Rapport Evaluation des OCR

Romain Karpinski, Abdel Belaid

Rapport

Evaluation des OCR

Romain Karpinski
romain.karpinski@loria.fr

Encadrant : Abdel Belaid
abdel.belaid@loria.fr

Année 2015–2016

Table des matières

Table des matières **i**

1 Introduction générale **1**

1.1 Introduction . . . 1
1.2 Contexte . . . 2
1.2.1 Les évaluations . . . 2
1.2.2 La segmentation . . . 2
1.2.3 La reconnaissance de caractères . . . 6
1.3 Contrainte . . . 7
1.3.1 Sortie de l'OCR - Entrées du système . . . 7
1.4 Objectifs . . . 8

2 Evaluation Texte / Texte **9**

2.1 Enoncé du problème . . . 9
2.2 Etat de l'art . . . 10
2.3 Choix de la méthode . . . 12
2.4 Corpus . . . 12
2.5 Evaluation . . . 13
2.5.1 Evaluation de la méthode . . . 13
2.5.2 Evaluation des OCR . . . 16
2.6 Discussion . . . 16
2.6.1 Conclusion . . . 16
2.6.2 Améliorations . . . 17

3 Evaluation XML / XML **18**

3.1 Enoncé du problème . . . 18
3.2 Etat de l'art . . . 18
3.3 Choix de la méthode d'évaluation . . . 26
3.3.1 Segmentation . . . 26
3.3.2 Reconnaissance de caractères . . . 30

3.4 Corpus . . . 31
3.5 Evaluation . . . 32
3.5.1 Segmentation . . . 32
3.5.2 Reconnaissance de caractères . . . 34
3.6 Discussion . . . 35
3.6.1 Conclusion . . . 35

4 Evaluation Texte / XML **37**
4.1 Enoncé du problème . . . 37
4.2 Etat de l'art . . . 37
4.3 Choix de la méthode . . . 37
4.4 Corpus . . . 40
4.5 Evaluation . . . 41
4.5.1 Evaluation de la méthode . . . 41
4.5.2 Evaluation des OCR . . . 42
4.6 Discussion . . . 42
4.6.1 Conclusion . . . 42
4.6.2 Améliorations . . . 43

5 Conclusion générale **44**

Bibliographie **46**

Liste des illustrations **47**

Liste des tableaux **49**

Annexes 51

6 Algorithme Zonemap+ **51**

7 Sortie XML de l'OCR "OCRopus" en format HOCR **52**

8 Document n°7 du corpus$_1$ **53**

9 Corpus$_2$ **54**

10 Exemple de document complexe - Corpus$_2$ **57**

1 Introduction générale

1.1 Introduction

De nos jours, nous utilisons énormément de documents papier (de type administratif, rapport, publicité...). Le développement important de l'informatique a créé le besoin de dématérialiser les informations contenues dans ces documents afin de pouvoir les classifier et les analyser. On trouve aujourd'hui une large variété de systèmes ayant des objectifs bien définis, comme la segmentation de lignes manuscrites en mots ou la reconnaissance de mots.
Un système de reconnaissance de documents a pour objectif de transformer un document physique en document numérique. Par exemple, on peut vouloir effectuer des recherches de contenus ou traiter les informations contenues dans ces documents, d'où l'intérêt d'extraire leur contenu. Dans le cas des documents image de type imprimés, on trouve une catégorie de systèmes permettant d'extraire le contenu et la mise en page : les OCR (**O**ptical **C**haracter **R**ecognition). Dans le contexte d'un OCR, la reconnaissance d'un document s'effectue généralement en quatre étapes :

1. **Pré-traitement** : traitement de l'image afin de faciliter la reconnaissance. Elle peut prendre la forme d'une augmentation du contraste (binarisation pour une image de niveau de gris), ou d'un redressement du document (correction de l'orientation), ou d'un nettoyage de bruit.
2. **Segmentation** : Découpage de l'image en zones d'information, potentiellement à plusieurs niveaux différents : paragraphe, ligne, mot, image, figure, tableau...
3. **Reconnaissance de caractères** : Extraction du texte à proprement parler. Il s'agit de reconnaître chaque caractère grâce à ses caractéristiques typographiques, ou en le comparant à une base de caractères.
4. **Post-traitement** : Correction des erreurs éventuelles, évaluations des performances dy système pour toutes les étapes de traitement.

Avec le développement de ces systèmes, est venu le besoin de les évaluer. Lorsqu'on parle d'évaluation, on peut se poser trois questions :

1. Quels sont les aspects évaluables ?
2. Quelles sont les contraintes à respecter ?
3. Comment évaluer ces aspects ?

Nous allons voir dans la section 1.2 les aspects évaluables des OCR puis nous verrons en section 1.3, les contraintes imposées. Enfin le reste du rapport tentera, après un état de l'art, d'apporter une réponse à la troisième question : comment les évaluer ?

1.2 Contexte

Dans le cadre du développement du logiciel d'évaluation d'OCR intitulé "Performance Evaluation", l'équipe READ cherche à lister et à mettre en oeuvre des méthodes générique permettant l'évaluation des OCR. Ces méthodes doivent être applicables sur tous les OCR pour pouvoir comparer correctement leurs résultats. Nous allons tout d'abord énoncer les types d'évaluations possibles sur ces systèmes, les décrire et montrer leurs limites.

1.2.1 Les évaluations

Elles sont mises en place à la dernière étape (**Post-traitement**) de l'OCR, c'est-à-dire lorsqu'il a terminé de reconnaître les caractères et produit une sortie. Il n'est pas prévu d'évaluer l'étape de **Pré-traitement** car elle n'est pas connue, c'est-à-dire cachée par l'OCR. Il nous reste donc deux étapes à évaluer, la **Segmentation** et la **Reconnaissance de caractères**.

1.2.2 La segmentation

Définition

L'objectif de la segmentation des documents image est d'isoler des régions (ensemble de pixels) d'une image. La segmentation est une étape importante dans l'extraction des informations d'un document image, car un mauvais repérage des de l'information mènera probablement à un accroissement de la difficulté lors de l'étape de reconnaissance.

La segmentation peut être effectuée à plusieurs niveaux :

- **Bloc** : zones souvent rectangulaire contenant soit une succession de lignes de textes, soit un graphique ou une photographie.
- **Ligne** : ensemble de mots
- **Mot** : chaîne de caractères
- **Caractère** : ensemble de pixels connexes

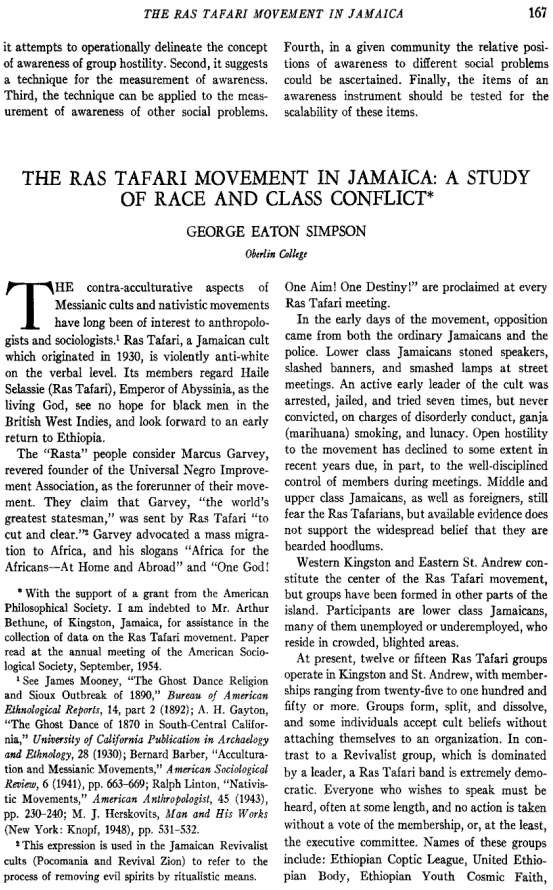
À l'heure actuelle, les OCRs ne sont pas très habiles pour extraire des structures complexes telles que des tableaux ou des équations. Pour les photographies ou les logos, souvent, ils sont segmentés en blocs et ne sont pas passés à l'étape de reconnaissance.

La figure 1.1 est un exemple d'un document image avec ses zones correspondant à la segmentation en mots. Sur cette figure, la zone est représentée par sa boite englobante. Chaque zone sur cette image a été labelisée comme étant un mot.

FIGURE 1.1 – Exemple de document image avec sa segmentation au niveau mots

Les erreurs

Une erreur possible, en lien direct avec la segmentation est l'inversion de certaines zones par rapport à l'ordre logique de lecture du document. Sur l'exemple de la Figure 1.2 le paragraphe en bas à gauche correspond à des notes de bas de page. Si on regarde le document de loin, sans prendre en compte son contenu, l'ordre logique de lecture qui apparait est de lire la colonne de gauche puis la colonne de droite.
On voit clairement que la disposition physique joue un rôle important dans la définition de l'ordre de lecture mais qu'elle peut être trompeuse. C'est par le contenu que nous pourrons faire la différence et retrouver le bon ordre.



THE RAS TAFARI MOVEMENT IN JAMAICA 167

it attempts to operationally delineate the concept of awareness of group hostility. Second, it suggests a technique for the measurement of awareness. Third, the technique can be applied to the measurement of awareness of other social problems. Fourth, in a given community the relative positions of awareness to different social problems could be ascertained. Finally, the items of an awareness instrument should be tested for the scalability of these items.

THE RAS TAFARI MOVEMENT IN JAMAICA: A STUDY OF RACE AND CLASS CONFLICT*

GEORGE EATON SIMPSON

Oberlin College

THE contra-acculturative aspects of Messianic cults and nativistic movements have long been of interest to anthropologists and sociologists.[1] Ras Tafari, a Jamaican cult which originated in 1930, is violently anti-white on the verbal level. Its members regard Haile Selassie (Ras Tafari), Emperor of Abyssinia, as the living God, see no hope for black men in the British West Indies, and look forward to an early return to Ethiopia.

The "Rasta" people consider Marcus Garvey, revered founder of the Universal Negro Improvement Association, as the forerunner of their movement. They claim that Garvey, "the world's greatest statesman," was sent by Ras Tafari "to cut and clear."[2] Garvey advocated a mass migration to Africa, and his slogans "Africa for the Africans—At Home and Abroad" and "One God! One Aim! One Destiny!" are proclaimed at every Ras Tafari meeting.

In the early days of the movement, opposition came from both the ordinary Jamaicans and the police. Lower class Jamaicans stoned speakers, slashed banners, and smashed lamps at street meetings. An active early leader of the cult was arrested, jailed, and tried seven times, but never convicted, on charges of disorderly conduct, ganja (marihuana) smoking, and lunacy. Open hostility to the movement has declined to some extent in recent years due, in part, to the well-disciplined control of members during meetings. Middle and upper class Jamaicans, as well as foreigners, still fear the Ras Tafarians, but available evidence does not support the widespread belief that they are bearded hoodlums.

Western Kingston and Eastern St. Andrew constitute the center of the Ras Tafari movement, but groups have been formed in other parts of the island. Participants are lower class Jamaicans, many of them unemployed or underemployed, who reside in crowded, blighted areas.

At present, twelve or fifteen Ras Tafari groups operate in Kingston and St. Andrew, with memberships ranging from twenty-five to one hundred and fifty or more. Groups form, split, and dissolve, and some individuals accept cult beliefs without attaching themselves to an organization. In contrast to a Revivalist group, which is dominated by a leader, a Ras Tafari band is extremely democratic. Everyone who wishes to speak must be heard, often at some length, and no action is taken without a vote of the membership, or, at the least, the executive committee. Names of these groups include: Ethiopian Coptic League, United Ethiopian Body, Ethiopian Youth Cosmic Faith,

* With the support of a grant from the American Philosophical Society. I am indebted to Mr. Arthur Bethune, of Kingston, Jamaica, for assistance in the collection of data on the Ras Tafari movement. Paper read at the annual meeting of the American Sociological Society, September, 1954.

[1] See James Mooney, "The Ghost Dance Religion and Sioux Outbreak of 1890," *Bureau of American Ethnological Reports*, 14, part 2 (1892); A. H. Gayton, "The Ghost Dance of 1870 in South-Central California," *University of California Publication in Archaeology and Ethnology*, 28 (1930); Bernard Barber, "Acculturation and Messianic Movements," *American Sociological Review*, 6 (1941), pp. 663–669; Ralph Linton, "Nativistic Movements," *American Anthropologist*, 45 (1943), pp. 230–240; M. J. Herskovits, *Man and His Works* (New York: Knopf, 1948), pp. 531–532.

[2] This expression is used in the Jamaican Revivalist cults (Pocomania and Revival Zion) to refer to the process of removing evil spirits by ritualistic means.

FIGURE 1.2 – Ordre de lecture complexe dans un document multi-colonnes où la note de bas de page prolonge la première colonne à partir de sa moitié

Les erreurs de segmentation se situent majoritairement au niveau des lignes et des blocs. Les erreurs de segmentation des mots sont plus rares en général. En ce qui concerne les caractères, le cas est trop rare pour être envisagé. Il peut survenir dans le cas de documents de mauvaise qualité où les caractères ont été abimés, coupés etc. On distingue 5 types d'erreur de segmentation :

1. **Fusion**
   (a) **Verticale**
   Cette erreur ne modifie pas l'ordre de lecture en général mais elle peut le faire si l'ordre est particulier. Si cette erreur se produit sur un bloc, alors elle est indétectable au niveau des lignes (les lignes ne sont pas modifiées). Si elle se produit sur une ligne alors la reconnais-

sance va sûrement échouer puisqu'on tentera de reconnaître une ligne dans une image en contenant deux. Le premier cas est bénin tandis que le dernier cas est grave puisqu'il est quasiment certain qu'on aura une mauvaise reconnaissance.

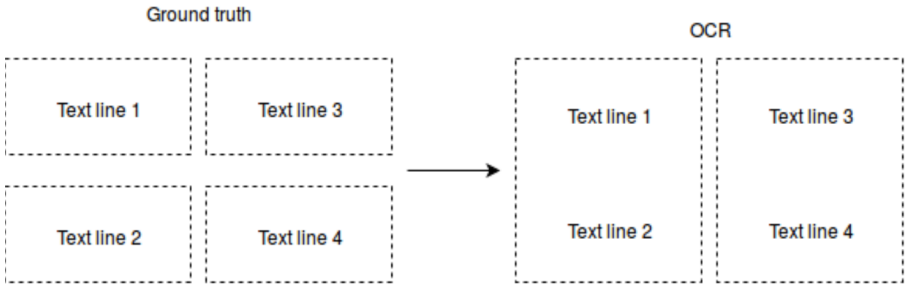


FIGURE 1.3 – Erreur de segmentation : Fusion verticale de blocs. Les numéros donnent l'ordre de lecture des lignes

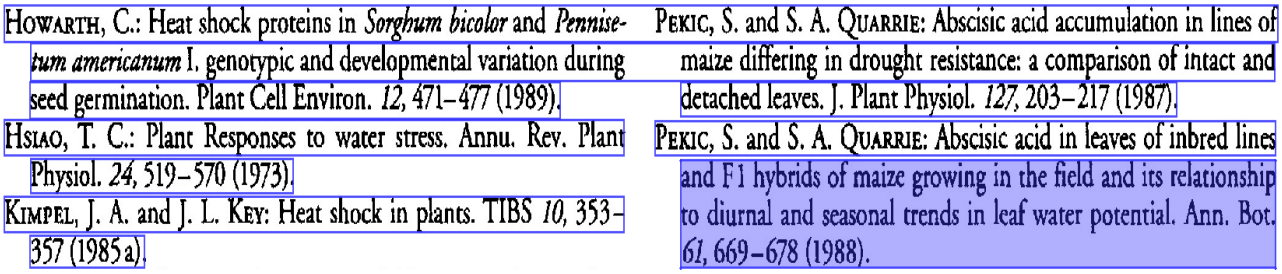


FIGURE 1.4 – Exemple de cas réél d'erreur de segmentation : Fusion verticale de deux lignes par l'OCR OCRopus

(b) **Horizontale**
Cette erreur conduit à la fusion des lignes adjacentes horizontalement. Elle modifie l'ordre de lecture du document.
C'est une erreur de segmentation grave car elle associe deux zones séparées physiquement. De plus elle peut être difficilement détectable car elle dépend du contenu qui lui aussi peut être erroné.

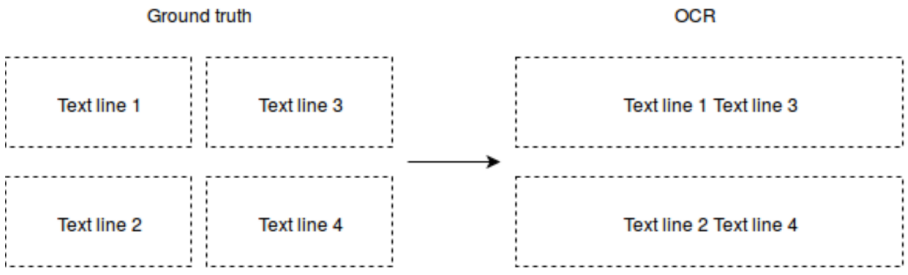


FIGURE 1.5 – Erreur de segmentation : Fusion horizontale

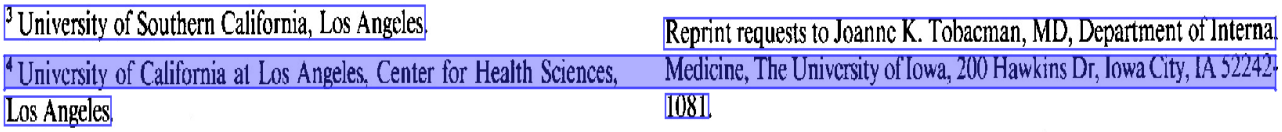


FIGURE 1.6 – Exemple de cas réél d'erreur de segmentation : Fusion horizontale de deux lignes par l'OCR OCRopus

2. **Fission**

(a) **Verticale**

La fission verticale des blocs au même titre que la fusion verticale est bénigne. Elle est aussi non visible au niveau des lignes (lorsqu'elle est sur les blocs). Si elle s'effectue sur des lignes alors cette erreur est grave car on devra reconnaître une ligne en ayant la moitié ou un morceau. Dans ce dernier cas, on est quasiment sûr que la reconnaissance échouera.

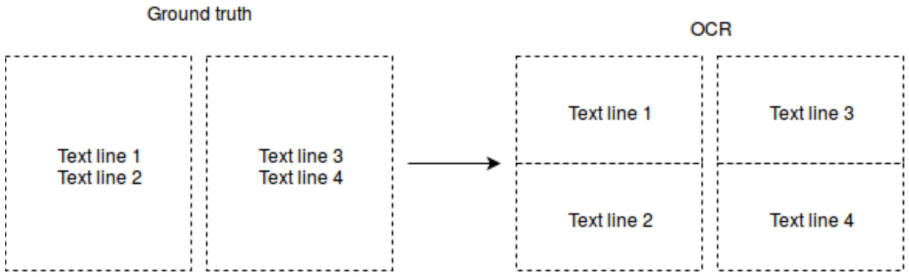


FIGURE 1.7 – Erreur de segmentation : Fission verticale

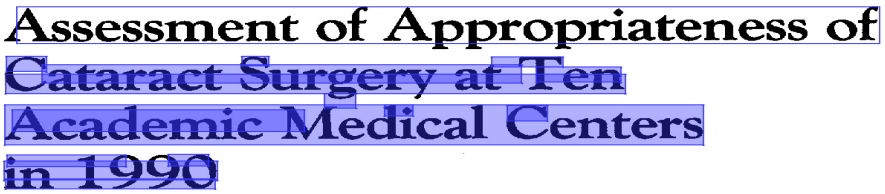


FIGURE 1.8 – Exemple de cas réél d'erreur de segmentation : Fission verticale de deux lignes par l'OCR OCRopus

(b) **Horizontale**

Si on se place au niveau des lignes, la fission horizontale découpera la ligne concernée en au moins 2 morceaux. On peut détecter cette erreur si la fission est contenue dans un bloc et que les lignes avant et après font la taille du bloc. Sinon, une vérification par le contenu est nécessaire pour la détecter. On peut aussi trouver des mots coupés en 2 le plus souvent, ce qui peut poser problème lors d'une tâche requérant une comparaison de contenu. Elle ne modifie pas l'ordre de lecture lorsqu'elle est encadré par deux lignes, cependant on peut penser à un paragraphe coupé entièrement en deux, ce qui provoquera une vrai perturbation de l'ordre.

