherche et Résumé\recherche.docx

Leur OCR était donc capable de prendre en entrée une image au format TIF, de spécifier la langue soit ici le français, le type de texte (ticket de caisse), de ne pas détecter les tableaux et d’afficher les coordonnées de l’image redressé dans un fichier XML ALTO de sortie.

Leur rendu OCR n’étais pas parfait il était possible de visualiser ces erreurs suivantes :

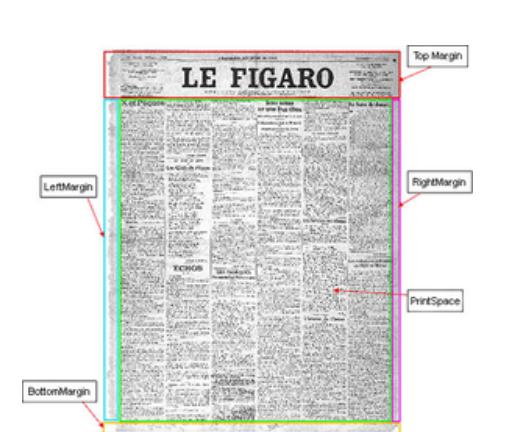
* Les signes euro de fin de lignes qui sont transformer en C ou en f
* Les G signifiant « gramme » sui sont interprété comme des chiffres comme le chiffre 8
* Un espace ajouté à l’avant dernière ligne
* Des chiffres oublier
* Des étoiles de début de ligne transformé en accent circonflexe ou non reconnue
* Des points reconnus en virgule
* Des lignes horizontales encadrant le total reconnu comme du texte

**Recherche sur XML alto**

Le format ALTO peut être utilisé pour exploiter les résultats de l’OCR, c’est l’un des formats les plus utilisés pour la conversion des textes contenu dans des images de document. Il conserve toutes les coordonnées géométriques des contenus (textes, illustrations, graphiques) dans l’images et permet la superposition de l’image et du texte (par exemple dans un PDF multicouche) ainsi que la surbrillance des mots recherchés lors d’une requête.

Le format ALTO permet également la segmentation d’une page en différents éléments composé de sous éléments.

(Extrait d’une page de presse le Figaro)



**Exemple de segmentation d’une page de presse avec le format ALTO.**

L’élément page peut contenir cinq éléments:

* TopMargin : désigne la zone supérieure de la page du bord gauche au bord droit hors zone de texte. Quand c’est possible, il s’agit de la zone contenant le titre courant, l’ours, etc.
* BottomMargin : désigne la zone inférieure de la page du bord gauche au bord droit hors zone de texte.
* LeftMargin : désigne la zone gauche de la page hors zone supérieure, zone inférieure et zone de texte
* RightMargin : désigne la zone droite de la page hors zone supérieure, zone inférieure et zone de texte.
* PrintSpace : désigne la zone de texte. Cet élément est obligatoire. Il contient au moins un élément BlockGroup.

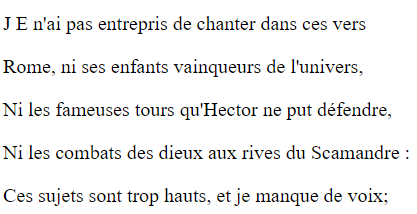
Dès qu’un de ces éléments contient une information (texte, illustration… ), cette information est décrite dans un ou plusieurs éléments BlockGroup.

