Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 31e édition),
Traitement Automatique des Langues Naturelles (TALN, 27e édition),
Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition)
Nancy, France, 08-19 juin 2020

DOING@DEFT : cascade de CRF pour l'annotation d'entités cliniques imbriquées

Anne-Lyse Minard[2] Andréane Roques[1] Nicolas Hiot[1]
Mirian Halfeld Ferrari Alves[1] Agata Savary[3]
(1) Université d'Orléans, LIFO, Orléans, France
(2) Université d'Orléans, LLL-CNRS, Orléans, France
(3) Université François Rabelais Tours, LIFAT, Tours, France
anne-lyse.minard@univ-orleans.fr, mirian@univ-orleans.fr

RÉSUMÉ

Cet article présente le système développé par l'équipe DOING pour la campagne d'évaluation DEFT 2020 portant sur la similarité sémantique et l'extraction d'information fine. L'équipe a participé uniquement à la tâche 3 : "extraction d'information". Nous avons utilisé une cascade de CRF pour annoter les différentes informations à repérer. Nous nous sommes concentrés sur la question de l'imbrication des entités et de la pertinence d'un type d'entité pour apprendre à reconnaître un autre. Nous avons également testé l'utilisation d'une ressource externe, MedDRA, pour améliorer les performances du système et d'un pipeline plus complexe mais ne gérant pas l'imbrication des entités. Nous avons soumis 3 runs et nous obtenons en moyenne sur toutes les classes des F-mesures de 0,64, 0,65 et 0,61.

ABSTRACT

DOING@DEFT : cascade of CRF for the annotation of nested clinical entities

In this paper, we present the participation of the DOING team in the DEFT 2020 shared task. DEFT 2020 focuses on semantic similarity and fine-grained information extraction. The DOING team has only participated in the 3rd task : "information extraction". We have used a method based on a cascade of CRF for annotating the requested information. Our work focus on the issue of nested entities and the impact of entity types to each other. We have also experimented the use of an external resource, MedDRA, which enables us to improve the performances of our system, and the use of an extraction pipeline which do not deal with nested entities. We have submitted 3 runs and we have obtained on overall the following F1 : 0.64, 0.65 and 0.61.

MOTS-CLÉS : extraction d'information fine ; cas cliniques ; entités cliniques ; entités imbriquées ; apprentissage automatique ; CRF.

KEYWORDS: fine-grained information extraction ; clinical cases ; clinical entities ; nested entities ; machine learning ; CRF.

1 Introduction

Dans le cadre du groupe de travail DOING (cf. section 1.1), nous avons développé un système pour participer à la tâche 3 de DEFT 2020 (Cardon *et al.*, 2020). Cette tâche est centrée sur l'extraction d'information fine dans un corpus de cas cliniques, et en particulier de l'annotation d'entités cli-

niques (examen, anatomie, substance, pathologie, etc.) et d'informations associées (dosage, mode d'administration, etc.).

Le schéma d'annotation utilisé pour cette tâche admet l'annotation d'entités imbriquées. Nous nous sommes concentrés sur cet aspect pour mettre en place notre méthode d'extraction. Nous proposons d'utiliser une méthode basée sur une cascade de CRF (champs aléatoires conditionnels, (Lafferty, 2001)) qui nous permet d'extraire d'abord les entités les plus imbriquées et de terminer avec les entités pouvant englober plusieurs types d'entité.

Dans la suite de cette section, nous présentons le projet DOING dans le cadre duquel nous avons participé à cette campagne d'évaluation, puis un état de l'art du domaine. Ensuite, nous décrivons la méthode utilisée (section 2), notre système à base d'apprentissage automatique (section 3) et les résultats obtenus (section 4).

1.1 Projet DOING

Le travail ici présenté a été développé dans le cadre des activités DOING (Données Intelligentes). En effet, le groupe de travail DOING, proposé en 2018 dans le cadre du réseau régional DIAMS (Données, Intelligence Artificielle, Modélisation et Simulation) [1], a commencé ses rencontres en février 2019 autour d'une collaboration entre chercheurs en bases de données, intelligence artificielle et traitement automatique de la langue. Aujourd'hui DOING a évolué : non seulement il représente un groupe actif au sein de DIAMS, mais propose une ouverture nationale comme atelier MADICS (janvier 2020) [2] et internationale dans l'organisation de DOING'2020 [3] – *workshop* au sein de la conférence ADBIS-TPDL-EDA [4]. Dans tous ces différents formats, DOING s'intéresse à la transformation des données en information, puis en connaissance. Le groupe vise en particulier deux grandes lignes de discussions ainsi que leur mise en relation : (1) la transformation des données en information, c'est-à-dire, l'extraction de l'information des données textuelles pour peupler une base de connaissances et (2) la transformation de l'information en connaissance, c'est-à-dire, l'interrogation intelligente et efficace, et la maintenance des bases de connaissances. Le domaine de la santé s'est présenté comme la première cible d'application de DOING.