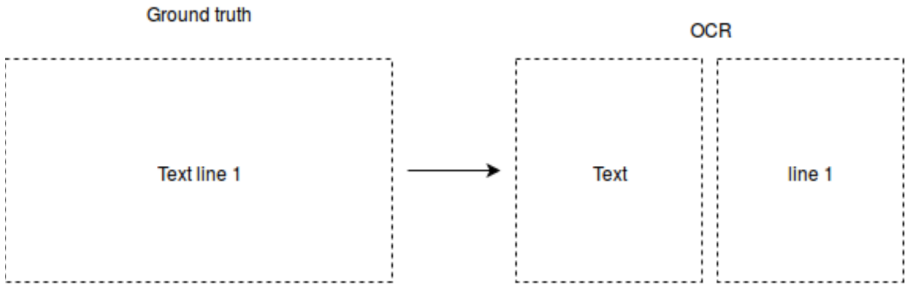


FIGURE 1.9 – Erreur de segmentation : Fission horizontale

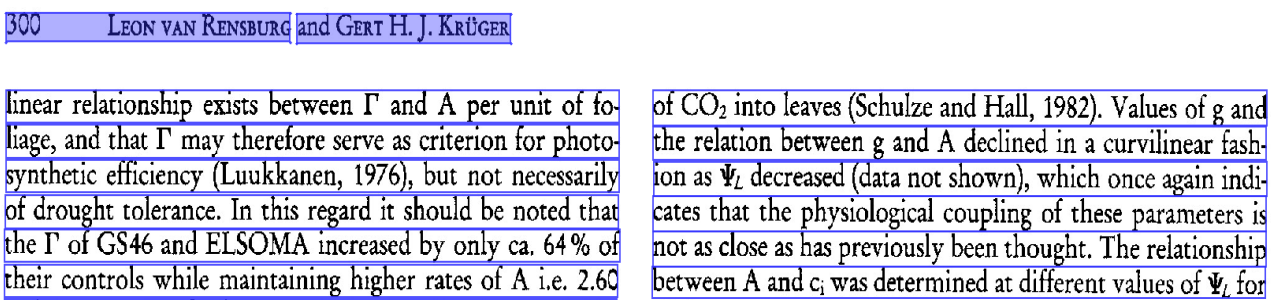


FIGURE 1.10 – Exemple de cas réel d'erreur de segmentation : Fission horizontale de deux lignes par l'OCR OCRopus

1.2.3 La reconnaissance de caractères

Définition

La reconnaissance de caractères consiste à identifier la forme à partir de image. Certaines méthodes ne vont pas reconnaître les caractères indépendamment les uns des autres mais au contraire, reconnaître des mots entiers (dans le cas où le nombre de mots à reconnaître est réduit).

Les erreurs

Les erreurs de reconnaissance se situent donc au niveau des caractères.

On distingue 3 types d'erreurs :

- **Délétion** : caractère manquant

$$word \longrightarrow wrd$$

- **Substitution** : un caractère remplacé par un autre (voire plusieurs)

$$word \longrightarrow w0rd$$
$$word \longrightarrow ivord$$

- **Insertion** : caractère inséré dans le texte

$$word \longrightarrow wordsd$$

La plupart des erreurs de reconnaissance de caractères proviennent d'une mauvaise interprétation de la morphologie des caractères à reconnaître conduisant parfois à une confusion avec d'autres caractères proches. Par exemple : "M" peut être compris comme un ensemble de lettre qui serait "I V I" qui correspondent au découpage de la lettre d'origine. Ces erreurs sont donc fonction de la police du texte et bien sûr de la connaissance de l'OCR sur cette police.

1.3 Contrainte

Dans le but d'évaluer les sorties des OCR, nous avons besoin de données de référence afin d'effectuer des alignements et des comparaisons. La sortie de référence (ou vérité) est soit du même format que la sortie de l'OCR soit un texte linéaire.

1.3.1 Sortie de l'OCR - Entrées du système

Les entrées du système d'évaluation sont deux fichiers contenant :
- La sortie de l'OCR.
- La vérité.

Ces deux entrées peuvent être un simple **texte** (suite de lignes) ou quelque chose de plus formaté avec une structure de lignes (au minimum) donné en **XML**.

Cas de l'entrée texte

On considèrera que le texte de l'image est donné sous forme de lignes qui sont les lignes de l'OCR. Sans ces lignes, nous aurons plus de difficultés à retrouver les erreurs de segmentation (elles se produisent majoritairement sur les lignes). Le listing 1.1 montre qu'il y a une ligne OCR par ligne du fichier.

```
1 REVIEWS
2 THE ROAD TO JUSTICE. By the RT. HON. Sir ALFRED DENNING.
3 [London: Stevens &amp;amp; Sons, Ltd. 1955. viii and 118 pp.
4 10s. 6d. net.]
5 In the last two years Lord Justice Denning has delivered a number of
```

Listing 1.1 – Exemple de fichier texte avec une segmentation en lignes

Cas de l'entrée XML

Lorsque l'entrée est un fichier XML, celui-ci peut contenir différentes informations sur le contenu telle que la disposition physique des paragraphes, la police, etc. Ces informations sont extraites par le système (ou corrigées dans le cas de la vérité).

La liste des tags et attributs donnés par tous les OCR dans le fichier XML est :
- Bloc ou ligne :
  - Position (coordonnées)
  - Contenu (suite de caractères)

La liste des tags et des attributs optionnels est :

1. Page :
   - Dimension de la page (hauteur x largeur)
   - Langage
2. Bloc :
   - Alignement (centré, à gauche, à droite)
   - Langage
3. Mot :

- Taux de confiance
- Langage

4. Caractère :
   - Taux de confiance

Nous disposons au moins de la boite englobante des lignes et des blocs. Si le système le permet, nous pouvons avoir plus de précisions sur ces blocs et lignes. De plus, nous avons parfois des informations concernant les mots ou même les caractères. Il est important de noter que nous ne pouvons utiliser que les informations fournies par tous les OCR, autrement, nous ne pourrons pas les comparer avec les mêmes critères si la méthode d'évaluation dépend de l'information disponible.

L'annexe 7 montre un extrait du XML donné par l'OCR "OCRopus". Ce fichier XML est construit suivant le format HOCR qui est une représentation standardisée des informations extraites par l'OCR. On y retrouve pour chaque ligne, la boite englobante ainsi que le contenu reconnu.

1.4 Objectifs

Les évaluations sont importantes car elles vont permettre de mieux comprendre le fonctionnement global d'un OCR. C'est en ayant les évaluations les plus fines que nous aurons les données nécessaires à une analyse en profondeur des limites de chaque étape de l'OCR étudié. Par exemple, on peut combiner deux OCR ensemble en conjuguant leurs forces et leurs faiblesses. L'objectif de ce rapport est de proposer des méthodes d'évaluation les plus étayées possible. Pour ce faire, nous proposons un état de l'art consistant avec une analyse des limites des méthodes actuelles. Puis nous dégagerons des méthodes d'évaluation qui porteront sur la segmentation et la reconnaissance de caractères.

Afin d'évaluer la segmentation et la reconnaissance des caractères, il est essentiel d'effectuer un alignement des zones et/ou des contenus (suivant le cas texte ou XML). L'alignement consiste à faire correspondre des éléments de la vérité avec des éléments du sytème. Ces éléments peuvent être des zones (boites englobantes, polygones) ou du contenu (mots, lignes, caractères...).

Nous allons maintenant passer à l'étude des méthodes d'évaluation pour les trois cas suivant d'entrées :

	Vérité	Système
1	Texte	Texte
2	XML	XML
3	Texte	XML

Table 1.1 – Les 3 types d'entrées donnés au système d'évaluation

2 Evaluation Texte / Texte

2.1 Enoncé du problème

Les entrées du système d'évaluation sont deux textes, identiques au format du listing 2.1. Il s'agit donc de comparer deux ensembles de lignes et de faire correspondre ces lignes entre elles. Ces lignes peuvent avoir subi des modifications à cause d'erreurs de segmentation. Le listing 2.1 correspond à la vérité, c'est-à-dire au texte tel qu'il se trouve sur l'image.

```
1 REVIEWS
2 THE ROAD TO JUSTICE. By the RT. HON. Sir ALFRED DENNING.
3 [London: Stevens &amp;amp; Sons, Ltd. 1955. viii and 118 pp.
4 10s. 6d. net.]
5 In the last two years Lord Justice Denning has delivered a number of
```

Listing 2.1 – Exemple de vérité de type texte

Le listing 2.2, qui est un exemple de sortie d'un OCR, correspond à une fission horizontale des 2ème et 3ème lignes du listing 2.1.

```
1 REVIEWS
2 THE ROAD TO JUSTICE.
3 By the RT. HON. Sir ALFRED DENNING.
4 [London: Stevens &amp;
5 Sons, Ltd. 1955. viii and 118 pp.
6 10s. 6d. net.]
7 In the last two years Lord Justice Denning has delivered a number of
```

Listing 2.2 – Exemple de fission horizontale de l'OCR OCRopus

Le listing 2.3 correspond à une inversion de l'ordre de lecture. Ici la ligne 1 du listing 2.1 est insérée entre la 4ème et la 5ème ligne.

```
1 THE ROAD TO JUSTICE. By the RT. HON. Sir ALFRED DENNING.
2 [London: Stevens &amp;amp; Sons, Ltd. 1955. viii and 118 pp.
3 10s. 6d. net.]
4 REVIEWS
5 In the last two years Lord Justice Denning has delivered a number of
```

Listing 2.3 – Exemple d'inversion de l'ordre de lecture de l'OCR OCRopus

Les lignes peuvent avoir subi toutes les erreurs de segmentation définies précédemment. On peut donc voir que la difficulté d'aligner les lignes découle directement de la qualité de la segmentation : plus la segmentation donnée par l'OCR est éloignée de celle de la vérité, plus il sera difficile de retrouver avec certitude la segmentation d'origine.

2.2 Etat de l'art

Feng et Manmatha [4] proposent d'utiliser les textes de vérité issus du site Project Gutenberg pour estimer les errreurs OCR. Ces livres publiques ont été vérifiés par des volontaires et sont dénués d'erreurs OCR. Un problème avec ces livres est qu'on ne dispose pas des lignes et des coupures de pages. Le livre correspond à une longue chaîne pouvant facilement faire 1/2 million de caractères. Les auteurs ont donc approché le problème comme étant un alignement entre la sortie de l'OCR et le texte du livre en utilisant un modèle de Markov caché (MMC ou HMM en anglais). Ils proposent d'utiliser les mots uniques du livre comme points de repères pour segmenter la chaîne de caractères du texte en de plus petites chaînes. Les points de repères correspondent aux mots uniques ayant une correspondance dans la vérité. Pour assurer la robustesse de la méthode, une vérification de la correspondance des n-grammes autour de ces mots uniques est nécessaire. En effet, les effets des erreurs de reconnaissance peuvent rendre un mot faux et transformer un mot en un autre. Les deux erreurs combinées mènent à une correspondance des mots uniques erronée (faux positif). Les segments ainsi obtenus sont alignés individuellement en utilisant un HMM et sont concaténés dans leur ordre d'origine. Cette technique découpe le problème en une multitude de sous problèmes qui requièrent moins de calcul et de mémoire. Cette technique fonctionne bien car on trouve beaucoup de mots uniques malgrè les erreurs de l'OCR. Une des limites est que dans certaines situations, la distance entre deux repères peut être longue, ce qui rend les calculs plus coûteux. De même, ces repères dépendent des mots uniques qui ne sont pas obligatoirement disposés dans le même ordre. Donc si l'ordre est entièrement perturbé, l'alignement échouera. La seconde remarque que nous pouvons faire est que le MMC requiert des probabilités qui doivent être estimées par un entrainement, ce qui nécessite une grande quantité d'information. Aussi, ces données peuvent changer suivant l'OCR et le corpus utilisé.

*
* *

L'approche [5] est très similaire au précédent modèle HMM. La différence est que ce modèle est hiérarchique, et donc aligne le texte hiérarchiquement en prenant les positions comme des états. On calcule la probabilité de générer une sortie de l'OCR en donnant toutes les positions possibles de la vérité. Il y a une correspondance des états pour chaque position dans la vérité. En comparaison avec un algorihme d'alignement basé sur la distance d'édition entre deux chaînes, cette méthode permet de prendre en compte les modifications effectuées par l'OCR sur un corpus (par sa phase d'apprentissage). Cette méthode n'utilise pas les mots uniques ce qui permet d'utiliser cette méthode lorsqu'ils ne sont pas présent en grand nombre dans les documents. Cependant, il reste la nécessité d'un apprentissage important qui peut être difficile lorsqu'on dispose de peu de données. De plus, les auteurs ne discutent pas des inversions qui peuvent être sources d'erreurs.

*
* *

Les auteurs de la méthode [1] utilisent un framework **W**eighted **F**inite **S**tate **T**ransducer (automate) pour l'alignement de la vérité avec des documents historiques difficiles. Ils représentent le texte comme étant un FST avec des variations, permettant au système de gérer les erreurs OCR. On peut noter, que cette méthode requière une bonne segmentation, ou des erreurs ne modifiant pas trop les lignes d'origine. Cet article traite exactement le problème d'alignement de deux textes contenant des erreurs OCR et une segmentation perturbée. Ils utilisent un système de poids sur les transitions représentant les opérations afin de rendre l'automate plus flexible. Le problème concerne la façon

dont la structure est modifiée. Les auteurs tentent d'aligner des textes issus de livres, ce qui limite les erreurs de segmentation. Ils prennent en compte différentes coupures de lignes (fission horizontale) pour aligner. Ils ne prennent pas en compte la fusion verticale et horizontale. Concernant les inversions, ils prennent une fenêtre de quelques lignes sur lesquelles ils appliquent l'automate.

*
* *

La plupart de ces méthodes ne prennent pas en compte une forte perturbation de la segmentation ou d'une inversion de l'ordre de lecture. Ces méthodes sont principalement appliquées sur des livres qui sont des structures linéaires : leur ordre de lecture est clair et indiscutable. Dans ce cas, l'OCR ne peut pas faire d'inversion de lignes ou de paragraphes. De plus, on ne trouve pas de fusion horizontale (par exemple : fusion de 2 colonnes dans un document multi-colonnes), ce qui limite les cas d'erreurs majeures.

*
* *

L'objectif de l'article [10] est d'évaluer la précision des OCR sur des livres scannés.

L'approche est la suivante : dans un premier temps, les auteurs identifient les mots uniques contenus dans le livre. Les mots uniques de la vérité et de la sortie OCR sont alignés en utilisant l'algorithme de la plus longue chaîne commune (**L**ongest **C**ommon **S**ubsequence). Les textes sont ensuite segmentés relativement aux mots uniques (chaque segment est un morceau de texte entre deux repères). Chaque segment peut maintenant être pensé comme étant un document entier. Si on regarde le contenu de chaque segment, nous allons pouvoir trouver de nouveau des mots uniques. Chaque segment peut donc être aligné en utilisant ses propres mots uniques. Cette méthode est récursivement appliquée jusqu'à ce que les segments soient d'une taille assez petite pour utiliser un algorithme d'alignement de chaînes, basé sur une distance d'édition. Cette approche est très rapide en terme de coût de calcul. Il est important de noter que les mots uniques doivent avoir le même ordre dans la vérité et dans la sortie de l'OCR sinon ils seront ignorés. Un exemple de la méthode est décrit par la figure 2.1. La première limite évidente est l'inversion d'ordre entre les deux textes qui n'est pas prise en compte. L'algorithme échouera lorsqu'il y aura une sévère perturbation de la segmentation. Comme précédemment, il est aussi basé sur les mots uniques (qui sont présent dans une quantité dépendante des performances du système).

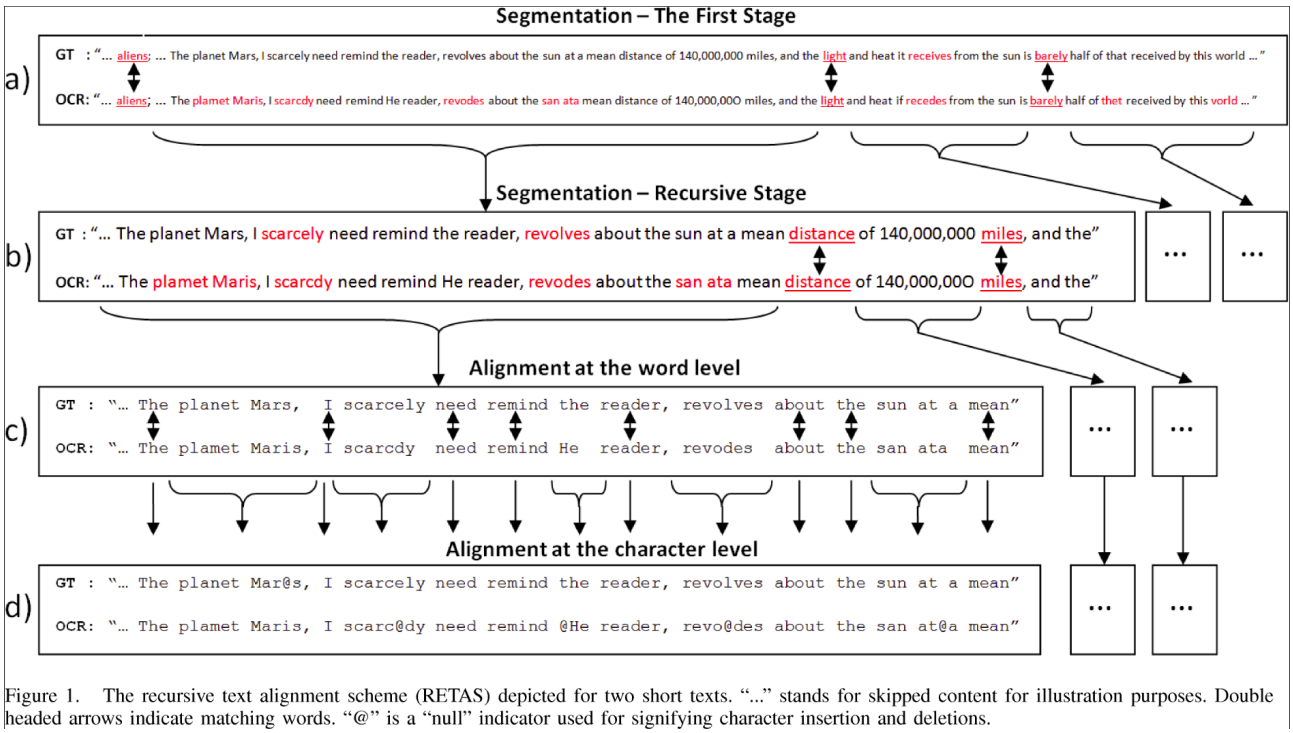


Figure 1. The recursive text alignment scheme (RETAS) depicted for two short texts. "..." stands for skipped content for illustration purposes. Double headed arrows indicate matching words. "@" is a "null" indicator used for signifying character insertion and deletions.

FIGURE 2.1 – Algorithme RETAS - Exemple d'alignement

2.3 Choix de la méthode

Dans notre cas, nous ne prenons pas en compte toutes les erreurs de segmentation car il est très difficile de faire l'alignement quand la structure est entièrement détruite. Il est surtout compliqué d'éviter de faire de mauvais alignements sans avoir d'informations sur la structure physique du document alors que celle-ci a été détruite par l'OCR.

Nous avons donc choisi d'utiliser le programme RETAS. Il nous permet d'évaluer les textes qui sont entièrement linéaires avec une très bonne précision. Malheureusement, il échoue dès qu'un cas correspond à une des limites présentées précédemment en donnant de faux résultats. Nous avons fait notre propre implémentation (en nous aidant des sources) pour pouvoir l'intégrer correctement dans notre logiciel d'évaluation.

2.4 Corpus

Afin de tester cette méthode et de montrer que les remarques que nous avons faites sont valides, nous nous proposons d'utiliser 2 corpus :

1. Le premier est composé de 49 documents **linéaires** de type livres, articles mono-page. Ce sont des documents ayant une structure physique très simple avec peu de variations possibles. Ils proviennent de plusieurs éditeurs et de plusieurs sources. Il y a des articles scientifique en français, en anglais, en allemand et en espagnol, de plusieurs époque différentes (de 1942 jusqu'en 2000), avec des polices différentes. Certains documents possèdent des zones graphiques
2. Le second corpus est composé de 5 documents **non linéaires** de type articles multi-colonnes de journaux scientifiques. Ce sont des documents ayant une structure variant de simple à très complexe. Quatre parmi cinq des documents sont des articles à 2 colonnes avec certains

possèdant des parties linéaires entre les colonnes. Le dernier document est un document à 3 colonnes. On a aussi des entêtes (première page d'un article) et des références (dernière page d'un article). L'ensemble du corpus peut être retrouvé en annexe 9.

2.5 Evaluation

Les corpus que nous avons, nous permettent d'évaluer la méthode en utilisant un OCR et en contrôlant sa sortie. Ils nous permettent aussi d'évaluer les OCR, qui est l'objectif principal. L'évaluation portera uniquement sur la reconnaissance des caractères car, actuellement, aucune technique ne permet d'évaluer la segmentation à partir de deux textes.

2.5.1 Evaluation de la méthode

Pour l'évaluation de la méthode, nous utilisons la sortie de l'OCR "Tesseract" qui est, parmi les OCR libres, le meilleur au niveau de la segmentation. De plus, nous allons aussi utiliser "OCRopus", qui, lui, possède un algorithme de segmentation moins performant, ce qui nous permettra de comparer les résultats et d'observer des variations. Nous avons réalisé l'alignement vérité pour chaque document et pour chaque OCR. Ces vérités nous permettent de calculer exactement le taux de reconnaissance des caractères, et ce, même lorsque la structure du document est détruite. Ce taux est appelé taux de reconnaissance théorique. En effet, lors de la réalisation de la vérité, nous corrigeons les erreurs de segmentation en changeant l'ordre des lignes si nécessaire et en rétablissant les fissions et fusions. Nous pouvons ainsi calculer une distance entre l'évaluation de la méthode et la vérité pour en observer la précision.

