Participation d'EDF R&D à DEFT 2019 : des vecteurs et des règles !

Philippe SUIGNARD[1] Meryl BOTHUA[1] Alexandra BENAMAR[1]
(1) EDF R&D, 7 Boulevard Gaspard Monge, 91120 Palaiseau
prenom.nom@edf.fr

RÉSUMÉ

Ce papier décrit la participation d'EDF R&D à la campagne d'évaluation DEFT 2019. Notre équipe a participé aux trois tâchés proposées : Indexation de cas cliniques (Tâche T1) ; Détection de similarité entre des cas cliniques et des discussions (Tâche T2) ; Extraction d'information dans des cas cliniques (Tâche 3). Nous avons utilisé des méthodes symboliques et/ou numériques en fonction de ces tâches. Aucune donnée supplémentaire, autre que les données d'apprentissage, n'a été utilisée. Notre équipe obtient des résultats satisfaisants sur l'ensemble des taches et se classe première sur la tache 2. Les méthodes proposées sont facilement transposables à d'autres tâches d'indexation et de détection de similarité qui peuvent intéresser plusieurs entités du groupe EDF.

ABSTRACT

EDF R&D submission to DEFT 2019

This paper describes the participation of EDF R&D at DEFT 2019 evaluation campaign. Our team participated in the tree proposed tasks : Clinical Cases Indexation (Task T1) ; Semantic Similarity Detection between Cases and Discussions (Task T2) ; Information Extraction in Clinical Cases (Task T3). We used both symbolic and numerical methods. No additional data other than the training data was used. Our team achieves satisfactory results on all tasks and ranks first on task 2. The proposed methods are easily transferable to other Indexation or Semantic Similarity Detection tasks and may interest several entities of the EDF group.

MOTS-CLÉS : données cliniques, indexation, détection de similarité sémantique, Word2Vec, détection de multimots, extraction d'information, clustering.

KEYWORDS: clinical data, indexation, semantic similarity detection, information extraction, Word2Vec, multiwords detection, clustering.

1 Introduction

Plusieurs éléments nous ont motivés à participer à l'édition 2019 du défi DEFT (Grabar *et al.*, 2019) :

— S'évaluer sur des données différentes comme les données médicales (Grabar *et al.*, 2018)
— Participer à DEFT était l'occasion de travailler sur plusieurs méthodes de détection de similarité dont les résultats contribueront directement à EDF Commerce et à d'autres entités du groupe EDF.

2 Tâche 1 : indexation, calcul de mots-clés

La tâche 1 est une tâche d'indexation, qui consiste à trouver les mots-clés associés à un document, ici le couple cas/discussion dans le domaine médical. La liste des mots clés est fournie, elle est composée de mots simples, de mots composés et de multi mots.

2.1 Méthode 1 : *embeddings* + similarités

La méthode proposée est basée sur les embeddings de mots ou vecteurs-mots. Après avoir entrainé un modèle Word2Vec sur le corpus d'apprentissage, la méthode consiste à calculer une représentation vectorielle des mots-clés potentiels ainsi que des documents puis à retenir les N mots-clés les plus similaires au document. Liste des étapes :

1. **Calcul des embeddings sur le corpus** : La méthode commence par entraîner un modèle Word2Vec ((Mikolov *et al.*, 2013)) sur les données. Les différents paramètres de W2V ont été choisis de manière à optimiser les résultats de la tâche 2.

2. **Calcul des embeddings pour les mots clés** : Quand le mot-clé est constitué d'un seul mot comme « vessie » ou de plusieurs mots, mais dont un seul fait partie du modèle W2V comme « fonction cognitive », l'embedding du mot-clé sera égal à l'embedding du mot. Quand le mot-clé est constitué de plusieurs mots, l'embedding du mot-clé est égal à la moyenne des embeddings de chaque mot. Deux variantes ont été considérées, soit en pondérant les mots clés par l'inverse de la fréquence (IDF) (Sparck Jones, 1972), soit sans pondération.

