Extraction de séquences numériques dans des courriers manuscrits

Numerical Sequence Extraction in Handwritten Mail Documents

G. Koch, L. Heutte and T. Paquet
Laboratoire PSI, Université de Rouen, FRANCE
Guillaume.Koch@univ-rouen.fr
Laurent.Heutte@univ-rouen.fr
Thierry.Paquet@univ-rouen.fr

Résumé
Dans cet article, nous proposons une méthode permettant d'extraire automatiquement des champs numériques dans des documents manuscrits. L'approche exploite la syntaxe connue des champs numériques à extraire ainsi que des caractéristiques morphologiques contextuelles afin de déterminer la meilleure étiquette pour chaque composante connexe. La localisation/extraction des champs d'intérêt est réalisée en appliquant un analyseur syntaxique à base de modèle de Markov sur l'ensemble du document. Les résultats obtenus lors de l'extraction de codes postaux, de numéros de téléphone et de codes clients démontrent l'intérêt de l'approche proposée.

Mots-clef
Analyse de documents, extraction de connaissances, traitement d'image, écriture manuscrite

Abstract
In this communication, we propose a method for the automatic extraction of numerical fields in handwritten documents. The approach exploits the known syntactic structure of the numerical field to extract, combined with a set of contextual morphological features to find the best label for each connected component. Applying a Markov model based syntactic analyzer on the overall document allows to localize/extract fields of interest. Reported results on the extraction of zip codes, phone numbers and customer codes from handwritten incoming mail documents demonstrate the interest of the proposed approach.

Keywords
Document analysis, knowledge extraction, image processing, handwriting

1 Introduction
Aujourd'hui, le traitement du courrier entrant est un problème pour les entreprises: réception du courrier, ouverture des enveloppes, reconnaissance du type de document (formulaire, facture, courrier, ...), acheminement des documents au service adéquat et enfin traitement du courrier. Bien qu'une partie de ce processus puisse être complètement automatisée (ouverture des enveloppes grâce à un équipement spécifique, lecture automatique des formulaires imprimés), une grande quantité de documents manuscrits ne peut être, à l'heure actuelle, traitée automatiquement. En effet, aucun système n'est capable de lire automatiquement une page complète de texte manuscrit cursif sans connaissance a priori. Ceci est dû à la complexité de cette tâche lorsqu'il faut gérer des documents avec une mise en page libre, de l'écriture cursive non contrainte, un contenu textuel inconnu [4]. Cependant, il est actuellement possible de considérer des documents plus spécifiques correspondant à des besoins industriels existants. L'extraction de données numériques (numéro de fichier, code client, numéro de téléphone, code postal, ...) dans des documents manuscrits dont le contenu est attendu (courrier entrant) en est un exemple.

Dans cet article, nous proposons une méthode réalisant l'extraction automatique de champs numériques dans des courriers entrants manuscrits. Pour cela, nous posons l'hypothèse que l'organisation spatiale des composantes connexes d'un champs numérique respecte une structure spécifique qui peut être exploitée pendant la phase d'extraction. Bien que cette hypothèse ne soit pas complètement réaliste, la méthode proposée est une alternative intéressante à l'utilisation d'un système de reconnaissance de chiffres sur l'ensemble du document. Elle vise essentiellement à détecter les champs numériques potentiels. En effet, les champs numériques contiennent fréquemment des chiffres liés et/ou fragmentés. De ce fait, la reconnaissance d'un champs numérique oblige à mettre en oeuvre une stratégie de segmentation-reconnaissance qui serait très pénalisante en charge de calcul si elle devait être mise en oeuvre indifféremment sur l'ensemble du document manuscrit. De plus, elle exigerait de développer des post-traitements pour rejeter les éléments non numériques. La méthode proposée servira de filtre syntaxique avant la reconnaissance. Deux étapes sont nécessaires à cette extraction. La première est dédiée à l'étiquetage des composantes connexes. Elle consiste à fournir des hypothèses sur la nature des composantes (chiffres, chiffres liés, séparateurs, mots, ...). La deuxième étape consiste à utiliser un analyseur syntaxique sur chaque ligne de texte afin de déterminer la meilleure séquence d'étiquettes connaissant la syntaxe du champs numérique que nous voulons extraire.

Ce papier est organisé comme suit. La section 2 est consacrée à la justification et aux motivations de la méthode. La section 3 présente les caractéristiques intrinsèques et contextuelles utilisées pour caractériser les composantes connexes, ainsi que les résultats de cet étiquetage. La section 4 décrit l'étape d'analyse syntaxique permettant d'extraire des champs numériques spécifiques. Les résultats expérimentaux réalisés sur une base de courriers manuscrits entrants sont présentés dans la section 5. Enfin, la section 6 présente les conclusions et les perspectives de ces travaux.

2 Présentation du système proposé

Le système proposé vise à extraire des champs numériques, tels que des codes postaux, numéros de téléphones ou des codes clients, dans des documents manuscrits non contraints. Deux exemples de courriers entrants sont présentés sur la figure 1. Nous pouvons constater que les champs d'intérêt que nous recherchons peuvent se situer n'importe où dans le document (entête, corps de texte, ...) ou même simplement ne pas être présents.