Corpus$_1$

OCR	Performance théorique	Performance réelle	Variation	Fission verticale	Fission horizontale	Fusion verticale	Fusion horizontale
Tesseract	97.70	96.94	0.76	0	0	0	0
OCRopus	96.44	95.80	0.64	5	20	5	0

Table 2.1 – Algorithme RETAS - Résultats théorique et réel de Tesseract et OCRopus sur le *corpus*$_1$

Le tableau 2.1 donne les résultats de deux OCR sur le *corpus*$_1$ avec d'une part un alignement théorique effectué à la main, et d'autre part, l'alignement effectué par la méthode RETAS. À la suite de cet alignement, un taux de reconnaissance est calculé en utilisant la distance d'édition. Les valeurs représentent la moyenne du corpus. La même technique est appliquée sur les deux alignements. On obtient des résultats assez similaires en terme de distance entre l'alignement théorique et celui effectué. On peut remarquer que malgrè la présence d'erreurs pour l'OCR OCRopus, la variation n'est pas plus importante. Il est important de noter que dans cette comparaison, il est possible de trouver des effets de bord qui vont conduire à des compensations. En effet, comme la distance d'édition va donner un score de correspondance, même si l'alignement est faux, le score ne sera pas obligatoirement nul. De la même façon, il est possible d'obtenir un meilleur score avec un faux alignement qu'avec l'alignement réel.

La figure 2.2 montre pour chaque image du corpus, la distance entre la précision théorique et celle réalisée, en rouge tandis que la performance réalisée est représentée en vert. Sur l'axe des ordonnées, on trouve le taux de caractères correctement reconnus et sur l'axe des abscisses le numéro du

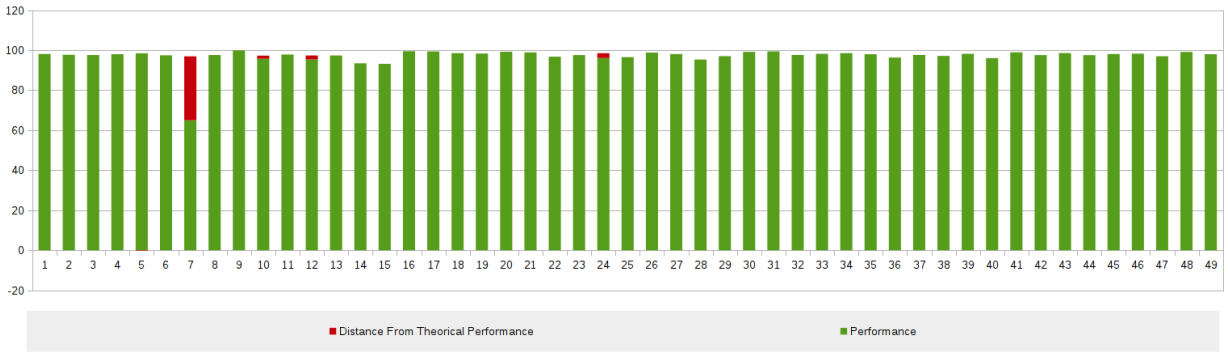


FIGURE 2.2 – Algorithme RETAS - Résultat de Tesseract $corpus_1$

document concerné. Ici, les données sont celles de l'OCR Tesseract. On peut facilement remarquer qu'à l'exception d'un document, il n'y a pas beaucoup d'écart entre la théorie et la réalité. Les erreurs sont dues à des inversions dans l'ordre de lecture. Il faut aussi noter que Tesseract n'a pas fait d'erreur de segmentation sur l'ensemble du corpus.

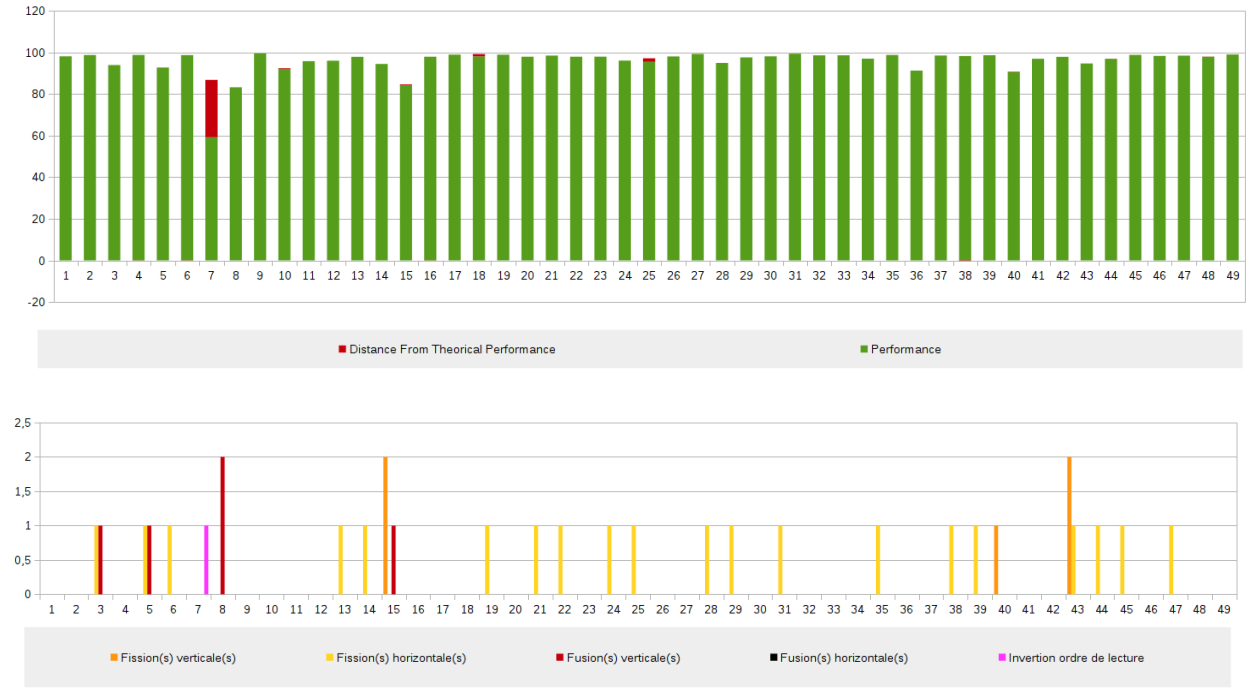


FIGURE 2.3 – Algorithme RETAS - Résultat d'OCRopus $corpus_1$ accompagné de ses erreurs de segmentation pour chaque document

La figure 2.3 correspond aux performances de l'OCR OCRopus. La différence entre Tesseract et OCRopus est minime par rapport aux résultats. De même que les erreurs de segmentation ne semblent pas avoir perturbé le bon alignement de la méthode. Ces erreurs dans le cadre d'un contexte linéaire ne sont pas graves car elles sont encadrées par des parties linéaires, donc elles se limitent à une position très restreinte.

La seule distance importante (pour les deux OCR) est due au document 7 présent en annexe 8. L'erreur sur ce document correspond à une inversion de paragraphes dans l'ordre de lecture (entre la vérité et le système). La présence de ce document dans le $corpus_1$ (documents linéaires) peut être remise en cause car on peut donner plus d'un ordre de lecture et il n'est donc pas indiscutable.

Concernant les autres erreurs, elles sont dues aux effets de bord de la distance de Levenshtein énoncés précédemment.

Corpus$_2$

OCR	Performance Théorique	Performance Réelle	Variation
Tesseract	96.23	89.10	7.13
OCRopus	94.68	71.75	22.93

TABLE 2.2 – Algorithme RETAS - Résultat de Tesseract et d'OCRopus sur le *corpus*$_2$ comparé aux résultats théoriques

Le tableau 2.2 montre les résultats obtenus sur le corpus$_2$. Ces résultats étaient attendus ; ils confirment les limites de la méthode que nous avons énoncées plus tôt, à savoir, la non prise en compte de possibles inversions de l'ordre de lecture, ainsi que des erreurs de segmentation.

La figure 2.4 représente les résultats de Tesseract. Les documents 1, 4 et 5 ont des performances, à quelques pourcents près, équivalentes à celles de l'alignement théorique. Seuls les documents 2 et 3 ont subi des variations au niveau de la méthode d'alignement. Les erreurs à la source de ces distances sont principalement des inversions dans l'ordre de lecture ainsi que quelques rares fusions horizontales.

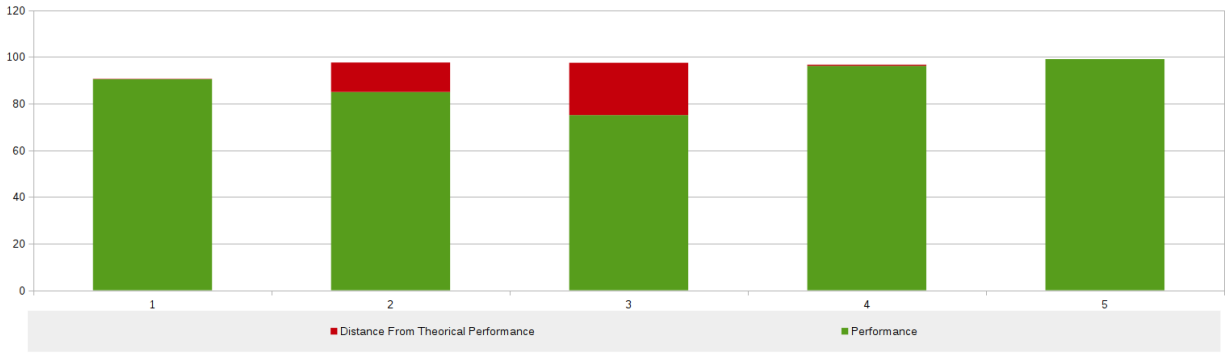


FIGURE 2.4 – Algorithme RETAS - Résultat de Tesseract sur le *corpus*$_2$

La figure 2.5 correspond à la représentation des résultats d'OCRopus. Contrairement à Tesseract, chaque document a vu sa structure physique modifiée par l'OCR. Ces erreurs, lorsqu'elles surviennent sur des documents non linéaires, sont très graves. Une fusion horizontale de deux lignes appartenant à deux colonnes différentes créera une double erreur : au moment d'aligner, la seconde ligne sera considérée comme du bruit par rapport à la première colonne, et elle ne sera pas trouvée pour la seconde colonne. C'est une multiplication de ces erreurs qui a conduit à un accroissement de la distance entre le taux de reconnaissance mesuré et réel.

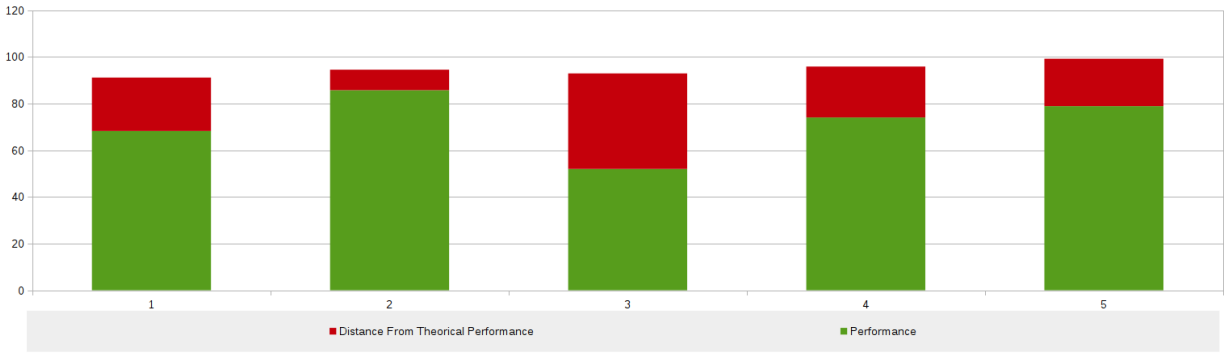


FIGURE 2.5 – Algorithme RETAS - Résultat d'OCRopus sur le $corpus_2$

2.5.2 Evaluation des OCR

Étant donné les résultats de l'évaluation de la méthode, on peut voir que si l'on souhaite évaluer la reconnaissance des caractères d'un OCR, il faut limiter les erreurs de segmentation qui fausseraient l'évaluation. Un choix évident serait d'utiliser des documents comme ceux du $corpus_1$ car nous avons vu que sur les documents linéaires, les erreurs de segmentation ne perturbent pas trop l'alignement. De plus, les évaluations de Tesseract et OCRopus ont déja été effectuées sur ce corpus.

Tesseract

Il semblerait que ce soit le meilleur OCR en terme de performances. La méthode de segmentation employée montre moins d'erreurs de segmentation alors qu'OCRopus va souvent altérer la structure physique du document. Concernant la reconnaissance, le $corpus_1$ étant celui ayant le plus de documents et le plus fiable, on peut voir que Tesseract atteint un taux de reconnaissance de 97.70 %.

OCRopus

Dans l'ensemble, OCRopus est en dessous des performances de Tesseract en termes de reconnaissance de caractères. Cependant, on peut noter qu'OCRopus fait beaucoup plus d'erreurs de segmentation. D'un point de vue de la reconnaissance, on peut noter 96.44 % de taux de reconnaissance sur le $corpus_1$. Il n'est pas loin de son concurrent mais reste quand même en dessous.

2.6 Discussion

2.6.1 Conclusion

Nous avons étudié la méthode RETAS et testée sur 2 corpus composés de documents linéaires et de documents non linéaires. Les résultats sur le premier corpus sont corrects et montrent que la méthode est efficace sur les documents linéaires. Les résultats sur le deuxième corpus, montrent que cette méthode ne fonctionne plus correctement lorsqu'il y a des inversions dans l'ordre de lecture mais aussi lorsque les erreurs de segmentation sont multiples sur une zone du document. Il est naturel de dire que plus la structure physique d'un document est altérée, plus il sera difficile de

recouper les morceaux entre eux. Cependant, plus la méthode d'alignement pourra retrouver la structure d'origine, plus la mesure de la reconnaissance sera précise et fiable.

Nous avons ensuite observé les résultats de Tesseract et OCRopus sur le corpus_1. Ces résultats nous ont permis de dire que le premier a un meilleur algorithme de segmentation, car contrairement à son concurrent, il fait très peu d'erreurs. De même, il est légèrement au-dessus d'OCRopus en terme de taux de reconnaissance avec un écart d'environ 1%. Concernant les performances d'OCRopus, il est important de noter qu'obtenir une bonne segmentation est nécessaire pour effectuer une bonne reconnaissance. Il serait interressant de voir sur une segmentation donnée, si OCRopus fait mieux que Tesseract en terme de reconnaissance pure.

2.6.2 Améliorations

Une amélioration possible serait d'avoir une méthode qui permet d'effectuer un alignement en prenant en compte toutes les erreurs de segmentation. Cette méthode ne serait pas basée sur les mots uniques car ils sont fortement dépendant des systèmes utilisés et du corpus, mais les utiliserait s'il sont présents. Le problème est très compliqué car il faut déduire la structure physique du document à partir du texte de vérité en utilisant le texte du système, en sachant que cette structure est potentiellement erronée.

3 Evaluation XML / XML

3.1 Enoncé du problème

Le problème dans le cas de deux fichiers XML en entrée est qu'il faut aligner les zones de ces XML. Cet alignement va permettre de faire la comparaison des zones (obtenir des données sur leurs précisions, leur types d'erreurs), donc l'évaluation de la segmentation. Puis, nous pourrons déterminer à partir de l'alignement précédent, les morceaux d'informations (lignes, mots...) qui correspondent entre eux. Cette dernière étape nous permettra d'évaluer la reconnaissance du contenu.

Nous verrons dans un premier temps l'état de l'art sur le problème, puis nous verrons plus en détail la méthode sélectionnée "Zonemap" ainsi que ses limites. Ensuite nous étudierons comment cette méthode a été adaptée et nous comparerons les deux méthodes obtenues, sur un corpus. Enfin nous analyserons les résultats des deux OCR étudiés par rapport aux résultats précédents (partie Texte / Texte).

3.2 Etat de l'art

Les auteurs de [7] présentent le logiciel PETS (**P**erformance **E**valuation **T**ool**S**) offrant plusieurs méthodes d'évaluation basées sur les pixels. Ces méthodes d'évaluation reposent sur un alignement des zones entre elles. Ces zones peuvent être évaluées suivant des critères définis comme suit :

Classification de la zone

Une zone est pénalisée lorsque son type n'est pas le même que celui de la vérité.

Segmentation de la zone

La précision de la segmentation de la page est calculée comme étant le pourcentage des lignes de la vérité contenues correctement dans des zones résultats sans modifications. Cette méthode ne tolère pas de fission verticale.

Correspondance de la zone

Comme la segmentation de zones mais doit avoir une contrainte de type de zone (classification de la zone) pour être candidat à la correspondance avant de pouvoir établir une fusion ou une intersection.

Les zones sont représentées par des rectangles ou des polygones. Dans le cas de superpositions de zones, les pixels sont associés à une zone en utilisant l'encodage "run-length". La métrique utilisée par PETS est basée sur les superpositions de pixels pour faire correspondre les zones de la vérité et celles du système. Le coût des méthodes basées sur le pixel est très important, c'est pourquoi les

auteurs calculent une boîte englobante de superposition. Si la superposition de la boite est supérieure à un seuil défini par l'utilisateur, alors on calcule la superposition au niveau des pixels.
Soient $G = \{g_1, g_2, ..., g_n\}$ l'ensemble des zones de la vérité et $R = \{r_1, r_2, ..., r_m\}$ l'ensemble des zones du système.

Les métriques sont les suivantes :

1. $TruePositive(TP) = \{p \mid p \in g_i \wedge p \in r_j\}$
2. $FalsePositive(FP) = \{p \mid p \notin g_i \wedge p \in r_j\}$
3. $FalseNegative(FN) = \{p \mid p \in g_i \wedge p \notin r_j\}$
4. $Precision = \frac{TP}{TP+FP}$
5. $Recall = \frac{TP}{TP+FN}$
6. $F1 = \frac{2*Precision*Recall}{Precision+Recall}$

Ces métriques sont utilisées pour construire une table de scores de correspondance entre les zones de la vérité et celles du système. Les zones ayant un score de correspondance au-dessus d'un seuil défini par l'utilisateur, sont dites chevauchantes. Dans le cas contraire, leurs valeurs sont remises à zéro (les zones ne correspondent pas). Les auteurs utilisent cette technique pour aligner les zones de la vérité avec les zones du système dans une des quatre configurations suivantes :

1. One-to-one
2. One-to-many
3. Many-to-one
4. Many-to-many

Une liste $M = \{G_m, R_m\}$ contiennent des paires $\{\cup Z_g, \cup Z_r\}$ des unions des zones qui correspondent. C'est en regardant la cardinalité de ces paires que nous obtenons une des 4 observations suivantes :

1. Manqué (Missed) : Z_g n'a pas de correspondance Z_r
2. Sur-detection (False Alarm) : Z_r n'a pas de correspondance Z_g
3. Match : la cardinalité de Z_g et Z_r est de 1
4. Multi-match : la cardinalité de Z_g ou Z_r est supérieure à 1 et aucun des deux est nul.

L'algorithme utilise beaucoup de seuils définis par l'utilisateur pour permettre une personnalisation. Cependant cela rend le système très rigide. Certes, on réussira à évaluer correctement la majorité des cas mais on aura aussi plus de faux positifs et de vrais négatifs qu'avec un système plus souple. De plus, cette méthode ne respecte pas nos contraintes car nous ne pouvons pas faire de calcul basé sur les pixels car nous ne possédons que les boites englobantes.

*
* *

Dans la compétition ICDAR 2007, les auteurs [2] proposent une méthode dont les zones sont représentées par des polygones. Ces zones sont converties en un ensemble d'intervalles afin de pouvoir effectuer des calculs rapidement.

Correspondance

On cherche à trouver les endroits de superpositions entre les éléments vérité et système.

Les superpositions sont détectées entre :
— Un intervalle de la segmentation et rien.

— Un intervalle de la segmentation et la vérité.
— Un intervalle de la vérité et rien.

Une fois les intersections trouvées, il est assez simple de déterminer les configurations suivantes :
— Une région système qui n'a pas d'intersection avec au moins une région de la vérité est un élément faussement détecté.
— Une région vérité qui est superposée entièrement par un élément système est une région correctement détectée.
— Une région vérité qui est superposée entièrement ou partiellement par plus d'un élément système est une région segmentée.
— Une région système qui superpose entièrement ou partiellement plus d'un élément vérité est une fusion de régions.
— Une région vérité qui n'est pas superposée entièrement par un élément système est une région correctement détectée de manière partielle.
— Une région vérité qui n'est pas superposée par au moins un élément système est une région manquée.

Qualification et quantification des erreurs

Les 5 types d'erreurs définis précédemment auront une importance différente suivant :
— Le contexte interne au document.
— Le scénario de l'application.

Exemples

— Contexte
  — Une fusion verticale entre deux blocs de texte est une erreur insignifiante.
  — Une fusion entre un bloc de texte et une image (figure) est une erreur importante.
  — Une fusion entre deux colonnes de texte est une erreur importante.
  — Une fusion entre un bloc de texte et une région graphique est une erreur importante.
— Scénario
  — Une fusion entre deux régions graphiques est une erreur insignifiante dans le cas d'un OCR.
  — Une fusion entre une entête et un bloc de texte n'est pas une erreur importante en général sauf lorsqu'on veut extraire l'entête.