3. **Calcul des embeddings pour les documents** : Pour calculer l'embedding d'un document, on commence par supprimer les mots appartenant à une « stop liste » constitué d'environ 1000 mots. Puis on moyenne les différents vecteurs-mots en les pondérant par l'inverse de la fréquence (IDF).

4. **Constitution d'une liste de mots-clés potentiels** : Sur le corpus d'apprentissage, on observe que certains documents ont pour mots-clés, des mots qui ne sont pas présents dans le document. Mais comme ces situations sont loin d'être majoritaires, pour éliminer le risque de bruit, on constitue une liste de mots-clés potentiels en gardant uniquement les mots clés constitués des mots présents dans le document lui-même.

5. **Similarité entre les mots-clés potentiels et les documents** : Pour chaque mot-clé potentiel, on calcule sa similarité avec le document. Puis on conserve les N mots clés ayant la similarité la plus élevée avec ce document. En comparant les mots-clés ainsi obtenus sur le corpus d'apprentissage à ceux attendus, on s'aperçoit d'une sur-représentation des mot-clés composés de 2 mots, 3 mots, etc. par rapport aux mots clés composés d'un seul mot. On va donc pondérer le calcul de similarité en fonction du nombre de mots du mot-clé :

$$sim = coef * \cos(Vec(Mot - cle), Vec(Document)) \quad (1)$$

avec $coef = 1$ si $n = 1$, $coef = 0,9$ si $n = 2$, $coef = 0,8$ si $n = 3$ et $coef = 0,7$ si $n >= 4$, et n le nombre de mots du mot-clé.

2.2 Méthode 2 : détection de multi-mots et exploitation des étiquettes morpho-syntaxiques

Les mots-clés à retrouver dans ce corpus sont des mots uniques (mots simples, par exemple « tumeur » et composés, par exemple « Wolff-Parkinson ») et des expressions multi-mots (comme, par exemple, « tumeur rénale ») . Après une analyse fréquentielle sur les mots-clés à retrouver dans le corpus, nous remarquons qu'il y a à peu près autant de mots simples que d'expressions multi-mots (Cf. Table 1).

	mots simples	multi-mots	total
sans doublons	358	329	687
total	684	445	1.129

TABLE 1 – Occurrence des mots-clés à retrouver dans le corpus d'entraînement

2.2.1 Détection des mots simples

La fréquence des mots simples dans le corpus est un indicateur pouvant être déterminant dans le filtrage des mots obtenus après indexation (Cf. Table 2). Les mots-clés obtenus (simples et composés) apparaissent légèrement plus dans la discussion que dans le cas associé, mais cela ne semble pas significatif. Il y a 46 mots-clés qui n'apparaissent pas dans le corpus, ce qui complexifie la tâche d'indexation : nous ne pourrons donc pas nous baser uniquement sur le contenu du texte. Des pré-traitements ont été réalisés pour le comptage des mots-clés : minusculisation et suppression des accents.

Mots simples			Total
Occurence cas	Occurence discussion	Déduction annotateur	-
247	277	46	358

TABLE 2 – Occurrence des mots à trouver par type de mots (simples ou multi-mots) dans le corpus

Afin de comprendre au mieux le choix d'attribution des mots-clés que nous devons retrouver dans le corpus, nous avons utilisé TreeTagger [1], qui permet d'étiqueter les mots sur le plan morpho-syntaxique (Cf. Table 3). Nous avons obtenu les résultats suivants : les mots-clés simples sont principalement des noms (« NOM ») et des adjectifs (« ADJ »). Cette information nous permettra d'effectuer un filtre sur notre corpus, et de supprimer 248.161 mots, soit 65% de nos données textuelles.

