Apprentissage automatique avec peu d'exemples pour l'extraction du contenu des documents non structurés

M. KANDI, L. NICOLAIEFF, Y. ZEGAOUI, C. BORTOLASO

Berger-Levrault, 64 Rue Jean Rostand, 31670 Labège, France

{mohamed.kandi, lina.nicolaieff,younes.zegaoui,christophe.bortolaso}@berger-levrault.com

Résumé

De nos jours, les entreprises déploient des mécanismes complexes pour automatiser la collecte, le stockage et le traitement de données. Néanmoins, certaines données sont sous un format non structuré : factures, bons de commande, ordonnances, etc. Il est possible de construire un modèle entraîné sur des exemples annotés pour extraire le contenu utile de ces documents. Cependant, dans de nombreux cas, il y a beaucoup de variations dans les types de documents. L'annotation d'exemples est une tâche fastidieuse et répétitive effectuée régulièrement lorsque de nouveaux types de documents arrivent. Pour minimiser ce travail de supervision, nous présentons dans ce papier une méthode permettant de sélectionner un sous-ensemble pertinent de documents non structurés à annoter. Pour évaluer la méthode, nous avons entraîné un modèle de type Faster R-CNN avec cinq jeux de données différents. Nous avons comparé les performances avec différents types de documents et taille de jeux de données. Nous montrons qu'un choix pertinent et automatisé d'exemples de documents peut éviter un effort considérable d'annotation.

Mots-clés

Détection et extraction de contenu, apprentissage avec peu d'exemples, Faster R-CNN, Triplet-loss.

Abstract

Nowadays, companies deploy complex mechanisms to automate data collection, storage, and processing. Some of this data is in an unstructured format : invoices, medical prescriptions... It is possible to build a model trained on annotated examples to extract useful information from these documents. However, in many cases, there is a lot of variation in document templates. Annotation is a tedious and repetitive task done regularly when new document templates arrive. We present in this paper a method to select a small and relevant subset of unstructured documents to annotate. To evaluate the method, we trained a model with five datasets. We compared the performance with different choices of document templates and dataset size. We show that a relevant and automated choice of document examples can avoid a huge annotation effort.

Keywords

Content localization and extraction, Few-shot learning, Faster R-CNN, Triplet-loss.

1 Introduction

Les données constituent un élément clé de la prise de décision. De nos jours, de nombreuses entreprises déploient des processus métier complexes pour automatiser la collecte, le stockage et le traitement d'une énorme quantité de données. Malheureusement, certaines de ces données ont un format non structuré : factures, emails, devis, bons de commande, tickets, documents scannés, cartes d'identité, ordonnances, etc. Cette représentation non structurée est difficile à exploiter par la machine, ce qui complique l'automatisation des processus métier et reporte un effort considérable sur les agents administratifs qui doivent ressaisir les informations dans les logiciels de gestion.

Le traitement intelligent de documents (Intelligent Document Processing, IDP) est un ensemble de moyens, méthodes et technologies permettent de capturer, extraire et traiter des données à partir de nombreux formats de documents [1]. Avec le traitement intelligent de documents, il est possible de transformer des données non-exploitables en données structurées facilement manipulables par un processus métier automatisé. Un framework IDP utilise des techniques d'analyse d'image, de traitement du langage naturel et d'apprentissage automatique profond pour remplir cette tâche. Ces derniers ont connu un grand succès ces dernières années, grâce à une grande quantité de données générées, à la disponibilité de capacités de calcul à la demande à des coûts raisonnables, et aux méthodes et architectures neuronales fournies par la communauté scientifique.

Les applications du traitement intelligent des documents sont nombreuses. Dans ce travail, nous prenons, comme cas d'usage, le contrôle de conformité des factures générées par les logiciels de gestion financière. En France, les factures générées au niveau des municipalités sont envoyées à une autorité centrale, qui les imprime et les envoie aux entités concernées. La réglementation impose de nombreuses règles concernant la forme et le contenu des factures. Par exemple, l'adresse de l'expéditeur et celle du destinataire doivent figurer dans des cases bien définies, certaines zones doivent être vides, le logo ne doit pas chevaucher sur les

marges, etc. Un document non-conforme est rejeté. Dans le cas contraire, cela entraîne des conséquences négatives pour de nombreuses personnes, comme le fait de ne pas recevoir un document à cause d'une adresse illisible, tronquée ou mal positionnée.

La réglementation évolue régulièrement, les éditeurs de logiciels de gestion financière doivent donc s'adapter rapidement aux changements. Pour cela, il est crucial de contrôler les factures et de détecter automatiquement les éléments pertinents (tels que l'adresse de l'expéditeur, l'adresse du destinataire et le logo) et d'extraire le contenu. Dans les travaux existants, il existe deux approches pour y parvenir [2] : le pattern-matching et l'apprentissage automatique. La première approche nécessite la mise en place de règles de correspondance qui sont difficiles à maintenir. De plus, cette approche ne se généralise pas. L'objectif d'un framework IDP est d'adopter une solution qui pourrait être généralisée à d'autres types de documents pour de futurs cas d'usage. C'est pourquoi nous avons décidé de baser notre travail sur les approches d'apprentissage automatique.