*
* *

Dans ICDAR 2011, la méthode [3] reprend la même représentation que celle d'ICDAR 2007. Cependant la correspondance est différente. Utiliser des intervalles est moins complexe pour trouver les superpositions qu'une approche par pixel. Les associations entre des éléments vérités et systèmes sont inscrits dans 2 tables :

1. La première marque les relations de la vérité par rapport au sytème.
2. La seconde marque les relations du système par rapport à la vérité.

Ces tables permettent d'identifier les 4 types de segmentation suivants :
— Merge : une zone système superpose plus d'une zone de la vérité.
— Split : une zone vérité superpose plus d'une zone du système.
— Miss : zone vérité non associée.
— False detection : zone système non associée.
— Match : une zone du système superpose entièrement une zone de la vérité.

Cette méthode utilise des poids (poids sur les types de zones, les types d'erreurs) pour permettre la personnalisation de l'évaluation (en fonction de l'importance de ces zones). L'approche par les intervalles permet de réduire les calculs par rapport à une approche basée sur les pixels. La limite de cette méthode est qu'elle ne prend pas en compte les erreurs multiples sur les zones. De même que si un intervalle du système est légérement sur un intervalle de la vérité, on considèrera l'intersection comme étant correct. De plus, cette technique n'explique pas comment elle gère les superpositions naturelles dans la vérité.

*
* *

Les auteurs dans [8] proposent un algorithme basé sur les pixels. Ils calculent un "weighted bipartite graph" qu'ils appellent "pixel-correspondance graph" dont chaque nœud représente une segmentation. Un bord est construit entre deux nœuds tel que le poids du bord soit égal au nombre de pixels dans l'intersection entre deux segments (comme un nœud représente un segment). Si les segments ne se superposent pas alors aucun bord n'est construit. Si la segmentation du système correspond parfaitement à celle de la vérité alors c'est un "Match" parfait. S'il y a des différences entre les segments, le graphe ne sera pas un "Match" parfait. Au contraire, un nœud représentant le système ou la vérité peut avoir plusieurs bords.

Soit P le nombre total de pixels correspondant à un nœud. Soit M le nombre de bords d'un nœud. Soit w_i , avec $i = 1, 2, ...M$, le poids de chaque bord.

Alors $P = \sum_{i=1}^{M} w_i$

Pour chaque nœud du graphe, w_i/P donne la fraction de pixels se superposant avec chacun de ses nœuds correspondants.

Un bord entre deux nœuds est considéré significatif si $w_i/P \geq t_r$ or $w_i \geq t_a$ ou t_r est un seuil relatif à la fraction de superposition. Ils utilisent une valeur de $t_r = 0{,}1$. Pour éviter de passer à côté de certaines zones, ils mettent en place un autre seuil t_a qui lui est absolu. Ils le font varier suivant le type de la segmentation et la taille de zones. Par exemple, ils utilisent $t_a = 500$ pixels pour une segmentation au niveau des zones et $t_a = 100$ pixels au niveau des lignes.

Pour un nœud de la vérité, s'il y a plus d'un bord du système alors le nœud est considéré comme étant une sur-segmentation. A l'inverse, si pour un nœud du système, il y a plus d'un bord de la vérité, alors le nœud est considéré comme étant une fusion.

Ils mettent en place les métriques suivantes :

- Total sur-segmentation (T_o) : le nombre total de bords significatifs de la vérité moins le nombre d'éléments de la vérité.
- Total sous-segmentation (T_u) : le nombre total de bords significatifs du système moins le nombre d'éléments du système.
- Éléments sur-segmentés (C_o) : le nombre d'éléments de la vérité ayant plus d'un bord significatif.
- Éléments sous-segmentés (C_u) : le nombre d'éléments du système ayant plus d'un bord significatif.
- Éléments manqués (C_m) : le nombre d'éléments de la vérité qui ne correspondent pas au système.
- Éléments sur-détectés (C_f) : le nombre d'éléments du système qui ne correspondent pas à la

vérité.

Les calculs sont coûteux car ils sont basés au niveau des pixels. Cependant la mesure est très précise. Ils n'expliquent pas comment ils opèrent lorsqu'il y a des intersections naturelles dans la vérité. De plus, ils ne prennent pas en compte les relations multiples (many - many) qui peuvent exister dans les cardinalités.

*
* *

Wolf et Jolion [9] proposent un framework pour l'évaluation de la position de lignes de texte. Le principe de cette méthode est que l'évaluation est effectuée au niveau des objets (boîtes englobantes) et prend en compte la qualité de chaque correspondance entre les boîtes du système et celles de la vérité. Les correspondances ("matchs") sont déterminées par la superposition des zones respectant un seuil de qualité de correspondance minimum. Ensuite, des poids pour les relations un-à-un, un-à-plusieurs et plusieurs-à-plusieurs, sont utilisés pour calculer les résultats. Deux seuils sur la précision de la surface (t_p) et sur le rappel de la surface (t_r), contrôlent la manière de faire correspondre les éléments. Ils proposent d'utiliser des valeurs par défaut soit t_r = 0.8 et t_p = 0.4. Afin de calculer le rappel et la précision sur l'ensemble, la méthode utilise toutes les correspondances de l'ensemble. On peut modifier les poids sur les relations, ce qui permet de ne pas pénaliser certains comportements qui sont bénins. Les principales limites ici, sont l'utilisation des deux seuils relatifs et le manque de détail sur la classification des erreurs des zones.

*
* *

La méthode Zonemap[6] a été développée par le LNE (Laboratoire National de Météorologie et d'Essais) dans le cadre de leur campagne d'évaluation d'OCR Maurdor.

Pour deux documents V (vérité) et S (système) , l'algorithme du ZoneMap se décompose en plusieurs étapes et va aboutir à un ensemble de groupes contenant des éléments de V et/ou de S :

1. On calcule la force du lien qui relie chaque élément de V à chaque élément de S. La force du lien entre deux boîtes englobantes b_v et b_s est donnée par la formule :

$$Lien(b_v, b_s) = (\frac{b_v \cap b_s}{b_v})^2 + (\frac{b_v \cap b_s}{b_s})^2$$

2. On ne conserve que les liens dont la force est non-nulle.
3. On trie les liens par ordre décroissant de force.
4. Pour chaque lien l qui associe les éléments b_v et b_s :
   — Si b_v est déjà dans un groupe, on essaye d'y insérer b_s
   — Si b_s est déjà dans un groupe, on essaye d'y insérer b_v
   — Sinon, on crée un groupe qui contient b_v et b_s
5. Une insertion ne peut réussir que si elle n'aboutit pas à un groupe qui contient plus d'un élément de V et plus d'un élément de S en même temps.
6. Si une zone n'a pas pu être insérée, on ne fait rien.
7. À la fin de l'algorithme, les éléments qui n'appartiennent à aucun groupe sont insérés dans un groupe à part.

Il faut ensuite calculer l'erreur E_i de chaque groupe i dans le but de calculer l'erreur globale $E_{ZoneMap}$.

$$E_{ZoneMap} = \frac{\sum_{i=1}^{N} E_i}{Area(R)} \text{ avec } E = (1 - \alpha_c)E_s + \alpha_c E_c$$

Ici E_s correspond à l'erreur de surface, E_c est l'erreur de classification de la zone avec $\alpha_c \in [0\,;1]$, le poids donné à l'erreur de classification. Le calcul de E_s et E_c dépend de la configuration du groupe.

Cet algorithme va permettre d'obtenir une série de groupes qui seront chacun dans l'une des cinq configurations suivantes :

1. Fausse alerte : quand le groupe ne contient qu'un élément de S. Cela signifie que l'on n'a pas pu aligner un élément de S, et que l'OCR a effectué une sur-détection.
$$E_s = Area(S_i)$$
$$E_c = E_s$$
2. Manqué : quand le groupe contient un élément de V. Cela signifie que l'OCR n'a pas reconnu la zone de référence.
$$E_s = Area(V_i)$$
$$E_c = E_s$$
3. Correspondance : quand le groupe contient exactement un élément de V et un élément de S. Cela signifie que l'élément de V a été correctement détecté.
$$E_s = Area(S_i \cup V_j - S_i \cap V_j)$$
$$E_c = d(t_S, t_V)Area(S_i \cap V_j) + E_s$$
Ici $d(t_S, t_V) \in [0; 1]$ correspond à la distance entre les classes des deux zones.
4. Séparation : quand le groupe contient exactement un élément de V et plus d'un élément de S. Cela signifie que l'élément de V a été segmenté en plusieurs boîtes englobantes.
$$E_s = Area(V_i \cap S_j) * \alpha_{MS} * |S_i|$$
avec $\alpha_{MS} \in [0; 1]$ le coefficient de Fusion/Fission (**M**erge/**S**plit en anglais) et $|S_i|$ la cardinalité de S_i
$$E_c = (|s_i| - 1 + min_{s \in S_i} d(t_s, t_v))$$
5. Fusion : quand le groupe contient plus d'un élément de V et exactement un élément de S. Cela signifie que les éléments de V ont été fusionnés en une seule boîte englobante.

Le calcul de l'erreur pour la fusion est le même que pour la séparation mais en inversant la zone système et la zone vérité.

La méthode Zonemap prend en compte certaines erreurs que l'OCR a pu commettre à la segmentation, mais ne prend pas tous les cas existants en compte. En particulier, elle ne fonctionne pas quand :

1. Des régions du document de vérité se chevauchent : l'algorithme va alors détecter des fusions ou des divisions de zones alors qu'il n'y en a pas. Voir figure 3.1

   Déroulement de l'algorithme

   - Liens : {A1} , {B1}
   - Liens triés : {B1} , {A1}
   - Iterations :
     - Iteration 0 : {A} , {B} , {1}
     - Iteration 1 : {B1} , {A}
     - Iteration 2 : {B1A}

— Résultat :
La zone 1 fusionne les deux zones A et B.
— Attendu :
La zone 1 correspond à la zone B. La zone A est manquée.

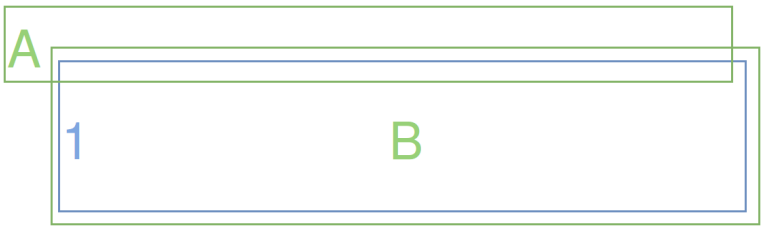


FIGURE 3.1 – Algorithme Zonemap : premier cas limite. En vert les éléments de référence, en bleu les éléments du système

2. Des régions sont à la fois fusionnées et divisées, c'est-à-dire quand une région va être divisée en plusieurs parties qui seront fusionnées à d'autres régions (voir figure 3.2).

Déroulement de l'algorithme

— Liens : {A1} , {A2} , {B1} , {B2}
— Liens triés : {A1} , {B2} , {A2} , {B1}
— Iterations :
  — Iteration 0 : {A} , {B} , {1} , {2}
  — Iteration 1 : {A1} , {B2}
— Résultat :
La zone 1 correspond à la zone A. La zone 2 correspond à la zone B.
— Attendu :
La zone 1 fusionne la zone A et la zone B. La zone 2 fusionne la zone A et la zone B. La zone A est sur-segmentée par la zone 1 et la zone 2. La zone B est sur-segmentée par la zone 1 et la zone 2.

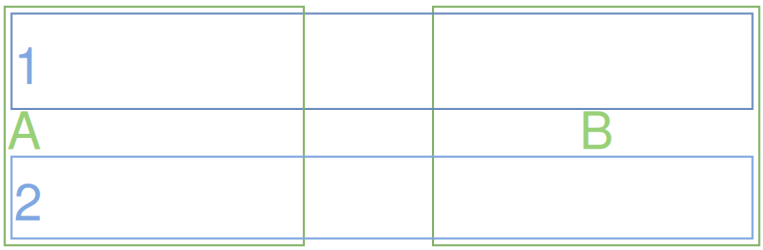


FIGURE 3.2 – Algorithme Zonemap : second cas limite. En vert les éléments de référence, en bleu les éléments du système

On peut, à la suite de ces cas limites, les regrouper dans un seul et unique cas qui synthétise ces limites (voir figure 3.3).

Déroulement de l'algorithme

- Liens : {A1} , {A2} , {B1} , {B2} , {C1} , {C2}
- Liens triés : {B1} , {B2} , {A1} , {A2} , {C1} , {C2}
- Iterations :
  - Iteration 0 : {A} , {B} , {C} , {1} , {2}
  - Iteration 1 : {B1} , {A} , {C} , {2}
  - Iteration 2 : {B12} , {A} , {C}
- Résultat :
  La zone B est sur-segmentée par la zone 1 et la zone 2. La zone A est manquée. La zone C est manquée.
- Attendu :
  La zone B est sur-segmentée par la zone 1 et la zone 2. La zone A est sur-segmentée par la zone 1 et la zone 2. La zone 1 fusionne la zone A et la zone B. La zone 2 fusionne la zone A et la zone B. La zone C est manquée.

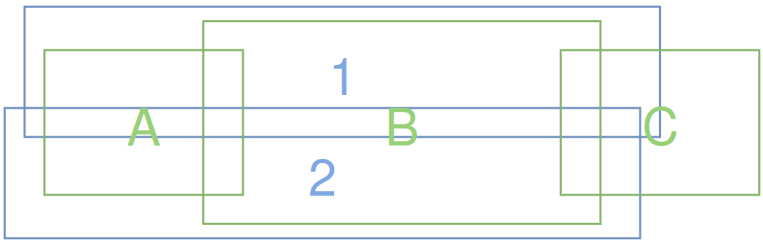


FIGURE 3.3 – Algorithme Zonemap : cas limite global. En vert les éléments de référence, en bleu les éléments du système

3.3 Choix de la méthode d'évaluation

La méthode **Zonemap** nous semble être la plus adaptée à notre travail, en termes d'entrées et sorties mais aussi en termes de précision de la mesure. En effet, elle donne beaucoup d'indications sur les surfaces des zones, erronées comme correctes. De plus, elle demande peu de calcul car il suffit simplement de calculer la force des liens et de les regrouper itérativement. Cette méthode permet aussi de prendre en compte les erreurs de classification des zones. Nous avons conservé l'évaluation de la classification dans cette méthode car elle ne posait pas de problèmes. Cependant, nous ne parlerons pas de classification dans les évaluations car elles portent sur des systèmes faisant peu d'erreurs de classifications. Par exemple, un OCR ne confondra pas un mot avec une ligne car une ligne est un ensemble de mots.
On se propose d'améliorer la méthode pour qu'elle prenne en compte les limites énoncées auparavant.

3.3.1 Segmentation

Les limites de l'algorithme Zonemap nous permettent de dégager deux points importants :

1. Les intersections entre les zones de référence ne sont pas prises en compte dans la façon d'évaluer les zones du système. Ces intersections interviennent, par exemple, quand un graphique chevauche une région textuelle ou encore dans le cas d'un document incliné, lorsque des régions textuelles se chevauchent.

2. Une zone est responsable d'une et d'une seule erreur alors qu'elle peut en provoquer plusieurs.

Un nouvel algorithme Zonemap+ a été conçu à partir de ces deux remarques. On peut le trouver en Annexe 6 et conserve la majorité des concepts de Zonemap.

Le principe est le suivant :
- On commence par effectuer le calcul des liens de force entre les zones ayant une intersection.
- Puis on trie ces liens toujours dans l'ordre décroissant. C'est à ce moment que le nouvel algorithme diffère de l'ancien.
- Pour chaque lien, on trouve 3 grandes conditions :

    1. Le polygone système est associé à au moins un élément de référence. Dans ce cas, on mémorise la fusion (le polygone système est associé à au moins un polygone de référence) et on retire les zones déjà comparées du polygone du système.

    2. Le polygone référence a été associé à au moins un élément du système. Dans ce cas, on mémorise la sur-segmentation (le polygone vérité est associé à au moins un polygone du système) et on retire les zones déjà comparées du polygone de vérité.

    3. L'intersection de la surface disponible du système avec celle du polygone de référence est suffisamment grande. On marque le lien entre les deux polygones avec la cardinalité de la relation (c'est-à-dire le nombre d'éléments système et vérité pour cette relation).

Nous avons introduit un nouveau groupe "Multiple" qui caractérise une zone ayant subi plusieurs opérations (fission **ET** fusion). Ce cas est assez rare en pratique, mais il est possible de le rencontrer et la méthode le traite correctement. Nous allons maintenant dérouler Zonemap+ sur le cas limite global représenté par la figure 3.4

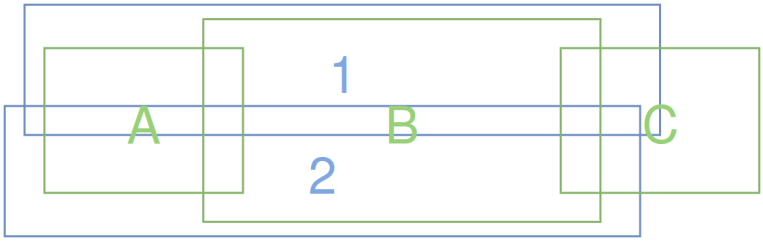


FIGURE 3.4 – Algorithme Zonemap+ : cas regroupant toutes les limites identifiées de l'algorithme Zonemap. En vert les éléments de référence, en bleu les éléments du système

Exemple

Avant de commencer l'algorithme, nous allons avoir besoin de données pour effectuer les calculs. Ces données sont présentées dans le tableau 3.1. On y retrouve les différents rectangles avec leur taille et position ainsi que la surface en pixel^2. Ces données nous seront utiles pour calculer la force des liens.

Rectangle	Gauche (x)	Haut (y)	Largeur	Hauteur	Surface
A	110	160	100	100	10 000
B	190	140	200	140	28 000
C	370	160	100	100	10 000
1	100	130	320	90	28 800
2	90	200	320	90	28 800

TABLE 3.1 – Algorithme Zonemap+ : données des rectangles du cas limite global

La première étape est donc de calculer la force des liens entres les rectangles de la vérité et ceux du système. On rappelle que la formule pour calculer un lien entre deux rectangles r_v vérité et r_s système est la suivante :

$$Lien(r_v, r_s) = (\frac{r_v \cap r_s}{r_v})^2 + (\frac{r_v \cap r_s}{r_s})^2$$

Nous allons détailler le calcul du premier lien A1. Il faut, dans un premier temps calculer le rectangle d'intersection entre le rectangle A et le rectangle 1. Le résultat est le rectangle $A1_{intersection}$ de position (110,160) et de dimension (100,60). Ce rectangle a une surface de 6000 pixel^2. Nous disposons maintenant de toutes les données pour calculer la force du lien. On pose le calcul :

$$Lien(A, 1) = (\frac{A1_{intersection}}{A})^2 + (\frac{A1_{intersection}}{1})^2$$

L'application numérique nous donne la force suivante :

$$Lien(A, 1) = (\frac{6000}{10000})^2 + (\frac{6000}{28800})^2 = (\frac{6}{10})^2 + (\frac{60}{288})^2 = \frac{36}{100} + \frac{3600}{82944} = 0.36 + 0.0434 = 0.4034$$

Nous venons de voir comment calculer la force d'un lien. Cette étape est répétée pour chaque paire de rectangles système et référence possédant une intersection non nulle. Les résultats sont reportés dans le tableau 3.2

La seconde étape de l'algorithme consiste à trier les liens par ordre décroissant de leur force. D'après le tableau 3.2, l'ordre des liens est le suivant :

Force des liens	1	2
A	0.4034	0.4034
B	0.6351	0.6351
C	0.1001	0.0645

TABLE 3.2 – Algorithme Zonemap+ : matrice de la force des liens entre chaque paire d'éléments systèmes et références

$$B1 > B2 > A1 > A2 > C1 > C2$$

La suite de l'algorithme est d'itérer sur ces liens triés.

Lien B1

Les rectangles B et 1 n'ont pas encore été associés à d'autres rectangles. Nous pouvons donc les mettre ensemble sans aucun conflit et la zone d'intersection entre les deux rectangles est un "match". On dit que cette zone est marquée par B et par 1 (ou zone de correspondance B1). La figure 3.5 montre le résultat de la première itération.

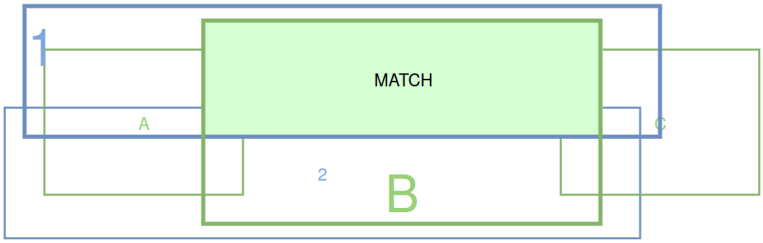


FIGURE 3.5 – Algorithme Zonemap+ : résultat du traitement du lien B1

Lien B2

On peut remarquer que l'élément B à déjà été associé avec l'élément 1. Par conséquent, la zone marquée précédemment ne peut pas être réutilisée. À cette étape, la zone disponible de B correspond à B auquel on retranche toutes les zones marquées par B précédemment. Ici, on a $B_{disponible}$ = B - $B1_{intersection}$ avec $B1_{intersection}$ la zone de correspondance B1 (Voir figure 3.6). De plus, on dit que la zone est un "split" car la zone de référence (B) est coupée par au moins deux éléments du système (1 et 2).