Les éléments BlockGroup peuvent être de quatre types différents :

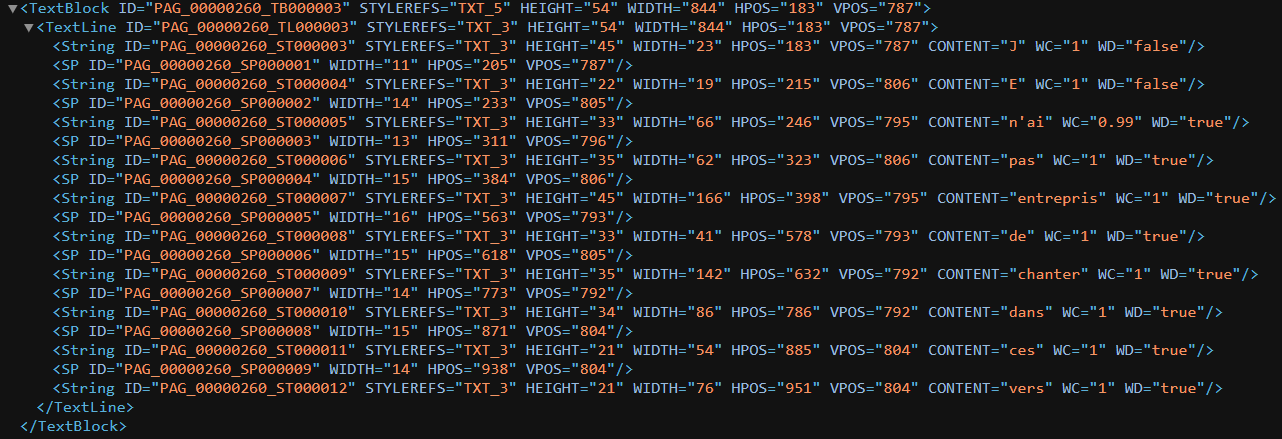
* TextBlock : désigne le bloc de texte. Cet élément est utilisé pour regrouper les lignes de textes en un ensemble cohérent ;
* Illustration : désigne une image ou un dessin ;
* GraphicalElement : désigne un élément graphique autre qu’une image ou un dessin. Il peut être utilisé pour décrire un élément de séparation intertextuel ou un élément textuel non reconnu en tant que tel par l’OCR ;
* ComposedBlock : est utilisé pour permettre l’imbrication d’éléments BlockGroup.

A l’intérieur d’un TextBlock, l’élément String rassemble les chaînes de caractères.

**Exemple de texte :**

****

**Voici un exemple ci-dessous d’un résultat XML ALTO :**

****

On peut voir que ce résultat reprend la première ligne du texte ci-dessus, puis donne différents résultats, comme le style qui fait référence à la police, la taille de la police, la position horizontal et vertical, le contenu (soit le caractère), WC pour Word Confidence allant de 0 à 1 cette valeur sert à calculer :

* Le taux de qualité de chaque page qui se calcule en faisant la somme des scores individuels pour chaque page, puis en divisant ce total par le nombre de mots de la page respective
* Calculer le taux de qualité de chaque document en faisant la somme de toutes les pages du document, divisée par le nombre de mots du document.

**Correction de la sortie OCR**

A l’aide d’expression régulière ils ont pu corriger l’erreur du signe « euro ». Ils ont tout simplement remarqué que le symbole pouvait être tromper avec : : 4, 5, 6, C, c, e, E, F, f, K, s, S, T donc quand ces caractères intervenaient en fin de ligne, suivant une séquence comprenant un chiffre, une virgule ou un point puis deux chiffre, c’est que sa correspond à un montant et sont donc remplacer par le symbole « euro ».

**Falsification de documents**

Comme dit ci-dessus ils ont engagé des personnes pour falsifiés les documents, les manipulations d’images de décomposait en 5 catégories :

* CPI : copier-coller a l’intérieur du document (ou copy-move forgery)
* CPO : copier-coller à l’extérieur du document (ou splicing)
* IMI : pour imitation, boite textuelle imitant la police de caractères du ticket
* CUT : suppression d’un ou plusieurs caractères
* Autres : ils ont laissé libre au cours aux personnes s’ils avaient des idées de falsification