L'un des principaux inconvénients des méthodes d'apprentissage automatique tient dans le besoin de documents annotés pour entraîner le modèle. L'annotation est une tâche fastidieuse et coûteuse. Nous appliquons des techniques d'apprentissage avec peu d'exemples (Few-Shot Learning ou *FSL*) pour réduire l'effort d'annotation. Pour sélectionner un sous-ensemble suffisant de documents pertinents, nous proposons une méthode basée sur le calcul d'embeddings avec un modèle Triplet-Loss [1] [3, 4, 5], puis un clustering avec une méthode de k-means. Lorsque les documents similaires se retrouvent dans le même cluster, nous considérons qu'il n'est pas pertinent de les annoter tous. Nous construisons notre jeu de données avec seulement quelques exemples de chaque cluster. Nous annotons ce jeu de données puis nous entraînons un modèle suivant une architecture Faster R-CNN [6] pour détecter les éléments pertinents dans les documents.

Ce papier est organisé comme suit. D'abord, nous donnons un aperçu des travaux existants et nous positionnons notre travail dans la section 2. Puis, nous présentons la description du système proposé dans la section 3. Ensuite, nous détaillons les expériences réalisées pour évaluer notre contribution dans la section 4. Enfin, nous concluons le papier et donnons nos perspectives de recherche dans la section 5.

2 Travaux antérieurs

Il existe deux approches pour extraire des données de documents non structurés [2] : le pattern-matching (sous-section 2.1) et l'apprentissage automatique (sous-section 2.2). L'approche de l'apprentissage automatique s'appuie sur l'apprentissage avec peu d'exemples pour construire des modèles efficaces avec peu d'effort d'annotation (sous-section 2.3). Pour positionner notre travail, nous présentons, dans ce qui suit, le principe et les limites des méthodes existantes.

1. https ://www.tensorflow.org/addons/tutorials/losses_triplet

2.1 Méthodes orientées pattern-matching

Ces méthodes consistent à identifier des motifs dans les documents et à les utiliser pour extraire des informations [7, 8]. Plusieurs types de documents sont prédéfinis, et le but est de vérifier dans quelle mesure les documents sources correspondent aux modèles cibles. Cependant, la création et la maintenance des types demandent du temps et de l'expertise. De plus, ces méthodes ne fonctionnent pas bien dans le cas de petites différences entre les documents sources et les modèles, qui sont difficiles à interpréter pour la méthode. Par exemple, si nous avons plusieurs formats de facture avec des éléments éventuellement mal placés, il pourrait être difficile de déterminer si un désalignement avec un modèle est dû au fait que le document n'appartient pas à ce modèle ou aux éléments mal placés. Une solution rapide serait d'ajouter des règles qui peuvent être évaluées à tout moment pour vérifier si chaque élément est bien placé. Néanmoins, l'établissement de toutes les règles possibles serait fastidieux et déraisonnable pour une approche de type pattern-matching.

2.2 Méthodes basées sur l'apprentissage automatique

Une autre approche consiste à utiliser l'apprentissage automatique. Il s'agit d'entraîner un modèle avec un ensemble d'exemples annotés. Certains travaux considèrent la tâche comme une classification de mots. Pour chaque mot du document, nous décidons de l'extraire ou non. Si nous devons détecter plusieurs éléments, la tâche devient une classification multi-classes. Ces travaux ont opté, pour résoudre le problème, soit des modèles classiques d'apprentissage automatique comme les SVMs [9], soit avec des réseaux de neurones [10].

Nous pouvons également considérer les documents comme des images. Dans ce cas, il est possible de tirer parti des architectures *CNN* bien établies pour la détection d'objets : YOLO [11], Single Shot MultiBox Detector [12], Fast R-CNN [13], Faster R-CNN [6], Feature Pyramid Networks [14], etc.

L'un des principaux inconvénients des méthodes d'apprentissage automatique c'est la nécessité de disposer de nombreux exemples annotés pour entraîner le modèle. L'apprentissage avec peu d'exemples est un domaine de recherche qui vise à résoudre ce problème.

2.3 Apprentissage avec peu d'exemples : Few-shot learning

L'annotation manuelle est une tâche coûteuse, et il est difficile de créer un grand ensemble de données annotées. Nous devons sélectionner le plus petit sous-ensemble possible de documents tout en étant pertinents et variés en termes d'exemples.

Les travaux existants sur l'apprentissage avec peu d'exemples se répartissent en trois grandes catégories [15]. La première catégorie se concentre sur les données. Il s'agit de commencer par l'ensemble des données disponibles. Ensuite, on utilise les connaissances antérieures pour appliquer des transformations qui génèrent un ensemble de don-