	Catégorie morpho-syntaxique			Total
	NOM	ADJ	Autres	-
mots-clés	293	19	46	358
corpus	91.152	45.412	248.161	384.725

TABLE 3 – Catégories morpho-syntaxique des mots simples à retrouver dans le corpus (obtenues avec TreeTagger)

1. http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/

Afin de réduire le bruit, nous avons ensuite calculé une pondération des mots obtenus en comparant la fréquence d'un mot-clé w_i dans un document d_j, à sa fréquence dans tout l'ensemble des documents d. La fonction de pondération f est définie par :

$$f(w_i) = \frac{freq(w_i, d_j)}{freq(w_i, d)} \tag{2}$$

On note $freq(w_i, d_j)$ la fréquence du mot dans le document et $freq(w_i, d)$ sa fréquence dans le corpus (équation 3).

$$freq(w_i|d_j) = \frac{count(w_i, d_j)}{count(w, d_j)}; freq(w_i|d) = \frac{count(w_i, d)}{count(w, d)} \tag{3}$$

L'objectif est alors de trier les mots-clés obtenus afin de maximiser f, ce qui revient à classer les mots selon leur importance dans le document par rapport à leur importance dans tout le corpus. Après quelques pré-traitements (minusculisation et suppression des accents), utilisation de la morpho-syntaxe (« NOM » et « ADJ ») et de la liste de référence fournie, nous obtenons les résultats suivants :

	Référence			
	Présents		Absents	Total
	Présents dans corpus	Déduction annotateur		
Trouvés	534	0	4.402	4.936
Non trouvés	104	46	-	150
Total	638	46	4.402	5.086

TABLE 4 – Matrice de confusion des mots simples retrouvés dans le corpus après filtrage morpho-syntaxique (« NOM » et « ADJ ») et filtrage sur la liste de mots-clés.

Il est important de noter que les 46 déductions d'annotateurs ne peuvent pas être retrouvés avec cette méthode, nous soustrayons donc que le nombre de mots-clés simples à obtenir au nombre de déductions annotateurs.

Afin de réduire le bruit, nous avons fixé un seuil de $w_i > 0,05$, ce qui a permis de réduire la liste de mots-clés trouvés (Cf. Table 5).

	Référence			
	Présents		Absents	Total
	Présents dans corpus	Déduction annotateur		
Trouvés	520	0	3.836	4.356
Non trouvés	118	46	-	164
Total	638	46	3.836	4.520

TABLE 5 – Matrice de confusion des mots simples retrouvés dans le corpus après filtrage morpho-syntaxique (« NOM » et « ADJ »), filtrage sur la liste de mots-clés et fixation d'un seuil.

Pour mesurer la qualité des résultats obtenus, nous avons utilisé trois indicateurs : précision, rappel et f-mesure (Cf. Table 6). Le seuil utilisé semble avoir légèrement amélioré les résultats obtenus en précision et en f-mesure, ce qui s'explique par la réduction du bruit généré. Cependant, on remarque

que le rappel diminue, ce qui signifie que nous avons supprimé de la liste des mots-clés qui étaient pertinents pour le cas clinique étudié.

	Précision	Rappel	F-mesure
Table 4	0,1082	0,7807	0,1900
Table 5	0,1194	0,7602	0,2063

TABLE 6 – Mesures de qualité calculées sur les matrices de confusion dans les tableaux 4 et 5

2.2.2 Détection des multi-mots

Comme pour la détection de mots simples, il existe des expressions multi-mots qui n'apparaissent pas dans le corpus : les déductions annotateurs. Une expression est une détection annotateur si elle n'apparaît pas telle quelle dans le corpus (après avoir réalisé un pré-traitement sur les données, identique à celui effectué précédemment). Il y a donc 101 expressions multi-mots qui sont des déductions annotateurs, parmi les 329 expressions à trouver, soit près d'un tiers des expressions (Cf. Table 7). Cela nous indique que la détection des expressions strictes dans les données ne sera pas suffisante pour obtenir de bonnes performances d'extraction de mots-clés.

Mots simples			Total
Occurence cas	Occurence discussion	Déduction annotateur	-
247	277	101	329

TABLE 7 – Occurrence des expressions multi-mots à trouver dans le corpus

Dans un premier temps, nous avons extrait les expressions multi-mots qui apparaissent dans les cas cliniques et discussions associés à chaque patient ; cette méthode basique est celle que l'on a nommée *hard matching*. Les résultats obtenus montrent que 20% des multi-mots obtenus sont pertinents pour le cas clinique associé. Parmi les multi-mots de référence, il y en a moins d'un tiers que l'on ne retrouve pas dans le corpus. Cela s'explique par le fait que 97% d'entre-eux n'apparaissent pas tels quels dans le corpus. Pour cette méthode, nous obtenons un faible rappel et une faible précision (cf. Table 8).

