DeFT 2019 : Auto-encodeurs, *Gradient Boosting* et combinaisons de modèles pour l'identification automatique de mots-clés. Participation de l'équipe TALN du LS2N

Mérième Bouhandi Florian Boudin Ygor Gallina
LS2N, Université de Nantes
prénom.nom@univ-nantes.fr

RÉSUMÉ

Nous présentons dans cet article la participation de l'équipe TALN du LS2N à la tâche d'indexation de cas cliniques (tâche 1). Nous proposons deux systèmes permettant d'identifier, dans la liste de mots-clés fournie, les mots-clés correspondant à un couple cas clinique/discussion, ainsi qu'un classifieur entraîné sur la combinaison des sorties des deux systèmes. Nous présenterons dans le détail les descripteurs utilisés pour représenter les mots-clés ainsi que leur impact sur nos systèmes de classification.

ABSTRACT

Autoencoders, gradient boosting and ensemble systems for automatic keyphrase assignment : The LS2N team participation's in the 2019 edition of DeFT

In this article, we present the participation of the TALN team at the LS2N in the clinical case indexing task (task 1). We propose two systems to identify for each clinical case/discussion pair its corresponding keywords in a given thesaurus, as well as a classifier trained on the the two systems outputs combination. We will present in detail the features used to represent the keywords and their impact on the given task.

MOTS-CLÉS : Identification automatique de mots-clés, autoencoders, gradient boosting, TAL.

KEYWORDS: Automatic keyword assignment, autoencoders, gradient boosting, NLP.

1 Introduction

Dans cet article, nous présentons nos travaux réalisés dans le cadre de l'édition 2019 du Défi Fouille de Texte (DeFT) (Grabar *et al.*, 2019). Portant sur l'analyse de cas cliniques rédigés en français, cette édition se compose de trois tâches autour de la recherche et de l'extraction d'information. Nous avons choisi de participer à la tâche d'indexation des cas cliniques (tâche 1) qui consiste à retrouver les mots-clés les plus pertinents, pour une paire de cas clinique/discussion donnée, dans une liste de mots-clés fournie.

Dans un premier temps (§2), nous détaillons l'ensemble des descripteurs utilisés pour représenter les mots-clés, ainsi que les différents modèles utilisés pour les identifier automatiquement, puis quelques expériences exploratoires. Dans un second temps (§3), nous présentons les résultats obtenus. Finalement (§4), nous concluons et discutons des perspectives de travaux futurs.

2 Approches

Le corpus mis à disposition cette année dans le cadre de la compétition DeFT est composé de cas cliniques associés à une discussion et des mots-clés (Grabar *et al.*, 2018). Les documents proposés sont liés à différentes spécialités médicales (cardiologie, urologie, oncologie, obstétrique, pneumologie, gastro-entérologie, etc.) et ont été tirés de publications francophones (France, Belgique, Suisse, Canada, pays africains, etc.).

Un cas clinique est tiré du dossier médical d'un patient et correspond à la description des symptômes, des diagnostics et propositions thérapeutiques d'un médecin. Les discussions médicales couvrent un sujet particulier de façon plus complète et sont généralement plus longues (cf. Figure 1.

Cas
Un échantillon de sérum a été prélevé (puis congelé) 30 minutes après son admission pour une demande de recherche d'amphétamines et d'acide gamma hydroxybutyrique, et une mèche de cheveux ...

Discussion
Par ailleurs, à la suite de prises croissantes de GHB sur une période de 28 jours (30, 45, 45 et 60 mg/kg de poids corporel) chez un volontaire, l'analyse des cheveux prélevés a présenté les résultats suivants : les segments témoins présentaient des concentrations moyennes de 0,62 ng/mg et ...

Mots-clés de réference :
analyse de cheveux ; acide gamma hydroxybutyrique ; intoxication sanguine.

FIGURE 1 – Exemple de couple cas clinique / discussion et mots-clés associés.

Les mots-clés associés sont variés. On y trouve des termes simples (*"cancer"*, *"prostate"*) ou des termes complexes, aussi bien morphologiquement complexes (*"urétéroscopie"*) que polylexicaux (*"syndrome de la fente médiane"*).