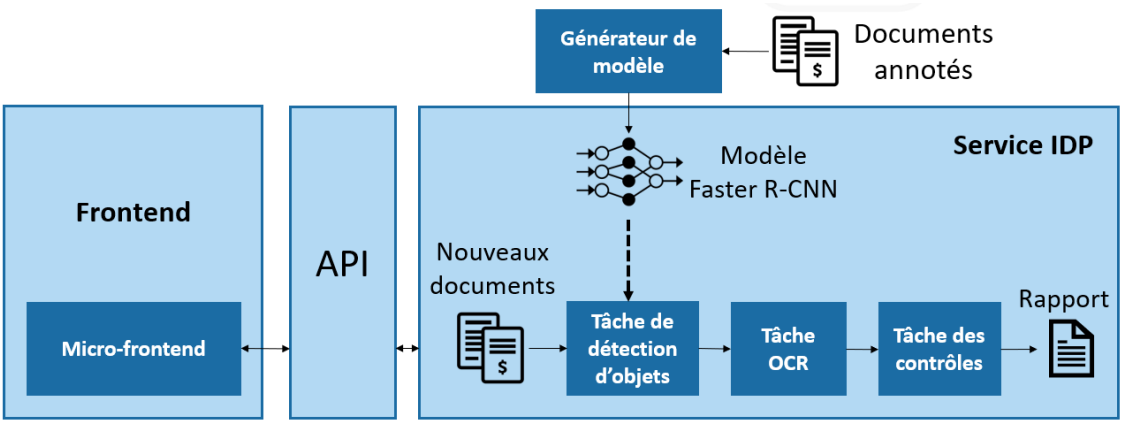


FIGURE 1 – Une vue globale du framework de traitement intelligent des documents

nées plus important. La deuxième catégorie se concentre sur le modèle. Il s'agit d'utiliser les connaissances préalables pour réduire l'espace des hypothèses et limiter la complexité du problème à résoudre. On peut résoudre ce dernier avec un jeu de données qui n'est pas volumineux. La troisième catégorie se concentre sur l'algorithme. Il s'agit d'utiliser les connaissances préalables pour modifier la stratégie de recherche des meilleurs paramètres du modèle : en donnant une bonne initialisation ou/et en guidant les étapes de la recherche.

La première catégorie de travaux applique des transformations pour augmenter les données. Nous pouvons faire l'augmentation avec une liste de règles faites à la main. Des exemples typiques sont l'augmentation d'images avec des traductions [16], des recadrages [17], des rognages [18], des rotations [18] etc. Cependant, les règles sont souvent spécifiques à l'ensemble de données disponible et sont difficilement applicables à d'autres ensembles de données. Par conséquent, l'augmentation des données ne nous permet pas de résoudre complètement le problème d'apprentissage avec peu exemples.

La deuxième catégorie de travaux tente de réduire l'espace des hypothèses et de limiter la complexité du problème. Nous pouvons classer les méthodes appartenant à cette catégorie en deux techniques [15] : (1) l'apprentissage multi-tâches et (2) l'apprentissage par embeddings. Lorsque nous avons plusieurs tâches liées, l'apprentissage multi-tâches permet d'apprendre plusieurs tâches simultanément en exploitant à la fois les informations génériques communes et les informations spécifiques à la tâche. Cette technique suppose que certaines tâches n'ont que quelques exemples alors que d'autres en ont beaucoup. Les paramètres à apprendre pour chaque tâche dépendent des autres tâches. Nous pouvons le faire avec le partage [19, 20] ou la fixation des paramètres [21, 22]. L'inconvénient de l'apprentissage multi-tâches est qu'il suppose l'existence de plusieurs tâches similaires, certaines avec de nombreux exemples. Ce qui n'est pas toujours le cas. Souvent, nous nous trouvons dans un cas où nous considérons une seule tâche ou un ensemble de tâches avec peu d'exemples. Il faut également noter que l'apprentissage simultané de toutes les tâches est nécessaire. Lorsqu'une nouvelle tâche arrive, l'ensemble du modèle multi-tâches doit être réentraîné, ce qui est coûteux et lent.

Avec l'apprentissage basé sur les embeddings, nous représentons chaque document dans un espace de plus petite dimension. Dans cet espace, les documents similaires sont proches, tandis que les documents différents sont éloignés. La fonction de transformation est entraînée sur des connaissances antérieures [23], ou avec des connaissances spécifiques de la tâche à accomplir [24, 25]. Il peut également être entraîné sur une combinaison des deux [26, 27, 28, 29]. Nous ne pouvons utiliser les méthodes d'embeddings que si nous disposons d'un grand ensemble de données, contenant suffisamment d'exemples de différentes classes génériques de connaissances préalables, ou d'un modèle pré-entraîné. Malheureusement, ce n'est pas toujours le cas. En outre, l'efficacité de ces méthodes est incertaine si les données spécifiques à la tâche ne sont pas liées aux données génériques. Enfin, la manière de combiner les informations génériques et spécifiques dépend de la nature des données, et il n'existe pas de stratégie bien établie.

La troisième catégorie de travaux utilise les connaissances préalables pour influencer l'algorithme d'exploration des paramètres du modèle. Il existe trois techniques [15] : (1) choisir des paramètres initiaux endossés à l'aide de données provenant d'autres tâches, puis affiner avec les données de la tâche cible [30, 31, 32, 33], (2) choisir des paramètres initiaux approuvés par un méta-algorithme à partir d'un ensemble de tâches qui ont la même distribution que la tâche cible [34], (3) trouver des méta-algorithmes pour guider intelligemment la direction de recherche ou l'étape d'itération en fonction des connaissances préalables [35]. La première technique est utile pour accélérer l'apprentissage du modèle, mais elle sacrifie la précision. Or, la précision est un objectif clé de nos cas d'utilisation. Les deuxième et troisième techniques présentent plusieurs problèmes ouverts et non résolus dans l'état de l'art : en particulier, comment gérer différentes granularités, comme la classification de documents au sens large, par opposition à la classification de modèles de factures, ou différentes sources de données, comme les images par opposition aux textes.