A première vue, une solution naïve pourrait consister à utiliser un système de reconnaissance de l'écriture manuscrite classique afin de reconnaître toutes les formes du document (chiffres, lettres, mots) puis de ne sélectionner que les informations d'intérêt (les chiffres). Cependant, l'utilisation d'un système de reconnaissance sur une page complète d'écriture manuscrite est coûteuse en temps de calcul et peu fiable quand il s'agit de prendre en compte un large vocabulaire [5]. En outre, comme les informations à extraire ne représentent qu'une faible part du document, la reconnaissance de ce document dans son intégralité n'est pas nécessaire. Une deuxième approche, probablement plus réaliste, serait d'utiliser un système de reconnaissance de chiffres sur les composantes connexes du document. Si l'on approfondit cette approche, la présence de chiffres liés impose notamment la mise en place d'une stratégie de segmentation-reconnaissance. Plusieurs études, en particulier sur la reconnaissance de montants numériques, ont montré la difficulté d'une telle approche [2].

En conséquence, nous avons choisi de nous concentrer sur une simple discrimination morphologique entre les chiffres et les mots. Cette discrimination va permettre de localiser les champs numériques sans pour autant faire appel à un système de reconnaissance de chiffres mais en utilisant les contraintes imposées par la structure syntaxique d'une séquence numérique. Cette discrimination servira donc de filtre syntaxique permettant de ne fournir au module de reconnaissance de chiffres, non intégré ici, que les hypothèses les plus vraisembables.

Notre système est composé des trois étapes suivantes, une fois que l'ensemble des composantes connexes a été extrait du document:

- Pré-traitement: les lignes de textes sont extraites grâce à une approche de regroupement de composantes connexes. Comme notre approche de discrimination repose sur des caractéristiques spatiales, la connaissance précise des alignements est essentielle.

- Étiquetage des composantes connexes: puisque notre méthode est dédiée à la détection de champs numériques à l'intérieur des lignes de textes, nous devons assigner à chaque composante connexe une classe qui peut être « Digit », ou « Rejet ». Une troisième classe est prise en compte (« séparateur ») car les champs numériques peuvent également contenir des séparateurs. Chaque composante connexe est décrite par 7 caractéristiques intrinsèques et contextuelles. La vraisemblance de chaque classe est alors estimée.

- Analyse syntaxique: cette dernière étape est cruciale pour le système car elle vérifie que certaines séquences de composantes connexes peuvent être retenues comme candidates. En effet, les séquences numériques que nous recherchons respectent toutes une syntaxe précise (cinq chiffres pour les codes postaux, dix chiffres pour un numéro de téléphone, ...). L'analyseur syntaxique sera utilisé comme un outils de localisation précis des champs numériques capable d'accepter les séquences syntaxiquement correctes et de rejeter les autres.

Bien que la détection des lignes de texte soit le point d'entré du système, elle est basée sur une approche plutôt classique. Nous décrivons sommairement les principales étapes. L'approche utilisée est inspirée de [3] où une méthode de détection des lignes de texte sans connaissance a priori de l'orientation est présentée. Plusieurs modifications ont été apportées afin de ne prendre en compte que les lignes horizontales. Les principales étapes sont les suivantes:

- regroupement des composantes connexes dont la taille est supérieure à un seuil donné. Ceci permet de ne prendre en compte que les composantes connexes correspondant à des mots ou parties de mot, alors que les signes de ponctuation et les accents sont ignorés. Le regroupement est réalisé selon un critère de distance.

- Fusion d'alignements trop proches: plusieurs alignements peuvent être détectés dans une même ligne de texte. La fusion de ces alignements est réalisée selon un critère de distance prenant en

compte la taille moyenne des inter-lignes sur l'ensemble du document.

- Affectation des composantes isolées à la ligne la plus proche: cette dernière phase permet de prendre en compte l'ensemble des composantes connexes ignorées lors de la première étape.

3 Étiquetage des composantes connexes

Un document manuscrit peut être vu comme un ensemble de lignes constituées de composantes connexes, chaque composante connexe pouvant être « résumé » par son rectangle englobant (comme l'illustre l'exemple de la figure 2).

Comme nous n'avons pas besoin de la connaissance symbolique que véhicule une composante connexe ("0", "3", "A", "Tel", ...) mais simplement d'être capable de discriminer les chiffres du reste du texte, il est nécessaire de regrouper les chiffres dans une seule et même classe. Par conséquent, il est indispensable de trouver des caractéristiques communes à toutes les formes de chiffres. Il est de plus important de constater qu'un grand nombre de champs (principalement les numéros de téléphone et les codes clients) contiennent plusieurs séparateurs (point ou tiret). Comme ces séparateurs sont souvent un point fort dans la syntaxe de ces champs, ils doivent être également identifiés. Enfin, un grand nombre de champs numériques contiennent des chiffres liés (figure 3). Du fait que notre approche n'utilise pas un classifieur chiffres, il n'est pas nécessaire de faire appel à un procédé de segmentation explicite sur ces composantes. Il est cependant nécessaire d'identifier ces composantes connexes correspondant à des chiffres liés.

Figure 3. Un exemple de champs contenant deux chiffres liés

En tenant compte des observations précédentes, nous avons défini quatre classes d'étiquettes appelées: chiffre ou « digit » (classe D), chiffres liés ou double digits (classe DD), séparateurs (classe S) et rejet, c'est-à-dire toute autre composante connexe restant dans le document (classe R).

Le premier problème est de déterminer des caractéristiques fiables qui puissent discriminer autant que possible ces quatre classes. Prenons comme exemple les champs numériques (numéro de téléphone et code client) de la figure 4. Il n'y a à première vue pas de ressemblance notable entre les différents chiffres.

