Une catégorisation de fins de lignes non-supervisée

Pierre Zweigenbaum[1] Cyril Grouin[1] Thomas Lavergne[1,2]
(1) LIMSI, CNRS, Université Paris-Saclay, F-91405 Orsay
(2) LIMSI, CNRS, Univ. Paris-Sud, Université Paris-Saclay, F-91405 Orsay
pz,grouin,lavergne@limsi.fr

RÉSUMÉ

Dans certains textes bruts, les marques de fin de ligne peuvent marquer ou pas la frontière d'une unité textuelle (typiquement un paragraphe). Ce problème risque d'influencer les traitements subséquents, mais est rarement traité dans la littérature. Nous proposons une méthode entièrement non-supervisée pour déterminer si une fin de ligne doit être vue comme un simple espace ou comme une véritable frontière d'unité textuelle, et la testons sur un corpus de comptes rendus médicaux. Cette méthode obtient une F-mesure de 0,926 sur un échantillon de 24 textes contenant des lignes repliées. Appliquée sur un échantillon plus grand de textes contenant ou pas des lignes repliées, notre méthode la plus prudente obtient une F-mesure de 0,898, valeur élevée pour une méthode entièrement non-supervisée.

ABSTRACT

End-of-line classification with no supervision

In some plain text documents, end-of-line marks may or may not mark the boundary of a text unit (e.g., of a paragraph). This problem is likely to impact subsequent natural language processing components, but is seldom addressed in the literature. We propose a fully unsupervised method to classify whether end-of-lines must actually be seen as simple spaces (soft line breaks) or as true text unit boundaries and test it on a corpus of clinical texts. This method achieves 0.926 F-measure on a random sample of 24 texts with soft line breaks. When applied to a larger sample of mixed texts which may or may not contain soft line breaks, our more conservative method achieves 0.898 F-measure, again a high value for a fully unsupervised method.

MOTS-CLÉS : Classification non-supervisée ; classification des fins de lignes ; paragraphes repliés.

KEYWORDS: Unsupervised classification ; end-of-line classification ; wrapped paragraphs.

1 Introduction

Contexte et motivation La segmentation de textes est une tâche « de bas niveau » qui contribue aux tâches d'analyse « de plus haut niveau » réalisées en traitement automatique des langues. Par exemple, Smith (2011, p. 5) écrit : « *If we build a language model on poorly segmented text, for instance, its predictive performance will suffer.* ». Nous nous intéressons ici à un type de découpage peu décrit dans la littérature, qui a des conséquences sur le découpage en phrases et donc sur la suite des traitements. Il peut être décrit comme la détermination des frontières de paragraphes, ou encore la *classification des fins de lignes*. Il est pertinent pour les textes dans lesquels les fins de lignes peuvent ou non jouer le rôle de frontière de paragraphe, donc de phrase. On rencontre ce phénomène dans divers types de documents, par exemple dans nombre de courriers électroniques (ex : corpus ENRON)

ou de comptes rendus hospitaliers (ex : corpus i2b2/UTHealth ou MIMIC II). Même si l'impact précis de la classification de fins de lignes reste à étudier et peut dépendre des types de traitements subséquents visés, le nombre de situations dans lesquelles on peut le rencontrer justifie l'étude de méthodes générales pour le traiter.

Travaux antérieurs Le problème de la classification des fins de lignes semble peu exploré en traitement automatique des langues, et Smith (2011) ne le mentionne pas. Quelques travaux sur l'analyse de textes hospitaliers ont récemment commencé à l'aborder par des heuristiques (Zweigenbaum & Grouin, 2014) ou de la classification supervisée (Miller *et al.*, 2015).