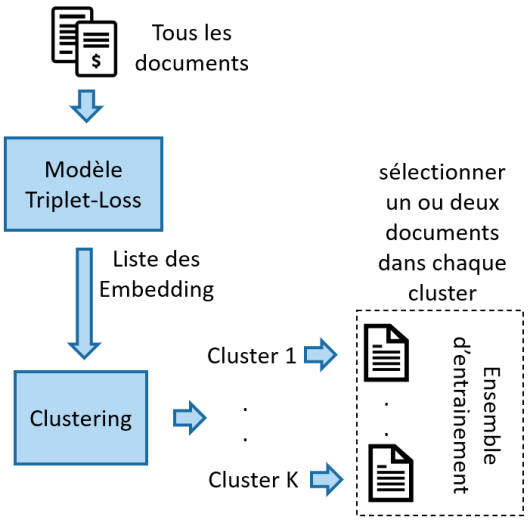


FIGURE 2 – Sélection de documents avec la Triplet-loss et le clustering

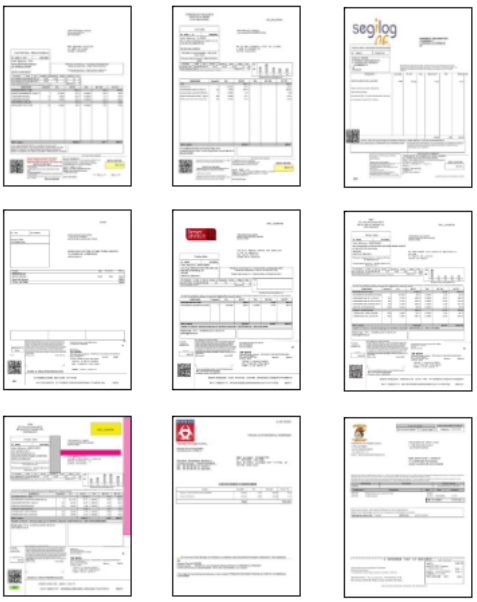


FIGURE 3 – Types de factures

3 Description du système et méthodologie

Dans nos travaux, nous considérons les documents en entrée comme des image et nous adoptons un réseau *CNN* détecteur d'objets standard. Pour entraîner (fine-tuning) ce modèle, nous avons besoin d'un ensemble de documents annotés. Nous montrons dans la figure 1, l'architecture de notre framework IDP. Celui-ci se présente sous la forme d'une API. Les clients peuvent soumettre de nouveaux documents sous forme de flux. Ces documents sont soumis à un module qui détecte et localise les éléments pertinents. Dans notre cas, nous avons limité l'étude à la détection et localisation de : (1) l'adresse expéditeur, (2) l'adresse destinataire, (3) le logo et (4) la datamatrix (QRCode contenant l'ensemble des informations de la facture). Une fois l'objet détecté, nous utilisons une méthode OCR pour extraire le texte qu'il contient. L'avantage est que le framework peut être généralisé pour de nombreux modèles de documents avec des jeux de données annotés. Enfin, le flux passe par une liste de contrôles qui permettant de vérifier le respect de la réglementation. Un rapport est généré qui contient la liste des éléments, leur emplacement dans le document et leur valeur. Le rapport contient également les résultats des contrôles. Pour les contrôles non respectés, on peut voir un commentaire qui aide à la correction.
La figure 4 montre un exemple de détection et extraction de contenu à partir d'une facture. Dans le résultat affiché, nous pouvons voir les éléments détectés, leur score de confiance, ainsi qu'un tableau des contrôles de conformité.

3.1 Détection d'éléments pertinents

Le module de détection d'éléments est basé sur un modèle *CNN* de détection d'objets et a pour objectif de produire plusieurs régions, ou imagettes, à partir de l'image du document en entrée. Ces régions doivent être centrées le plus possible sur les éléments d'intérêt du document. Les régions sont ensuite transmises en entrée à une méthode OCR pour extraire le contenu textuel. Les modèles *CNN* détecteur d'objets existants se répartissent en deux catégories : les méthodes en deux étapes et les détecteurs en un coup (single shot detector). Alors que les premiers sont censés produire des résultats plus robustes, les seconds peuvent traiter les images plus rapidement et réaliser une détection d'objets en temps réel. Dans notre cas, le temps réel n'est pas une contrainte car nous souhaitons généralement traiter les documents par lots. Nous avons alors décidé d'utiliser l'architecture Faster R-CNN [13] car c'est le détecteur à deux étapes le plus largement utilisé.

Le modèle Faster R-CNN [13] est composé de 2 modules :

— le premier, appelé le réseau de proposition de régions, traite l'image d'entrée et produit plusieurs régions d'intérêt avec différentes tailles et proportions, appelées propositions ;
— le second module prend en entrée ces propositions et vise à les ajuster au mieux autour de l'objet qu'elles contiennent, à l'aide d'une fonction de régression, ainsi qu'à leur attribuer une classe, à l'aide d'une fonction de classification classique.

En raison de la présence du premier module, nous nous attendons à ce que le modèle ait une meilleure précision par rapport aux détecteurs en un coup tout en obtenant des valeurs de rappel similaires. De plus, pour obtenir de meilleures performances, nous avons utilisé un modèle pré-entraîné sur le célèbre corpus COCO (Common Objects in Context) [36]. Il nous est possible d'obtenir un grand nombre de documents de factures, cependant le coût d'annotations serait trop important. Il est alors nécessaire de disposer d'un moyen pour sélectionner un petit sous-ensemble de documents aussi varié que possible afin d'optimiser au mieux le travail d'annotations.