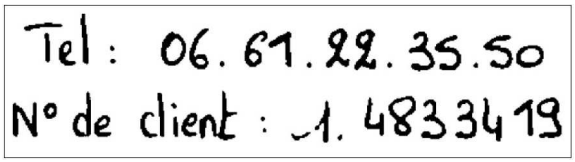


Figure 4. Exemple de deux champs numériques

Cependant, si nous considérons les rectangles englobants des composantes connexes (figure 5), nous pouvons mettre en évidence que les chiffres d'un même champs sont plutôt réguliers aussi bien dans leurs dimensions que dans leur espacement.

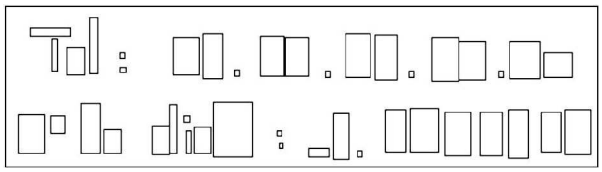


Figure 5. Rectangles englobants des composantes connexes de la figure 4

Puisque tous les chiffres ont pratiquement le même aspect, le rapport hauteur sur largeur semble être une caractéristique pertinente. Malheureusement, cette seule caractéristique n'est pas suffisamment discriminante. Nous avons donc ajouté d'autres caractéristiques afin de prendre en compte la régularité des composantes connexes en terme de hauteur, largeur et d'espacement au sein d'un champs numérique.

Ces régularités sont mesurées dans le voisinage de chaque composante connexe en considérant leurs voisins directs à gauche et à droite sur la même ligne de texte de la façon suivante. Soit C la composante connexe dont on calcule le vecteur de caractéristiques, C-1 et C+1 ses voisins gauche et droit respectivement. Soit H_C, W_C et G_C sa hauteur, sa largeur et l'abscisse de son centre de gravité. En prenant en compte la hauteur et la largeur de C-1 et de C+1 et la distance relative de C-1 et C+1 à C, la régularité/irrégularité en hauteur, en largeur et espacement dans le voisinage de C peut être mesurée par les caractéristiques suivantes:

$$f_1 = \frac{H_{C-1}}{H_C}\ ,\ f_2 = \frac{H_{C+1}}{H_C}\ ,\ f_3 = \frac{W_{C-1}}{W_C}\ ,$$
$$f_4 = \frac{W_{C+1}}{W_C}\ ,\ f_5 = \frac{|G_C - G_{C-1}|}{W_C}\ ,$$
$$f_6 = \frac{|G_C - G_{C+1}|}{W_C}\ ,\ f_7 = \frac{H_C}{W_C}\ .$$

f1 et f2 traduisent la régularité en hauteur, f3 et f4 en largeur, tandis que f5 et f6 traduisent la régularité dans l'espacement des rectangles englobants. Enfin, f7 caractérise l'aspect intrinsèque de la forme.

Finalement, chaque composante connexe C faisant partie d'un alignement sera caractérisée par un vecteur à 7 caractéristiques. La vraisemblance de chaque étiquette est alors estimée par un classifieur statistique (KPPV). Les expériences réalisées sur une base de 293 documents ont donné les résultats en première position (TOP1) (cf tableau 1).

	Reject	Digit	Separator	Double Digit	OK	Confusion
Reject	83 287	8 233	2 395	371	88,33%	11,67%
Digit	3 289	5 290	38	46	61,06%	38,94%
Separator	1 266	28	831	0	39,11%	60,89%
Double Digit	381	135	1	60	10,40%	89,60%

Table 1. Résultats TOP1 de l'étiquetage des composantes connexes sur 293 images

La première remarque que l'on peut faire est que la classe rejet est plutôt bien reconnue et que peu de composantes de cette classe sont confondues (11,67%). On peut également remarquer la faible confusion entre digits et séparateurs ainsi qu'entre doubles digits et séparateurs. Cependant, la confusion des digits, doubles digits et séparateurs avec le rejet est assez forte. Ceci provient du fait qu'un grand nombre de composantes connexes provenant de mots ou morceaux de mots ont des caractéristiques très proches de celles des digits, double digits et séparateurs.

Néanmoins, l'étape suivante va permettre de corriger ces mauvaises hypothèses en contextualisant les interprétations locales grâce à la syntaxe connue des champs numériques qu'on souhaite détecter.

4 Extraction des champs par analyse syntaxique

Rappelons notre hypothèse: dans une ligne de texte, chaque champs numérique possède des régularités morphologiques mises en évidence par le vecteur de caractéristiques et possède une structure syntaxique particulière qui correspond à une séquence de chiffres et de séparateurs. Cette syntaxe peut facilement être modélisée par un modèle de Markov pourvu que l'on dispose de séquences numériques étiquetées pour chacun des champs que l'on souhaite extraire. Finalement, l'extraction d'un champs numérique dans une ligne manuscrite consistera à rechercher la meilleure séquence d'étiquettes possible pour cette ligne, problème qui trouve sa solution par l'algorithme de Viterbi éventuellement modifié pour retenir les n meilleures solutions. Nous avons donc défini un modèle pour chaque syntaxe (pour chaque type de champs numériques que nous voulons extraire). Nous allons maintenant expliciter comment ces modèles sont construits en illustrant le cas du code postal. Les autres modèles sont construits de la même façon.