Ce problème a néanmoins été étudié en analyse de documents, après une reconnaissance de caractères ou pour reformater un texte obtenu à partir d'un fichier PDF. Radakovic *et al.* (2013) vérifient si une ligne débute avec certains symboles (puces) ou casse (majuscules) ou se termine par un nombre, ainsi que d'autres indices comme le nombre de mots dans la ligne, l'indentation et la taille de police. Fang *et al.* (2011) examinent la position des caractères dans la page et l'indentation, mais pas d'indices sur le contenu des lignes. Aiello *et al.* (2002) combinent des informations sur le positionnement spatial des lignes et paragraphes et sur la probabilité que la catégorie morphosyntaxique du premier mot de la ligne suivante suive celles des deux derniers mots de la ligne précédente. Dans notre situation, aucune information de positionnement spatial ou de taille de police n'est disponible, seul le contenu et la taille des lignes peuvent être employés.

Nous cherchons de plus à trouver une méthode qui soit aussi peu supervisée que possible. Nous verrons dans la section suivante que nous pouvons nous placer dans une situation d'apprentissage où l'on cherche à classer chaque « blanc » [1] (*white space*) d'un texte (fins de lignes incluses) comme un simple espace séparateur de mots (<SP>) ou une réelle marque de frontière d'unité textuelle (<FUT>), une partie des annotations disponibles (les fins de lignes) étant cependant incertaine. Nigam *et al.* (2000) utilisent l'algorithme espérance-maximisation (EM) pour améliorer itérativement un classifieur bayésien naïf en exploitant des données non annotées en plus de données annotées. Une façon de modéliser notre situation consisterait à considérer les fins de lignes comme des exemples non annotés et les autres blancs comme des exemples positifs pour <SP>. Diverses méthodes ont été proposées pour apprendre en présence d'exemples positifs et d'exemples non annotés. Elkan & Noto (2008) proposent une méthode non itérative pour cela, mais cette méthode suppose que les exemples annotés soient tirés au sort parmi les exemples positifs, ce qui n'est pas le cas dans notre situation. Une autre méthode consisterait à considérer les annotations de fins de ligne comme ambiguës et à appliquer les méthodes de Wisniewski *et al.* (2014). Cependant, cela créerait dans ces annotations une dépendance systématique entre ces deux classes, situation dans laquelle l'apprentissage n'est pas garanti (Bordes *et al.*, 2010).

2 Classification non supervisée des fins de lignes

Modélisation de la tâche Le problème de la classification des fins de lignes peut se décomposer en deux parties : (i) déterminer si un document est sujet au repliement des paragraphes, c'est-à-dire contient au moins une fin de ligne qui devrait être considérée comme un <SP>, et (ii) dans un document sujet au repliement des paragraphes, classer les fins de lignes comme <SP> ou <FUT>. Nous nous focalisons sur la seconde partie de la tâche, la classification des fins de lignes, en supposant

1. Nous regroupons dans cet article sous le terme « blanc » les espaces (y compris les tabulations) et les fins de ligne (marquées selon le système d'exploitation par les caractères LF, CR ou CR+LF).

que la première partie (plus simple) est résolue, par exemple de façon supervisée. Nous présentons aussi une direction pour traiter la première partie de façon non-supervisée, et l'intégrons dans notre méthode générale.

Détermination des textes sujets au repliement Un texte dans lequel de nombreuses lignes sont repliées a tendance à avoir des lignes de longueur similaire, donc proches de la moyenne des longueurs de lignes dans ce texte. La distribution des longueurs de lignes d'un texte par rapport à leur longueur moyenne est donc une information utile pour déterminer si un texte est sujet au repliement. L'information clé prise en compte dans (Zweigenbaum & Grouin, 2014) était le *coefficient de variation* des longueurs des lignes (plus bas dans l'équation 2), couplée à un seuil réglé de façon supervisée. Nous reprenons ici cette caractéristique mais sans supervision.

Traitement des lignes blanches Nous résolvons d'abord un cas facile : celui des lignes blanches (vides ou ne contenant que des espaces). Celles-ci marquent toujours une frontière d'unité textuelle : la fin de ligne qui précède et celle qui termine une ligne blanche sont toutes deux non-ambiguës et marquées comme telles automatiquement (classe '2' dans le tableau 2 plus bas).