Afin d'améliorer le système existant et de s'intéresser aux expressions multi-mots n'apparaissant pas dans le corpus, nous avons testé une méthode que nous avons appelé *fuzzy matching*. Cette méthode consiste à regarder, pour chaque expression multi-mots, si tous les mots qui la composent sont présents dans le corpus. Par exemple, si le multi-mot est « cancer du sein » et que « cancer » et « sein » sont dans le corpus, l'expression est retenue pour le cas clinique. Cette méthode permet de retrouver 78% des expressions multi-mots à trouver, soit 25% de plus qu'avec la méthode *hard matching*. Néanmoins, nous récupérons 3.405 expressions en trop avec cette méthode.

Nous avons ensuite cherché à attribuer un rang de pertinence des mots clés trouvés. Pour ce faire, nous avons traité séparément les expressions obtenues avec un *hard matching* et celles obtenues en *fuzzy matching*. Les expressions apparaissant telles quelles dans le texte seront en tête de liste d'expressions multi-mots. Ensuite, pour ordonner les autres mots-clés, nous leur avons donné la

	Précision	Rappel	F-mesure
Hard Matching	0,2210	0,5393	0,2536
Fuzzy Matching	0,0927	0,7820	0,1658

TABLE 8 – Mesures de qualité calculées avec la méthode *hard matching* et la méthode *fuzzy matching*

somme des occurences de chaque mot w_i qui les composent dans un document d_j. Une liste de mots vides a été construite afin de ne pas comptabiliser les mots de type préposition :

$$weight(w_{1..n}|d_j) = \frac{\sum_{i=1}^{n} count(w_i, d_j)}{n} \quad (4)$$

2.2.3 Ranking

Nous avons cherché à détecter séparément les mots-clés simples et les expressions multi-mots et nous les avons ordonnés différemment. L'objectif est maintenant de mélanger les deux sorties afin d'ordonner les mots simples et les expressions multi-mots ensemble. Pour cela, une règle évidente s'est imposée à nous : les expressions multi-mots qui apparaissent dans le texte tels quels (avec la méthode *hard matching*) sont placés en début de sortie. Il nous reste ensuite à ordonner les mots simples et les expressions multi-mots qui apparaissent uniquement avec la méthode *fuzzy matching*. Pour cela, nous avons utilisé une règle 50/50, qui consiste à récupérer un mot-clé simple et une expression multi-mot, jusqu'à ce qu'il n'y ait plus de mots-clés dans notre liste.

2.3 Résultats

Les meilleurs scores ont été obtenus avec la méthode 1, soit l'utilisation d'*embeddings* suivi d'un calcul de similarité. Malheureusement, nous sommes en dessous de la moyenne des résultats obtenus à cette compétition (qui est de 38,5%).

Méthode	MAP	R-Precision
Run 1 : *embeddings* + similarité	0.3617	0.3243
Run 2 : multi-mots et morpho-syntaxe	0.2732	0.2362

TABLE 9 – Scores obtenus pour la tâche 1

3 Tâche 2 : similarité entre documents

La tâche 2 est une tâche d'appariement entre documents. Les documents à apparier sont d'un côté des « cas » et de l'autre des « discussions ». Les méthodes proposées ici sont basée sur les embeddings de mots ou vecteurs-mots. Après avoir entrainé les embeddings sur le corpus d'apprentissage, la méthode consiste à calculer une représentation vectorielle des cas d'un côté et des discussions de l'autre, puis à calculer des similarités deux à deux entre ces cas et ces discussions, puis à choisir la configuration qui maximise les paires de similarités cas/discussion à l'aide de l'algorithme hongrois (ref). On commence par utiliser Word2Vec (Mikolov *et al.*, 2013) pour transformer les mots en