Un code postal français est composé de cinq chiffres, chacun correspondant à un état précis: D0, D1, D2, D3, D4. Comme une ligne de texte peut contenir, en plus du code postal, des mots que nous devons considérer comme du rejet, il est nécessaire d'introduire un état de rejet noté R. Une matrice de transition est alors construite, reflétant les probabilités de passer d'un état au suivant. Par exemple, si l'on se trouve en l'état D0, la seule transition possible dans un code postal est d'aller vers l'état D1, tous les autres étant interdits. En faisant de même sur les autres états, nous obtenons le modèle de la figure 6.

Pour l'intégration des états doubles digits, nous devons ajouter autant d'états qu'il y a de combinaisons de deux chiffres collés: dans notre cas, 4 états possibles pour une séquence de 5 chiffres. Cependant, une observation plus fine des codes postaux manuscrits permet de constater que les deuxième et troisième chiffres (D1 et D2) sont très rarement liés (à cause d'un espacement plus grand): cette habitude des scripteurs peut s'expliquer par la structure des codes postaux français puisque les deux premiers chiffres indiquent le département et les trois suivant indiquent la ville à l'intérieur de celui-ci. Cette observation nous permet de limiter le nombre d'états double digit à 3 (de DD0 à DD2) (figure 6).

Pour compléter la définition du modèle, nous devons de plus construire une matrice des états initiaux. Comme un code postal est plus souvent situé en début de ligne, les états D0 et DD0 seront favorisés. Cependant, comme une ligne de texte peut également commencer par autre chose qu'un code postal, la probabilité initiale pour l'état rejet doit être non nulle. De même, une matrice des états finaux est construite (seuls les états D4, DD2, et R ont des valeurs non nulles).

Il est important de remarquer que ce modèle ne s'applique qu'à des champs codes postaux parfaitement propres. Cependant, une quantité non négligeable de champs contient des petites composantes connexes qui proviennent du bruit ou de morceaux de chiffres comme le montre la figure 7. Dans ce cas, le modèle va rejeter les champs car la syntaxe d'un code postal n'est pas respectée.

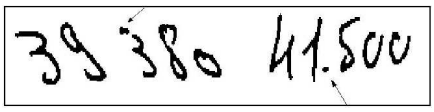


Figure 7. Présence de bruit à l'intérieur d'un code postal

Afin de rendre le modèle syntaxique plus tolérant au bruit, nous avons rajouté un état parasite P entre deux états. Notons que les probabilités de transition vers ces états parasites doivent rester faibles afin de ne pas augmenter inutilement la fausse alarme. Nous obtenons enfin le modèle syntaxique complet pour un champs de type code postal (figure 8).

Les modèles syntaxiques des numéros de téléphone et des codes clients sont construits en raisonnant de même. 24 états sont retenus pour les codes clients et 33 pour les numéros de téléphone.

L'apprentissage des différents modèles est réalisé par une simple estimation statistique des différentes matrices (matrices des états initiaux, finaux et matrice de transitions) à partir d'une base d'apprentissage étiquetée.

5 Résultats expérimentaux

Les expérimentations ont été réalisées sur deux bases distinctes d'images de courriers entrants manuscrits provenant du service de réception du courrier d'une grande entreprise: la première (292 images) a été utilisée comme base d'apprentissage pour la classification des composantes connexes ainsi que pour déterminer les probabilités de transitions des HMM et pour paramétrer le système; la seconde (293 documents) a servi à tester notre approche.

La détection des champs numériques est réalisée en effectuant l'analyse de chaque ligne d'un document. L'analyseur syntaxique se prononce pour la présence d'un champs (détection) ou contre (rejet) sur la ligne en cours d'analyse. Comme l'analyseur syntaxique peut proposer plusieurs hypothèses de localisation lors de la détection d'un même champs, la sortie de l'analyseur sera une liste de N solutions correspondant aux N meilleurs chemins dans le treillis d'observations. Par conséquent, un champs est considéré comme convenablement détecté si et seulement si aucune composante du champs étiqueté n'est rejetée et si toutes les composantes connexes dans le champs détecté appartiennent au champs étiqueté. Des exemples illustrent différents cas de localisation: localisation d'un code postal (figure 9), d'un code client (figure 10) et d'un numéro de téléphone mal détecté (figure 11). Pour chaque exemple, on présente l'image d'une ligne extraite d'un document manuscrit complet et les rectangles englobants issus de l'extraction des composantes connexes. La liste des différentes propositions fournies par l'analyseur syntaxique (grâce à l'algorithme de Viterbi) est donnée par ordre décroissant: les solutions les plus probables se trouve en tête de liste. Chaque étiquette correspond à une composante connexe de la ligne (lue de gauche à droite). Dans le cas de la figure 9 qui illustre l'extraction d'un code postal, l'étiquette « D0 » correspond au premier chiffre du code postal, « D1 » au deuxième, ..., et R au rejet.

La figure 12 donne le résultat de l'application de cette méthode pour l'extraction de numéros de téléphone sur un document manuscrit complet. Ici la liste de propositions d'alignement a été réduite à deux. Les lignes en caractères gras correspondent aux deux lignes de texte contenant des numéros de téléphone (composantes connexes en noir sur l'image). Nous pouvons remarquer sur cet exemple que le deuxième numéro de téléphone qui est situé dans le corps du document est aussi bien détecté que celui qui se trouve dans l'entête.

Le tableau 2 présente les résultats de la détection des trois types de champs numériques à extraire suivant que la solution se trouve placée première proposition (TOP1), dans les deux premières propositions (TOP2) ou dans dix premières (TOP10).