Dans ce contexte, l'extraction d'information est un aspect clé du travail du groupe DOING ; le point de départ pour la ligne de recherche (1) citée ci-dessus. Le défi DEFT 2020 est une opportunité de concrétisation d'une collaboration naissante autour d'un stage dont l'ambition repose sur une première approche permettant de peupler une base de données à partir des données textuelles. Les méthodes développées pour DEFT 2020 s'insèrent dans les premières étapes de la conception de cette approche.

1.2 État de l'art

La tâche d'extraction d'information (IE) consiste en l'extraction automatique d'information structurée à partir de documents numériques non structurés ou semi-structurés, par exemple dans le but d'alimenter une base de données ou faciliter les traitements consécutifs (Jurafsky & Martin, 2009). Ce domaine

1. https://www.univ-orleans.fr/lifo/evenements/RTR-DIAMS/
2. http://www.madics.fr/ateliers/doing/
3. http://www.univ-orleans.fr/lifo/evenements/doing/?page_id=77
4. http://eric.univ-lyon2.fr/adbis-tpdl-eda-2020/adbis/

a une longue tradition et une bibliographie très riche dans le domaine du TAL. La sous tâche la plus élémentaire de l'IE est l'extraction des entités d'intérêt, souvent appelées entités nommées (EN), d'où le terme consacré : *reconnaissance d'entités nommées* (REN). Dans la langue générale, il s'agit le plus souvent des noms propres (*Union européenne*) ou des dates (*25 mai*) et mesures (*57,5%*). Dans une langue de spécialité, telle que la biomédecine, les ENs incluent des termes (*occlusion intestinale*) et mesures spécifiques (*191/95 mm Hg*). Les EN de la langue générale apparaissent dans une très grande quantité de textes librement accessibles et ont fait l'objet de nombreux efforts d'annotation en beaucoup de langues. Ceci a permis un développement de benchmarks (Tjong Kim Sang, 2002; Tjong Kim Sang & De Meulder, 2003), ainsi que de méthodes supervisées de tagging séquentiel, y compris neuronales (Yadav & Bethard, 2018), souvent basées sur le codage du corpus selon le modèle BIO. [5] La REN spécifique à un domaine de spécialité souffre d'une accessibilité moindre de textes, tant bruts qu'annotés, et des benchmarks sont accessibles surtout pour l'anglais dans le domaine biomédical (Campos *et al.*, 2012). De ce point de vue, la campagne DEFT 2020 constitue un effort important pour le français.

Le défi majeur, commun à ces deux tâches – la REN en langue générale et celle dans le domaine biomédical – est la présence d'une grande quantité d'entités imbriquées les unes dans les autres. Par exemple – selon la terminologie de DEFT 2020 – dans *[anémie à [9g/dl]*$_1$ *d'[hémoglobine]*$_2$*]*$_3$ les entités 1 et 2 des catégories $valeur$ et $substance$, respectivement, sont englobées par l'entité plus large 3 de catégorie $pathologie$. Toutes ces trois entités doivent être automatiquement reconnues et catégorisées. L'imbrication d'entités reste un défi important en REN, car elle la rapproche de la tâche d'analyse syntaxique, où ce sont des mini-arbres d'entités qu'il faut désormais produire, et non seulement des séquences d'étiquettes (Finkel & Manning, 2009). Des méthodes de tagging séquentiel, telles que les CRF, peuvent tout de même être utilisées, surtout si les EN restent continues, mais soit les modèles, soit les jeux d'étiquettes s'en trouvent complexifiés (d'où la rareté de données accrue), pour tenir compte du fait qu'un mot peut appartenir à plusieurs entités à la fois (Alex *et al.*, 2007). Vu que les données annotées de la campagne DEFT 2020 comportent de nombreuses imbrications, le système que nous présentons y attache une attention particulière. Il s'inspire de l'une des architectures proposées par ces derniers auteurs.

2 Méthode

Les entités considérées dans la tâche 3 de DEFT 2020 sont de 10 types [6]. Dans le tableau 1, nous donnons la répartition des entités dans le corpus d'entraînement. Nous pouvons les regrouper en 2 groupes, les entités cliniques et les entités associées à ces entités cliniques :

1. $anatomie$, $examen$, $traitement$, $substance$, $sosy$ (signe ou symptôme), $pathologie$
2. $valeur$, $dose$, $mode$, $moment$ (+ $date$, $duree$, $frequence$)

Le schéma d'annotation permet l'imbrication des entités. Dans le tableau 2, nous présentons les imbrications les plus fréquentes dans le corpus. La première colonne contient le type de l'entité englobante, la deuxième colonne le type de l'entité imbriquée, la troisième colonne le nombre de fois où ces deux entités sont imbriquées et les deux dernières colonnes le pourcentage d'entités imbriquées

5. Chaque mot du corpus est ainsi encodé comme apparaissant au début (B), à l'intérieur (I) ou en dehors (O) d'une EN d'un certain type.