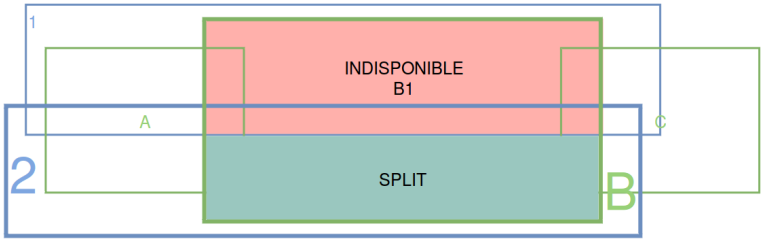


FIGURE 3.6 – Algorithme Zonemap+ : résultat du traitement du lien B2

Lien A1

La figure 3.7 représente le traitement de ce lien. Dans un premier temps, nous pouvons remarquer que l'élément A n'a pas encore été associé mais l'élement 1 l'est à B. On ne peut donc pas utiliser les zones marquées par 1 pour comparer A et 1. Ici $1_{disponible}$ = 1 - $B1_{intersection}$ et

correspond à la zone en rose. Le résultat est donc l'intersection entre A et $1_{disponible}$ représentée en bleu sur la figure. Cette zone est considérée comme une fusion (merge en anglais) car l'élément système (1) est associé à plusieurs éléments de référence (A B).

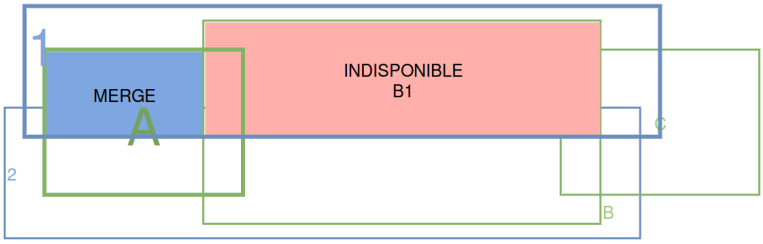


FIGURE 3.7 – Algorithme Zonemap+ : résultat du traitement du lien A1

Lien A2

Ce lien est intéressant car les deux éléments A et 2 ont tous les deux été associés. A a déjà été associé à 1 et B à 2. Par cette explication, nous voyons que nous nous retrouvons dans une situation d'association de plusieurs rectangles de la vérité à plusieurs rectangles du système. Cette relation est classée comme étant le résultat de plusieurs erreurs de segmentation : "Multiple". De la même manière que précédement, nous calculons $2_{disponible}$ et $A_{disponible}$ en retranchant respectivement des rectangles 2 et A, leurs précédentes zones marquées. Afin d'obtenir les zones en orange, nous calculons l'intersection entre $2_{disponible}$ et $A_{disponible}$. Il est important de noter que l'intersection entre deux rectangles donne au maximum 1 rectangle. Cependant, la différence entre deux rectangles peut donner plusieurs rectangles. On calcule donc ici des intersections entre des ensembles de rectangles. De plus, ces ensembles ont la particularité de ne pas posséder d'intersection entre eux. L'avantage de cette propriété est qu'elle permet de simplifier les calculs.

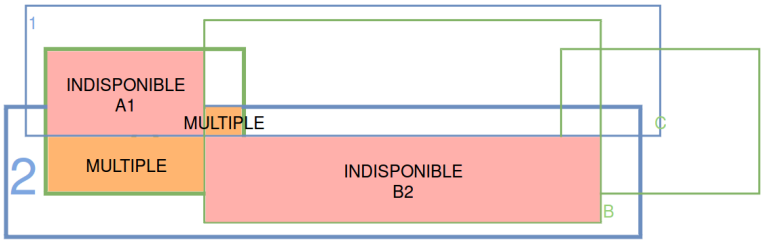


FIGURE 3.8 – Algorithme Zonemap+ : résultat du traitement du lien A2

Lien C1

Il n'y a pas beaucoup de chose à dire de plus que pour le lien A1. Nous sommes exactement dans le même cas car il y a une fusion. La différence est que l'élément 1 a une fusion en plus de A avec B.

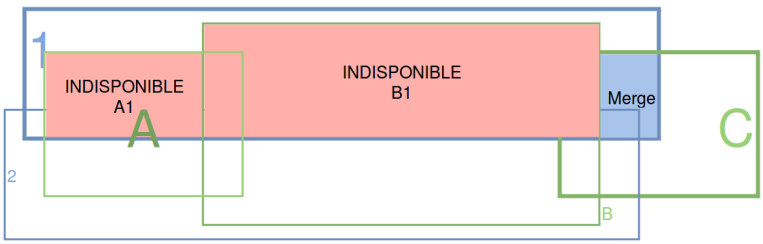


FIGURE 3.9 – Algorithme Zonemap+ : résultat du traitement du lien C1

Lien C2

Il est intéressant car il montre que l'association n'est pas systématique lorsqu'il y a eu association par un des deux éléments. Il faut que le taux de recouvrement de la surface d'intersection entre $C_{disponible}$ et $1_{disponible}$ par rapport à la surface disponible de la vérité $C_{disponible}$ dépasse un certain seuil afin de pouvoir considérer qu'il y a une correspondance entre deux zones. Ce seuil est calculé de la façon empirique et est fixé à 20%. Ici le taux de recouvrement n'est pas suffisant pour associer les zones. Si les zones 1 et C n'avaient pas été associées à d'autres rectangles alors on aurait autorisé leur association.

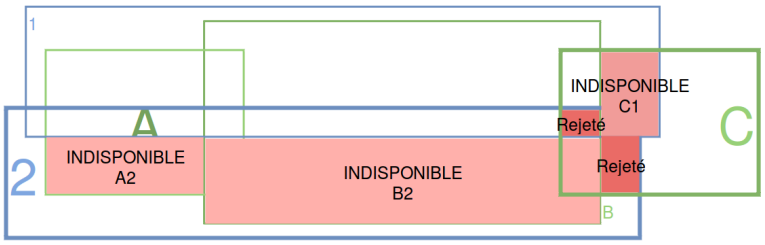


FIGURE 3.10 – Algorithme Zonemap+ : résultat du traitement du lien C2

Lorsque tous les liens ont été traités, les zones de référence n'ayant pas d'associations sont marquées comme étant des manqués (miss en anglais) et celles du système comme des fausses alarmes (false alarm en anglais). Les résultats de l'exemple sont inscrits dans le tableau 3.3. On peut remarquer que la méthode donne beaucoup de précision sur les erreurs commises par les zones. Chaque zone du tableau correspond à une sous-zone des zones concernées.

Zones	Référence	Système	Type
1	B	1	MATCH
2	B	1 2	SPLIT
3	A B	1	MERGE
4	A B	1 2	MULTIPLE
5	A B C	1	MERGE
6	A	-	MISS
7	C	-	MISS
8	-	1	FALSE ALARM
9	-	2	FALSE ALARM

TABLE 3.3 – Algorithme Zonemap+ : résultat du cas limite global

3.3.2 Reconnaissance de caractères

Afin d'évaluer la reconnaissance de caractères, nous nous aidons de l'algorithme Zonemap+ précédemment décrit. En effet, en plus de permettre de faire un alignement des zones, on peut aligner les contenus. Pour ce faire, nous nous basons sur l'alignement des boîtes. Lorsque nous associons deux zones entre elles, nous associons aussi les contenus. La seule différence est que pour associer le contenu d'une zone avec celui d'une autre, il faut au moins une zone n'ayant pas déjà été associée. Nous avons appelé cette méthode "Zonemap+ Alignment".

Exemple

Reprenons l'exemple de la segmentation du cas limite global. Pour le premier lien B1, les deux contenus vont être associés ensemble. Pour le lien B2, le contenu de 2 va être associé à B. Pour le lien A1, le contenu de 1 ne sera pas associé à 2. Cette méthode est répétée jusqu'à ce qu'il n'y ait plus de lien à traiter.

La raison principale derrière cette méthode est qu'on ne dispose pas d'information sur la position exacte du contenu issu des boîtes. Par exemple, si nous disposons d'une zone d'une ligne, il nous est impossible de découper le contenu de la même manière qu'une boîte. Si cette ligne est coupée en deux, on ne peut pas différencier précisément les mots appartenant à la première moitié de ceux appartenant à la seconde.

3.4 Corpus

Afin d'évaluer la segmentation, nous allons utiliser un corpus de documents linéaires et un corpus de documents non linéaires. Concernant l'évaluation de la reconnaissance de caractères, nous utiliserons aussi 2 corpus avec un corpus de documents non linéaires plus réduit. En effet, corriger une segmentation de façon semi-automatique est beaucoup plus rapide qu'une correction des contenus extraits. Par soucis de temps, nous avons donc décidé de n'utiliser pour l'évaluation de la reconnaissance, qu'un tiers des documents non linéaires destinés à l'évaluation de la segmentation.

Le tableau 3.4 décrit brièvement les corpus utilisés ainsi que les méthodes pour lesquelles ils seront utilisés. Les documents sont composés d'une page.

Corpus	Type de documents	Nombre de documents	Méthode(s) utilisée(s)
Corpus_1	Linéaires	49	Segmentation / Reconnaissance
Corpus_{1bis}	Linéaires	49	Segmentation / Reconnaissance
Corpus_2	Non linéaires	97	Segmentation
Corpus_{2bis}	Non linéaires	30	Reconnaissance

Table 3.4 – Description des corpus utilisés pour le cas XML/XML

Le corpus_1 correspond au même corpus que celui utilisé dans la section Texte / Texte. Le corpus_{1bis} correspond au corpus_1 auquel nous avons retiré des XMLs, les zones graphiques : les tableaux, les figures, les logos... Les OCRs utilisent des statistiques sur les composantes de l'image comme des espaces inter-mots ou taille de police afin d'effectuer leurs traitements. Or, si nous modifions l'image en retirant ces zones graphiques, la sortie de l'OCR risque d'être modifiée car les statistiques auront changé. Dans l'objectif de comparer l'influence de ces zones sur les performances anonncées, nous avons modifié uniquement la sortie de l'OCR.
Ce corpus va nous permettre de voir l'influence de ces zones sur les résultats, à la fois au niveau de la segmentation et de la reconnaissance du contenu.

Le corpus_2 est une extension du corpus_2 utilisé dans la section Texte / Texte. Il comporte des documents multi-colonnes avec des entêtes, des notes de bas de page et des références. Les documents ont été soigneusement choisis pour ne pas comporter les zones graphiques que nous avons retirées du corpus_{1bis}. La structure physique de ces documents peut-être un mélange entre des parties linéaires et non linéaires (Voir annexe 10).

3.5 Evaluation

3.5.1 Segmentation

Influence des zones graphiques

Les figures 3.11 et 3.12 ci-dessous montrent deux graphiques représentant les résultats de l'OCR OCRopus sur les deux corpus $corpus_1$ et $corpus_{1bis}$. On peut noter une grande différence entre les deux courbes. La figure 3.11 montre des pics de score plus importants que sur la figure 3.12. Ces différences sont dues aux zones graphiques qui sont propices aux erreurs de segmentation. Il est en effet difficile d'effectuer la vérité sur des éléments graphiques complexes en disposant uniquement de boîtes. De plus, les OCR ne sont pas faits pour reconnaître ce type de zone. Si ces dernières comportent du texte, il est fort probable que l'OCR tentera de reconnaître les caractères mais il n'arrivera pas à restituer la structure de la zone graphique. Afin de correctement évaluer les OCR, il est donc important d'écarter ce type de zones. On peut les écarter en employant deux méthodes :

1. Manuelle : en retirant les parties correspondant à une zone graphique de la sortie XML.
2. Automatique : par une méthode d'identification de ces zones qui nous permettra de rejeter tout contenu extrait dans ces zones par l'OCR.

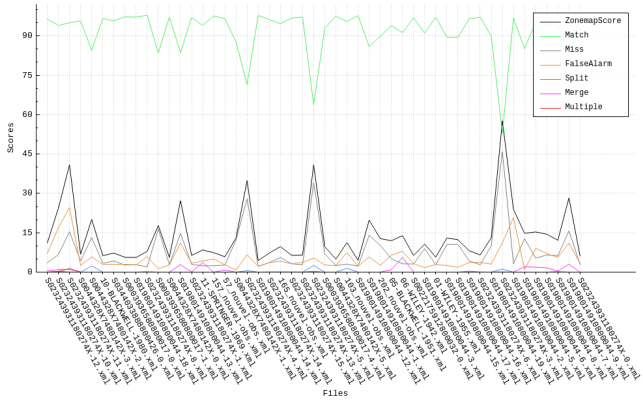


FIGURE 3.11 – Algorithme Zonemap+ - Résultat d'OCRopus sur le $corpus_1$

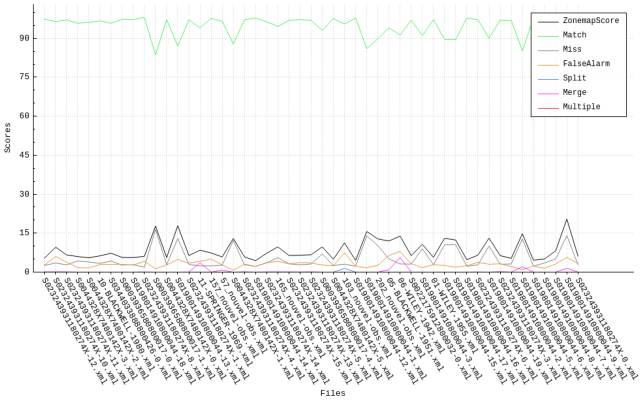


FIGURE 3.12 – Algorithme Zonemap+ - Résultat d'OCRopus sur le $corpus_{1bis}$

Evaluation des OCRs

Nous avons utilisé l'algorithme Zonemap+ sur les corpus : $corpus_1$, $corpus_1bis$ et $corpus_2$. Le tableau 3.5 nous donne les résultats de l'algorithme sur chaque corpus. D'un point de vue général, il apparait évident que Tesseract extrait une meilleure segmentation que son rival OCRopus, et ce pour chaque corpus. On peut remarquer que, comme énoncé précédemment, les zones graphiques perturbent l'évaluation en incluant plus d'erreurs de segmentation. Pour Tesseract et OCRopus, il y a un gain d'environ 7 points entre le $corpus_1$ et le $corpus_{1bis}$. Les résultats du $corpus_2$ par rapport au $corpus_{1bis}$, nous montrent que les structures non linéaires sont plus difficiles à identifier correctement. La présence de zones "Multiple" indique que la segmentation de l'OCR a subi de lourdes modifications. Elles sont toutes négligeables execpté pour OCRopus dans le $corpus_2$, ce qui nous indique qu'il fait beaucoup d'erreurs de segmentation. En effet, cette idée est confortée par la proportion de "Split" et "Merge" qui est aussi plus importante que toutes les autres évaluations.

Corpus	OCR	Zonemap	Total	Match	FalseAlarm	Miss	Split	Merge	Multiple
$Corpus_1$	Tesseract	8.53896	100% 105 847 757	93,44% 98 909 128	0,25% 269 892	0,01% 11 599	5,81% 6 158 955	0,44% 473 423	0,02% 24 760
	Ocropus	13.7744	100% 109 313 147	87,25% 95 378 176	0,43% 474 858	0,15% 163 463	4,55% 4 976 836	7,58% 8 289 449	0,03% 30 365
$Corpus_{1bis}$	Tesseract	1.20342	100% 100 903 181	98,71% 99 611 312	0,10% 102 822	0% 0	0,96% 975 991	0,21% 213 056	0% 0
	Ocropus	8.58964	100% 102 941 994	91,43% 94 120 256	0,3% 303 803	0,01% 13 931	2,95% 3 040 328	5,31% 5 463 676	0% 0
$Corpus_2$	Tesseract	3.90983	100% 345 365 274	96,39% 332 904 992	0,29% 1 006 123	0% 0	3,04% 10 510 397	0,27% 914 989	0,01% 28 773
	Ocropus	16.4732	100% 369 469 254	81,98% 302 876 320	5,44% 20 109 912	0,02% 80 279	5,85% 21 615 296	6,51% 24 065 472	0,2% 721 975

TABLE 3.5 – Algorithme Zonemap+ - Résultat de segmentation des OCRs OCRopus et Tesseract sur plusieurs corpus

Les résultats des deux OCRs pour le $corpus_{1bis}$ sont présentés dans les figures 3.13 et 3.14. Concernant OCRopus sur le $corpus_{1bis}$, les erreurs de "Miss" et "FalseAlarm" sont dues à des différences minimes entre les zones de vérité et les zones système. Le cumul de ces différences peut représenter une erreur importante mais elle est bénigne car elle ne perturbera pas la reconnaissance. Aussi, le nombre de Merges et Splits est faible. Concernant Tesseract, les résultats sont satisfaisants et très proches de la vérité. Les rares cas d'erreurs sont des cas isolés ou des différences entre la vérité et l'OCR. Ces différences ne sont pas obligatoirement des erreurs à proprement parler, mais peuvent correspondre à une autre vérité. Dans notre cas, c'est un mélange de ces deux erreurs.

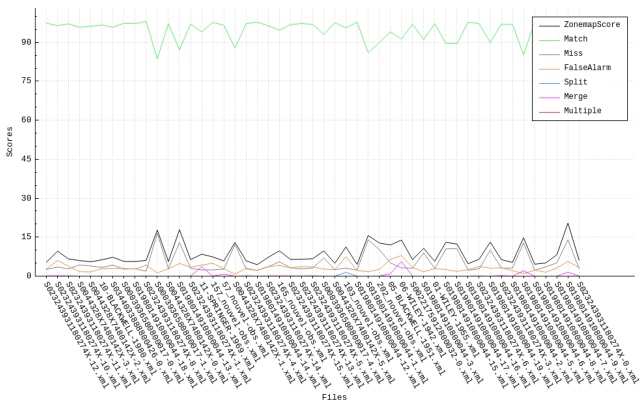


FIGURE 3.13 – Algorithme Zonemap+ - Résultat d'OCRopus sur le $corpus_{1bis}$

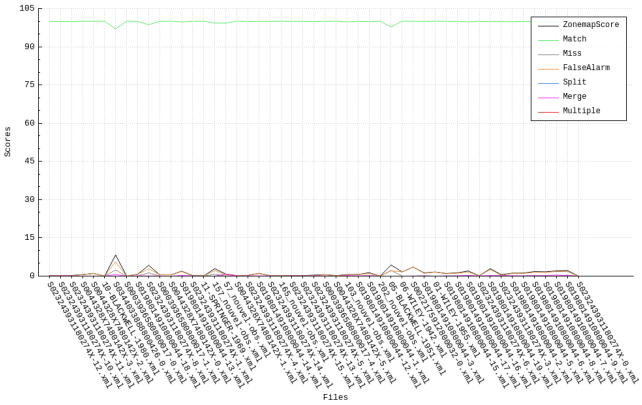


FIGURE 3.14 – Algorithme Zonemap+ - Résultat de Tesseract sur le $corpus_{1bis}$

Les résultats des deux OCRs pour le $corpus_2$ sont inscrits dans les figures 3.15 et 3.16. La différence entre OCRopus et Tesseract sur ce corpus est flagrante. D'un côté, nous avons une bonne segmentation avec Tesseract. De l'autre, OCRopus commet beaucoup d'erreurs, notamment des erreurs de "Merge" et "Split". Le pic de "False Alarm" de Tesseract correspond à une zone englobant l'image en entier. Cette erreur est causée par un contour noir sur les bords de l'image. Ce qu'on remarque, c'est que les erreurs sont très dépendantes de l'image. En effet, les résulats ne sont pas stables (pour OCRopus). On trouve des images correctement segmentées, mais aussi des images pauvrement segmentées. Il apparait que ces erreurs interviennent sur les documents à dominance non linéaires.

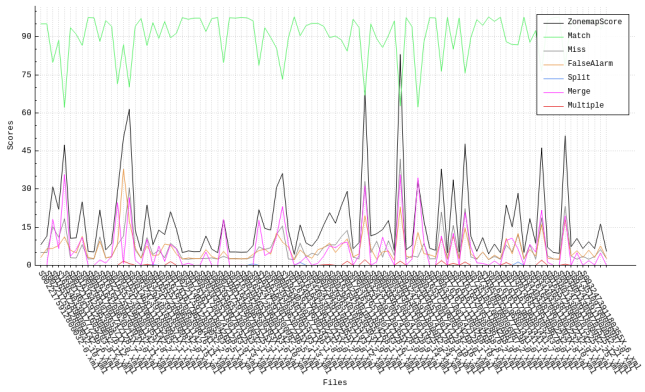


FIGURE 3.15 – Algorithme Zonemap+ - Résultat d'OCRopus sur le corpus$_2$

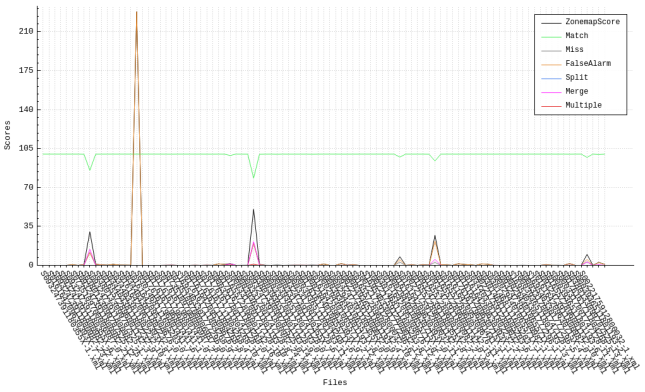


FIGURE 3.16 – Algorithme Zonemap+ - Résultat de Tesseract sur le corpus$_2$

3.5.2 Reconnaissance de caractères

Evaluation de la méthode

Nous allons maintenant évaluer la méthode d'alignement du contenu des zones. Pour ce faire, nous allons comparer les alignements effectués, un par un. Un alignement est caractérisé par le type de l'erreur ainsi que par le contenu de l'OCR et celui du système. Un alignement de référence a été mis en place en corrigeant semi-automatiquement l'alignement du système. Un alignement système est correct uniquement s'il possède une correspondance **exacte** avec un alignement de référence. Nous avons évalué la méthode d'alignement sur l'ensemble des documents du corpus$_{2bis}$ avec l'OCR OCRopus. Les résultats sont regroupés dans le tableau 3.6.