Désambiguïsation des fins de lignes : méthode générale Nous partons de textes qui ont été segmentés en mots, et où les ponctuations ont été séparées des mots. Ces textes contiennent des caractères d'espacement qui jouer le rôle de simples espaces (<SP>) ou de frontières d'unité textuelle (<FUT>). Nous nous attaquons à la tâche consistant à décider, *pour chaque fin de ligne*, s'il s'agit d'un <SP> (classe positive, ou '1') ou d'un <FUT> (classe négative, ou '0'), de la façon suivante :

1. Les espaces sont annotés de façon certaine comme <SP> (exemples positifs), mais les fins de lignes sont ambiguës : nous convertissons cette situation en une annotation bruitée où les fins de lignes sont annotées comme <FUT> (exemples négatifs ; ce qui n'est bien sûr pas toujours correct).
2. Puisque seules les fins de lignes sont ambiguës, nous nous intéressons seulement à l'application de notre classifieur sur elles, pas sur les espaces.
3. L'entraînement du classifieur est effectué sur ces annotations bruitées, avec des attributs liés aux mots comme indiqué plus bas ($A_1 \ldots A_4$), et résulte en un modèle M_A pour la classification des fins de lignes.
4. Ce modèle est appliqué aux fins de lignes du corpus d'entraînement lui-même, fournissant des étiquettes de fins de lignes désambiguïsées (auto-apprentissage) : certaines sont maintenant des <SP> (exemples positifs), les autres restant des <FUT> (exemples négatifs).
5. Un second modèle M_B est entraîné sur ces nouvelles annotations de fins de lignes, positives (<SP>) et négatives (<FUT>), en employant des attributs différents (B_1, B_2 : coapprentissage).
6. Le modèle M_B (ou une combinaison $M_{A \cdot B}$ fondée sur le produit des rapports de vraisemblance calculés pour chacun des modèles bayésiens naïfs M_A et M_B) est appliqué aux fins de lignes du corpus d'entraînement, ce qui fournit des étiquettes de fins de lignes modifiées.

Nous obtenons ainsi, de façon entièrement non-supervisée, trois modèles et leurs annotations de fins de lignes associées. Nous laissons des essais d'itération du processus, comme dans (Nigam *et al.*, 2000), à des travaux futurs.

Classifieur bayésien naïf Nous employons un classifieur très simple, le classifieur bayésien naïf (*Naive Bayes*), connu pour sa robustesse et sa rapidité. Un problème classiquement rencontré est la présence d'exemples dans l'ensemble de test dont des attributs n'ont pas été rencontrés dans l'ensemble d'entraînement. Ce problème peut être traité par lissage. Nous employons le lissage de

lignes	position $b_i \downarrow$	A_1	A_2	A_3	A_4	B_1	B_2
a	no complaint	no	complaint	min	min	s/o	s/o
a	complaint .	complaint	.	min	p_forte	s/o	s/o
a,b	. REVIEW	.	REVIEW	p_forte	MAJ	– 0.32	0.59
b	REVIEW OF	REVIEW	OF	MAJ	MAJ	s/o	s/o
b	no shortness	no	shortness	min	min	s/o	s/o
b,c	shortness of	shortness	of	min	min	0.13	0.59

TABLE 1 – Exemples de valeurs d'attributs pour quelques positions des lignes 1a–1c ; min = débute par une minuscule, MAJ = tout en majuscules, p_forte = ponctuation forte, s/o = sans objet

Laplace (équation 1), qui ajoute 1 au nombre d'occurrences de l'attribut a_i avec la classe c_j, et ajoute le nombre de classes C au dénominateur :

$$\hat{P}(a_i|c_j) \approx \frac{occ(a_i, c_j) + 1}{occ(c_j) + |C|} \tag{1}$$

Attributs Nous définissons ci-dessous les attributs calculés pour chaque blanc. Le tableau 1 illustre les valeurs qu'ils prennent sur quelques positions des lignes (plausibles mais inventées) 1a–1c.

