Indexation et appariements de documents cliniques pour le Deft 2019

Davide Buscaldi[1] Dhaou Ghoul[2] Joseph Le Roux[1] Gaël Lejeune[1]
(1) LIPN UMR 7030, Université Paris XIII, 99 avenue Jean-Baptiste Clément, 93430 Villetaneuse, France
(2) STIH EA 4509, Sorbone Université, 1 rue Victor Cousin, 75005 Paris, France
(1) prenom.nom@lipn.univ-paris13.fr, (2)
prenom.nom@sorbonne-universite.fr

RÉSUMÉ

Dans cet article, nous présentons nos méthodes pour les tâches d'indexation et d'appariements du Défi Fouile de Textes (Deft) 2019. Pour la tacĥe d'indexation nous avons testé deux méthodes, une fondée sur l'appariemetn préalable des documents du jeu de tset avec les documents du jeu d'entraînement et une autre méthode fondée sur l'annotation terminologique. Ces méthodes ont malheureusement offert des résultats assez faible. Pour la tâche d'appariement, nous avons dévellopé une méthode sans apprentissage fondée sur des similarités de chaînes de caractères ainsi qu'une méthode exploitant des réseaux siamois. Là encore les résultats ont été plutôt décevant même si la méthode non supervisée atteint un score plutôt honorable pour une méthode non-supervisée : 62% .

ABSTRACT

Indexing and pairing texts of the medical domain

In this paper, we propose different methods for the indexation and pairing tasks of the 2019 edition of the french text mining challenge (Deft). For indexing texts we proposed two different methods, one exploit a pairing method developed for task 2 in order to use keywords found in similar documents.. The second method takes advantage of terminological annotation obtained via the MeSH (Medical Subject Headings) resource. For the pairing task, we also proposed two methods. The first one is unsupervised and relies on similarities at character-level, in the second one we exploited siamese networks. In both tasks our results were quite disappointing except for task 2 where the unsupervised method achieved an honourable 62% score.

MOTS-CLÉS : Appariement, Indexation, Réseaux Siamois, MESH, Modèles en Caractères.

KEYWORDS: Pariring, Indexation, Siamese Networks, MESH, Character-level Models.

1 Introduction

L'édition 2019 du Défi Fouille de Textes (Grabar *et al.*, 2019) proposait trois tâches différentes, exploitant toutes un corpus de textes du domaine médical, en l'espèce des cas cliniques et de discussions à propos de ces cas cliniques. Le développement d'outils de TAL pour ce type particulier de données textuelles est une perspective de recherche très important puisqu'il s'agit d'exploiter des données très riches et très fournies en terminologie afin d'extraire des connaissances et de faciliter

le stockage et l'échange d'information entre les praticiens. Ce sujet de recherche est très actif et intéresse différentes branches du TAL parmi lesquelles la Recherche et l'Extraction d'Information.

Nous avons travaillé sur la tâche d'indexation (Tâche 1) et sur la tâche d'appariement (Tâche 2) et nous avons laissé de côté la partie extraction d'Information (tâche 3). Pour la tâche d'indexation (Section 2), nous avons conçu une méthode endogène fondée sur la réutilisation des résultats d'un système d'appariements de textes développé pour la tâche 2. Nous présentons également une méthode exogène qui exploite la ressource MESH (*Medical Subject Headings*). Pour la tâche d'appariement de cas cliniques et de discussions (Section 3), nous avons proposé deux approches : une approche non-supervisée fondée sur des similarités de chaînes de caractères (Section 3.1) et une approche d'apprentissage profond exploitant des réseaux siamois (Section 3.2).

2 Méthodes pour la tâche d'indexation (Tâche 1)

2.1 Indexation fondée sur des appariements de documents (run1)

Notre hypothèse initiale consistait à factoriser le travail effectué sur la tâche 2. Il s'agissait d'utiliser l'appariement préalable des textes pour associer à un texte nouveau (texte du jeu de test) les mots-clés figurant dans un texte similaire et déjà indexé (présent donc dans le jeu d'entraînement). Notre hypothèse qu'un cas clinique doit être indexé peu ou prou de la même manière que la discussion à laquelle il se rapporte (et inversement pour les discussions). Indexer un cas clinique C (resp. une discussion D) du jeu de test revient à l'apparier avec une discussion D' (resp. un cas clinique C') du jeu d'entraînement et lui assigner les mots-clés correspondants. Étant donné que la tâche consistait à indexer les paires (cas clinique – discussion), nous avons exploité les mots-clés des deux appariements obtenus.