Méthode	Valeur	%
Rappel	2465 / 2477	99.5155
Precision	2465 / 2511	98.168

TABLE 3.6 – Algorithme Zonemap+ Alignment - Résultat de l'évaluation de la méthode sur le corpus$_{2bis}$ avec l'OCR OCRopus

À part quelques erreurs, les performances de la méthode sont satisfaisantes. Concernant les erreurs, elles proviennent de l'incapacité de la méthode à détecter les fissions verticales. En effet avec l'algorithme Zonemap+, les fissions verticales ne sont pas détectés comme tel. La figure **??** montre un exemple de résultat qui illustre le problème. Dans ce cas, si on ne considère que la ligne "Academic Medical Centers", on peut remarquer qu'il n'y a pas de zone classée comme étant une fission. Ce phénomène est un effet de bord de l'utilisation des rectangles :
On dispose d'un rectangle X et d'un rectangle Y. Sachant que X est contenu entièrement dans Y, si on retire le contenu de Y à l'emplacement X, le rectangle Y sera inchangé. La conséquence de ce phénomène est que les éléments de la fission verticale seront classés comme étant des sur-détections.

Ces erreurs nous montrent une nouvelle limite de l'algorithme d'alignement.

Evaluation des OCRs

La figure 3.7 regroupe les taux de reconnaissance des deux OCR pour chaque corpus. Le taux de reconnaissance de Tesseract entre les corpus $corpus_{1bis}$ et $corpus_2$ a augmenté alors que le score de Zonemap+ a lui baissé. Ce phénomène peut s'expliquer par le faible nombre de caractères spéciaux dans le $corpus_2$ ainsi que la bonne qualité des images. On peut donc émettre l'hypothèse qu'Ocropus aurait eu un meilleur taux de reconnaissance (correspondance des chaînes) s'il avait eu un score de zonemap beaucoup moins important. Ocropus a tellement détruit la structure que la reconnaissance a obligatoirement une baisse de performance. Il faut donc noter que les deux OCRs sont très proches en termes de reconnaissance, même si OCRopus est toujours en dessous. On soulignera la très bonne performance de Tesseract sur le $corpus_{2bis}$ avec un taux de reconnaissance de 99.37%.

Corpus	OCR	Recognition Accuracy (%)
Corpus1	Tesseract	97.24
	Ocropus	95.56
Corpus1bis	Tesseract	97.71
	Ocropus	96.46
Corpus2bis	Tesseract	99.37
	Ocropus	94.99

TABLE 3.7 – Algorithme Zonemap+ Alignment - Taux de reconnaissance de caractères des OCRs OCRopus et Tesseract

3.6 Discussion

3.6.1 Conclusion

L'algorithme Zonemap+ introduit une nouvelle classe d'erreur "Multiple", qui correspond à une zone responsable d'une erreur de segmentation de fusion et de fission. Il s'avère que cette erreur en pratique est faiblement représentée lorsque le système de segmentation est performant. À l'inverse, moins le système est performant, plus les erreurs "Multiple" seront nombreuses. De plus, les zones non linéaires des documents sont plus propices à ce genre d'erreurs car leur structure physique est souvent plus complexe. Zonemap+ prend maintenant en compte des intersections entre les éléments de la vérité. L'algorithme Zonemap+ est plus coûteux en calcul que Zonemap mais il offre une meilleure précision.

L'algorithme d'alignement des contenus utilisant la méthode Zonemap+ s'est révélée être à la fois simple et efficace. Simple, car elle consiste à utiliser les résultats de chaque itération de Zonemap+. Elle est aussi efficace car elle permet un alignement proche des 99%.

Cette méthode d'alignement est donc fiable même lorsque les documents sont abîmés ou avec des structures physiques détruites. De plus, elle bénéficie des concepts d'alignement de la Zonemap comme le calcul de la force des liens ainsi que de l'alignement itératif.

Les commentaires des résultats convergent tous vers la même idée : Tesseract segmente mieux les images qu'OCRopus. L'écart entre les performances des deux OCR est assez important. De plus,

OCRopus fait beaucoup d'erreurs sur les document non linéaires. Tesseract est, quand à lui, beaucoup plus stable et prévisible que son concurrent.

La reconnaissance étant dépendante de la segmentation, OCRopus n'est pas avantagé. Cependant, la segmentation comme la reconnaissance doivent être les meilleures possible dans le cadre d'un OCR. Les conclusions sont les mêmes que pour la section Texte / Texte, avec OCRopus toujours en dessous de Tesseract.

Améliorations

L'algorithme Zonemap+ pourrait être amélioré en allant encore plus loin dans l'identification des erreurs. Au lieu d'avoir uniquement une classe "Fission/Split", on pourrait la séparer en deux pour faire la différence entre une fission verticale et une fission horizontale. Il faudrait aussi conserver la classe "Fission" car certains cas de fission peuvent être difficilement mis dans une des deux nouvelles classes. Il en va de même pour la fusion.

L'algorithme "Zonemap+ Alignment" pourrait être amélioré en détectant les erreurs de fission verticales. Au vue des expériences menées, cette limite est la seule que nous avons détectée.

4 Evaluation Texte / XML

4.1 Enoncé du problème

Le problème soulevé par ce cas est une combinaison des deux cas précédemment présentés. Il faut aligner les contenus en utilisant à la fois la segmentation et le contenu lui-même. La tâche d'alignement est fonction de la segmentation donné par l'OCR qui elle est fonction de la structure du document. L'objectif principal sera d'aligner correctement les contenus puis dans un second temps il serait souhaitable d'identifier précisément les erreurs de segmentation.

Nous verrons dans un premier temps l'état de l'art concernant ce problème, puis nous développerons les choix conceptuels qui nous ont conduits à l'établissement d'une nouvelle méthode. Enfin nous analyserons les performances de la méthode résultante ainsi que les données qu'elles fournit sur les corpus en utilisant les deux OCR : Ocropus et Tesseract.

4.2 Etat de l'art

L'état de l'art sur ce problème exact est quasiment inexistant.Cependant, nous pouvons nous référer à l'état de l'art 2.2 pour avoir une idée des techniques employées pour effectuer un alignement texte à texte. On peut aussi se référer à l'état de l'art de la partie précédente 3.2 pour effectuer un alignement xml à xml (zone à zone).

4.3 Choix de la méthode

Comme il n'y a pas de méthode traitant exactement de ce problème, nous avons mis en place une nouvelle méthode. Pour aligner, cette méthode doit être un mélange entre l'alignement par la segmentation et un alignement par le contenu. Les problèmes principaux qui apparaissent dans ce contexte sont la modification de l'ordre de lecture ainsi que la perturbation de la structure physique d'origine. Il faut pouvoir corriger ces deux sources d'erreur afin de pouvoir effectuer l'alignement.

L'algorithme 1 décrit les étapes principales de la méthode proposée. Soit l'ensemble des lignes de la vérité : $V = \{v_0, v_1, ..., v_i\}, i \in \mathbb{N}$. Soit l'ensemble des lignes du système : $S = \{s_0, s_1, ..., s_j\}, j \in \mathbb{N}$.

Nous allons maintenant expliquer chaque étape plus en détails.

Marquage des chaînes uniques

On trouve des chaînes uniques dans le texte avec un paramètre n représentant un nombre de

Algorithm 1: Text/Xml - Algorithme d'alignement

```
1 begin
2     MarquageDesChainesUniques(V, S)
3     DetectionEtCorrectionDesFusionsFissions(V, S)
4     Boolean matched ⟵ true
5     while matched do
6         corrigerDerniersAlignements
7         matched ⟵ alignerLesNouveauxElements
8     end
9 end
```

mots consécutifs. Par exemple pour n = 2, on va sélectionner toutes les combinaisons de 2 mots consécutifs issus des textes et marquer celles qui sont uniques. On commence de n jusqu'à 1 car les chaînes uniques composées de 3 mots sont plus fiables qu'un mot de 2 ou 3 lettres. Une limite de ce marquage est la possibilité qu'un caractère d'un mot m1 soit substitué et qu'ils en résulte un mot m2 existant dans le texte. Le mot m2 d'origine, est quand à lui modifié en un mot m3, ce qui a pour conséquence que le mot m1 (maintenant m2) représente un autre mot. Il est donc possible d'effectuer un mauvais alignement car l'erreur de l'OCR est grave. Les chaînes proviennent uniquement des lignes, c'est-à-dire qu'une chaîne donnée correspond à une seule ligne. Cependant, une ligne peut avoir plusieurs chaînes uniques. Le but de cette méthode est de trouver le plus de lignes ayant une chaîne unique. Suite au marquage des mots, un alignement dit "naturel" se dégage. Lorsqu'on associe 2 chaînes ensemble, nous associons aussi deux lignes. Ce premier traitement nous fournit un brouillon d'alignement qu'il faut vérifier et qui nous servira de base pour réaliser l'ensemble de l'alignement.

Traitement de l'alignement "naturel"

Il y a deux étapes pour le traitement de l'alignement correspondant aux deux configurations possibles. Ces deux configurations sont les erreurs de fusions et de fissions que nous pouvons détecter facilement. En effet, si une ligne du système est alignée avec au moins 2 lignes de la vérité alors c'est une fusion. À l'inverse, si une ligne de la vérité est alignée avec au moins 2 lignes du système alors c'est une fission. Ces deux cas sont traités dans l'algorithme 2.

- Dans le cas d'une fusion, nous devons couper et réaligner la ligne du système par les lignes de la vérité avec lesquelles elle est alignée. Ce cas est plus complexe à traiter que la fission car il est toujours plus facile d'assembler que de couper. Pour rétablir la fusion, on itère sur chacun des éléments vérité correspondant à l'élément système. On prend deux lignes et on tente de faire correspondre les contenus en rejetant une extrémité. Si l'extrémité est non nulle alors on fait correspondre les deux chaînes précédentes privées de l'extrémité et on remplace la ligne du système par l'extrémité (qui est le reste). Si l'extrémité est nulle alors tout le contenu est aligné et on s'arrête. Si, à la fin, il reste une extrémité qui n'a pas été alignée, alors on la place dans l'ensemble des lignes du système S. Il faut remarquer que lorsqu'on coupe un élément du système, on effectue une estimation de l'emplacement de la coupure en comptant le nombre de caractère à gauche et à droite. Cette méthode, même si elle est peut précise car elle dépend de la police et de la taille des caractères, est suffisante pour séparer les éléments fusionnés.
- Dans le cas d'une fission, nous devons assembler les lignes du système associées à la ligne de vérité concernée. Cette tâche est assez simple puisqu'il suffit de fusionner les lignes du système en utilisant leur positions. La fission est une fission horizontale car dans le cas d'une fission verticale, le texte n'est pas correctement reconnu étant donné que cette dernière modifie les caractères. Comme la fission est horizontale, il faut ordonner les lignes de

gauche à droite et les fusionner. Avant de fusionner, il faut vérifier que les lignes associées ne soient pas dues à des effets de bord des chaînes uniques. Pour ce faire, nous regardons si la taille total des lignes systèmes assemblé est proche de celle de la vérité. Le seuil de proximité est fixé à 20% de tolérance d'écart entre la taille totale des lignes du système en caractère et celle de la vérité. Si le seuil est respecté alors nous fusionnons itérativement les lignes de gauche à droite.

Algorithm 2: Text/Xml - Detection Et Correction Des Fusions / Fissions

```
1  begin
2      for v_i ∈ V do
3          if |v_i.matches| ≥ 1 then
4              Entier tailleVerite ⟵ getCharNumber(v_i)
5              Entier tailleMatches ⟵ getCharNumber(v_i.matches)
6              Entier distanceTaille ⟵ abs(tailleVerite−tailleMatches) / min(tailleVerite,tailleMatches)
7              if distanceTaille < 0.2 then
8                  Trier les matches de gauche à droite
9                  for s_k ∈ v_i.matches , k ∈ [1; |v_i.matches|] do
10                     fusionneLignes(v_i.matches(0), v_i.matches(k))
11                 end
12             end
13         end
14     end
15     for s_j ∈ S do
16         if |s_j.matches| ≥ 1 then
17             Trier les matches de gauche à droite
18             for m_k ∈ s_j.matches , k ∈ [0; |s_j.matches|] do
19                 Ligne restant ⟵ matchLignes(s_j, m_k)
20                 if !restant.empty then
21                     associerLignes(s_j, m_k)
22                     s_j.valeur ⟵ restant
23                 end
24             end
25         end
26     end
27 end
```

Vérification des lignes alignées

Cette étape vérifie qu'une ligne n'a subit aucune fission ou fusion de la même manière que précédemment à la seule différence que nous ne savons pas quelles sont les lignes concernées. Une relation de cardinalité 1-1 ne signifie pas qu'il n'y a pas d'erreurs de segmentation. Il est donc important de vérifier chaque alignement que nous avons effectué. Cette méthode est décrite dans l'algorithme 3.

Algorithm 3: Text/Xml - Correction des derniers alignements

```
1  begin
2      for v_i ∈ V do
3          if !v_i.verifie then
4              Ligne restant for m_k ∈ v_i.matches do
5                  restant ⟵ matchLignes(v_i, m_k)
6                  if !restant.empty then
7                      associerLignes(s_j, m_k)
8                      s_j.valeur ⟵ restant
9                  end
10                 v_i.verifi ⟵ vrai
11             end
12             if !restant.empty then
13                 ajouteLignes(S, restant)
14             end
15         end
16     end
17 end
```

Alignement des lignes non alignées

Cette étape est décrite dans l'algorithme 4. Suite aux corrections, de nouvelles lignes peuvent apparaître et ont besoin d'être alignées. Nous itérons sur les lignes de la vérité en constituant des intervalles entre deux lignes matchés. Ces intervalles forment des listes de lignes de vérité non alignées. Aussi, elle nous donne, grâce aux matches des deux lignes formant l'intervalle, une zone de recherche. Cette zone correspond au rectangle formé entre les deux matches. La zone ainsi définie nous permet d'utiliser les positions des lignes du système afin de les sélectionner pour les aligner. Nous disposons de deux ensembles de lignes qu'il nous faut comparer pour les associer entre elles. Cette dernière étape est réalisé par une méthode récursive de recherche de meilleur correspondance entre les lignes. Cela nous permet de trouver un alignement respectant l'ordre d'origine des lignes tout en rejetant si besoin certaines lignes dont le contenu est trop éloigné.

Enfin, tous les éléments qui ont été extrait des lignes du système lors d'une supposé fusion sont remis à leur place dans leur ligne d'origine. Cette heuristique permet de recoller les morceaux qui ont été mal découpé.

4.4 Corpus

Le corpus utilisé pour évaluer la méthode et les deux OCR est le corpus$_{2bis}$ décrit dans la table 3.4.

Algorithm 4: Text/Xml - Aligner les nouveaux éléménts

```
1  begin
       Result: Boolean aligne
2      Entier indexDebut ⟵ −1
3      Entier indexFin ⟵ 0
4      Entier indexMax ⟵ |V|
5      Boolean aligne ⟵ faux
6      Boolean init ⟵ faux
7      for i ∈ [0; maxIndex] do
8          if |v_i.matches| ≠ 0 then
9              init ⟵ vrai
10             indexFin ⟵ i
11             if abs(indexFin − indexDebut) > 1 then
12                 if aligneLineaireMatch(indexDebut, indexMax) then
13                     aligne ⟵ vrai
14                 end
15             end
16             indexDebut ⟵ i
17         end
18     end
19     if indexDebut ≠ indexMax&init then
20         if aligneLineaireMatch(indexDebut, indexMax) then
21             aligne ⟵ vrai
22         end
23     end
24 end
```

4.5 Evaluation

4.5.1 Evaluation de la méthode

Nous évaluons, pour cette méthode, uniquement l'alignement des contenus car les erreurs de segmentation sont difficiles à obtenir dans cette configuration.

L'évaluation de l'alignement des lignes peut être parfois trompeuse car un alignement peut-être "faux" mais ne pas générer de différence lors de l'évaluation de la reconnaissance de caractères. Cette évaluation nous donne une idée de la fiabilité de la méthode.

La figure 4.1 représente les performances de l'alignement de la méthode sur le corpus$_{2bis}$. Les principales erreurs surviennent lorsque la structure physique du document est particulière et que nous ne trouvons plus les bons candidats à aligner.

Afin d'évaluer plus précisément l'impact des différences d'alignements entre la méthode et la vérité, on se propose de calculer les taux de reconnaissance de caractères en utilisant ces deux alignements. La différence des deux taux nous permettra de juger de l'importance des erreurs d'alignement. Les résultats sont inscrits dans la table 4.2.

La différence entre les deux taux de reconnaissance est très faible (0.37% d'écart). On peut donc dire

-	Alignments corrects	%
Rappel	2540 / 2629	96.61
Precision	2540 / 2710	93.73

Table 4.1 – Texte / Xml - Rappel et précision de l'alignement de la sortie d'OCRopus sur le corpus$_{2bis}$

-	Reconnaissance
Alignement - vérité	96,40 %
Alignement - méthode	96,03 %

Table 4.2 – Texte / Xml - Comparaison des taux de reconnaissance entre la vérité et la méthode sur le corpus$_{2bis}$ avec l'OCR OCRopus

que les erreurs d'alignement produites par la méthode sont des erreurs faibles en conséquences sur l'évaluation finale. Cependant, des différences existent et elles peuvent être potentiellement amplifiées sur des documents difficiles ou avec des systèmes peu performants.

4.5.2 Evaluation des OCR

La figure 4.3 montre les résultats de reconnaissance de caractères des OCR Ocropus et Tesseract sur le corpus$_{2bis}$. Ils ont été calculés à partir de la méthode et non de la vérité. En effet, par manque de temps, nous disposons uniquement de l'alignement de vérité d'OCRopus. On distingue une différence de plus de 3% entre les deux OCRs. Comme énoncé précedemment, les résultats d'OCRopus peuvent être en partie expliqués par ses erreurs de segmentation qui sont plus fréquentes que chez son concurrent.

-	% Reconnaissance
Tesseract	99,27
Ocropus	96,03

Table 4.3 – Texte / Xml - Comparaison des taux de reconnaissance entre OCRopus et Tesseract sur le corpus$_{2bis}$

4.6 Discussion

4.6.1 Conclusion

La méthode d'alignement nécessite un contenu **et** une segmentation d'une qualité minimum. En effet, de part sa nature mixte, il va exister une interdépendance entre le contenu et la segmentation. Si la segmentation est trop mauvaise, même si le contenu est parfaitement reconnu il sera parfois difficile d'aligner. L'inverse est aussi vrai : si le contenu est trop mauvais, nous n'aurons pas de points de repère dans le document et nous ne pourrons pas utiliser la segmentation. Cependant, si ces erreurs sont ponctuelles et non généralisées alors la méthode permet de les compenser. Une

limite de cette méthode est la détermination de la zone de recherche des candidats à l'alignement par la position. Cette zone peut être entièrement erronnée de part la position des deux lignes la définissant. En effet, ne connaissant ni les blocs physique, ni l'ordre de lecture sur le document, il est parfois difficile de trouver une zone satisfaisante. Une autre limite est l'utilisation des chaînes uniques comme points de repères entre le texte de la vérité et les boites du système. Lorsqu'aucune chaîne ou très peu sont trouvées, la méthode perd grandement en efficacité. Cette méthode n'est donc pas adaptée aux documents difficiles ou aux systèmes peu performants. Même si elle n'est encore mature, les résultats qu'elle propose, par rapport à d'autres méthodes plus précises comme dans la partie Xml/Xml, sont satisfaisants car très proches de la vérité.

4.6.2 Améliorations

Pour améliorer la méthode actuelle, il faudrait mettre en place un moyen permettant d'obtenir les blocs physiques d'un document afin de mieux cibler les lignes par leur position. De plus, une classification et une détection des erreurs de segmentation seraient possibles en utilisant la cardinalité de chaque relation. Cependant, cette dernière amélioration serait très sensible car les erreurs d'interprétation ou erreurs d'alignement sont faciles.