vecteurs. Plusieurs paramètres ont été testés : la version Skip-Gram et CBOW (qui consiste à prévoir les contextes d'un mot à partir du mot lui-même ou qui consiste à prévoir le mot à partir de son contexte), le voisinage de mot à gauche et à droite (de 2 à 5), la taille de la couche cachée (de 25 à 300) et le nombre d'itérations (500 ou 1000). Les meilleurs résultats sont obtenus avec : Skip-Gram, un voisinage de 5 mots, une couche cachée de taille 300 et 1000 itérations. La fréquence minimale pour les mots est fixée à 3. Plusieurs méthodes de passage des embeddings de mots aux embeddings de documents ont été testées correspondant aux différents run : SWEM-aver, DoCov et DoCov + P-Mean. Une fois calculées les similarités deux à deux entre les cas et les discussions, la configuration optimale est choisie en faisant appel à « l'algorithme hongrois ou méthode hongroise, aussi appelé algorithme de Kuhn-Munkres, algorithme d'optimisation combinatoire, qui résout le problème d'affectation en temps polynomial. C'est donc un algorithme qui permet de trouver un couplage parfait de poids maximum dans un graphe biparti dont les arêtes sont valuées. » [2].

3.1 Run 1 : SWEM-average

La première méthode pour passer des embeddings de mots aux embeddings de documents est la plus simple, elle consiste à moyenner tous les vecteurs-mots du document. Elle est appelée SWEM-aver dans (Shen *et al.*, 2018) pour « Simple Word Embedding Model – average » :

$$z = \frac{1}{L}\sum_{i=1}^{L} v_i \tag{5}$$

avec z, la représentation vectorielle du document, L le nombre de mots du document, v_i le vecteur mot du i^{eme} mot du document. Les meilleurs résultats ont été obtenus avec la variante consistant à pondérer les mots par leur IDF.

3.2 Run 2 : DoCov

La méthode DoCoV (Torki, 2018) pour « Document Co Variance » va calculer une représentation vectorielle du document à partir du triangle supérieur de la matrice de covariance. Ce vecteur aura pour taille $d * \frac{d+1}{2}$ si d est la taille des embeddings (cf. Figure 1).

$$O = \begin{pmatrix} x_{11} & \cdots & x_{1d} \\ x_{21} & \cdots & x_{2d} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nd} \end{pmatrix} \quad \sigma_{X,Y} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{N}$$

$$C = \begin{pmatrix} \sigma^2_{X_1} & \sigma_{X_1X_2} & \cdots & \sigma_{X_1X_d} \\ \sigma_{X_1X_2} & \sigma^2_{X_2} & \cdots & \sigma_{X_2X_d} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{X_1X_d} & \sigma_{X_2X_d} & \cdots & \sigma^2_{X_d} \end{pmatrix} \quad v = \mathbf{vect}(C) = \begin{cases} \sqrt{2}C_{p,q} & \text{if } p < q \\ C_{p,q} & \text{if } p = q \end{cases}$$

FIGURE 1 – DoCov

2. https://fr.wikipedia.org/wiki/Algorithme_hongrois

De même que précédemment, nous avons modifié ces formules pour introduire l'IDF. La taille des embeddings résultant est assez impressionnante. Avec d=300, on obtient un vecteur de taille 45150 !

3.3 Run 3 : DoCov + Pmean

La méthode des « Power Mean » présentée dans (Rücklé *et al.*, 2018) consiste à concaténer plusieurs types de moyennes :

$$\left(\frac{x_1^p + ... + x_n^p}{n}\right)^{1/p} ; p \in \mathbb{R} \cup \{\pm\infty\} \tag{6}$$

Calcul réalisé pour chaque composante des vecteurs, avec x_i la i^{eme} composantes des n vecteurs composant le document. Avec $p = 1$, on retrouve la mesure SWEM-aver précédente. De la même manière que précédemment, nous avons introduit le coefficient IDF. Les valeurs des composantes x_i doivent obligatoirement être positives. Comme ce n'est pas le cas avec nos données, nous avons choisi des valeurs paires pour p (sauf pour $p = 1$). Il suffit ensuite de concaténer les vecteurs obtenus avec p=1 et p=2 pour obtenir un vecteur de taille $2 * d$. L'idée intuitive est de « capter » plus d'informations par ces concaténations. Nous avons fait des tests avec $p = 1, 2$ puis $p = 1, 2, 4$ puis $p = 1, 2, 4, 6$ puis $p = 1, 2, 4, 6, 8$. Comme DoCoV obtenait de bons résultats, on vient ajouter aux embeddings obtenus avec DoCoV ceux obtenus avec PMean avec p=1,2.