Les mots-clés ne sont pas toujours composés que de caractères alphanumériques. On y retrouve, par exemple, des mots contenants des caractères spéciaux (*"fighter®"*) ou plusieurs termes séparés par des virgules *"atropine, scopolamine, hyoscyamine, hallucinogène"*. Certains mots-clés sont rpésents sous différentes variantes morphologiques d'un mots-clés peuvent être utilisés (*"analyse de cheveux"*, *"analyse des cheveux"*). De plus, un mot-clé n'est pas toujours présent dans les deux documents cas/discussion et il n'est parfois présent dans aucun des deux.

2.1 Pré-traitements effectués

Certains mots étant coupés en deux, *"lésion"* devenant *"lési on"* ou *"chimiothérapie"* devenant *"chi miothérapie"* par exemple, nous avons appliqué une normalisation du corpus en utilisant le vocabulaire présent dans le corpus ainsi que les mots de la liste de mots-clés fournie : pour deux mots qui se suivent dans le corpus, si la concaténation des deux est présente dans le vocabulaire issu du corpus ou la liste de mots-clés fournie, alors ils sont fusionnés. De plus, nous avons observé que ces césures apparaissent généralement lorsque le premier mot (*"lési"* dans l'exemple) se termine par une des

lettres de la suite *"ïïǐkltv"*. Ce problème est probablement dû à une mauvaise gestion des césures dans les documents originaux.

Aussi, nous avons fait le choix de ne pas pré-traiter le texte, en lui appliquant des normalisations telles la suppression des mots vides ou une lemmatisation, pour éviter de perdre de l'information.

2.2 Descripteurs

Nous avons essayé de caractériser les particularités linguistiques et stylistiques qui caractérisent les mots-clés au moyen des descripteurs suivants :

— **Descripteurs fréquentiels**

1. Fréquence du terme (TF) : fréquence d'un terme dans la paire de documents
2. Fréquence inverse de document (IDF) : logarithme de l'inverse de la proportion de documents du corpus qui contiennent le terme
3. $TF - IDF$

— **Descripteurs positionnels**

1. Position de la première occurrence (Occ_{first}) : position (en nombre de caractères) de la première occurrence du mot dans la paire de documents. Si le mot est absent de la paire de documents, $Occ_{first} = -1$
2. Mesure d'étalement ($Spread$) : Nombre de caractères entre la première et la dernière occurrence dans la paire de documents.

— **Descripteurs statistiques**

1. Présence dans les deux documents (P_{both}) : vrai si le mot est présent dans les deux documents de la paire cas/discussion, faux sinon
2. Mesure du lien du mot et son contexte (W_{rel}) : mesure de la singularité du mot dans le corpus. Plus un mot candidat co-occure avec des termes différents, plus ce mot candidat est susceptible d'être peu important dans le document. On le calcule comme suit :

$$W_{rel} = (0.5 + ((WL \cdot \frac{TF(w)}{MaxTF}) + PL)) + (0.5 + ((WR \cdot \frac{TF(w)}{MaxTF}) + PR))$$

   avec $TF(w)$ la fréquence du terme dans la paire de document, $MaxTF$ la fréquence du terme le plus fréquent, WL [ou WR] le rapport entre le nombre de mots différents qui co-occurrent avec le mot candidat à gauche (ou à droite) et le nombre de mots total qui co-occurrent avec celui-ci et PL (ou PR) mesure le rapport entre le nombre de mots différents qui co-occurrent avec le terme candidat à gauche (ou à droite) et le $MaxTF$.
3. Occurrence de sous-parties ($Occ_{subparts}$) : somme de la fréquence de chaque mot composant le terme
4. Occurrence de variantes ($Occ_{variants}$) : compte des variantes du terme dans la paire de documents. Pour *"analyse des cheveux"*, nous prenons aussi en compte *"analyses des cheveux"* et *"analyse de cheveux"*, par exemple.
5. Longueur de la paire de documents normalisée (D_{len}) : somme de la longueur de la paire de document