Pour chaque paire à indexer (C_i, D_i), on calcule une paire appariée (D_j, C_j). On obtient donc deux jeux de mots-clés candidats : KW_{Di} et KW_{Cj}. Dans un premier temps on conserve l'intersection : $KW_{Di} \cup KW_{Cj}$, si la taille de l'intersection est inférieure au nombre de mots-clés attendus alors on utilise l'union : $KW_{Di} \cap KW_{Cj}$. Les mots-clés ainsi obtenus sont rangés par ordre d'importance en fonction de leur longueur en caractères.

Si l'idée semblait séduisante, et permettait de factoriser le travail effectué sur la tâche 2, il s'est avéré que son efficacité était significativement inférieure à une simple *baseline* vérifiant la présence des mots-clés de la référence et les rangeant dans l'ordre inverse de leur longueur (run 2 de la tâche 1).

2.2 Indexation fondée sur l'annotation terminologique (run4)

2.2.1 Annotation hybride

Cette méthode, déjà présentée pour DEFT2016 (Buscaldi & Zargayouna, 2016) combine une annotation fondée sur le volet terminologique avec une annotation supervisée pour maximiser le rappel.

L'annotation terminologique utilise un moteur de recherche d'information. D'abord, la ressource sémantique (MeSH français) est indexée en utilisant Whoosh [1]. L'index relie chaque identifiant de

1. https ://whoosh.readthedocs.io/en/latest/

concept à ses représentations lexicales ou étiquettes (principales ou autres) qui constituent le volet terminologique de la ressource. Les fragments textuels qui constituent les étiquettes peuvent ainsi être cherchés par le moteur de recherche.

En phase d'annotation, le texte du document à annoter d est analysé pour extraire les fragments à soumettre au moteur de recherche en tant que requêtes. Annoter un document revient donc à interroger la ressource indexée en utilisant des séquences de mots extraites à partir du texte (dans notre cas, on avance trigramme de mots par trigramme de mots). Le moteur de recherche renvoie une liste ordonnée de résultats contenant les identifiants des concepts avec leurs poids. Le meilleur résultat est ajouté à l'ensemble d'annotations du documents.

Cet algorithme d'annotation est utile quand les étiquettes sont présentes dans le texte même partiellement. Le bruit peut provenir des étiquettes partageant la même racine ainsi que des concepts retournés en premier mais avec un faible score.

2.2.2 Annotation fondée sur l'Information Mutuelle

Quand les concepts n'ont pas de représentations lexicales dans le texte, les algorithmes d'annotation fondés sur la terminologie échouent. Cet algorithme se fonde sur la co-occurrence entre le concept et les termes du documents. L'idée est que si, dans un corpus de textes annotés, un ou plusieurs termes co-occurrent souvent avec le même concept c, alors si on retrouve le même terme ou la même combinaison de termes dans un document sans annotation, on peut l'annoter avec le concept c.

Nous avons d'abord procédé à quelques filtrages. Nous avons éliminé tous les cas pour lesquels les co-occurrences sont dues au pur hasard. Pour cela, dans la phase d'entraînement nous avons pris en compte seulement les concepts qui ont servi pour annoter au moins 3 documents dans le corpus de référence. Au final, nous avons calculé l'information mutuelle uniquement si le terme et le concept co-occurrent dans au moins 2 documents. Par rapport à l'édition 2016, on a réduit les deux paramètres (originellement fixés respectivement 5 et 3) à cause du nombre inférieur de documents dans le corpus d'entrainement.

L'information mutuelle ponctuelle est définie comme :

$$IP(t,c) = \log \frac{p(t,c)}{p(t)p(c)}$$

Où t est un terme (substantif ou adjectif dans notre cas) et c un concept ; $p(t,c)$ est la probabilité conjointe entre t et c, calculée comme $freq(t,c)/N$, $p(t) = freq(t)/N$ et $p(c) = freq(c)/N$, où N est le nombre de documents dans le corpus d'entraînement.

2.2.3 Construction du modèle

Nous construisons une matrice M avec $|C|$ lignes et $|T|$ colonnes, où C est l'ensemble des étiquettes dans le corpus d'entraînement avec fréquence > 5, et T est l'ensemble des mots du dictionnaire (donc tous les substantifs et adjectifs observés dans le corpus) avec fréquence > 3. Chaque élément $M_{i,j}$ est calculé comme suit :

$$M_{i,j} = \begin{cases} IP(t_i, c_j) & if \;\; IP(t_i, c_j) > 0 \\ 0 & else \end{cases}$$

Sur cette matrice, nous appliquons l'analyse sémantique latente, en décomposant M en valeurs singulières (algorithme SVD) $M = U\Sigma V^T$, et en approximant M avec $\hat{M} = U_k\Sigma_k V_k^T$, avec les meilleurs k valeurs singulières sélectionnés ($k = 100$).