3.4 Résultats

Le meilleur score est obtenu avec la méthode DoCoV, qui est le meilleur résultat obtenu à cette compétition.

Méthode	Score
Run 1 : SWEM-Aver	88,79 %
Run 2 : DoCov	95,32 %
Run 3 : DoCov + Pmean (1,2)	93,45 %

TABLE 10 – Scores obtenus pour la tâche 2

4 Tâche 3 : extraction d'information

Dans cette tâche d'extraction d'information, il est nécessaire de repérer, dans les cas cliniques, les informations démographiques et cliniques.

4.1 Détection du genre des patients

A partir des cas cliniques, l'objectif est de déterminer si la personne est de genre « féminin », « masculin » ou s'il s'agit de plusieurs personnes, auquel cas « féminin|masculin ». Il y a également une catégorie d'individus non classés.

4.1.1 Méthode

La distribution des genres des patients est présentée dans la Table 11. Bien qu'il y ait plus de « masculin » que de « féminin », les proportions ne sont pas significativement différentes pour les utiliser dans l'étude. On remarque que « féminin » désigne à la fois un cas clinique sur une personne de genre « féminin », mais aussi sur plusieurs personnes de ce genre, tout comme « masculin ».

féminin	masculin	féminin|masculin	NA
117	168	4	1

TABLE 11 – Répartition des genres dans le corpus

Nous avons également remarqué un cas difficile à détecter dans le corpus : on parle d'un patient de 57 ans, et on précise bien, à de nombreuses reprises, qu'il s'agit d'un patient de sexe masculin, et le genre associé au patient est « féminin ».

Pour extraire les informations sur le genre, nous avons utilisé deux lexiques :

1. un lexique qui contient des mots relatifs à une désignation d'une personne de genre « féminin » comme « fillette », « femme », « patiente », « sexe féminin » etc. Nous avons également ajouté les situations qui sont plus fréquentes chez une personne de genre « féminin » : « violée », « ménopausée », « enceinte » etc. Enfin, nous avons ajouté les verbes conjugués qui reviennent très souvent dans le corpus : « âgée de » et « née le » ainsi que les parties du corps présentent le plus souvent chez ces personnes : « vagin », « vulve », « ovaire » etc.

2. un lexique qui contient des mots des mots relatifs à une désignation d'une personne de genre « masculin » comme « garçon », « monsieur », « patient », « sexe masculin » etc. On a également ajouté des situations qui sont plus fréquentes chez une personne de genre « masculin » : « infertilité », « circoncit », « éjaculer », « flaccide » etc. Enfin, nous avons ajouté des verbes conjugués qui reviennent très souvent dans le corpus : « âgé de » et « né le » ainsi que des parties du corps présentent le plus souvent chez ces personnes : « pénis », « testicule », « prostate » etc.

Ces lexiques ont été utilisés pour nous permettre de comptabiliser les mots significatifs pouvant apparaître dans un cas clinique. Si on ne trouve que des mots appartenant au lexique « féminin », on associera ce genre au cas clinique, de même pour le genre masculin. En revanche, si on voit des mots appartenant aux deux lexiques, le mot majoritaire « gagne », et la classe « féminin|masculin » est obtenue uniquement s'il y a autant de mots des deux lexiques dans le cas clinique.

4.2 Détection de l'âge des patients

A partir des cas cliniques, l'objectif est de déterminer l'âge du ou des patients. Il y a également une catégorie d'individus pour lesquels il n'y a pas d'âge (non classés). Pour cette tâche, l'objectif est d'utiliser un ensemble de règles d'extraction pour pouvoir déterminer l'âge du ou des patients en utilisant le cas clinique.