Descripteur	**Références**
Fréquence du terme (TF)	(Jones, 2004)
Fréquence inverse de document (IDF)	(Jones, 2004)
$TF - IDF$	(Jones, 2004)
Position de la première occurrence (Occ_{first})	(Aquino *et al.*, 2014)
Mesure d'étalement ($Spread$)	(Hasan & Ng, 2014)
Présence dans les deux documents (P_{both})	-
Mesure du lien du mot et son contexte (W_{rel})	(Campos *et al.*, 2018)
Occurrence de sous-parties ($Occ_{subparts}$)	-
Occurrence de variantes ($Occ_{variants}$)	(Claveau & Raymond, 2012)
Longueur du document normalisée (D_{len})	-
Z-Score du mot (W_z)	(Aquino *et al.*, 2014)

TABLE 1 – Récapitulatif des descripteurs utilisés pour caractériser les mots-clés.

6. Z-Score du mot (W_z) : fréquence normalisée du terme en utilisant sa fréquence moyenne dans le corpus et son écart-type

Ainsi, pour chaque paire de cas/discussion, nous représentons chaque mot de la liste de mots-clés fournie en utilisant ces descripteurs. Ils sont normalisés en utilisant le *StandardScaler* de la librarie python *scikit-learn*, qui transforme une valeur x en une valeur $z = \frac{x-u}{s}$, avec u la moyenne des x et s son écart-type.

Ces descripteurs sont plus ou moins importants pour caractériser les mots-clés (Figure 2). Après plusieurs essais, nous avons empiriquement déterminé que seuls les éléments qui corrèlent à plus de 0.125 avec notre classe cible sont conservés.

Ces descripteurs (Table 2) seront utilisés en entrée de tous nos systèmes sauf celui présenté en §2.3.

Descripteur	**Corrélation avec la cible**
$TF - IDF$	0,44
Fréquence du terme (TF)	0,36
Fréquence inverse de document (IDF)	0,32
Présence dans les deux documents (P_{both})	0,28
Mesure d'étalement ($Spread$)	0,21
Mesure du lien du mot et son contexte (W_{rel})	0,21
Position de la première occurrence (Occ_{first})	0,13

TABLE 2 – Récapitulatif des descripteurs utilisés pour caractériser les mots-clés.

2.3 Vecteurs multi-hot & TF-IDF

Ce système est basé sur la comparaison entre la paire cas/clinique et chaque mot-clés fournis. Les paires cas/discussions sont représentés par leur vecteur Tf-Idf et les mots-clés par un vecteur multi-hot dans le même espace vectoriel. La mesure cosinus est alors utilisée pour calculer la similarité ,et ainsi la pertinence, du mot-clé par rapport à la paire de document.