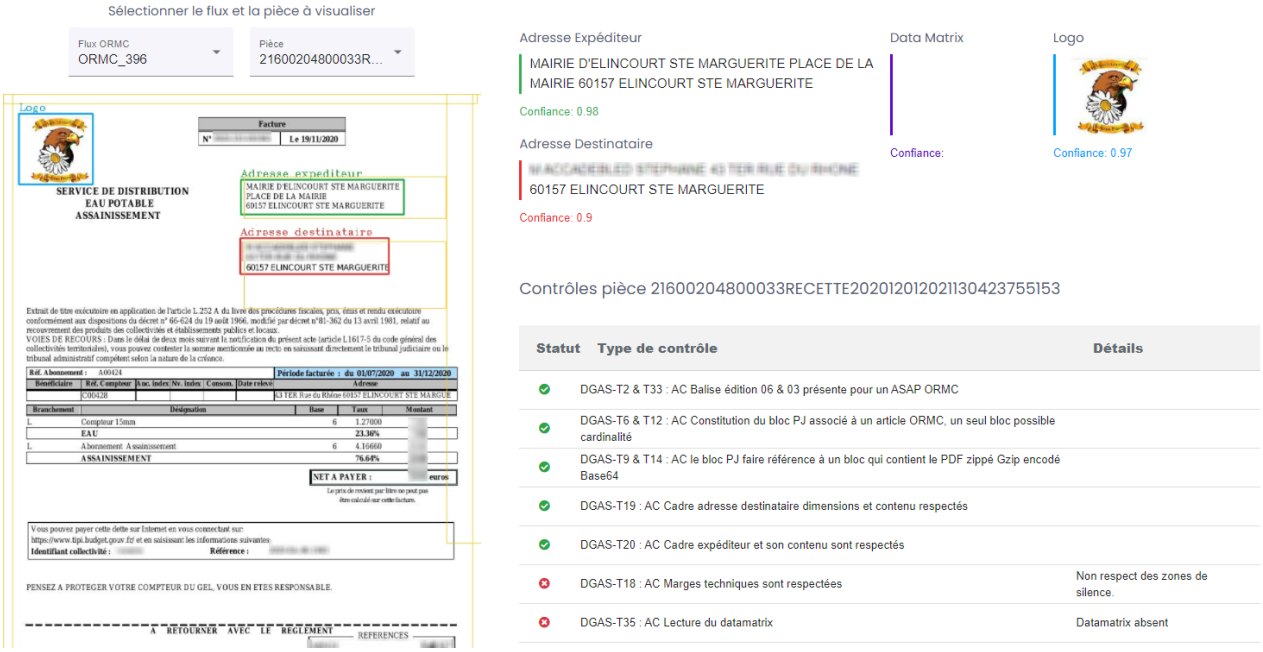


FIGURE 4 – Exemple d'extraction d'extraction de contenu d'une facture

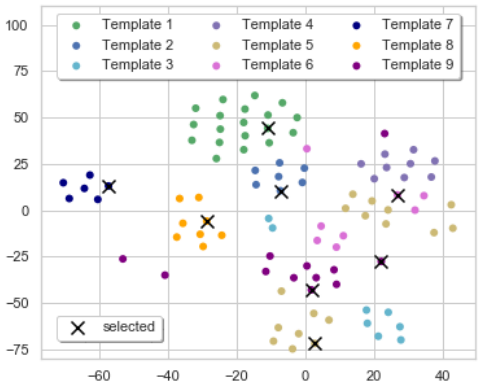
3.2 Sélection des meilleurs candidats pour l'entraînement

Bien qu'il puisse être coûteux en temps d'annoter chaque document à notre disposition avec des boîtes englobantes autour des éléments d'intérêt, il est intéressant de regrouper les documents en fonction des similarités de leurs structure graphique (template). Étant donné que les documents de chaque groupe sont très similaires, notre objectif est de sélectionner les exemples qui capturent le mieux les caractéristiques sous-jacentes du modèle afin que le réseau puisse en tirer parti lors de l'entraînement sans avoir à traiter de nombreux documents quasi-identiques. La figure 2 illustre le processus de sélection des documents candidats. Tout d'abord, nous utilisons un modèle Triplet-loss [37] pour projeter les documents dans un espace vectoriel. Le but du modèle est que les vecteurs embeddings associés aux documents d'un même template soient proches les uns des autres dans l'espace latent tout en étant éloignés des documents des autres templates. Le réseau *CNN* est entraîné comme tel : pour chaque image en entrée, appelée ancre, deux autres images sont passées en entrée au modèle. L'une est l'exemple positif, appartenant au même template que l'ancre, l'autre étant l'exemple négatif, appartenant à un template différent. La fonction de coût calculée minimise la distance dans l'espace latent entre l'ancre et l'exemple positif tout en maximisant la distance entre l'ancre et l'exemple négatif. Une fois le réseau entraîné, nous calculons un vecteur embedding pour chaque document de notre ensemble de données et utilisons l'algorithme du k-means pour regrouper ceux-ci en clusters. Nous pouvons alors vérifier à quel point les clusters obtenus recoupent la répartition des documents en templates. Il est ainsi utile de montrer à quel point nous pouvons utiliser le même espace latent afin d'y projeter des documents appartenant à template jamais vu sans avoir à entraîner de nouveau le modèle Triplet-loss. Enfin, nous sélectionnons pour chaque cluster des documents candidats. Les documents sélectionnés sont alors annotés manuellement et constituent notre ensemble d'apprentissage pour le modèle de détection d'éléments Faster R-CNN. Pour un cluster donné, nous pouvons choisir le document le plus proche du centroïde ou la paire de documents qui ont une distance maximale entre eux.