Nous espérons améliorer la couverture en appliquant LSA sur la matrice. En effet, LSA permet de trouver des termes fortement associés avec d'autres termes qui sont très caractéristiques pour certaines étiquettes, même si dans le corpus d'entraînement l'étiquette n'apparaît pas avec ce mot. Par exemple, si « Paris » et « français » co-occurrent souvent, mais dans la collection on ne trouve que « France » associé au terme « français », on peut déduire que « France » est une annotation plausible si dans le texte on trouve « Paris », même si on ne les a jamais vus ensemble dans le corpus l'entraînement.

Finalement, nous appliquons un filtre sur la matrice $\hat{M}$ de la façon suivante :

$$\hat{M}^{\sigma}_{i,j} = \begin{cases} \hat{M}_{i,j} & if \;\; \hat{M}_{i,j} \geq \Big(0.5 * \sigma'(\hat{M}_{*,j}) + \mu'(\hat{M}_{*,j})\Big) \\ 0 & else \end{cases}$$

Où $\hat{M}_{*,j}$ est le $j-$ème vecteur colonne de la matrice $\hat{M}$, $\sigma'(\mathbf{x})$ est l'écart type des éléments non nuls du vecteur $\mathbf{x}$ et $\mu'(\mathbf{x})$ est la moyenne des éléments non nuls du vecteur $\mathbf{x}$. La matrice $\hat{M}^{\sigma}$ ainsi obtenue est utilisée comme *modèle* pour l'annotation des documents.

2.2.4 Annotation

Nous associons au document d à annoter un vecteur binaire $\mathbf{b}$ de taille $|T|$ (taille du vocabulaire) où chaque élément $\mathbf{b}_i$ est à 1 si le terme correspondant est dans le texte du document d, 0 autrement. Nous calculons pour chaque étiquette l_i le score $s(l_i) = \mathbf{b} \cdot \hat{M}^{\sigma}_{i,*}$. Le document est annoté finalement avec les 20 étiquettes avec les meilleurs scores > 0. S'il y a moins de 20 étiquettes avec des scores positifs, alors une annotation est générée avec toutes les étiquettes ayant un score positif. Le choix de 20 a été défini empiriquement après les tests sur le corpus de développement.

3 Méthodes pour la tâche d'appariement (Tâche 2)

3.1 T2 : Appariements fondés sur des distributions de chaînes de caractères (run1)

Afin d'apparier les cas cliniques et les discussions nous avons eu l'idée d'exploiter le phénomène de recopie, de réutilisation de séquences langagières. de manière à rendre la méthode aussi endogène et générique que possible, nous nous sommes efforcés de nous affranchir d'une approche purement lexicale. Nous avons fait l'hypothèse que les discussions pouvaient être vues comme des prolongements, des développements des cas cliniques. Ceci nous a amené à nous intéresser à une proposition faite pour le Défi Fouille de Textes en 2011 dans une autre tâche d'appariement visant cette fois à associer des résumés et des articles scientifiques. Dans cet article(Lejeune *et al.*, 2011), nous proposions de considérer que l'association entre un résumé et un article était liée à des correspondances uniques dans le corpus nommées "affinités". Une affinité y était définie comme une sous-chaîne de caractère saillante dans l'article (répétée à des positions clés) et partagée par un seul résumé du corpus.

Une chaîne de caractère était considérée comme saillante dans l'article si elle était présente dans l'introduction ou la conclusion et dans le corps de l'article.

D'un point de vue fonctionnel, l'expérience consistait à considérer que les articles étaient des célibataires et que les résumés étaient des prétendants. Chaque célibataire était présenté tour à tour à tous les prétendants et il s'agissait de calculer les sous-chaînes de caractères communes à un célibataire et à un seul prétendant. Autrement dit, il s'agit de sous-chaînes qui sont hapax dans le sous-corpus des prétendants. On cherche donc toutes les affinités entre le célibataire et chacun des prétendants (critère $Card - Aff$) et on s'intéresse aussi à l'affinité la plus longue (critère $Aff - Max$).