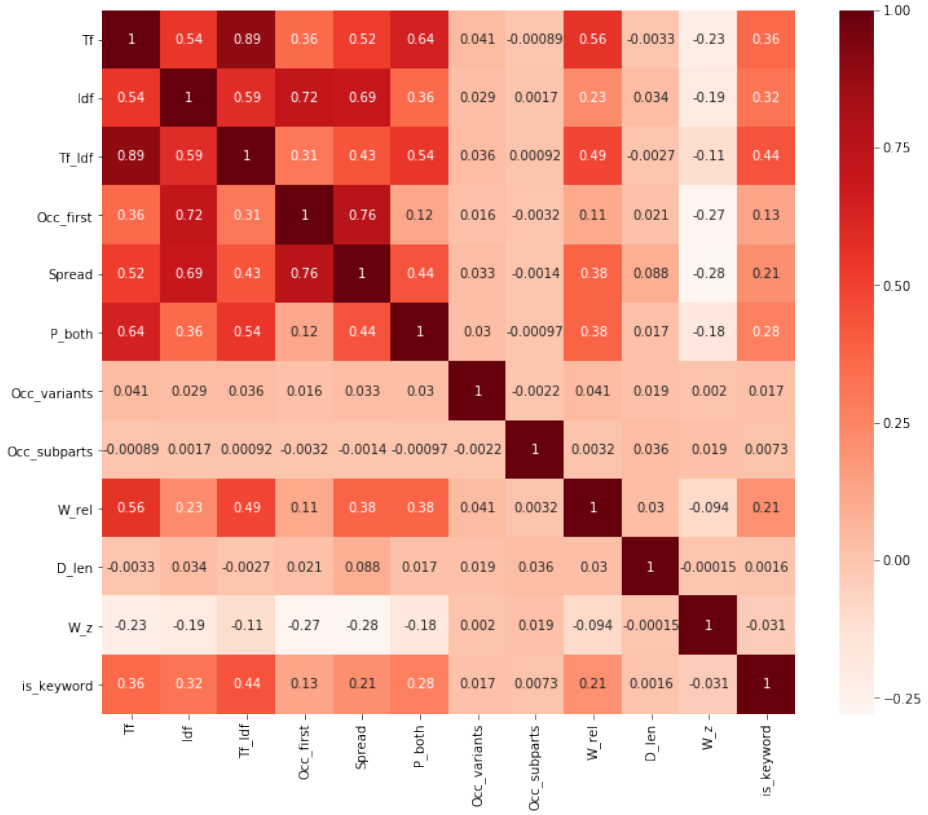

FIGURE 2 – Corrélation des différents descripteurs et de la cible (*est/n'est pas un mot-clé*).

Nous montrons dans la Table 3 comment sont construits ces vecteurs pour les mots-clés et pour les paires cas/discussions.

	analyse des cheveux	*acide gamma hydroxybutyrique*	*intoxication sanguine*	*cas + discussion*
	Compte			TFIDF
acide	0	1	0	0,0147
analyse	1	0	0	0,0147
cheveux	1	0	0	0,0294
gamma	0	1	0	0,0147
hydroxybutyrique	0	1	0	0,0147
intoxication	0	0	1	0
sanguine	0	0	1	0

TABLE 3 – Représentations vectorielles des mots de la liste de mots-clés fournie ainsi que des paires cas/discussions.

2.4 Gradient Boosting

Le *boosting* est une méthode d'apprentissage ensembliste qui consiste apprendre itérativement plusieurs classifieurs dont les poids des individus sont corrigés au fur et à mesure pour mieux prédire les valeurs difficiles. Les classifieurs sont alors pondérés selon leurs performances et agrégés itérativement.

Nous utilisons le modèle *XGBoost* (*eXtreme Gradient Boosting*) qui est une implémentation très populaire, notamment lors des compétitions *Kaggle*, du modèle *Gradient Boosting*. La principale différence entre *boosting* classique (*AdaBoost*) et le *Gradient Boosting* se trouve au niveau de la fonction de coût : ce dernier utilise des gradients dans sa fonction de coût alors que le premier se contente d'appliquer des poids plus importants aux individus mal classifiés.

Nous prenons en entrée les descripteurs présentés dans la Table 1, labellisés. Le classifieur nous renvoie en sortie la probabilité que le mot soit effectivement mot-clé de la paire de documents donnée. Les paramètres utilisés sont présentés dans la Table 4.

Paramètres	**Valeurs**
alpha	10
colsample_bytree	0.3
early_stopping_rounds	10
learning_rate	0.1
max_depth	5
metrics	rmse
nfold	3
num_boost_round	50
objective	reg :linear
seed	123

TABLE 4 – Paramètres utilisés pour l'entraînement du classifieur XGBoost.

2.5 Auto-encodeur

La deuxième stratégie mise en place est l'utilisation d'un *auto-associative neural network* ou auto-encodeur. Plutôt que de classifier, l'objectif de l'auto-encodeur est de reconstruire en sortie l'ensemble de données d'entrée.

Ainsi, étant donné un ensemble d'entrée X et un ensemble de sortie X', on peut mesurer l'erreur de reconstruction commise par l'auto-encodeur en calculant la somme des différences au carré :

$$e(X) = \sum_{i=1}^{n} (X_i - X'_i)^2$$

L'intuition derrière l'utilisation de cette méthode pour l'identification automatique des mots-clés est la suivante : nous n'avons pas suffisamment de données pour entraîner un système à associer à chaque document l'ensemble de mots-clés correspondants. De plus, les classes (*est/n'est pas un mot-clé*) sont fortement non balancées. En effet, pour chaque paire de documents, sur les 1311 mots

de la liste de mots-clés fournie, seuls 4 mots sont en moyenne associés à chaque document et sur 38k combinaisons liste mots-clés/documents, seules 765 correspondent à des mots-clés. Il est donc plus facile d'apprendre à reconnaître un mot qui n'est pas un mot-clé, plutôt qu'un terme l'étant. En traitant cette tâche comme une tâche de détection d'évènements rares, l'erreur de reconstruction sur les mots-clés sera particulièrement élevée par rapport à celle de mots qui ne le sont pas.