		TOP1		TOP2		TOP10	
ZIP Code	detected	138	41,82%	220	66,67%	292	88,48%
	not detected	192	58,18%	110	33,33%	38	11,52%
Phone number	detected	169	69,55%	195	80,25%	223	91,77%
	not detected	74	30,45%	48	19,75%	20	8,23%
Customer code	detected	91	60,26%	116	76,82%	133	88,08%
	not detected	60	39,74%	35	23,18%	18	11,92%

Table 2. Résultats de la détection des trois types de champs

Nous pouvons remarquer que les meilleurs résultats sont obtenus pour les syntaxes les plus contraignantes. En effet, un code postal n'est composé que de cinq chiffres et d'aucun séparateur alors qu'un numéro de téléphone ou un code client sont des séquences de chiffres plus longues susceptibles de contenir des séparateurs ce qui contraint la syntaxe de ces séquences.

Une analyse plus précise du comportement du système est illustrée par le tableau 3 qui présente la matrice de confusion de chaque analyseur syntaxique pour la détection en TOP1. Il est important de remarquer que cette matrice ne comptabilise pas le nombre de champs correctement détectés ou non, mais les lignes. Ainsi, une ligne contenant un code client mais étant détectée comme contenant un code postal apparaîtra dans la matrice de confusion à la ligne « code client » et à la colonne « code postal ». Enfin, certaines cases contiennent deux chiffres: le premier correspond à un champs bien détecté et bien positionné dans la ligne, exemple des figures 9 et 10, alors que le second correspond à une détection du bon type de champs mais mal localisé dans la ligne (un champs mal aligné, comme le montre l'exemple de la figure 11 pour un numéro de téléphone).

	ZIP Code	Reject	Phone Number	Reject	Customer Code	Reject
ZIP Code	137 59	127	45	278	38	285
Phone number	226	14	167 51	22	188	52
Customer code	127	20	55	92	90 21	36
Other numerical fields	219	756	63	912	56	919
Reject	264	3756	45	3975	44	3976

Table 3. Matrice de confusion TOP1 de chaque analyseur syntaxique

A partir des résultats de cette matrice de confusion, nous pouvons constater que les lignes ne contenant pas de champs numériques (lignes à rejeter) sont effectivement rejetées. Sur l'exemple de la figure 12, les lignes ne contenant pas de champs numériques d'intérêt sont toutes étiquetées comme étant une suite

de composantes connexes de type rejet. Nous pouvons également remarquer que les champs numériques qui ne sont ni des codes postaux, ni des codes clients, ni des numéros de téléphone, mais qui peuvent être une date, un numéro de compte, un numéro de carte SIM, ..., sont mieux rejetés par les syntaxes « numéro de téléphone » et « code client » que par la syntaxe « code postal ». Ceci montre que la confusion est moindre pour les syntaxes contraignantes.

6 Conclusion et perspectives

Dans ce papier, nous avons présenté une méthode originale permettant de localiser avec précision des champs numériques dans des courriers entrants manuscrits. L'originalité de la méthode repose sur son principe de localisation qui est réalisée grâce à une analyse syntaxique des rectangles englobants des composantes connexes. Comme il n'est pas nécessaire de reconnaître les chiffres, l'extraction des caractéristiques est simple et rapide et le nombre de classes à discriminer réduit. De plus, le système a été implémenté pour l'extraction de codes postaux, codes clients et numéros de téléphone dans des courriers manuscrits, mais rien n'empêche d'appliquer le même principe pour extraire d'autres types de champs numériques dans des documents manuscrits non contraints pour peu que les champs respectent une syntaxe précise.

Les perspectives d'améliorations du système portent sur deux optimisations du processus de localisation. La première concerne l'apprentissage automatique des différents modèles de syntaxes en utilisant l'algorithme de Baum-Welch car les matrices d'états initiaux, finaux et de transition ont été estimées sur la base d'apprentissage d'après les observations d'un opérateur humain. Le second point concerne la combinaison des solutions fournies par les trois analyseurs syntaxique afin de diminuer le taux de fausse alarme et d'augmenter le taux de bonne détection.

7 Références

[1] G.D. Forney, “The Viterbi algorithm”, Proc. of the IEEE 61(1973), pp. 268-278.

[2] L. Heutte, P. Pereira, O. Bougeois, J.V. Moreau, B. Plessis, P. Courtellemont, Y. Lecourtier, “Multi-bank check recognition system: consideration on the numeral amount recognition module”, IJPRAI 11 (1997), pp. 595-618.

[3] L. Likforman-Sulem, C. Faure, « Une méthode de résolution des conflits d'alignements pour la segmentation des documents manuscrits », Traitement du Signal 12 (1995), pp. 541-549.

[4] L. Lorette, “Handwriting recognition or reading ? What is the situation at the dawn of the third millenium”, IJDAR (1999), pp. 2-12.

[5] A. Nosary, T. Paquet, L. Heutte, A. Bensefia, “Handwritten text recognition through writer adaptation”, IEEE Proceedings (2002), IWFHR'02, Ontario, pp. 363-368.