**Des données textuelles aux connaissances**

Dans l’optique de vérifier les informations des documents pour les authentifier, ou au contraire, les détecter comme faux, ils doivent d’abord extraire les informations du document, c’est-à-dire reconnaître le type d’information de chaque partie du texte. Dans le cas des tickets de caisse, il s’agira de repérer dans le texte ce qui est un nom de produit, ce qui est un prix, les adresses, les noms de magasins, les totaux, les numéros de SIRET et de nombreuses autres informations. Ces informations, que l’on peut assimiler à des entités nommées, doivent donc être extraites et sauvegardées de façon à conserver et à mettre en évidence les liens qu’elles ont entre elles. Ces liens permettront par la suite de créer des requêtes que nous enverrons sur les moteurs de recherche afin de comparer les informations du document et les informations du Web. Cette représentation nous permettra également d’enregistrer les informations de tous les tickets et de pouvoir les comparer entre elles.

**Les différentes méthodes employées par les concurrents**

**Méthode 1** : La première méthode n’utilise que les images et combine de l’apprentissage profond avec des techniques de détection de la fraude pour obtenir plus de 85% de bonnes suppositions sur le corpus d’apprentissage. Les images sont toutes d’abord prétraitées en utilisant une combinaison de méthodes de détection d’images trafiquées :

* Analyse du niveau d’erreur (utilisée pour identifier différents niveaux d’artefacts de compression dans une image, ces artefacts peuvent être des indications de manipulation ou de falsification).
* Transformée en ondelette discrète
* Niveaux de gris

Ces matrices à trois dimensions, pour les trois méthodes, ont ensuite été fournies au réseau de neurones Resnet152 (He et al. 2016).

**Méthode 2** : Cette méthode vise à détecter des parties de l’image d’un document qui sont dupliquées, par exemple en cas de modification d’une chaîne de caractères par copier-coller de quelques-uns des caractères. Sur la base de quelques travaux antérieurs appliqués sur des images de scènes naturelles (Fridrich et al. 2003), la méthode développée est basée sur la transformée en cosinus discrète, qui est souvent utilisée dans les algorithmes de compression d’images en raison de sa capacité à projeter une image (ou une partie d’une image) avec d’excellentes propriétés pour regrouper les niveaux d’énergie. Cela permet par conséquent d’avoir des informations majeures sur seulement quelques coefficients. Cet algorithme utilise cette propriété pour détecter et identifier les zones des images qui possèdent des coefficients similaires, signifiant que l’information est identique.

**Méthode 3** : L’approche proposée ici se compose de neuf modules de contrôle, chacun concernant un type spécifique de fraude. Ces modules sont basés soit sur le texte soit sur l’image :

* Modules basés sur le texte :
* Vérification de variation des prix : recherche des prix aberrants
* Vérification du total à payer : examen des incohérences dans les prix des articles et le montant à payer
* Vérification de texte manquant : recherche des mots-clés qui impliquent un élément d’information spécifique, mais cette information est manquante
* Vérification des remises : recherche des incohérences dans les promotions
* Vérification des quantités : recherche des incohérences dans la formule quantité × prix de l’article = somme
* Vérification des dates : recherche des dates non valides
* Modules basés sur l’image (utilisant OpenCV) :
* Contrôle des couleurs : recherche de saturation artificielle, de noirceur, ou de « bruits de poivre » (pixels noirs)
* Vérification des parties effacées : recherche des zones blanches non naturelles ou de grandes zones homogènes (qui n’ont pas de bruit)
* Vérification des copier-coller : recherche des composantes connexes identiques dans les images binarisées

Chaque module renvoie une valeur de probabilité de fraude comprise entre 0 et 1. La fusion des modules reporte une fraude si la somme de toutes les valeurs est supérieure ou égale à 1. Par conséquent, la fraude est détectée si l’un ou l’autre des modules est très confiant ou si beaucoup de modules ont une petite valeur. Le bruit dans les données textuelles, venant de la sortie OCR partiellement corrigée, constituait un véritable défi. Une normalisation du texte a été effectuée (suppression des espaces dans les prix, correction des points dans les décimales, etc.) mais ceci pourrait être étendu pour couvrir davantage d’incohérences. Les paramètres ont été réglés manuellement, mais cela pourrait être automatisé à l’avenir.