4.3 Méthode

Afin d'extraire l'âge du ou des patients, nous avons choisi d'utiliser une méthode à base de règles d'extraction. Pour cela, il a fallu prendre en compte différents cas :

— l'âge de l'individu est indiqué en années : nous avons seulement à récupérer l'âge associé à l'individu.
— l'âge de l'individu est indiqué en mois : une conversion en années s'impose, en arrondissant le résultat obtenu à l'âge inférieur auquel il correspond. Par exemple, pour un enfant de 3 mois, nous arrondirons l'âge à 0 ans.
— l'âge de l'individu, en mois ou en années, est écrit en lettres : nous utilisons un dictionnaire qui nous permet d'associer un chiffre ou un nombre écrit en toutes lettres à l'écriture chiffrée correspondante.
— un adjectif, comme « quinquagénaire », est donné pour indiquer l'âge du patient : un dictionnaire est également utilisé pour parer à cette éventualité.
— plusieurs patients sont présents dans le cas clinique : extraction de déclencheurs comme « âgées de », « âges respectifs » etc.

4.4 Détection de l'issue

A partir des cas cliniques, l'objectif est de déterminer l'état du patient parmi : amélioration, stable, détérioration ou décès. Il y a également une catégorie d'individus non classés.

4.4.1 Méthode

Dans cette partie, l'objectif est d'essayer de retrouver les classes en regroupant les documents avec des techniques de *clustering*. Cela nous permet d'évaluer la similarité des documents entre plusieurs matrices. Nous avons utilisé une méthode de *clustering* de type *spherical k-means*, utilisant la distance cosinus.

La classification sera réalisée à partir d'une matrice **document-vecteur** obtenue après entraînement d'un **modèle doc2vec** PV-DBOW (Le & Mikolov, 2014) sur le corpus : prédit le mot cible à partir de plusieurs mots du contexte. Exemple : prédiction de « lit » à partir de la séquence « le chat s'asseoit sur le ».

Nous avons utilisé un lexique de mots nous permettant, avant le calcul de doc2vec, de repérer les cas cliniques ayant une issue « décès », ces cas étant assez faciles à détecter à l'aide de règles d'extraction.

4.5 Résultats

Malheureusement, les résultats obtenus sont en dessous de ceux obtenus durant la compétition.

Age		Genre		Issue	
Precison	Rappel	Precison	Rappel	Precison	Rappel
0.93925	0.46744	0.96667	0.47209	0.36150	0.18033

TABLE 12 – Scores obtenus pour la tâche 3

5 Conclusion

Participer à la campagne DEFT 2019, nous a permis de tester plusieurs méthodes basées sur des règles linguistiques et des plongements de mots. Aucune donnée supplémentaire, autre que les données d'apprentissage, n'a été utilisée. Les résultats obtenus sont satisfaisants. Les méthodes que nous avons mises en œuvre sont facilement transposables à d'autres tâches et peuvent intéresser plusieurs entités du groupe EDF.

Références

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*.

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*.

LE Q. & MIKOLOV T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning*, p. 1188–1196.

MIKOLOV T., SUTSKEVER I., CHEN K., CORRADO G. S. & DEAN J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, p. 3111–3119.

RÜCKLÉ A., EGER S., PEYRARD M. & GUREVYCH I. (2018). Concatenated power mean word embeddings as universal cross-lingual sentence representations. *arXiv preprint arXiv :1803.01400*.

SCHAKEL A. M. & WILSON B. J. (2015). Measuring word significance using distributed representations of words. *arXiv preprint arXiv :1508.02297*.

SHEN D., WANG G., WANG W., MIN M. R., SU Q., ZHANG Y., LI C., HENAO R. & CARIN L. (2018). Baseline needs more love : On simple word-embedding-based models and associated pooling mechanisms. *arXiv preprint arXiv :1805.09843*.

SPARCK JONES K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, **28**(1), 11–21.

TORKI M. (2018). A document descriptor using covariance of word vectors. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, p. 527–532.

WILSON B. J. & SCHAKEL A. M. (2015). Controlled experiments for word embeddings. *arXiv preprint arXiv :1510.02675*.