(1) a. `SUBJECTIVE: He has no complaint.`
b. `REVIEW OF SYSTEMS: Negative. There is no shortness`
c. `of breath, chest pain or orthopnea. All the other`

Nous caractérisons une position b_i (blanc : espace ou fin de ligne) entre deux mots d'un document d comprenant N lignes par les quatre attributs discrets suivants :

A_1 **(resp** A_2**)** : le mot à gauche (resp. à droite) : certains mots ou ponctuations sont souvent des fins (resp. débuts) de paragraphes, d'autres rarement ;

A_3 **(resp** A_4**)** : forme typographique du mot à gauche (resp. à droite) : tout en majuscules, débute par une majuscule, débute par une minuscule, est un nombre, est constitué uniquement de ponctuations (en distinguant ponctuations fortes et faibles), est un nombre suivi d'au moins une ponctuation et optionnellement précédé d'une ponctuation (forme typique des introducteurs d'énumérations) ; nous supposons que certaines formes typographiques, comme les majuscules, sont plus fréquentes en début de paragraphe, alors que d'autres, comme les ponctuations fortes, sont plus fréquentes en fin de paragraphe).

Nous caractérisons de plus une position b_i en fin de ligne par les deux attributs suivants :

B_1 l, longueur en caractères de la ligne terminée par l'espace b_i, centrée et réduite (l_{norm}) à l'intérieur du document d (équation 2) ; nous supposons qu'une ligne très courte a peu de chances de devoir être « recollée » à la ligne suivante, de même qu'une ligne très longue ;

$$l_{norm} = \frac{l - \mu_d}{\sigma_d}, \qquad \mu_d = \frac{1}{N}\sum_{l \in d} l, \qquad \sigma_d = \sqrt{E[(l - \mu_d)^2]}, \qquad cv_d = \frac{\sigma_d}{\mu_d} \tag{2}$$

B_2 cv_d, coefficient de variation de la longueur l des lignes dans le document d (équation 2). Cet attribut est commun à toutes les positions du document d. Nous supposons qu'un document dont les paragraphes sont repliés est susceptible d'avoir de nombreuses lignes de longueur

similaire (proches de la largeur maximale de l'écran de saisie). Cela devrait se traduire par un écart-type de la longueur des lignes faible par rapport à la moyenne de la longueur des lignes. Le coefficient de variation mesure précisément ce rapport.

Les deux derniers attributs l_{norm} et cv_d sont à valeurs numériques, nous les discrétisons en dix plages de valeurs entre leurs valeurs minimales et maximales relevées sur le corpus. Si un document du test a des valeurs hors de ces bornes, elles sont ramenées dans la plage la plus proche.

3 Expériences, résultats et discussion

Corpus d'évaluation Nous avons effectué des expériences sur le corpus d'entraînement fourni à l'occasion de la campagne d'évaluation i2b2/UTHealth 2014 (Stubbs *et al.*, 2015). Ce corpus est constitué de 790 comptes rendus cliniques rédigés en anglais (178 patients). Certains de ces textes sont en double interligne (une ligne blanche a été insérée toutes les deux lignes) : nous avons détecté et normalisé automatiquement ces textes. Le phénomène de « repliement de lignes » se produit dans certains de ces documents, mais pas dans tous. Notre algorithme doit s'appliquer à tous ces documents, sans qu'on lui indique si le phénomène est présent ou pas. Nous rappelons par ailleurs que notre algorithme est non-supervisé et n'utilise donc pour son entraînement aucune annotation humaine.

À des fins d'évaluation, nous avons annoté manuellement deux échantillons de textes du corpus. Un premier jeu de 49 textes (Dev : 11 dossiers patient entiers), avec double annotation suivie d'une phase de consensus, nous a servi pour évaluer le système lors du développement de la méthode. Un second jeu distinct de 64 textes tirés aléatoirement (Test : appartenant à 53 dossiers), annoté en simple (très peu de divergences ayant été constatées lors de l'annotation du corpus Dev), a servi à évaluer le système une fois la méthode stabilisée. L'annotation indique la classe de chaque fin de ligne : 0 (<FUT>), 1 (<SP>) ou 2 (frontière de ligne blanche, voir plus haut la section 2).

corpus	classe	0	1	2	total	documents	avec repliement
Dev	avant annotation	2101	0	2576	4677	49	–
Dev	annoté (consensus)	1877	224	2576	4677	49	9 (18 %)
Test	avant annotation	3070	0	2549	5619	64	–
Test	annoté	2468	602	2549	5619	64	24 (38 %)

TABLE 2 – Corpus Dev et Test : nombre de fins de lignes avec leur classe d'annotation, nombre de documents avec au moins un repliement (classe '1')

Évaluation Comme nous n'employons aucune supervision, nous entraînons notre système sur le corpus entier (790 textes), y compris les échantillons sur lesquels nous l'évaluons. Cela n'introduit pas de biais dans les résultats obtenus sur le corpus Test, qui n'ont pas été employés pour modifier des paramètres de la méthode.