**Méthode 4** : Cette méthode utilise trois approches pour détecter les images falsifiées :

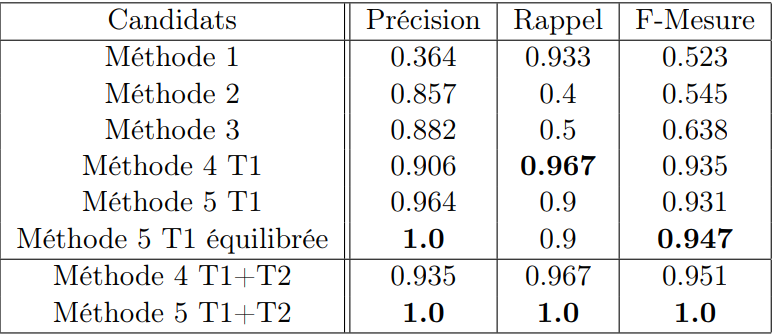
* Détection des fraudes de copier-coller et de dessin sur l’image par des techniques de zones denses (dense-field techniques) (Cozzolino et al. 2015)
* Empreinte de bruit : il s’agit d’extraire la signature de la caméra à travers un réseau profond qui enlève le contenu haut-niveau de l’image (Cozzolino & Verdoliva 2018b,a). Si une image a été trafiquée, une anomalie peut être découverte en comparant l’empreinte de bruit de cette image avec l’empreinte de référence extraite d’un ensemble d’images authentiques.
* Indices de stéganalyse : cette approche, proposée par Cozzolino et al. (2014), détecte les falsifications grâce à des indices locaux de l’image et à de la classification par des machines à vecteurs de support (SVM) linéaires. Les indices locaux, proposés à l’origine dans des travaux de stéganalyse par Fridrich & Kodovsky (2012), captent des micro-patrons expressifs dans l’image sur laquelle a été appliquée un filtre passe-haut.

**Méthode 5 :** Cette approche est basée sur les caractéristiques stéganographiques extraites à partir de l’image entière et utilisées pour former un ensemble de classificateurs SVM pour faire la distinction entre les images altérées et non altérées. Fridrich & Kodovsky (2012) ont présenté un ensemble de 39 filtres stéganographiques . Cozzolino et al. (2014) appliquent un ensemble de ces filtres sur l’image, et un descripteur de matrice de co-occurrences est formé pour l’image filtrée en entier. Chaque filtre est évalué par validation croisée, et les caractéristiques produites par les filtres les plus performants sont concaténées en un classificateur final. La méthode 5 suit une approche similaire : une validation croisée est effectuée sur le corpus d’entraînement pour trouver les filtres de Fridrich & Kodovsky (2012) les plus performants pour le corpus, mais ici, les classificateurs individuels sont entraînés pour chaque indice. Le résultat final est obtenu par un vote majoritaire sur toutes les sorties du classificateur. Les filtres stéganographiques de Fridrich & Kodovsky (2012) qui ont démontré les meilleures performances dans la validation croisée sont :

* S5x5 spam14hv q1
* S5x5 minmax22v q1
* S3x3 minmax22v q1
* S3x3 minmax24 q1
* S3 spam14hv q1
* S3 minmax34v q1
* S3 minmax22v q1
* S2 spam12hv q1
* S1 spam14hv q1

Chaque modèle est entraîné en utilisant bagging. La sortie de chaque modèle est calculée par la moyenne de toutes les sorties de sacs, et le résultat final est issu d’un vote majoritaire sur l’ensemble des modèles.