L'erreur de reconstruction est alors utilisée comme mesure pour classer les mots-clés par ordre de pertinence : plus cette erreur est élevée, plus important est le mot-clé. Nous récupérons alors les N premiers pour chaque paire de documents, avec N le nombre de mots-clés attendu.

Les paramètres utilisés sont récapitulés dans la Table 5.

Paramètres	**Valeurs**
batch_size	128
epoch	100
loss	mean squared error
optimizer	adam
learning_rate	0.001
encoder (input) $\rightarrow$ *activation*	tanh
encoder (hidden) $\rightarrow$ *activation*	relu
decoder (hidden) $\rightarrow$ *activation*	tanh
decoder (output) $\rightarrow$ *activation*	relu
EarlyStopping $\rightarrow$ *monitor*	val_loss
EarlyStopping $\rightarrow$ *patience*	10

TABLE 5 – Paramètres utilisés pour l'entraînement de l'auto-encodeur (utilisation de la librairie *Keras*).

2.6 Combinaison des systèmes

Dans l'optique de tirer partie des résultats obtenus avec les deux modèles, nous proposons deux approches pour les combiner.

2.6.1 Combinaison des scores

La première approche se fonde sur le calcul d'un score moyen pour un mot m de vecteur x en combinant la sortie $p(c|x)$ de XGBoost et l'erreur de reconstruction $e(x)$ de l'auto-encodeur au moyen d'une moyenne harmonique, pour éviter de sur-estimer le score si l'un des deux est significativement plus élevé que l'autre :

$$\overline{H}(x,c) = \frac{2 \cdot p(c|x) \cdot e(x)}{p(c|x) + e(x)}$$

2.6.2 Stacking

La seconde approche – *stacking* – est une technique d'apprentissage ensembliste permettant de combiner plusieurs modèles de classification entraînés chacun sur l'ensemble des données d'apprentissage. Les sorties de ces modèles sont alors fournies en données d'entrée d'un méta-classificateur pour prédire un résultat final. Nous utilisons pour cela XGBoost qui prend en entrée la sortie $p(c|x)$ du précédent XGBoost sur l'intégralité des données ainsi l'erreur de reconstruction $e(x)$ de l'auto-encodeur.

3 Résultats

Notre travail a donné lieu à de nombreuses expérimentations, notamment plusieurs combinaisons et variantes de nos systèmes. Nous présentons ici les systèmes les plus performants. Les mesures d'évaluation utilisées sont la MAP et la P@N (précision rang N, avec N le nombre de mots-clés attendus).

3.1 Résultats tâche 1

Nous reportons Table 6 les résultats obtenus pour nos différents systèmes sur le corpus d'apprentissage et Table 7 ceux obtenus pour les 3 runs sur le corpus de test. Pour la classification avec XGBoost, nous avons choisi une validation croisée en 3 *folds*. Pour les autres, nous avons découpés le corpus d'apprentissage en 80% pour le *train* et 20% pour le *test*.

Nous remarquons que la majorité des systèmes peinent à atteindre les 50% de MAP et de précision. Le modèle basé sur XGBoost donne les meilleurs résultats seul et gagne 3 points en étant associé avec l'auto-encodeur. Seul, ce dernier est assez médiocre, ce qui est lié à la petite taille du corpus d'apprentissage (Figure 3).

Modèle	**MAP**	**P@N**
TF-IDF	16,1	20,5
Auto-Encodeur (AE)	22,6	29,2
XGBoost (XGB)	**43,4**	**45,1**
TF-IDF + AE (moyenne harmonique)	**30,9**	**35,3**
TF-IDF + AE (XGB sur sorties)	27,2	**35,1**
AE + XGB (moyenne harmonique)	**45,3**	**48,7**
AE + XGB (XGB sur sorties)	44,8	**48,4**

TABLE 6 – Résultats sur le corpus d'apprentissage.