L'évaluation est faite sur la tâche de détection des fins de lignes qui sont en réalité des espaces <SP> (classe '1', repliement de paragraphe), avec les mesures classiques de précision (P : proportion des repliements proposés par le système qui sont corrects), rappel (R : proportion des repliements corrects qui sont trouvés par le système), et F-mesure (F : moyenne harmonique de P et R).

Nous avons examiné les questions suivantes sur le corpus Dev : (i) peut-on ignorer les fins de lignes non ambiguës (classe '2') pour entraîner le modèle M_A (–2 dans les tableaux de la section suivante) ? (ii) est-il utile de distinguer les ponctuations fortes (-fortes) ? (iii) devrait-on passer en minuscules

les mots pour calculer A_1 et A_2 (+minus) ? (iv) quel modèle parmi $\{M_A, M_B$ et $M_{A \cdot B}\}$ est meilleur en termes de P, R et F ? (v) cet article se focalise sur la classification des fins de lignes dans un texte dont on sait qu'il a subi des repliements de paragraphes ; est-ce que l'introduction des attributs liés à la longueur des lignes B_1B_2 (modèles M_B et $M_{A \cdot B}$), qui incluent le coefficient de variation, aide à traiter la tâche globale de classification des fins de lignes dans un document quelconque ?

Puis nous avons retenu les meilleurs réglages et évalué nos modèles sur le corpus Test : (vi) Est-ce que les conclusions obtenues sur Dev valent encore sur Test ? (vii) Y a-t-il une baisse de performance entre Dev et Test ?

corpus	variante	vp	fp	fn	vn	exa	p	r	f
Dev : +repl	replie-aucun	0	0	220	106	0.3252	—	0.0000	0.0000
Dev : +repl	replie-tous	220	106	0	0	0.6748	0.6748	1.0000	0.8059
Dev : +repl	M_A –2, –fort	208	9	12	97	0.9356	0.9585	**0.9455**	0.9519
Dev : +repl	M_A –fort, +min	206	7	14	99	0.9356	0.9671	0.9364	0.9515
Dev : +repl	M_A –fort	205	6	15	100	0.9356	**0.9716**	0.9318	0.9513
Dev : +repl	M_A	205	7	15	99	0.9325	0.9670	0.9318	0.9491
Dev : +repl	M_B	193	27	27	79	0.8344	0.8773	0.8773	0.8773
Dev : +repl	$M_{A \cdot B}$	208	7	12	99	**0.9417**	0.9674	**0.9455**	**0.9563**
Dev : tous	replie-aucun	0	0	224	1877	0.8934	—	0.0000	0.0000
Dev : tous	replie-tous	224	1877	0	0	0.1066	0.1066	1.0000	0.1927
Dev : tous	M_A –2, –fort	212	611	12	1266	0.7035	0.2576	**0.9464**	0.4050
Dev : tous	M_A –fort, +min	210	555	14	1322	0.7292	0.2745	0.9375	0.4247
Dev : tous	M_A –fort	209	441	15	1436	0.7830	0.3215	0.9330	0.4783
Dev : tous	M_A	209	377	15	1500	0.8134	0.3567	0.9330	0.5160
Dev : tous	M_B	193	27	31	1850	**0.9724**	**0.8773**	0.8616	**0.8694**
Dev : tous	$M_{A \cdot B}$	212	197	12	1680	0.9005	0.5183	**0.9464**	0.6698
Test : +repl	M_A	560	69	42	442	0.9003	0.8903	0.9302	0.9098
Test : +repl	M_B	544	57	58	454	0.8967	0.9052	0.9037	0.9044
Test : +repl	$M_{A \cdot B}$	564	52	38	459	**0.9191**	**0.9156**	**0.9369**	**0.9261**
Test : tous	M_A	560	431	42	2037	0.8459	0.5651	0.9302	0.7031
Test : tous	M_B	544	65	58	2403	**0.9599**	**0.8933**	0.9037	**0.8984**
Test : tous	$M_{A \cdot B}$	564	260	38	2208	0.9029	0.6845	**0.9369**	0.7910