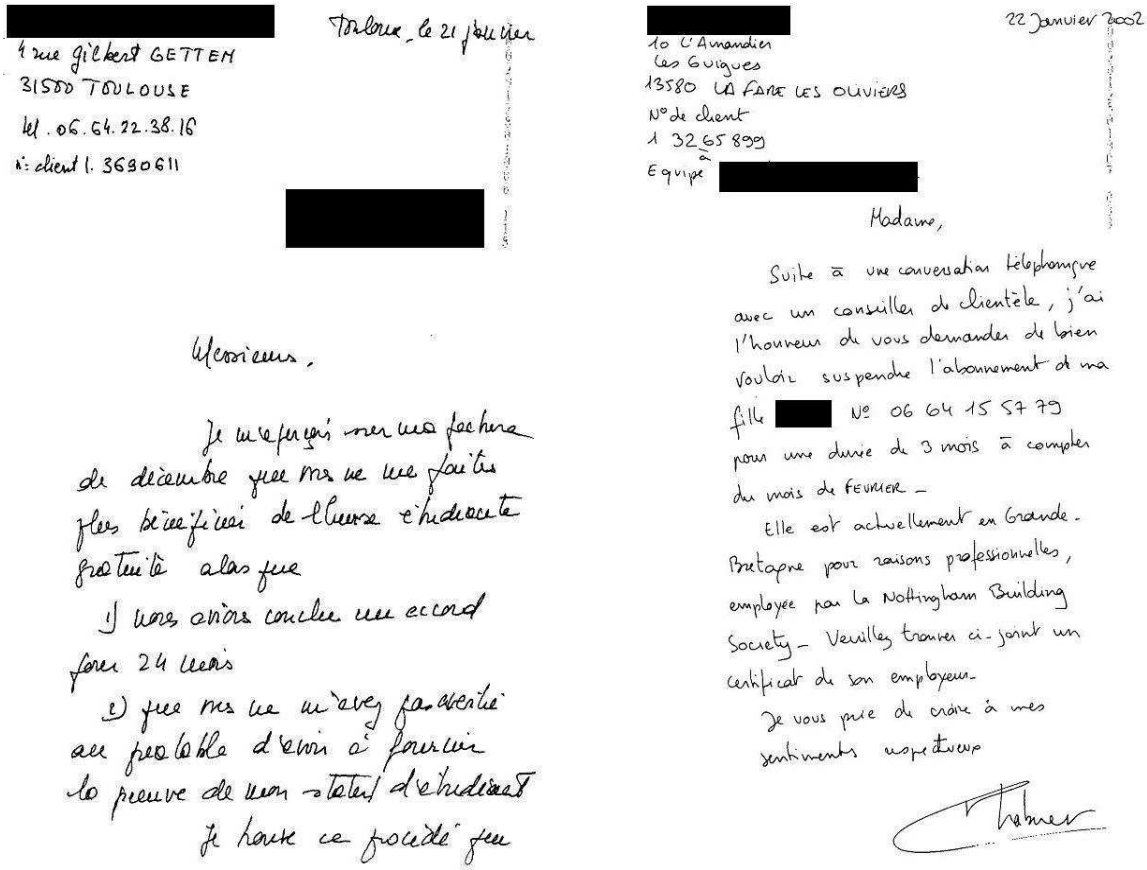


Figure 1. Deux exemples de courriers entrants manuscrits

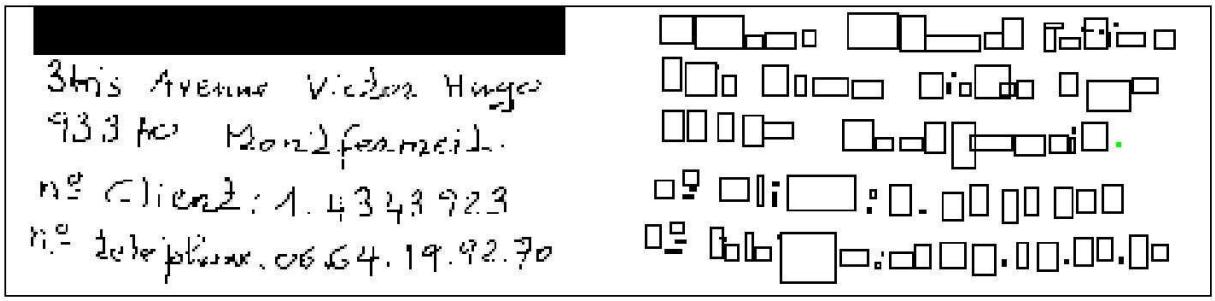


Figure 2. Entête d'un document manuscrit et ses composantes connexes

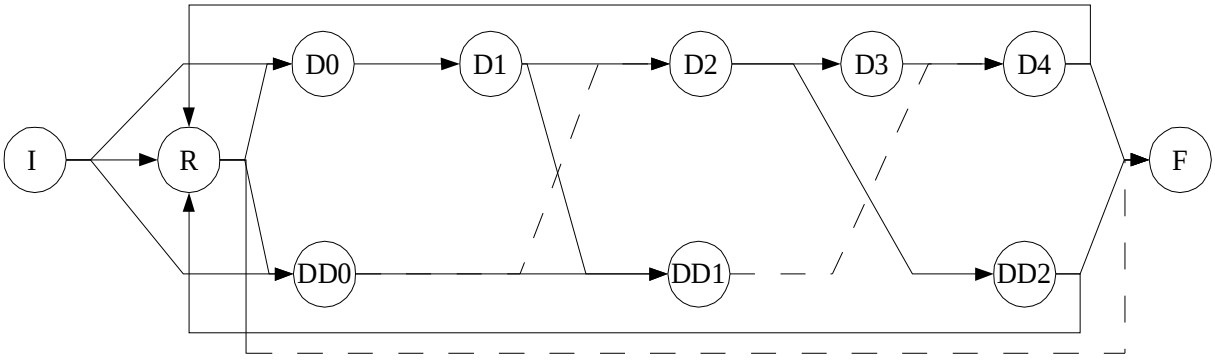


Figure 6. Modèle syntaxique pour un code postal

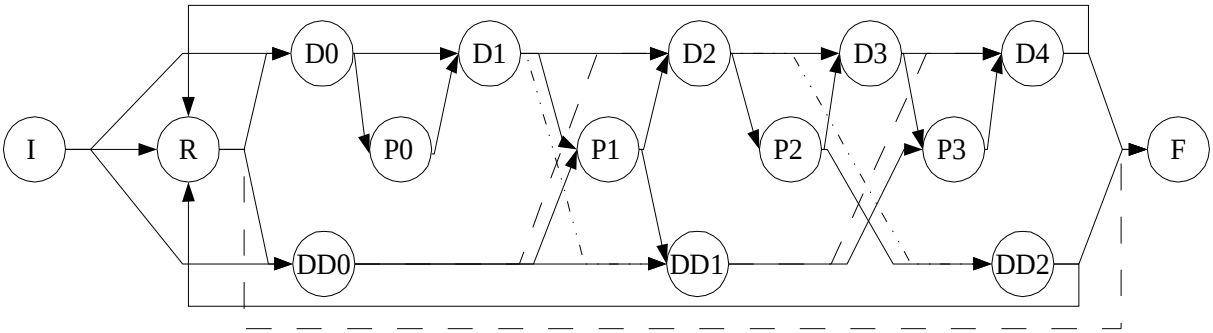


Figure 8. Modèle définitif pour la syntaxe d'un code postal

0 :	**D0**	**D1**	**D2**	**D3**	**D4**	**R**	**R**	**R**	**R**	**R**	**R**	**R**	**R**	**R**	**R**	**-4.2148**
1 :	D0	D1	D2	D3	P3	D4	R	R	R	R	R	R	R	R	R	-4.52807