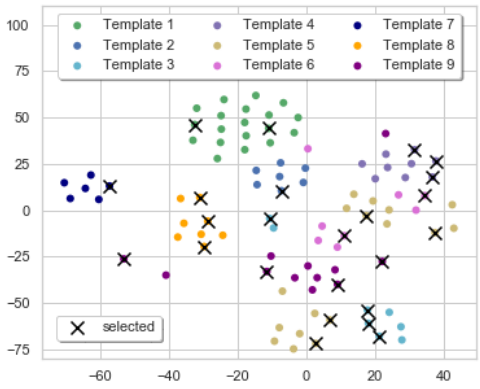
4 Experimentations

Pour évaluer notre approche, nous avons dans un premier temps pré-entraîné un modèle Triplet-loss avec le jeu de données RVL-CDIP [2] [38]. Puis dans un second temps, nous avons affiné le modèle avec un ensemble de données spécifique à notre cas d'utilisation (27 factures issues de 9 types de structures différentes). La figure 3 montre un exemple de chaque type de facture. Puis, nous avons calculé la représentation d'un ensemble de test contenant 87 nouvelles factures, avec le modèle Triplet-loss déjà entraîné. Enfin, nous avons exécuté un algorithme de clustering K-means sur les documents de l'ensemble de test. Nous avons fait varier la stratégie de sélection de documents (le document le plus proche du centroïde par rapport à la paire de documents présentant la distance maximale entre eux) et le nombre de documents sélectionnés (8, 16 et 24 documents). Pour visualiser le résultat, nous avons projeté les représentations sur un espace bi-dimensionnel avec T-SNE [39]. La figure 6 montre les résultats de cette méthode. Chaque point correspond à une facture et les couleurs permettent de distinguer les types de facture. On constate que les documents appartenant à un même type sont, dans la plupart des cas, proches dans l'espace des représentations. Chaque docu-

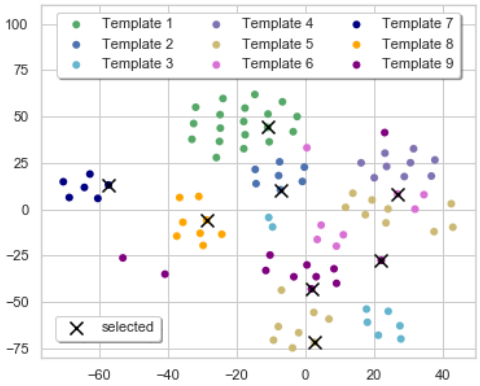
2. https ://www.cs.cmu.edu/ aharley/rvl-cdip/



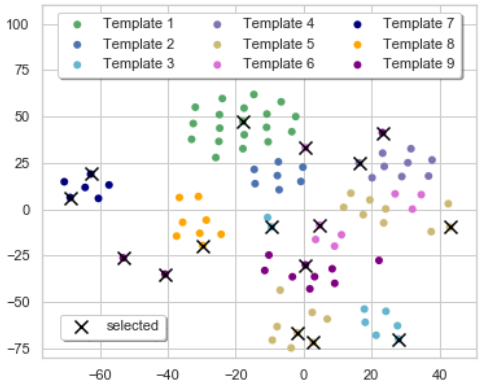
(a) Document le plus proche du centroïde - distance euclidienne - 8 documents sélectionnés



(b) Document le plus proche du centroïde - distance euclidienne - 24 documents sélectionnés

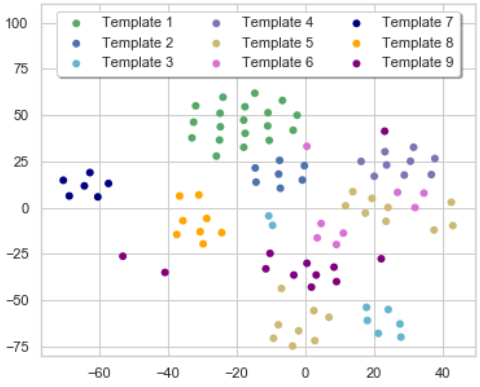


(c) Document le plus proche du centroïde - similarité cosinus - 8 documents sélectionnés

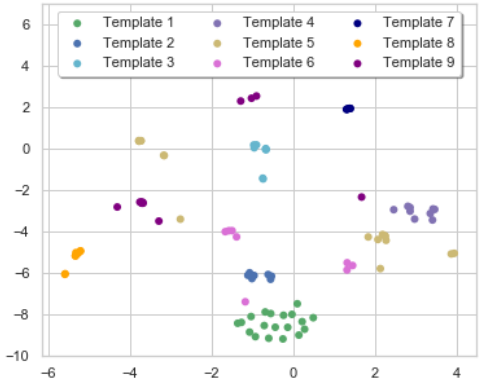


(d) Paire de documents avec distance maximale - distance euclidienne - 16 documents sélectionnés

FIGURE 5 – Projection des vecteurs de documents par T-SNE selon la fonction de distance adoptée (cosinus ou euclidienne), le nombre de documents sélectionnés et le choix des documents (plus proche du centroïde ou paire la plus éloignée).



(a) 1 epoch



(b) 5 epochs

FIGURE 6 – Projection des vecteurs embeddings à l'aide du T-SNE : après 1 epoch d'entraînement et après 5 epochs.