TABLE 3 – Évaluations sur les corpus Dev et Test (+repl = fichiers qui ont au moins un repliement) ; vp = vrais positifs, fp = faux positifs, fn = faux négatifs, vn = vrais négatifs ; exa = exactitude : proportion de classes (<SP> ou <FUT>) correctement prédites parmi tous les blancs

Résultats Le tableau 3 montre les résultats sur le corpus Dev, d'abord uniquement sur les fichiers qui ont au moins un repliement de paragraphe (+*repl* = « à repli »), puis sur tous. Deux bases de comparaison sont indiquées : laisser les textes tels quels (*replie-aucun*) est très mauvais sur les textes à repli, mais obtient une exactitude (*exa*) élevée sur l'ensemble des textes ; inversement, considérer chaque fin de ligne comme un espace (*replie-tous*) obtient une F-mesure bonne (0,80) sur les textes à repli, mais très mauvaise sur l'ensemble. Le modèle M_A et ses variantes obtiennent une F-mesure bien plus élevée, stable autour de 0,95 pour +*repl* mais variant de 0,40 à 0,51 pour *tous* : sans les fins de lignes non ambiguës (–2), sans les ponctuations fortes (–fort), ou en passant les mots en minuscules (+min), P et F se dégradent. Le modèle M_A est donc bien meilleur sur *tous* sans beaucoup

réduire les résultats sur *+repl*, nous le retenons donc par rapport à ses variantes. Le modèle M_B appris par coapprentissage avec les attributs de longueur de ligne obtient de moins bons résultats sur les textes à repli, mais améliore énormément P et F sur *tous*. Enfin, le modèle combiné $M_{A \cdot B}$ obtient les meilleurs R et F sur *+repl* et se situe entre M_A et M_B sur *tous* tout en y conservant le meilleur rappel. On voit donc que si l'on travaille sur des textes à repli, le modèle $M_{A \cdot B}$ a les meilleurs R et F et parmi les meilleures précisions ; et sur des textes tout venant, les modèles fondés sur la longueur des lignes réduisent énormément les faux-positifs dus au modèle M_A. L'analyse des erreurs de M_A sur Dev montre que la plupart des faux-négatifs sont des lignes se terminant par un point ou un deux-points, souvent rencontrés en fin de ligne : ces non-repliements ne gêneront donc pas une segmentation en phrases. En revanche, les faux positifs (replis intempestifs) rencontrés restent une source de problèmes pour la segmentation en phrases car ces lignes ne se terminent pas par une ponctuation.

Tous ces résultats se retrouvent dans le corpus Test (tableau 3). Sur les textes à repli, le modèle combiné perd trois points de F-mesure en passant de Dev à Test, une perte limitée, en revanche M_B en gagne trois. Sur le corpus global (*tous*), tous les résultats augmentent de Dev à Test du fait de la plus grande proportion de textes à repli dans Test (38 % vs. 18 %).

Par ailleurs, on peut se demander comment se comportent les modèles dans le cas où aucune fin de ligne n'est ambiguë (cas d'un corpus bien segmenté) : c'est la situation des documents présents dans *tous* mais pas dans *+repl*. D'après le tableau 3, sur le Test, en comparant les lignes *+repl* et *tous*, on voit que M_A trouve 69 FP sur *+repl*, puis 431 FP sur *tous*, soit 362 FP de plus pour 1695 VN de plus : il a donc tendance à sur-replier les lignes des textes qui n'ont aucun repliement. En revanche, lorsque l'on passe de *+repl* à *tous*, M_B ajoute très peu de FP (+8) pour 1949 VN de plus : il bloque efficacement les repliements intempestifs, tout en diminuant peu le nombre de repliements corrects (–16) par rapport à M_A dans *+repl*. Enfin, $M_{A \cdot B}$ se situe à une position intermédiaire dans l'ajout de FP (+208) mais augmente un peu le nombre de repliements corrects (+4) par rapport à M_A.