2 :	D0	D1	DD1	D4	R	R	R	R	R	R	R	R	R	R	R	-5.01316
3 :	D0	D1	D2	P2	D3	D4	R	R	R	R	R	R	R	R	R	-5.15132
4 :	D0	D1	D2	P2	D3	P3	D4	R	R	R	R	R	R	R	R	-5.15671
5 :	DD0	D2	D3	D4	R	R	R	R	R	R	R	R	R	R	R	-5.19361
6 :	D0	D1	DD1	P3	D4	R	R	R	R	R	R	R	R	R	R	-5.28962
7 :	D0	D1	D2	DD2	R	R	R	R	R	R	R	R	R	R	R	-5.45325
8 :	DD0	D2	D3	P3	D4	R	R	R	R	R	R	R	R	R	R	-5.47007
9 :	D0	D1	D2	P2	DD2	R	R	R	R	R	R	R	R	R	R	-5.85465

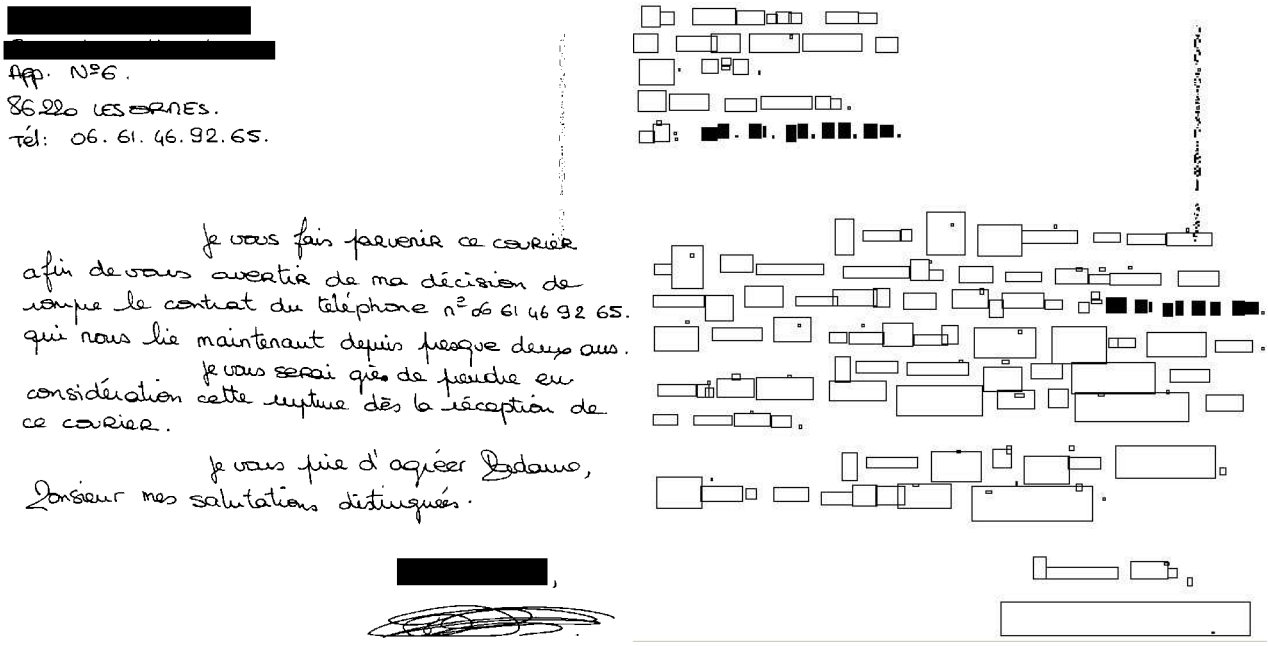
Figure 9. Exemple de localisation correcte d'un code postal
(La bonne solution se trouve en première position dans la liste)