On apparie un prétendant P_i à un célibataire C s'il respecte les deux conditions :

— Maximiser $Card - Aff$: P_i est le prétendant avec lequel C partage le plus d'affinités. Si plusieurs prétendants partagent le même nombre d'affinités alors nous considérons que l'appariement ne peut être validé.

— Maximiser $Aff - Max$: P_i et C partagent la plus longue affinité détectée. De la même manière, l'appariement n'est effectué que si un seul prétendant partage C une affinité de taille N.

En d'autres termes, pour chaque célibataire on apparie le prétendant qui possède à la fois le plus grand nombre d'affinités ainsi que l'affinité la plus longue. Les célibataires sont présentés tout à tour mais il apparaît expérimentalement que l'ordre d'apparition des célibataires a un impact marginal voire nul sur la qualité des appariements.

Nous avons utilisé le code disponible en ligne [2] pour reproduire cette méthode. Par rapport à l'exploitation qui en a été faite en 2011 nous avons opéré deux modifications. La première est que nous n'avons pas utilisé le critère de saillance. En effet, les cas cliniques comme les discussions sont moins richement structurés que des articles scientifiques de sorte que la pertinence de la notion de saillance était moins évidente. D'autre part, ce sont des documents beaucoup plus courts et plus denses que des articles scientifiques de sorte que les phénomènes de répétition sont moins prégnants. Les textes sont très condensés et n'appartiennent pas au genre très normé qui est celui de l'article scientifique. La seconde est que dans le Deft 2011 chaque article correspondait à un seulr ésumé et vice-versa. Une fois un appariement effectué entre un célibataire et un prétendant P_i, ce prétendant était écarté pour la suite du processus. Autrement dit, il s'agissait d'un tirage sans remise. Ici, cette stratégie n'était pas possible pour deux raisons. La première c'est que la même discussion peut correspondre à plusieurs cas cliniques. Ecarter une discussion déjà appariée n'était donc pas pertinent. La deuxième raison est plus prosaïque, la tâche du Deft 2019 était singulièrement plus difficile. Là où sur le Deft 2011 la première phase d'appariement permettait d'obtenir 80% de rappel avec une précision proche de 100%, sur le défi de cette année le rappel était de l'ordre de 20% avec une précision de 80%. Les premiers appariements étant moins nombreux et moins sûrs, le tirage avec remise était donc plus adapté.

3.2 T2 : Réseau siamois et algorithme hongrois (run2 et run3)

Les réseaux siamois (Bromley *et al.*, 1993) sont des architectures neuronales spécialisées pour l'appariement de structures similaires [3]. Ils sont composés de deux sous-réseaux identiques qui permettent de transformer deux vecteurs d'entrée, représentant les documents à évaluer, vers un espace de caractéristiques commun. Une dernière couche prend en entrée ces deux vecteurs de caractéristiques et calcule une énergie, censée représenter la proximité entre les deux structures.

2. `https://github.com/rundimeco/deft2011`
3. Ils ont été développés pour la reconnaissance automatique de signatures de chèques.

Plus concrètement, notre seconde méthode fonctionne comme suit.

1. Les documents sont filtrés via SpaCY pour ne garder que les noms communs [4]. À chaque document, on attribue comme vecteur représentatif la moyenne des vecteurs associés à ses mots après filtrage. Les méthodes par réseaux récurrents n'ont rien apporté.
2. Pour quantifier la proximité entre deux documentsd d_1 et d_2, on donne leur vecteur représentatif comme entrée au sous-réseau du réseau siamois, implémentée comme un perceptron à une couche cachée d'activation par rectificateur linéaire (ReLU). On obtient alors en sortie deux vecteurs v_1 et v_2
3. À la prédiction, on utilise comme énergie simplement la distance euclidienne entre v_1 et v_2.
4. Pour prédire globalement les appariements d'un ensemble de documents, on réduit le problème à trouver le couplage parfait de poids minimal dans un graphe bi-partite complet $G = (V_1 \cup V_2, V_1 \times V_2)$ où les éléments de V_1 et V_2 représentent respectivement les cas cliniques et les discussions et où les poids des arcs (v_1, v_2) sont données par la distance euclidienne. On utilise pour cela l'algorithme de (Munkres, 1957), aussi appelé algorithme *hongrois*, de complexité en temps $O(n^3)$ où n est le nombre de sommets.