Voici un tableau qui résume les résultats des différentes méthodes qui ont été proposées :



On peut voir que la première méthode détecte la plupart des bons documents (très bon rappel) mais classifie également comme faux beaucoup de documents authentiques (mauvaise précision), contrairement a la deuxième méthode qui ne cherche et trouve que les documents qui contiennent des fraudes de type duplication de contenu au sein d’un document.

Le dernier résultat du tableau montre un score de détection parfaite : la méthode utilisée trouve parfaitement les 30 documents frauduleux . Ce résultat surprenant est surement dû au fait que le corpus est très spécialisé. En effet, les documents ont tous été numérisée par la même caméra, avec des paramètres presque identiques. Il serait donc intéressant de voir si cette méthode permet d’obtenir des scores équivalents sur un corpus composé d’images provenant de différents appareils photographiques dans différentes conditions d’éclairage et d’inclinaison.

**Leur approche**

Donc jusqu’ici nous avons vu toutes les approches proposées par les différentes personnes voyons ce qu’eux ils ont utilisé.

Ils ont d’abord coupé en trois les approches, en premier nous auront donc la vérification interne au document, puis la vérification entre les documents d’un même corpus, et la vérification externe. D’abord il cherche à vérifier que les informations contenues dans un même document sont cohérentes les unes avec les autres, en effet les fraudeurs ne sont pas toujours rigoureux et oubliais régulièrement de modifier toutes les informations au sein du corpus, c’est-à-dire le fait que les informations d’un document ne sont pas aberrantes par rapport a celles des autres documents. Par exemple nous pouvons vérifier qu’un produit a toujours le même prix, ou que la variation est minime. Pour finir, nous proposons de chercher à estimer la vraisemblance des informations en les comparant a des informations externes au corpus.

Ils proposent donc une liste de 7 indices, qui sont tous binaires ils retournent 0 si la condition est satisfaite et 1 si la condition ne l’est pas.

**Somme des prix = total** Nous calculons d’abord la somme des prix de chaque produit du ticket et nous la comparons au montant total que nous avons extrait. En effet, il est fréquent dans notre corpus que les fraudeurs aient oublié de modifier les totaux en conséquence du changement des prix des produits.

**Somme d’articles = Nombre articles** Le deuxième indice porte sur le nombre extrait d’articles, comparé au nombre calculé de produits extraits. Il est possible que les fraudeurs ne fassent pas attention à ce genre de détails quand ils suppriment ou ajoutent un produit. La suppression de produit peut par exemple avoir lieu dans le cas de remboursement de frais de mission, où il n’est pas forcément bien vu de prendre certaines boissons alcoolisées. Cette information n’est cependant pas relevée sur tous les tickets de caisse, ce qui implique que cet indice ne pourra être pertinent que pour les tickets Carrefour.

**Total = paiement** Le troisième indice concerne le montant payé extrait et le montant total extrait. En effet, nous avons remarqué dans notre corpus qu’il est fréquent que les fraudeurs oublient de reporter le montant total fraudé sur le montant payé, probablement parce que les abréviations utilisées rendent difficile l’identification de cette information pour les profanes, qui ne voient pas dans des « CB EMV », « TR » et autres « ESP » la signification que notre outil, lui, peut relever.