5 Conclusion générale

Sur les méthodes

Dans ce rapport, nous avons proposé 3 méthodes visant à effectuer des évaluations de segmentation et de reconnaissance de caractères sur des OCR. Nous avons évalué chaque méthode en utilisant des corpus ayant des documents présentant des caractéristiques différentes pour montrer leur stabilité. Ces 3 méthodes ont le même objectif mais pas les mêmes contraintes. Suivant la forme des entrées du système, une méthode sera employée plutot qu'une autre car elle est plus adaptée. Nous allons maintenant résumer chacune des 3 méthodes, en commençant de la moins précise à la plus précise :

1. Partie Texte / Texte
   La méthode RETAS développé par [10] à été utilisé. Nous savons que les limites de la méthode reposent dans la structure physique des documents des corpus utilisés. Lorsque la structure est linéaire, la méthode fonctionne correctement. Cependant, lorsqu'elle devient complexe et que des erreurs d'ordre de lecture ou des fusions inter-colonnes apparaissent, l'alignement échoue.
2. Partie Texte / Xml : Nous avons créé une méthode pour ces contraintes particulières qui est un compromis entre un texte et un texte avec sa représentation spatiale. La méthode proposée fonctionne lorsque la segmentation et le contenu permettent de poser des repères fiables, dans le cas contraire, elle échouera.
3. Partie Xml / Xml : Nous avons modifié la méthode Zonemap [6] tout en conservant ses concepts de base. De plus, nous avons ajouté une nouvelle classe d'erreur de segmentation, ainsi que mis en place une méthode permettant d'aligner le contenu. Cette méthode s'est révélée être la plus précise. D'une part car elle permet d'évaluer la segmentation, ce que les autres méthodes ne permettent pas ou difficilement. Et d'une autre part elle permet de prendre en compte ces erreurs de segmentation dans l'alignement des contenus.

Sur les OCRs

Nous avons évaluer les performances de deux OCRs : OCRopus et Tesseract. De plus, nous les avons comparés afin de savoir lequel était meilleur que l'autre sur les deux critères suivants : la segmentation et la reconnaissance de caractères. Concernant la conclusion finale sur les résultats, nous utilisons la méthode la plus précise, à savoir Zonemap+ (partie Xml / Xml). Les résultats concernant la segmentation sont inscrits dans le tableau 3.5 et ceux concernant la reconnaissance sont répertoriés dans le tableau 3.7. Dans l'ensemble, indépendamment des corpus utilisés, l'OCR Tesseract obtient de meilleurs résultats à la fois pour la segmentation et pour la reconnaissance des caractères. Le second étant dépendant du premier, il serait interessant d'observer la variation du taux de reconnaissance de caractères en leur donnant la même segmentation. De plus, nous pouvons remarquer qu'OCRopus fait des erreurs de segmentation modifiant le contenu à reconnaître (fusion/fission verticale), et donc que les taux de reconnaissance subissent ces effets de bord. En conclusion, utilisés tels qu'ils sont donnés, les OCRs OCRopus et Tesseract ont des performances proches mais suffisament éloignés pour dire que Tesseract est meilleur qu'OCRopus. Il est important de noter qu'il n'y a eu aucun en-

trainement de ces OCRs sur les corpus. Une suite à ce rapport serait de déterminer les taux de reconnaissance indépendamment de la segmentation. De plus, on pourrait observer la variation des performances en les entrainant sur les corpus. La moyenne pondérée (au nombre de documents) du taux de reconnaissance de caractères de Tesseract sur les corpus corpus_1, corpus_{1bis} et corpus_{2bis} est de 97.91 %.

Bibliographie

[1] Mayce Al Azawi, Marcus Liwicki, and Thomas M Breuel. Wfst-based ground truth alignment for difficult historical documents with text modification and layout variations. In *IS&T/SPIE Electronic Imaging*, pages 865818–865818. International Society for Optics and Photonics, 2013. 10

[2] Apostolos Antonacopoulos and David Bridson. Performance analysis framework for layout analysis methods. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 2, pages 1258–1262. IEEE, 2007. 19

[3] Christian Clausner, Stefan Pletschacher, and Apostolos Antonacopoulos. Scenario driven in-depth performance evaluation of document layout analysis methods. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 1404–1408. IEEE, 2011. 20

[4] Shaolei Feng and R Manmatha. A hierarchical, hmm-based automatic evaluation of ocr accuracy for a digital library of books. In *Digital Libraries, 2006. JCDL'06. Proceedings of the 6th ACM/IEEE-CS Joint Conference on*, pages 109–118. IEEE, 2006. 10

[5] Andreas Fischer, Emanuel Indermühle, Volkmar Frinken, and Horst Bunke. Hmm-based alignment of inaccurate transcriptions for historical documents. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 53–57. IEEE, 2011. 10

[6] Olivier Galibert, Juliette Kahn, and Ilya Oparin. The zonemap metric for page segmentation and area classification in scanned documents. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 2594–2598. IEEE, 2014. 22, 44

[7] Wontaek Seo, Mudit Agrawal, and David Doermann. Performance evaluation tools for zone segmentation and classification (pets). In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 503–506. IEEE, 2010. 18

[8] Faisal Shafait, Daniel Keysers, and Thomas M Breuel. Pixel-accurate representation and evaluation of page segmentation in document images. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 872–875. IEEE, 2006. 21

[9] Christian Wolf and Jean-Michel Jolion. Object count/area graphs for the evaluation of object detection and segmentation algorithms. *International Journal of Document Analysis and Recognition (IJDAR)*, 8(4) :280–296, 2006. 22

[10] Ismet Zeki Yalniz and Raghavan Manmatha. A fast alignment scheme for automatic ocr evaluation of books. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 754–758. IEEE, 2011. 11, 44

Liste des illustrations

1.1 Exemple de document image avec sa segmentation au niveau mots 2

1.2 Ordre de lecture complexe dans un document multi-colonnes où la note de bas de page prolonge la première colonne à partir de sa moitié 3

1.3 Erreur de segmentation : Fusion verticale de blocs. Les numéros donnent l'ordre de lecture des lignes 4

1.4 Exemple de cas réél d'erreur de segmentation : Fusion verticale de deux lignes par l'OCR OCRopus 4

1.5 Erreur de segmentation : Fusion horizontale 4

1.6 Exemple de cas réél d'erreur de segmentation : Fusion horizontale de deux lignes par l'OCR OCRopus 5

1.7 Erreur de segmentation : Fission verticale 5

1.8 Exemple de cas réél d'erreur de segmentation : Fission verticale de deux lignes par l'OCR OCRopus 5

1.9 Erreur de segmentation : Fission horizontale 6

1.10 Exemple de cas réel d'erreur de segmentation : Fission horizontale de deux lignes par l'OCR OCRopus 6

2.1 Algorithme RETAS - Exemple d'alignement 12

2.2 Algorithme RETAS - Résultat de Tesseract $corpus_1$ 14

2.3 Algorithme RETAS - Résultat d'OCRopus $corpus_1$ accompagné de ses erreurs de segmentation pour chaque document 14

2.4 Algorithme RETAS - Résultat de Tesseract sur le $corpus_2$ 15

2.5 Algorithme RETAS - Résultat d'OCRopus sur le $corpus_2$ 16

3.1 Algorithme Zonemap : premier cas limite. En vert les éléments de référence, en bleu les éléments du système 24

3.2 Algorithme Zonemap : second cas limite. En vert les éléments de référence, en bleu les éléments du système 24

3.3 Algorithme Zonemap : cas limite global. En vert les éléments de référence, en bleu les éléments du système 25

3.4 Algorithme Zonemap+ : cas regroupant toutes les limites identifiées de l'algorithme Zonemap. En vert les éléments de référence, en bleu les éléments du système 27

3.5 Algorithme Zonemap+ : résultat du traitement du lien B1 28

3.6 Algorithme Zonemap+ : résultat du traitement du lien B2 28

3.7 Algorithme Zonemap+ : résultat du traitement du lien A1 29

3.8 Algorithme Zonemap+ : résultat du traitement du lien A2 29

3.9 Algorithme Zonemap+ : résultat du traitement du lien C1 29

3.10 Algorithme Zonemap+ : résultat du traitement du lien C2 30

3.11 Algorithme Zonemap+ - Résultat d'OCRopus sur le corpus_1 32

3.12 Algorithme Zonemap+ - Résultat d'OCRopus sur le corpus_{1bis} 32

3.13 Algorithme Zonemap+ - Résultat d'OCRopus sur le corpus_{1bis} 33

3.14 Algorithme Zonemap+ - Résultat de Tesseract sur le corpus_{1bis} 33

3.15 Algorithme Zonemap+ - Résultat d'OCRopus sur le corpus_2 34

3.16 Algorithme Zonemap+ - Résultat de Tesseract sur le corpus_2 34

8.1 Document 7 du $corpus_1$ 53

10.1 Exemple de document complexe du corpus_2 avec à la fois une partie linéaire et une partie non linéaire. 57

Liste des tableaux

1.1 Les 3 types d'entrées donnés au système d'évaluation 8

2.1 Algorithme RETAS - Résultats théorique et réel de Tesseract et OCRopus sur le $corpus_1$ 13

2.2 Algorithme RETAS - Résultat de Tesseract et d'OCRopus sur le $corpus_2$ comparé aux résultats théoriques 15

3.1 Algorithme Zonemap+ : données des rectangles du cas limite global 27

3.2 Algorithme Zonemap+ : matrice de la force des liens entre chaque paire d'éléments systèmes et références 28

3.3 Algorithme Zonemap+ : résultat du cas limite global 30

3.4 Description des corpus utilisés pour le cas XML/XML 31

3.5 Algorithme Zonemap+ - Résultat de segmentation des OCRs OCRopus et Tesseract sur plusieurs corpus 33

3.6 Algorithme Zonemap+ Alignment - Résultat de l'évaluation de la méthode sur le $corpus_{2bis}$ avec l'OCR OCRopus 34

3.7 Algorithme Zonemap+ Alignment - Taux de reconnaissance de caractères des OCRs OCRopus et Tesseract 35

4.1 Texte / Xml - Rappel et précision de l'alignement de la sortie d'OCRopus sur le $corpus_{2bis}$ 42

4.2 Texte / Xml - Comparaison des taux de reconnaissance entre la vérité et la méthode sur le $corpus_{2bis}$ avec l'OCR OCRopus 42

4.3 Texte / Xml - Comparaison des taux de reconnaissance entre OCRopus et Tesseract sur le $corpus_{2bis}$ 42

Annexes

6 Algorithme Zonemap+

Soit la liste des rectangles de la vérité : $V = \{V_0, V_1, .., V_i\}, i \in \mathbb{N}$
Soit la liste des rectangles du système : $S = \{S_0, S_1, .., Sj\}, j \in \mathbb{N}$
Soit l'ensemble L des liens : $\{S_i, V_j\}, i, j \in \mathbb{N}$
Soit l'ensemble L' des liens validés : $\{S_i, V_j\}, i, j \in \mathbb{N}$

Algorithm 5: Zonemap+ Algorithm

```
1  begin
2      Compute links
3      Sort links by descending order
4      for L_k ∈ L do
5          PolygonS ⟵ L_k.S
6          PolygonV ⟵ L_k.V
7          if {L_k.S, V_j} ∈ L' then
8              PolygonS ⟵ PolygonS − (PolygonS ∩ ( ⋃_{V_i ∈ L_k.S} V_i))
9              PolygonV ⟵ PolygonV − (PolygonV ∩ ( ⋃_{V_i ∈ L_k.S} V_i))
10             Config.Merge ⟵ card(V_i ∈ L_k.S)
11         end
12         if {S_i, L_k.V} ∈ L' then
13             PolygonV ⟵ PolygonV − (PolygonV ∩ ( ⋃_{S_i ∈ L_k.S} S_i)
14             Config.Split ⟵ card(S_i ∈ L_k.V)
15         end
16         if Area(PolygonS ∩ PolygonV) > threshold then
17             if not Config.Split & not Config.Merge then
18                 Config.Math ⟵ 1
19             end
20             ComputeError(config,PolygonS,PolygonT)
21             L_k ∈ L'
22             L_k.S ⟵ L_k.S − L_k.S ∩ PolygonS
23             L_k.V ⟵ L_k.V − L_k.V ∩ PolygonV
24         end
25     end
26 end
```