Nous suivons la méthode de (Chopra *et al.*, 2005) avec à l'apprentissage une énergie dite *contrastive* qui permet de diminuer l'énergie pour les paires de structures similaires et de l'augmenter pour les paires de structures dissimilaires. Pour tous les triplets (d_1, d_2, l) constitués de documents d_1, d_2 (cas cliniques ou discussions) et d'un label $l \in \{0, 1\}$ indiquant si les documents sont similaires ou non. Deux documents sont similaires (label 0) si : ils sont identiques, ils sont une paire cas clinique/discussion de références (ou contiennent le même texte), sont deux cas cliniques associés à la même discussion. Dans tous les autres cas, le label est 1. On peut ainsi définir la perte constrastive : $L(d_1, d_2, l) = (1 - l)\|v(d_1) - v(d_2)\|_2^2 + l\frac{1}{2}\max(0, m - \|v(d_1) - v(d_2)\|_2)^2$. Cette fonction est parfaitement minimisée, et donc égale à zéro pour toute paire, si la distance euclidienne des paires similaires est nulle et si la distance entre des documents non similaires est au moins m.

4 Résultats et discussion

4.1 Tâche 1 : Indexation

Run	MAP	R-Precision
Run1 (Appariements)	0.126	0.122
Run2 (Baseline)	0.220	0.240
Run4 (MeSH)	0.044	0.034

TABLE 1 – Résultats officiels sur la tâche 1

Nous présentons dans le tableau 1 les résultats que nous avons obtenu sur la tâche d'indexation. Les deux hypothèses que nous avions formulé à savoir l'utilisation de l'appariement de documents (Run1) et l'exploitation du MESH (Run4) se sont avérées inadaptées. En effet, ces méthodes se situent très nettement en retrait de la *baseline* que nous avions développé.

4. Nous avons essayé sans filtrage, ou avec d'autres parties du discours, mais avec de moins bons résultats.

4.2 Tâche 2 : Appariements

Pour la tâche 2, nous avons également obtenu des résultats plutôt décevants (Table 2). Nos méthodes fondées sur les réseaux siamois ont assez vite plafonné, que ce soit pour la variante *average* qui utilise un modèle moyenné sur les différentes itérations et le jeu de développement ou pour la variante *single* qui correspond simplement au modèle obtenant la meilleur évaluation sur le jeu de développement. Notre système fondé sur la similarité de chaînes de caractères s'est avéré plus satisfaisant, avec l'avantage de ne pas subir le phénomène de sur-apprentissage.

Run	Précision
Run1 (Similarité en caractères)	0.617
Run2 (Réseau Siamois *average*)	0.107
Run3 (Réseau Siamois *single*)	0.126

TABLE 2 – Résultats officiels sur la tâche 2

4.3 Discussion

Il est toujours ardu de constater que l'on s'est appuyé sur des hypothèses inadaptées, en particulier lorsque la différence de résultat est aussi grande. Sur la tâche 1, nous sommes bons derniers avec 16 points de pourcentage de moins que la moyenne et 18 de moins que la médiane. C'est peut être sur cette tâche que nous aurions pu améliorer nos résultats. En effet, sur la tâche 2 notre méthode sans apprentissage est encore plus loin de la moyenne (19 points) et de la médiane (25 points). Toutefois, il est difficile d'imaginer comment nous pourrions l'améliorer sans en dénaturer l'esprit.

Références

BROMLEY J., BENTZ J. W., BOTTOU L., GUYON I., LECUN Y., MOORE C., SÄACKINGER E. & SHAH R. (1993). Signature verification using a siamese time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, **7**(04), 669–688.

BUSCALDI D. & ZARGAYOUNA H. (2016). LIPN@DEFT2016 : Annotation de documents en utilisant l'Information Mutuelle. In *DÉfi Fouille de Texte 2016 – DEFT2016*, Paris, France.

CHOPRA S., HADSELL R., LECUN Y. *et al.* (2005). Learning a similarity metric discriminatively, with application to face verification. In *CVPR (1)*, p. 539–546.

GRABAR N., GROUIN C., HAMON T. & CLAVEAU V. (2019). Recherche et extraction d'information dans des cas cliniques. présentation de la campagne d'évaluation deft 2019. In *Actes de DEFT*, Toulouse, France.

LEJEUNE G., BRIXTEL R., GIGUET E. & LUCAS N. (2011). Deft 2011 : appariements de résumés et d'articles scientfiques fondés sur des distributions de chaînes de caractères. In *Proceedings of DEfi Fouille de Texte (DEFT'11)*, p. 53–64.

MUNKRES J. (1957). Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, **5**(1), 32–38.