6. La tâche proposée par DEFT n'inclut pas l'annotation des entités $date$, $duree$ et $frequence$. Comme les annotations étaient disponibles dans le corpus d'entraînement, nous avons travaillé également sur la reconnaissance de ces entités.

	nombre d'entités
sosy	1831
anatomie	1608
examen	1218
substance	1024
valeur	588
moment	451
dose	392
traitement	374
pathologie	369
mode	243

TABLE 1 – Nombre d'entités par type dans le corpus d'entraînement.

par rapport au nombre d'entités 1 ou 2 dans le corpus. [7] Nous remarquons entre autres que les entités *sosy* englobent très souvent d'autres entités, en particulier des entités *anatomie* et que les entités *anatomie* sont très souvent imbriquées (au total 1496 sont imbriquées, parfois dans plusieurs entités à la fois, ce qui représente 93% des entités *anatomie*). Ces entités *sosy* sont souvent longues, avec en moyenne 4,9 mots par entité, contre 1,5 mots par entité pour les entités *anatomie*. Cet aspect nous a incités à construire un système en cascade, en suivant les travaux de (Alex *et al.*, 2007). Ainsi, différents modèles seront entraînés pour apprendre à reconnaître un ou deux types d'entités.

entité 1	entité 2	ent2 imbriquée dans ent1	proportion[7] ent1	proportion[7] ent2
sosy	anatomie	980	54%	61%
pathologie	anatomie	151	41%	9%
examen	anatomie	370	30%	23%
traitement	anatomie	111	30%	7%
sosy	examen	440	24%	36%
sosy	valeur	409	22%	70%
pathologie	valeur	18	5%	3%
sosy	substance	80	4%	8%
examen	substance	23	2%	2%

TABLE 2 – Nombre d'entités imbriquées dans le corpus d'entraînement pour les paires les plus fréquentes.

L'apprentissage sera effectué dans l'ordre suivant :

1. *dose* et *valeur*
2. *duree* et *frequence*
3. *date* et *moment*
4. *anatomie* et *mode*
5. *traitement* et *examen*
6. *substance*
7. *sosy* et *pathologie*

7. `proportion ent1 = nombre ent2 imbriquée dans ent1 / nombre ent1; proportion ent2 = nombre ent2 imbriquée dans ent1 / nombre ent2`

En faisant ces regroupements, nous cherchons à apprendre ensemble des entités proches sémantiquement, morphologiquement et qui ne sont pas imbriquées ou très peu. [8] Par exemple, nous avons constaté que les entités *dose* et *valeur* se présentent majoritairement sous la forme d'un nombre suivi d'une unité (exemples : *dose* "40 mg" ; *valeur* "191/95 mm Hg"). En les apprenant ensemble, nous gagnons en rappel (+ 1,1 et + 1,7 respectivement pour *dose* et *valeur*) avec une légère perte de précision. [9] Les entités *anatomie* et *mode* peuvent être proches sémantiquement et/ou morphologiquement et apparaître dans des contextes similaires (exemples : *mode* "entérale" ; *anatomie* "abdominal"). Apprendre à les reconnaître avec le même modèle permet d'améliorer le rappel de +3 points et +0.3 points respectivement pour la reconnaissance des entités *mode* et *anatomie*. De même, l'entité *moment* tend par exemple à apparaître en début de phrase ("À son admission", "Une semaine plus tard"), tout comme l'entité *date* ("[En] 2001"). Dans ce contexte, les prépositions sont par ailleurs fréquentes. Comme évoqué précédemment, nous avons également effectué nos regroupements en considérant les entités imbriquées. Les entités *anatomie* et *mode* sont ainsi souvent imbriquées ou associées aux entités *traitement* et *examen*. C'est pourquoi deux paires ont été créées.

Cet apprentissage en cascade offre également la possibilité d'utiliser les annotations produites par le système comme traits supplémentaires pour l'annotation du niveau suivant. Par exemple le modèle numéro 5 appris pour *traitement* et *examen* utilisera les annotations en *valeur*, *dose*, *anatomie*, etc. et le modèle appris pour *sosy* et *pathologie* utilisera les annotations produites par tous les modèles précédents. Cette configuration permet par exemple d'améliorer le rappel et la précision pour la classe *pathologie* respectivement de +2,3 et +1,8. Ces traits sont décrits dans la section 3.3.1.