7 Sortie XML de l'OCR "OCRopus" en format HOCR

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <title>OCR Results</title>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <meta name="Description" content="OCRopus Output" />
  <meta name="ocr-system" content="ocropus-0.4" />
  <meta name="ocr-capabilities" content="ocr_line ocr_page" />
</head>
<body>
  <div class='ocr_page' title='file results/temp/0001.bin.png'>
    <span class='ocr_line' title='bbox 755 2421 1023 2459'>REVIEWS</span><br />
    <p />
    <span class='ocr_line' title='bbox 267 2325 1510 2367'>TE Roan ro JusrtcE.
   By the Rr. Hos. Sm Aasasn DENNINa.</span><br />
    <span class='ocr_line' title='bbox 352 2283 1511 2326'>[London : Stevens \&
   amp; Sons, Lt. 1955. viii and 118 pp-</span><br />
    <span class='ocr_line' title='bbox 349 2243 594 2283'>10s. 6d. net.]</span><
   br />
    <span class='ocr_line' title='bbox 273 2168 1515 2202'>Is the last two years
    Lord Justlce Denning has delivered a number of</span><br />
    </div>
</body>
</html>
```

Listing 7.1 – Exemple de sortie XML de l'OCR "Ocropus" en format HOCR

8 Document n°7 du $corpus_1$


SPÉCIAL NEIGE

LE MASQUE DE CHAMPION
C'est le masque d'Enak Gavaggio, athlète en ski cross en équipe de France. Vision grand large et verre anti-buée photochromique, à savoir qu'il varie d'intensité selon la luminosité.
140 €, Julbo, modèle Dark Lord, rens. : 03-84-34-14-14 ou www.julbo-eyewear.com

LE SNOWBOARD « BACKCOUNTRY »
C'est la planche *freeride backcountry* par excellence. Traduction, c'est celle des *freeriders* qui, loin dans la montagne, construisent eux-mêmes leurs aires de bosses. Ce « snow » est à double cambre, classique et inversé. Le « shape » est en forme de vague, donc. Traduction : la planche est concave sous les pieds et convexe à l'avant, à l'arrière et entre les pieds. Il est parfait pour la vitesse et les figures.
550 €, Burton, modèle Flying V, rens. : Urban Surfer, 01-53-10-85-88 et www.urbansurfers.fr

LE SAC À DOS SECOURISTE
On ne sait jamais… Muni d'un embout permettant de respirer l'oxygène circulant sous la neige en cas d'avalanche, ce sac à dos hyperrésistant, en toile balistique comme celle des gilets pare-balles, est ce que l'on fait de mieux pour les sorties sécurisées en *freeride*.
179 €, Blackdiamond, modèle Avalung, rens. : 04-90-68-68-68.

202 • LE NOUVEL OBSERVATEUR

FIGURE 8.1 – Document 7 du $corpus_1$

9 Corpus$_2$

Neuroradiology 14, 67–71 (1977)

Neuroradiology
© by Springer-Verlag 1977

The Glabella-Inion Line as a Baseline for CT Scanning of the Brain

A. Tokunaga[1], M. Takase[2], and K. Otani[1]

[1] The Third Department of Anatomy and the [2] Department of Neurological Surgery, School of Medicine, Chiba University, Chiba, Japan

Summary. The optimal position of the head for CT scanning of the brain is discussed according to the statistical data of roentgencraniometry. The glabella-inion (GI) line was shown to be parallel to the frontal-occipital pole (FO) line which was substituted for a line joining the deepest gyral impressions in the frontal and occipital parts of the cranial cavity. On the other hand, Reid's baseline and the canthomeatal (CM) line intersected the GI line at various angles from person to person. There was a significant difference between each average inclination angle of males and females. Regarding using either Reid's or the CM line for a baseline, it should be noted that the brains were scanned in almost random direction. Therefore it may be convenient to apply the GI line for comparing the findings between individuals, and for correlating them to the anatomical diagnosis.

Key words: Cranial CT – Head position – Glabella-inion line

Introduction

A new innocuous radiological technic, computed tomography (CT), was developed by Hounsfield [2] in 1973. It is possible to analyze attenuation coefficients within a cross section of the brain and to display the detailed plot of coefficients in a tomogram as in a section of a brain atlas. The technic has proved to be very important for clinical neurology.

In order to interpret the scanning picture, it is necessary to master the precise anatomy of brain sections which are cut at the same angle for the scanning. Customarily either Reid's or the canthomeatal (CM) line which joins the outer canthus to the center of the external auditory meatus, has been used as the baseline for cranial scanning. In many recently published atlases of the brain for computed tomography, the direction of slicing appears to be based on either of these lines without considering the positional relation between the extracranial lines and the brain. When comparing individual brains, it is convenient to have a certain standard line. We chose the glabella-inion (GI) and the frontal-occipital pole (FO) lines as the reference for both the head and brain, in order to correlate the baseline of the cranium with that of the brain.

Materials and Methods

Skulls of 15 males and 7 females (ranging from 24 to 70 years) were used. Their cranial cavities were opened and a right lateral roentgenograph was made in order to investigate a correlation between the GI and cranial FO lines. The skull was fixed by the Martin craniophore in Reid's horizontal plane (so called ear-eye plane or Frankfurt horizontal plane). The X-ray tube was set on the lateral extension of the sella turcica in the same plane, and the film was perpendicular to the plane. The frontal and occipital poles of the brain were substituted for the deepest gyral impressions in the left anterior cranial fossa and the upper part of the left transverse sinus sulcus in the posterior cranial fossa. These regions could be highly contrasted by putting a small piece of Oxycell which was immersed in the Pantopaque. The glabella was set at the anterior tip of its protrusion [4] and the inion was regarded as a midpoint of the base of the external occipital protuberance on the roentgenogram [6]. The distance between the GI and cranial FO lines was indicated by the length of a per-

it attempts to operationally delineate the concept of awareness of group hostility. Second, it suggests a technique for the measurement of awareness. Third, the technique can be applied to the measurement of awareness of other social problems. Fourth, in a given community the relative positions of awareness to different social problems could be ascertained. Finally, the items of an awareness instrument should be tested for the scalability of these items.

THE RAS TAFARI MOVEMENT IN JAMAICA: A STUDY OF RACE AND CLASS CONFLICT*

GEORGE EATON SIMPSON

Oberlin College

THE contra-acculturative aspects of Messianic cults and nativistic movements have long been of interest to anthropologists and sociologists.[1] Ras Tafari, a Jamaican cult which originated in 1930, is violently anti-white on the verbal level. Its members regard Haile Selassie (Ras Tafari), Emperor of Abyssinia, as the living God, see no hope for black men in the British West Indies, and look forward to an early return to Ethiopia.

The "Rasta" people consider Marcus Garvey, revered founder of the Universal Negro Improvement Association, as the forerunner of their movement. They claim that Garvey, "the world's greatest statesman," was sent by Ras Tafari "to cut and clear."[2] Garvey advocated a mass migration to Africa, and his slogans "Africa for the Africans—At Home and Abroad" and "One God! One Aim! One Destiny!" are proclaimed at every Ras Tafari meeting.

In the early days of the movement, opposition came from both the ordinary Jamaicans and the police. Lower class Jamaicans stoned speakers, slashed banners, and smashed lamps at street meetings. An active early leader of the cult was arrested, jailed, and tried seven times, but never convicted, on charges of disorderly conduct, ganja (marihuana) smoking, and lunacy. Open hostility to the movement has declined to some extent in recent years due, in part, to the well-disciplined control of members during meetings. Middle and upper class Jamaicans, as well as foreigners, still fear the Ras Tafarians, but available evidence does not support the widespread belief that they are bearded hoodlums.

Western Kingston and Eastern St. Andrew constitute the center of the Ras Tafari movement, but groups have been formed in other parts of the island. Participants are lower class Jamaicans, many of them unemployed or underemployed, who reside in crowded, blighted areas.

At present, twelve or fifteen Ras Tafari groups operate in Kingston and St. Andrew, with memberships ranging from twenty-five to one hundred and fifty or more. Groups form, split, and dissolve, and some individuals accept cult beliefs without attaching themselves to an organization. In contrast to a Revivalist group, which is dominated by a leader, a Ras Tafari band is extremely democratic. Everyone who wishes to speak must be heard, often at some length, and no action is taken without a vote of the membership, or, at the least, the executive committee. Names of these groups include: Ethiopian Coptic League, United Ethiopian Body, Ethiopian Youth Cosmic Faith,

* With the support of a grant from the American Philosophical Society. I am indebted to Mr. Arthur Bethune, of Kingston, Jamaica, for assistance in the collection of data on the Ras Tafari movement. Paper read at the annual meeting of the American Sociological Society, September, 1954.

[1] See James Mooney, "The Ghost Dance Religion and Sioux Outbreak of 1890," *Bureau of American Ethnological Reports*, 14, part 2 (1892); A. H. Gayton, "The Ghost Dance of 1870 in South-Central California," *University of California Publication in Archaeology and Ethnology*, 28 (1930); Bernard Barber, "Acculturation and Messianic Movements," *American Sociological Review*, 6 (1941), pp. 663–669; Ralph Linton, "Nativistic Movements," *American Anthropologist*, 45 (1943), pp. 230–240; M. J. Herskovits, *Man and His Works* (New York: Knopf, 1948), pp. 531–532.

[2] This expression is used in the Jamaican Revivalist cults (Pocomania and Revival Zion) to refer to the process of removing evil spirits by ritualistic means.

En revanche, les fibroblastes, après avoir été activés par l'IL-4 et/ou le TNF-α, deux cytokines connues pour être libérées sur le site des réactions allergiques.
Ces résultats suggèrent donc que les fibroblastes dermiques humains sont la principale source d'éotaxine, et sont probablement les principaux responsables de l'éosinophilie cutanée qui caractérise la DA.

C.P.

La cyclosporine A induit une diminution du taux sérique du SCD30 dans la dermatite atopique : un possible mécanisme d'action

V. Bottari, A. Frezzolini, M. Ruffelli, et al.
Cyclosporin (A) (CyA) reduces SCD30 serum levels in atopic dermatitis: a possible new immune intervention
Allergy 1999 ; 54 : 507-10

dermatite atopique

La molécule CD30 est un marqueur d'activation des lymphocytes Th2, et est libérée, sous forme soluble (SCD30), après activation de ces cellules. Les résultats d'une étude récente ont montré l'existence d'une augmentation du taux sérique du SCD30 chez les enfants atteints de DA, cette augmentation étant corrélée avec la sévérité de la DA. Enfin, les résultats d'autres études ont montré que la cyclosporine A induisait une amélioration clinique significative chez les patients atteints de DA sévère.
Les résultats de cette étude, effectuée chez des adultes atteints de DA sévère, montrent que, par rapport aux traitements classiques (corticoïdes locaux et émollients), les traitements par la cyclosporine A (3,5 mg/kg/24 h x 3 mois) induisent une amélioration clinique significative, qui est plus ou moins bien corrélée avec une diminution des taux sériques du SCD30 et de l'IL-4, ainsi qu'avec une diminution du taux sérique du SIL-2R (récepteur soluble de l'IL-2), un marqueur général d'activation lymphocytaires).
Ces résultats suggèrent que, chez les patients atteints de DA sévère, la cyclosporine A inhibe la génération et/ou l'activation des lymphocytes Th2, et, de ce fait, induit une amélioration clinique significative. Toutefois, il ne faut pas oublier les résultats d'autres études, effectuées dans l'espèce humaine et chez l'animal, qui ont montré que la cyclosporine A agissait essentiellement sur les lymphocytes Th1, et pouvait induire une augmentation des taux des IgE sériques totales et spécifiques.

C.P.

Médiateurs chimiques de la dermatite atopique : mise en évidence du rôle joué par le LTB4, libéré au cours de la réaction allergique immédiate, dans la pathogénie de la dermatite atopique

O. Koro, K. Furutani, M. Hide, et al.
Chemical mediators in atopic dermatitis: involvement of leukotriene B4 released by a type I allergic reaction in the pathogenesis of atopic dermatitis
J Allergy Clin Immunol 1999 ; 103 : 663-70

dermatite atopique / inhibiteurs des leucotriènes

La dermatite atopique (DA) est une maladie inflammatoire de la peau, où divers médiateurs (histamine notamment) et, surtout, diverses cytokines (IL-4, IL-5, etc.), semblent jouer un rôle important, les premiers dans l'exacerbation initiale des poussées, et les secondes dans l'induction et la prolongation de l'inflammation cutanée.
Les auteurs ont étudié les concentrations de LTB4 (leucotriène B4) libérées, après contact avec l'allergène, par des fragments de peau de patients atteints de DA (étude in vitro), ainsi que dans le liquide de succion des TC (étude in vivo) chez des patients également atteints de DA. Par rapport aux résultats obtenus avant le contact avec l'allergène, une importante augmentation des concentrations d'histamine et de LTB4 a été observée, tant in vitro qu'in vivo, en peau lésée, après contact avec l'allergène. Une augmentation de la concentration de ces deux médiateurs, bien que plus discrète, a également été observée en peau saine, après injection par voie intradermique de l'allergène.
Ces résultats suggèrent donc que les leucotriènes jouent un rôle dans la pathogénie de la DA, et suggèrent aussi que les inhibiteurs et les antagonistes des leucotriènes pourraient représenter une classe thérapeutique intéressante dans le traitement de la DA.

C.P.

Étude de l'expression des cytokines IL-1β, TNF-α et IL-12 sur le site d'application épicutanée des acariens domostiques chez les patients atteints de dermatite atopique

V. Junghas, C. Gutgesell, T. Jung, C. Neumann
Epidermal cytokines IL-1β, TNF-α, and IL-12 in patients with atopic dermatitis: response to application of house dust mite antigens
J Invest Dermatol 1998 ; 111 : 1184-8

dermatite atopique / IL-1 / IL-12 / TNF

Les cytokines telles l'IL-1, le TNF-α et l'IL-12 jouent un rôle déterminant dans l'induction et l'expression des eczémas de contact, et ont été mise en évidence dans les lésions cutanées d'hypersensibilité retardée et dans les patch-tests effectués avec des haptènes divers. Par ailleurs, la dermatite atopique (DA) est une affection comportant une composante d'hypersensibilité immédiate et une composante d'hypersensibilité retardée, avec des patch-tests aux allergènes souvent positifs.
Les auteurs ont effectué des patch-tests aux acariens chez des patients atteints de DA, et étudié l'expression des ARN messagers (mARN) de l'IL-1, du TNF-α et de l'IL-12 sur le site des patch-tests, 8 heures après l'application des allergènes : les résultats ont été comparés avec ceux obtenus chez des témoins non atopiques. Aucune modification significative n'a été observée chez ces derniers, alors qu'une augmentation significative de l'expression des mARN de l'IL-1β, du TNF-α et de l'IL-12 a été mise en évidence chez les atopiques, à l'état basal, et, surtout, après application des allergènes. Chez les atopiques, l'application de substances irritantes, telles le lamyl-sulfate de sodium, a également induit une légère augmentation de l'expression des mARN de l'IL-1β et du TNF-α.
Ces résultats confirment donc le rôle joué par la composante d'hypersensibilité retardée dans la DA : cette composante, déjà détectable à l'état basal, est majorée de façon significative par les expositions cutanées aux allergènes.

C.P.

Hépatite C
- ARN qualitatif RT-PCR
- ARN quantitatif RT-PCR
- génotypage (LIPA)

Hépatite B
ADN quantitatif PCR

VIH-1
- ARN quantitatif plasmatique RT-PCR (200 et 20 copies/ml)
- ARN quantitatif plasmatique BDNA
- ARN quantitatif ganglionnaire * (ARN totaux et messagers)
- ADN proviral quantitatif sur PBMC et LNMC *
- Résistance génotypique aux antirétroviraux : LIPA et séquençage
- ADN *gag* PCR
- ADN *pol* PCR

* Réservé aux patients sous protocole

CMV
- ADN quantitatif - hybridation dans les leucocytes et le plasma
- ADN qualitatif plasmatique PCR

HSV1 - HSV2 ADN PCR
HHV8 ADN PCR
VZV ADN PCR
EBV ADN PCR
JC ADN PCR
Entérovirus ARN RT-PCR
Papillomavirus hybridation

Autres agents infectieux
- *Chlamydia trachomatis* PCR
- Mycobactéries PCR

Hémostase
Mutation Q506 PCR

du virus de l'immunodéficience humaine (VIH), la nécessité de créer une unité de biologie moléculaire s'est très vite imposée et ce secteur a depuis « explosé » avec le souci de développer en permanence de nouveaux paramètres et d'améliorer la qualité des tests déjà en place.

L'installation de la biologie moléculaire a été possible grâce à une confiance réciproque établie avec la direction du CHITS qui a alloué une enveloppe permettant l'acquisition du matériel de base. De plus, l'adhésion unanime de l'ensemble de l'équipe, prêt à suivre les formations nécessaires à la pratique de ces techniques délicates, a permis l'intégration progressive de ces tests dans la routine du laboratoire.

Dès que cela a été possible, certains tests (VHC qualitatif, *Chlamydia*, mycobactéries) ont été effectués sur Cobas Amplicor (Roche), semi-automate qui réalise l'amplification et la détection des produits de PCR, avec un contrôle interne d'amplification. Les quantifications virales sur cet appareil sont à l'étude actuellement.

L'acquisition la plus récente est un séquenceur automatique consacré actuellement à l'étude de la résistance génotypique du VIH.

Les champs d'application de la biologie moléculaire en médecine sont vastes. Nous avons principalement développé les paramètres virologiques et bactériologiques *(tableau I)* :

- les tests de biologie moléculaire pour le diagnostic et le suivi des hépatites B et C ont été les premiers mis en place ;
- le VIH a particulièrement bénéficié du développement de ces techniques, que ce soit pour le diagnostic chez le nouveau-né et surtout pour le suivi de l'infection à VIH, mais aussi pour le diagnostic des principales maladies opportunistes : herpès virus 8 (HHV 8) lié au sarcome de Kaposi, virus de la leucoencéphalite multifocale progressive (LEMP), cytomégalovirus (CMV) et mycobactéries ;
- la mise en évidence des agents infectieux responsables d'infection génitale, ou des virus neurotropes par la technique de polymerase chain reaction (PCR), permet d'apporter une réponse rapide au clinicien. D'autres domaines sont concernés comme l'hématologie avec la recherche de la mutation Q506, facteur de risque thrombotique et l'accès aux techniques de séquençage devrait permettre d'élargir encore nos centres d'intérêt.

2. Virus de l'hépatite C (VHC)

La sérologie du virus de l'hépatite C ne fournit pas de renseignement sur l'état de réplication du virus dans l'organisme. Les tests de validation type RIBA n'apportent aucun élément supplémentaire et sont actuellement pratiquement abandonnés. Il semble plus intéressant, lorsque la sérologie est positive, confirmée par une deuxième technique, d'effectuer la recherche de l'ARN du VHC, réalisée ici sur Cobas Amplicor, dont le seuil actuellement à 1 000 copies/ml sera prochainement abaissé à 100 copies.

Les indications actuelles de l'ARN VHC par PCR sont [12] :

- si la sérologie est positive : le bilan pré-thérapeutique, l'évaluation du traitement, le bilan du nouveau-né de mère séropositive pour le VHC, la recherche d'une réplication chez le patient à transaminases normales de façon répétée ou dans le cadre d'une hépatopathie ayant plusieurs causes possibles ;
- si la sérologie est négative ou discordante : hépatopathie aiguë d'étiologie indéterminée, hépatopathie chronique chez le sujet immunodéprimé, exploration d'une maladie systémique ;
- bilan d'accident d'exposition au sang (AES) : si le sujet source est virémique pour le VHC ou si son statut est inconnu (PCR VHC tous les mois jusqu'au 3e mois) [6, 7].

Lorsqu'un traitement est envisagé, les investigations pourront être poursuivies avec :

- le typage du virus : nous effectuons la détermination du génotype viral par la méthode LIPA (Innogenetics) avant d'aborder prochainement le séquençage de la région 5' non codante, moins coûteux [8]. Dans les deux cas, il est possible de travailler sur les produits d'amplification obtenus sur Cobas Amplicor, d'où une réduction du coût et un gain de temps ;
- la quantification de l'ARN viral plasmatique : la virémie peut être déterminée par la technique bDNA (Chiron) ou par PCR quantitative type Amplicor Monitor Roche, dont le seuil est à 1 000 copies/ml.

Plusieurs schémas sont possibles pour le suivi thérapeutique, avec par exemple une recherche de réplication virale tous les 3 mois pendant un an de traitement et enfin 3 à 6 mois après l'arrêt de celui-ci à la recherche d'une rechute éventuelle.

3. Virus de l'hépatite B (VHB)

L'infection chronique par le virus de l'hépatite B est un phénomène de santé publique car, malgré l'existence d'un vaccin efficace, 300 millions de personnes sont contaminées dans le monde et près de 200 000 en France. Le suivi virologique de ces patients a longtemps été réservé aux laboratoires de recherche (hybridation avec marquage radio-actif, PCR qualitative ou quantitative « maison ») jusqu'à la com-

La biologie moléculaire dans les hémopathies lymphoïdes malignes

J.-P. MAGAUD, J.-P. ROUAULT, I. TIGAUD, H. LAPILLONNE, C. SAMARUT, P.-A. BRYON et R. RIMOKH *

RÉSUMÉ

L'introduction des techniques de biologie moléculaire en hématologie depuis une dizaine d'années a permis une avancée considérable dans l'étude des hémopathies lymphoïdes malignes avec des implications dans le domaine de la recherche fondamentale et de la recherche clinique.

La biologie moléculaire a en premier lieu conduit à une meilleure compréhension des mécanismes génétiques de la tumorogenèse lymphoïde : activation de proto-oncogènes, inactivation de gènes suppresseurs de tumeur consécutives aux anomalies chromosomiques spécifiques fréquemment associées aux hémopathies lymphoïdes. L'étude des réarrangements des gènes codant pour les immunoglobulines ou pour le récepteur antigénique des cellules T dans les prolifération lymphoïdes malignes a permis de situer ces tumeurs par rapport aux différentes voies de différenciation des cellules immunocompétentes. Les marqueurs moléculaires associés à la différenciation ou à l'oncogenèse des cellules lymphoïdes sont autant d'éléments dont il faut actuellement tenir compte lors du diagnostic et du suivi thérapeutique des hémopathies lymphoïdes malignes.

Enfin, la recherche de séquences virales dans certaines hémopathies lymphoïdes, l'étude de la reconstitution hématopoïétique après greffe de moelle ont également bénéficié de l'apport de la biologie moléculaire.

MOTS-CLÉS

hémopathies lymphoïdes malignes – biologie moléculaire – tumorogenèse – maladie résiduelle – proto-oncogènes – gènes suppresseurs de tumeurs.

SUMMARY

Recent developments in molecular biology have produced considerable advances in our understanding of the principles underlying lymphoid tumorogenesis.

In particular, molecular biology has shown that many chromosome changes in human leukemia are highly specific and result in the activation or mutation of genes (proto-oncogenes, tumor suppressor genes) involved in cell proliferation and its regulation. The existence of antigen receptor genes recombination during the generation of mature B and T cells has been used to design molecular tools which can be used for the characterization of the clonal status of lymphoid disorders and for the detection of minimal residual disease. These differentiation markers as well as the above-defined tumoral markers have indeed implications for both the diagnosis and the clinical management of malignant lymphoproliferations.

Other applications of molecular biology in hematology are the search for viral sequences in some lymphoid tumors, the study of hematopoiesis restauration after bone marrow engrafment.

KEY-WORDS

lymphoid malignancies – molecular biology – tumorogenesis – minimal residual disease – proto-oncogenes – tumor suppressor genes.

Introduction

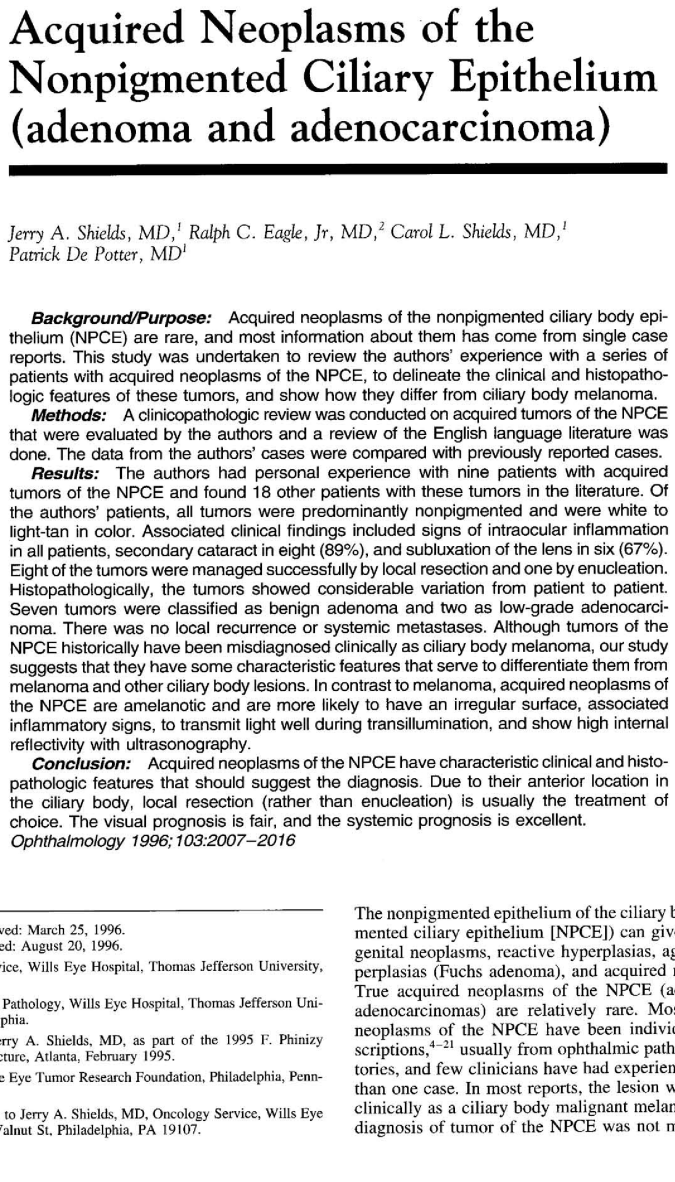
L'hématologie, comme la plupart des disciplines cliniques et biologiques, a largement bénéficié de l'apport des techniques de biologie moléculaire depuis le début des années 1980. Dans cette synthèse, nous développerons l'exemple des hémopathies lymphoïdes malignes, affections les plus fréquentes en hémato-oncologie. Comme nous le verrons, certaines des applications de la biologie moléculaire dans ce domaine restent du ressort de la recherche appliquée ou fondamentale. D'autres ont déjà trouvé une place en biologie clinique tant au niveau de l'aide diagnostique que de la surveillance

* *Unité d'hémato-oncologie moléculaire INSERM CJF 93-07*
Laboratoire central d'hématologie et de cytogénétique
Pavillon E – Hôpital Edouard-Herriot
69437 LYON CEDEX 03

TIRÉS A PART :
M. le Dr J.-P. MAGAUD

article reçu le 14 décembre 1994, accepté le 26 janvier 1995.

10 Exemple de document complexe - Corpus$_2$



Acquired Neoplasms of the Nonpigmented Ciliary Epithelium (adenoma and adenocarcinoma)

Jerry A. Shields, MD,[1] Ralph C. Eagle, Jr, MD,[2] Carol L. Shields, MD,[1] Patrick De Potter, MD[1]

Background/Purpose: Acquired neoplasms of the nonpigmented ciliary body epithelium (NPCE) are rare, and most information about them has come from single case reports. This study was undertaken to review the authors' experience with a series of patients with acquired neoplasms of the NPCE, to delineate the clinical and histopathologic features of these tumors, and show how they differ from ciliary body melanoma.

Methods: A clinicopathologic review was conducted on acquired tumors of the NPCE that were evaluated by the authors and a review of the English language literature was done. The data from the authors' cases were compared with previously reported cases.

Results: The authors had personal experience with nine patients with acquired tumors of the NPCE and found 18 other patients with these tumors in the literature. Of the authors' patients, all tumors were predominantly nonpigmented and were white to light-tan in color. Associated clinical findings included signs of intraocular inflammation in all patients, secondary cataract in eight (89%), and subluxation of the lens in six (67%). Eight of the tumors were managed successfully by local resection and one by enucleation. Histopathologically, the tumors showed considerable variation from patient to patient. Seven tumors were classified as benign adenoma and two as low-grade adenocarcinoma. There was no local recurrence or systemic metastases. Although tumors of the NPCE historically have been misdiagnosed clinically as ciliary body melanoma, our study suggests that they have some characteristic features that serve to differentiate them from melanoma and other ciliary body lesions. In contrast to melanoma, acquired neoplasms of the NPCE are amelanotic and are more likely to have an irregular surface, associated inflammatory signs, to transmit light well during transillumination, and show high internal reflectivity with ultrasonography.

Conclusion: Acquired neoplasms of the NPCE have characteristic clinical and histopathologic features that should suggest the diagnosis. Due to their anterior location in the ciliary body, local resection (rather than enucleation) is usually the treatment of choice. The visual prognosis is fair, and the systemic prognosis is excellent.

Ophthalmology 1996;103:2007–2016

Originally received: March 25, 1996.
Revision accepted: August 20, 1996.

[1] Oncology Service, Wills Eye Hospital, Thomas Jefferson University, Philadelphia.

[2] Department of Pathology, Wills Eye Hospital, Thomas Jefferson University, Philadelphia.

Presented by Jerry A. Shields, MD, as part of the 1995 F. Phinizy Calhoun, Jr, Lecture, Atlanta, February 1995.

Supported by the Eye Tumor Research Foundation, Philadelphia, Pennsylvania.

Reprint requests to Jerry A. Shields, MD, Oncology Service, Wills Eye Hospital, 900 Walnut St, Philadelphia, PA 19107.

The nonpigmented epithelium of the ciliary body (nonpigmented ciliary epithelium [NPCE]) can give rise to congenital neoplasms, reactive hyperplasias, age-related hyperplasias (Fuchs adenoma), and acquired neoplasms.[1–3] True acquired neoplasms of the NPCE (adenomas and adenocarcinomas) are relatively rare. Most reports on neoplasms of the NPCE have been individual case descriptions,[4–21] usually from ophthalmic pathology laboratories, and few clinicians have had experience with more than one case. In most reports, the lesion was diagnosed clinically as a ciliary body malignant melanoma, and the diagnosis of tumor of the NPCE was not made until the

FIGURE 10.1 – Exemple de document complexe du corpus$_2$ avec à la fois une partie linéaire et une partie non linéaire.