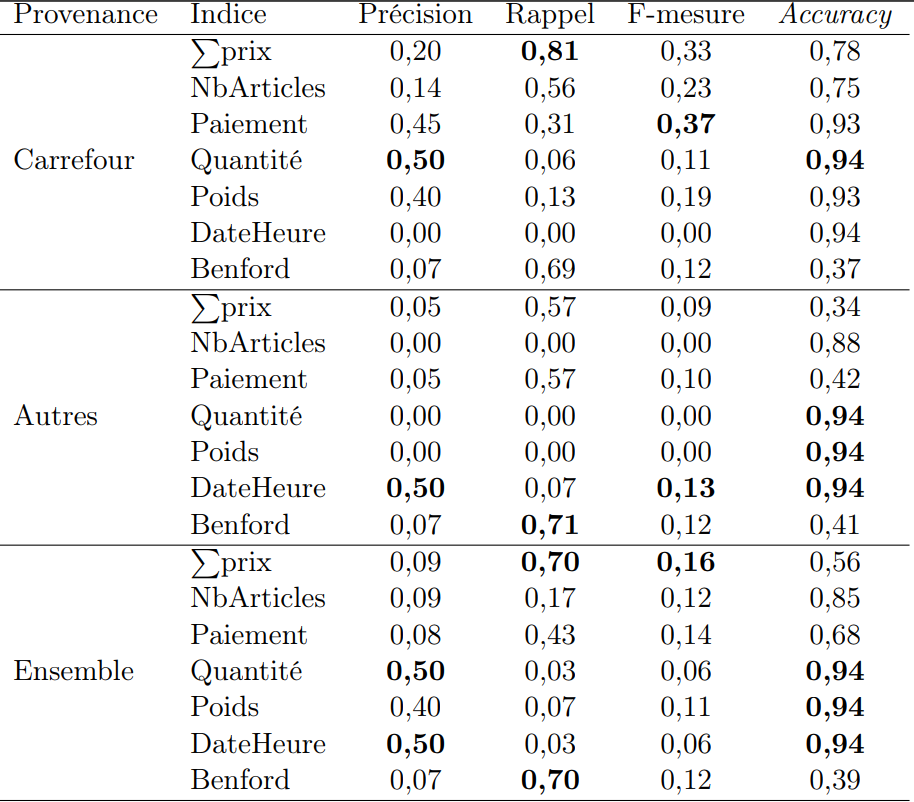
**Quantité × prix unitaire = prix** Sur les tickets de Carrefour, nous relevons la quantité et le prix unitaire des produits. Certaines fraudes portent sur ces informations et nous cherchons donc à vérifier que les fraudeurs n’ont pas fait d’erreurs dans la multiplication de ces deux informations : le prix total du produit doit en effet correspondre au résultat de cette multiplication.

**Poids × prix au kilogramme = prix** Nous vérifions également, dans le cinquième indice, l’égalité du résultat de la multiplication du poids d’un produit par son prix au kilogramme et du prix total, quand ces informations existent et sont relevées.

**Date et heure** Le sixième indice porte sur la vérification du format de la date et de l’heure : si une date ou une heure ne correspondent pas à une date ou à une heure possible, une propriété est\_suspect est ajoutée à l’ontologie pour le concept TicketCaisse lors de l’extraction. Par exemple si le numéro du mois est supérieur à 12, ou si le fraudeur a oublié qu’il n’y a que 28 ou 29 jours en février, la date ne peut pas être entrée dans l’ontologie car elle ne correspond pas au format attendu. Elle est donc rentrée en tant que chaîne de caractères.

**Loi de Benford** Le dernier indice cherche à vérifier la « loi de Benford » sur les prix des produits. La loi généralisée de Benford établit que la distribution de certains chiffres dans de nombreuses séquences de chiffres de la vie réelle ne suit pas une distribution uniforme. Cela signifie que, dans un ensemble de données sur n’importe quel domaine (comptabilité, démographie, presse, articles scientifiques...), il est fréquent qu’un chiffre soit sur-représenté par rapport aux autres (Nigrini 2012). D’après Durtschi et al. (2004), cette loi est souvent utilisée pour détecter les fraudes, avec plus ou moins de résultats, lors d’audits comptables. Le constat est que la répartition du premier chiffre significatif d’un nombre est logarithmique : il y a ainsi plus de nombres commençant par 1 que par 2, plus par 2 que par 3... Nous avons donc testé cette loi sur chacun des tickets de notre corpus comme septième indice, ainsi que sur l’ensemble de notre corpus (voir l’annexe C).

Selon ce tableau on peut voir que seulement la loi de Benford, somme des prix et paiement.



Un récapitulatif est disponible sur le mémoire qui a été conçu de la page 167 à 169.