4 Conclusion

Dans cet article, nous avons présentés nos travaux réalisés dans le cadre de l'édition 2019 du Défi Fouille de Texte (DeFT) pour la tâche 1, qui consistait à retrouver les mots-clés les plus pertinents,

Modèle	Run	MAP	P@N
Auto-Encodeur (AE)	1	23,2	28,3
XGBoost (XGB)	2	**40,4**	46,0
AE + XGB (moyenne harmonique)	3	**40,4**	**46,7**

TABLE 7 – Résultats sur le corpus de test.

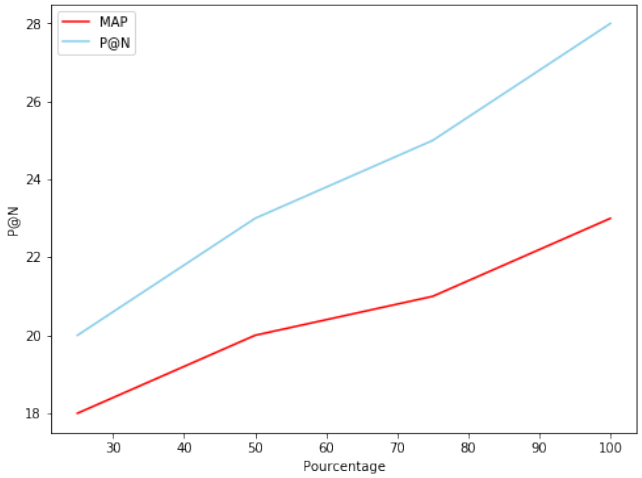

FIGURE 3 – MAP et Précision rang N pour différentes tailles de corpus d'apprentissage pour l'auto-encodeur)

pour une paire de cas clinique/discussion donnée, dans une liste de mots-clés fournie.

Nous avons obtenu des résultats moyens en utilisant des méthodes classiques telles les méthodes à base de *boosting* et d'arbres de décision, ce qui nous laisse une nette marge de progression. Les méthodes neuronales ont elles démontrés de moins bons résultats, en partie dus à la taille du corpus qui ne permettaient pas un apprentissage optimal. Nous restons cependant, avec notre meilleur système, dans la moyenne des résultats, puisque sur 6 participants, nous nous situons au dessus de la moyenne (38,5%) et de la médiane (40,1%) avec notre système *AE-XGBoost* (40,4%).

Finalement, nous avons montré qu'une combinaison de systèmes n'apportait finalement qu'une très légère amélioration.

Références

AQUINO G., HASPERUÉ W. & LANZARINI L. (2014). Keyword extraction using auto-associative neural networks.

CAMPOS R., MANGARAVITE V., PASQUALI A., JORGE A. M., NUNES C. & JATOWT A. (2018). A text feature based automatic keyword extraction method for single documents. In G. PASI, B. PIWOWARSKI, L. AZZOPARDI & A. HANBURY, Eds., *Advances in Information Retrieval*, p. 684–691, Cham : Springer International Publishing.

CLAVEAU V. & RAYMOND C. (2012). Participation de l'IRISA à DeFT2012 : recherche d'information et apprentissage pour la génération de mots-clés. In *JEP-TALN-RECITAL 2012, Workshop*

DEFT 2012 : DÉfi Fouille de Textes (DEFT 2012 Workshop : Text Mining Challenge), p. 49–60, Grenoble, France, France : ATALA/AFCP.

GRABAR N., CLAVEAU V. & DALLOUX C. (2018). CAS : French Corpus with Clinical Cases. In *LOUHI 2018 - The Ninth International Workshop on Health Text Mining and Information Analysis*, Ninth International Workshop on Health Text Mining and Information Analysis (LOUHI) Proceedings of the Workshop, p. 1–7, Bruxelles, France.

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 TALN 2019 (Traitement automatique des langues naturelles)*, ateliers DEFT 2018.

HASAN K. S. & NG V. (2014). Automatic keyphrase extraction : A survey of the state of the art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 1262–1273, Baltimore, Maryland : Association for Computational Linguistics.

JONES K. S. (2004). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, **60**(5), 493–502.