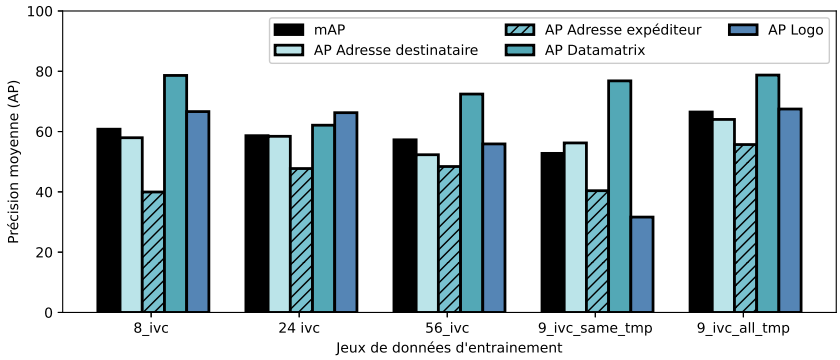


FIGURE 7 – Résultat de la précision moyenne (AP) du modèle de détection d'éléments Faster R-CNN

Entraînement	**mAP**	AP Destinataire	AP Expéditeur	AP Datamatrix	AP Logo
8_ivc	**60.789**	57.954	39.953	78.620	66.627
24_ivc	**58.634**	58.428	47.734	62.111	66.264
56_ivc	**57.267**	52.315	48.406	72.444	55.901
$9\_ivc\_same\_tmp$	**52.766**	56.230	40.383	76.818	31.634
$9\_ivc\_all\_tmp$	**66.486**	64.025	55.696	78.742	67.479

TABLE 1 – Résultats détaillés de la précision moyenne (AP)

Entraînement	Adresse destinataire	Adresse expéditeur	Datamatrix	Logo
8_ivc	8	8	6	4
24_ivc	24	24	19	12
56_ivc	56	56	36	38
$9\_ivc\_same\_tmp$	9	9	5	6
$9\_ivc\_all\_tmp$	9	9	4	9

TABLE 2 – Nombre de factures contenant chaque objet dans les jeux d'entrainement

ment d'un type se rapproche des autres dans l'espace après cinq époques d'entrainement en comparaison à une seule époque.

Sur les figures 5, les factures sélectionnées pour l'annotation sont identifiées par une croix. Nous remarquons que la similarité cosinus sélectionne les mêmes documents que la distance euclidienne. Les points sélectionnés sont bien répartis dans l'espace des représentations, ce qui garantit la variété des documents.

D'un autre côté, nous avons évalué la sensibilité du modèle de détection d'objets Faster R-CNN au jeu de données d'entraînement. Pour ce faire, nous avons calculé la performance avec la métrique de précision moyenne (AP) sur plusieurs sous-ensembles en faisant varier le nombre de documents et la diversité des types de structure des documents.

Pour évaluer l'importance de la diversité, nous avons construit cinq jeux d'entraînement :

- Les jeux d'entrainement 8_ivc, 24_ivc, 56_ivc contiennent 8 types de factures. Ce qui les différencie, c'est leur taille. Le premier contient 8 factures, le second 24 documents, et le troisième 56 documents.
- Nous disposons alors d'un ensemble d'apprentissage $9\_ivc\_all\_tmp$ contenant une facture pour chaque type de documents (neuf types).
- Enfin, un ensemble d'apprentissage $9\_ivc\_same\_tmp$ contenant neuf factures du même type.

Notre jeu de test contient des exemples des neuf types de facture. Nous avons calculé la métrique *mAP* sur quatre objets (adresse du destinataire, adresse de l'expéditeur, DataMatrix et logo). Les résultats sont présentés dans la Figure 7 et le Tableau 1. Nous remarquons des scores *mAP* similaires lorsqu'on utilise un sous-ensemble de 8 factures seulement (60,789%) par rapport à un sous-ensemble de 24 factures (58,634%), et qu'il est même légèrement inférieur lorsqu'on utilise 56 factures (57,267%). Nous pensons que cela est dû au fait que la plupart des exemples du même format dans l'ensemble de données sont fortement similaires les uns aux autres, n'apportant donc pas plus d'informations au modèle. Nous notons que les factures ne contiennent pas le même nombre d'objets, ce déséquilibre peut aussi expliquer la diminution du score *mAP* lorsque l'on augmente le nombre de factures (Tableau 2).

Cette hypothèse est accentuée par l'expérience sur la diversité des sous-ensembles : le score *mAP* est le plus élevé (66,486%) lors de l'utilisation d'un sous-ensemble comprenant seulement 9 factures sélectionnées selon notre critère

de diversité. Ces résultats nous amènent à penser que dans le cas spécifique des factures où les données ont tendance à être homogènes, il est préférable de trouver quelques exemples avec des structures graphiques variées plutôt que d'ajouter simplement des exemples aléatoires au jeu de données d'entraînement.

5 Conclusion

Dans cet article, nous avons montré que le modèle basé sur la Triplet-loss combiné au clustering peut être utilisé pour sélectionner un sous-ensemble de documents pertinents pour annoter et former un modèle de location d'objets. Dans des travaux futurs, nous mènerons des expériences sur un plus grand nombre de types de documents. Nous prévoyons également d'utiliser les vecteurs calculés pour surveiller les performances du modèle. Lorsqu'un nouveau document arrive, nous calculons la similarité de sa représentation avec les documents de l'ensemble d'apprentissage courant. Si la similarité est faible, alors nous déclenchons une alerte pour permettre à un validateur de vérifier que l'élément est correctement détecté. Si ce n'est pas le cas, il annote le nouveau document et l'ajoute au jeu de données de la prochaine mise à jour du modèle.
Nous prévoyons également d'étendre notre travail en concevant de nouvelles expériences qui pourraient nous aider à obtenir de meilleurs résultats sur la partie module de détection d'éléments de notre architecture : (1) unifier le modèle Triplet-loss avec le modèle de détecteur *CNN* en leur faisant partager certaines de leurs caractéristiques, (2) comparer le modèle Triplet-loss + k-means avec une approche unifiée de clustering à intégration profonde (*DEC*) [40], (3) aller plus loin dans la direction *FSL* en tirant parti des méthodes existantes telles que les réseaux de correspondance [41] pour aider notre modèle à obtenir le plus d'informations de notre ensemble de données, à la fois annotées et brutes, pendant l'entraînement.