0 :	**R**	**R**	**R**	**R**	**R**	**R**	**R**	**R**	**D0**	**S0**	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**	**D7**	**-7.24293**
1 :	R	R	R	R	R	R	R	R	D0	S0	DD0	D3	P4	D4	D5	D6	D7	-9.19281
2 :	R	R	R	R	R	R	R	R	D0	S0	D1	D2	P3	DD2	D5	D6	D7	-9.19281
3 :	R	R	R	R	R	R	R	R	D0	S0	D1	D2	P3	D3	DD3	D6	D7	-9.19281
4 :	R	R	R	R	R	R	R	R	D0	S0	DD0	D3	D4	D5	D6	P7	D7	-9.23857
5 :	R	R	R	R	R	R	R	R	D0	S0	D1	D2	D3	D4	DD4	P7	D7	-9.23857
6 :	R	R	R	R	R	R	R	R	D0	S0	D1	D2	D3	DD3	D6	P7	D7	-9.23857
7 :	R	R	R	R	R	R	R	R	D0	S0	DD0	D3	D4	D5	D6	D7	R	-9.2689
8 :	R	R	R	R	R	R	R	R	D0	S0	D1	D2	D3	D4	DD4	D7	R	-9.2689
9 :	R	R	R	R	R	R	R	R	D0	S0	D1	D2	D3	DD3	D6	D7	R	-9.2689

Figure 10. Exemple de localisation correcte d'un code code client
(la bonne solution se trouve en première proposition dans la liste)

0 :	R	D0	P0	D1	D2	D3	D4	D5	D6	D7	D8	D9	R	-7.97471
1 :	R	D0	P0	D1	D2	D3	D4	D5	D6	P9	D7	D8	D9	-8.4215
2 :	**R**	**R**	**R**	**D0**	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**	**D7**	**D8**	**D9**	**-8.51302**
3 :	D0	D1	P2	D2	D3	D4	D5	D6	D7	P11	D8	D9	R	-8.8778
4 :	R	D0	P0	D1	D2	P3	D3	D4	D5	D6	D7	D8	D9	-8.89862
5 :	R	D0	P0	D1	D2	D3	D4	P6	D5	D6	D7	D8	D9	-8.89862
6 :	R	D0	P0	D1	D2	D3	D4	D5	D6	D7	D8	P12	D9	-8.89862
7 :	D0	D1	P2	D2	P3	D3	D4	D5	D6	D7	D8	D9	R	-9.05389
8 :	D0	D1	P2	D2	D3	D4	P6	D5	D6	D7	D8	D9	R	-9.05389
9 :	D0	D1	P2	D2	D3	D4	D5	D6	P9	D7	D8	D9	R	-9.05389

Figure 11. Exemple de localisation mal alignée d'un numéro de téléphone
(la bonne solution se trouve en troisième position de la liste)



ligne 0 : 0 : R R R R R R R R R -1.94143
1 : DD0 D2 P3 D3 D4 P6 D5 DD3 DD4 -11.1048
ligne 1 : 0 : R R R R R R -1.27327
1 : DD0 DD1 DD2 DD3 DD4 R -10.1291
ligne 2 : 0 : R R R R R -2.60336
1 : DD0 DD1 DD2 DD3 DD4 -10.6136
ligne 3 : 0 : R R R R R R R -1.54087
1 : DD0 DD1 DD2 DD3 D8 D9 R -10.8604
ligne 4 : 0 : R R -0.0743003
1 : D9 R -999.941
ligne 5 : 0 : R R D0 D1 S0 D2 D3 S1 D4 D5 S2 D6 D7 S3 D8 D9 R -8.91556
1 : R D0 P0 D1 S0 D2 D3 S1 D4 D5 S2 D6 D7 S3 D8 D9 R -10.25
ligne 6 : 0 : R R R R R -0.394603
1 : DD0 DD1 DD2 DD3 DD4 -11.7527
ligne 7 : 0 : R R -0.47224
ligne 8 : 0 : R R R R R R R R R R R R R -1.92125
1 : D0 P0 D1 DD1 D4 P6 D5 D6 D7 DD4 R R R -12.1089
ligne 9 : 0 : R R R R R R R R R R R R DD0 D2 D3 D4 D5 D6 D7 D8 D9 R -8.32686
1 : R R R R R R R R R R R R R R R R R R R R R -8.73184
ligne 10 : 0 : R R R R R R R R -0.952718
1 : DD0 D2 P3 D3 DD2 DD3 DD4 R -11.291
ligne 11 : 0 : R R R R -0.742421
ligne 12 : 0 : R R R R R -0.829571
1 : DD0 DD1 DD2 DD3 DD4 -13.9666
ligne 13 : 0 : R R R R R R R R R R -1.02211
1 : R R DD0 DD1 DD2 DD3 DD4 R R R -10.0594
ligne 14 : 0 : R R R R -0.634081
ligne 15 : 0 : R R R R R R -1.13937
1 : DD0 DD1 DD2 DD3 DD4 R -12.1255
ligne 16 : 0 : R R R R R R R -0.707206
1 : DD0 DD1 DD2 DD3 DD4 R R -10.1748
ligne 17 : 0 : R R R R -0.613117

Figure 12. Exemple de localisation de numéro de téléphone sur une page complète