Conclusion Nous avons présenté une méthode entièrement non-supervisée pour la classification des fins de lignes. Le modèle $M_{A \cdot B}$ obtient des F-mesures élevées dans les textes avec repliement de paragraphes, et le modèle M_B détecte bien les textes qui ne sont pas sujet à ce phénomène. Les résultats obtenus sont relativement stables de Dev à Test, montrant la robustesse de la méthode. La combinaison $M_{A \cdot B}$ que nous avons testée se comporte moins bien sur les textes sans repliement : au vu des bons résultats du modèle M_B pour détecter ces textes, son usage comme filtre avant l'application de $M_{A \cdot B}$ est à envisager. Le test d'autres attributs est également à effectuer, comme des catégories morphosyntaxiques, des bigrammes de mots ou formes typographiques. De même, d'autres classifieurs (par exemple, par régression logistique) pourront être étudiés.

Une autre limitation de ce travail est son expérimentation sur un seul corpus : des tests sur d'autres corpus de textes cliniques et sur des corpus d'autres natures (courriers électroniques, journaux ocrisés) sont prévus. De même il sera intéressant de produire automatiquement des corpus avec repliement des paragraphes pour tester nos algorithmes de façon contrôlée.

Remerciements

Cet article a été partiellement réalisé dans le cadre des projets Accordys (ANR-12-COORD-0007-03) et CABeRneT (ANR-13-JS02-0009-01) financés par l'ANR.

Références

AIELLO M., MONZ C., TODORAN L. & WORRING M. (2002). Document understanding for a broad class of documents. *IJDAR*, **5**, 1–16.

BORDES A., USUNIER N. & WESTON J. (2010). Label ranking under ambiguous supervision for learning semantic correspondences. In *27th International Conference on Machine Learning (ICML 2010)*, p. 103–110.

ELKAN C. & NOTO K. (2008). Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, p. 213–220, New York, NY, USA : ACM.

FANG J., TANG Z. & GAO L. (2011). Reflowing-driven paragraph recognition for electronic books in PDF. In G. AGAM & C. VIARD-GAUDIN, Eds., *DRR*, volume 7874 of *SPIE Proceedings*, p. 1–10 : SPIE.

MILLER T. A., FINAN S., DLIGACH D. & SAVOVA G. (2015). Robust sentence segmentation for clinical text. In *Proc AMIA Symp*, p. 112–113, San Francisco, Ca. : AMIA.

NIGAM K., MCCALLUM A. K., THRUN S. & MITCHELL T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning*, **39**(2-3), 103–134.

RADAKOVIC B., GALIC S. & UZELAC A. (2013). *Paragraph Recognition in an Optical Character Recognition (OCR) Process*. United States Patent 8,565,474 B2, US Patent Office.

SMITH N. A. (2011). *Linguistic Structure Prediction*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool.

STUBBS A., KOTFILA C., XU H. & UZUNER O. (2015). Identifying risk factors for heart disease over time : Overview of 2014 i2b2/UTHealth shared task Track 2. *J Biomed Inform*.

WISNIEWSKI G., PÉCHEUX N., GAHBICHE-BRAHAM S. & YVON F. (2014). Cross-lingual part-of-speech tagging through ambiguous learning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1779–1785, Doha, Qatar : Association for Computational Linguistics.

ZWEIGENBAUM P. & GROUIN C. (2014). Reformatting clinical records based on global layout statistics. In O. BODENREIDER, J. L. OLIVEIRA & F. RINALDI, Eds., *Proceedings 6th International Symposium for Semantic Mining in Biomedicine (SMBM 2014)*, p. 53–60, Aveiro : University of Aveiro.