Références

[1] "Grobid," https ://github.com/kermitt2/grobid, 2008–2021.

[2] R. B. Palm, F. Laws, and O. Winther, "Attend, copy, parse end-to-end information extraction from documents," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019, pp. 329–336.

[3] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, "Person re-identification by multi-channel parts-based cnn with improved triplet loss function," in *Proceedings of the iEEE conference on computer vision and pattern recognition*, 2016, pp. 1335–1344.

[4] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification." *Journal of machine learning research*, vol. 10, no. 2, 2009.

[5] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet : A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn : Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, pp. 91–99, 2015.

[7] E. Riloff, "Automatically constructing a dictionary for information extraction tasks," in *Proceedings of the eleventh national conference on Artificial intelligence*, 1993, pp. 811–816.

[8] I. Muslea *et al.*, "Extraction patterns for information extraction tasks : A survey," in *The AAAI-99 workshop on machine learning for information extraction*, vol. 2, no. 2. Orlando Florida, 1999.

[9] Y. Li, K. Bontcheva, and H. Cunningham, "Svm based learning system for information extraction," in *International Workshop on Deterministic and Statistical Methods in Machine Learning*. Springer, 2004, pp. 319–339.

[10] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proceedings of NAACL-HLT*, 2016, pp. 260–270.

[11] J. Redmon and A. Farhadi, "Yolo9000 : better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd : Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[13] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[14] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[15] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples : A survey on few-shot learning," *ACM Computing Surveys (CSUR)*, vol. 53, no. 3, pp. 1–34, 2020.

[16] S. Benaim and L. Wolf, "One-shot unsupervised cross domain translation," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, pp. 2108–2118.

[17] H. Qi, M. Brown, and D. G. Lowe, "Low-shot learning with imprinted weights," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5822–5830.

[18] Y. Zhang, H. Tang, and K. Jia, "Fine-grained visual categorization using meta-learning optimization with

sample selection of auxiliary data," in *Proceedings of the european conference on computer vision (ECCV)*, 2018, pp. 233–248.

[19] S. Motiian, Q. Jones, S. M. Iranmanesh, and G. Doretto, "Few-shot adversarial domain adaptation," *arXiv preprint arXiv :1711.02536*, 2017.

[20] Z. Hu, X. Li, C. Tu, Z. Liu, and M. Sun, "Few-shot charge prediction with discriminative legal attributes," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 487–498.

[21] W. Yan, J. Yap, and G. Mori, "Multi-task transfer methods to improve one-shot learning for multimedia event detection." in *BMVC*, 2015, pp. 37–1.

[22] Z. Luo, Y. Zou, J. Hoffman, and L. Fei-Fei, "Label efficient learning of transferable representations across domains and tasks," *arXiv preprint arXiv :1712.00123*, 2017.

[23] E. Triantafillou, R. Zemel, and R. Urtasun, "Few-shot learning through an information retrieval lens," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 2252–2262.

[24] G. Koch *et al.*, "Siamese neural networks for one-shot image recognition," 2015.

[25] L. Yan, Y. Zheng, and J. Cao, "Few-shot learning for short text classification," *Multimedia Tools and Applications*, vol. 77, no. 22, pp. 29 799–29 810, 2018.

[26] L. Bertinetto, J. Henriques, P. Torr, and A. Vedaldi, "Meta-learning with differentiable closed-form solvers." International Conference on Learning Representations, 2019.

[27] L. Bertinetto, J. F. Henriques, J. Valmadre, P. Torr, and A. Vedaldi, "Learning feed-forward one-shot learners," in *Advances in neural information processing systems*, 2016, pp. 523–531.

[28] B. N. Oreshkin, P. Rodriguez, and A. Lacoste, "Tadam : task dependent adaptive metric for improved few-shot learning," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, pp. 719–729.

[29] F. Zhao, J. Zhao, S. Yan, and J. Feng, "Dynamic conditional networks for few-shot learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 19–35.

[30] S. Ö. Arik, J. Chen, K. Peng, W. Ping, and Y. Zhou, "Neural voice cloning with a few samples," in *NeurIPS*, 2018.

[31] R. Keshari, M. Vatsa, R. Singh, and A. Noore, "Learning structure and strength of cnn filters for small sample size training," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9349–9358.

[32] D. Yoo, H. Fan, V. N. Boddeti, and K. M. Kitani, "Efficient k-shot learning with regularized deep networks," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[33] J. Kozerawski and M. Turk, "Clear : Cumulative learning for one-shot one-class image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3446–3455.

[34] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1126–1135.

[35] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas, "Learning to learn by gradient descent by gradient descent," in *Advances in neural information processing systems*, 2016, pp. 3981–3989.

[36] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco : Common objects in context," in *European Conference on Computer Vision*, 2014, pp. 740–755.

[37] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Similarity-Based Pattern Recognition*, A. Feragen, M. Pelillo, and M. Loog, Eds. Cham : Springer International Publishing, 2015, pp. 84–92.

[38] A. W. Harley, A. Ufkes, and K. G. Derpanis, "Evaluation of deep convolutional nets for document image classification and retrieval," in *International Conference on Document Analysis and Recognition (ICDAR)*, 2015.

[39] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[40] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ser. ICML'16. JMLR.org, 2016, p. 478–487.

[41] O. Vinyals, C. Blundell, T. Lillicrap, k. kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016.