3 Système

Le premier module de notre système est un module de pré-traitement, il est présenté dans la section 3.1. Lors du pré-traitement, les annotations sont transformées selon le format BIO, format standard pour les CRF.

Des modèles sont ensuite appris pour chaque niveau d'annotation (cf. section 3.2) et appliqués au fur et à mesure aux données non annotées. Pour apprendre ces modèles, il est nécessaire de définir les traits à utiliser dans des templates, un par modèle.

Les templates sont des fichiers contenant des traits (caractéristiques d'un token, d'un segment, etc. utiles pour apprendre à reconnaître les entités) définis sous la forme de patrons, selon une syntaxe particulière. Pour un token courant donné (noté 0), il convient de détailler les informations à prendre en compte pour l'étiqueter. À partir d'un fichier en entrée au format tabulaire (un token par ligne), il est ainsi possible d'indiquer, pour ce token, combien de tokens précédents (exemple : 2) et/ou suivants (exemple : 1) considérer, ainsi que la colonne dans laquelle se trouvent les caractéristiques pertinentes (exemple : le lemme en colonne 3). Dans cette situation, les traits à définir sont les suivants :

- %x[-2,3] = deuxième token précédent, colonne 3
- %x[-1,3] = token précédent, colonne 3
- %x[0,3] = token courant, colonne 3

8. Dans le corpus d'entraînement, pour certaines entités que nous avons regroupées, il existe des cas où elles sont imbriquées. Par exemple, à 9 reprises il y a une entité *pathologie* imbriquée dans une entité *sosy*. Lorsque ces cas occurrent moins de 10 fois dans le corpus, nous les avons ignorés.

9. Les résultats présentés dans cette section ont été obtenus en validation croisée à 10 plis, avec les traits définis dans la section 3.3.

— %x[1,3] = token suivant, colonne 3

Ici, la fenêtre définie (c'est-à-dire le contexte pris en compte) se présente sous la forme de l'intervalle [-2,1]. Tout au long de cet article, nous utiliserons cette notation pour évoquer la taille des fenêtres choisie pour les traits utilisés. Ces derniers sont présentés dans la section 3.3.

Le dernier module permet d'effectuer la transformation des annotations au format BIO vers le format BRAT. [10]

3.1 Pré-traitement

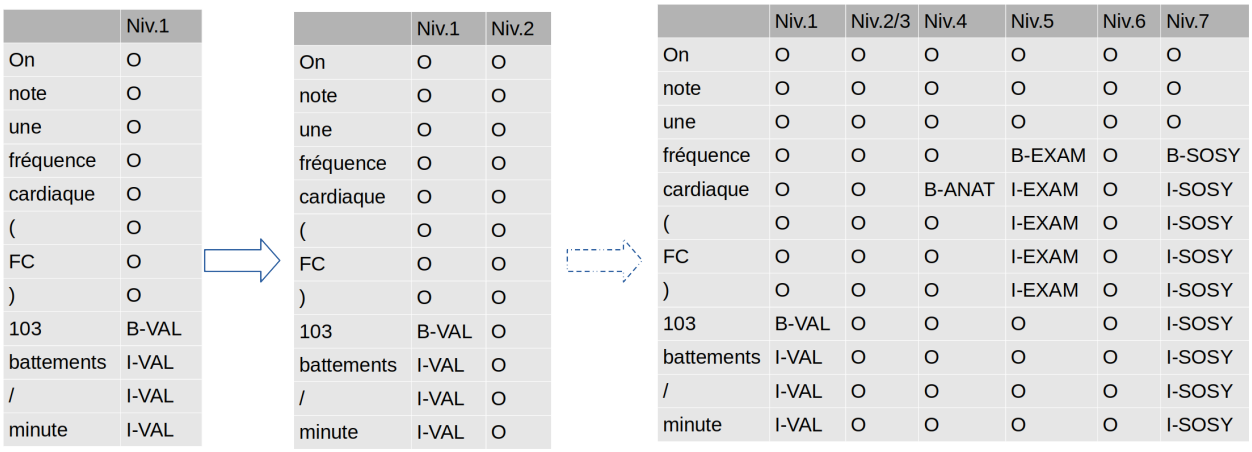
Pour le pré-traitement des fichiers, nous avons principalement utilisé SpaCy [11] (Honnibal & Montani, 2017), excepté pour le découpage en phrases car les performances du module pour le français ne nous satisfaisaient pas. En effet, il considère qu'un tiret entre deux mots indique une fin de phrase, ce qui produit un découpage non exploitable. Nous avons donc utilisé l'outil sentence-splitter développé pour le traitement du corpus Europarl [12].

Le modèle français de SpaCy [13] nous a permis d'effectuer la tokenisation et d'obtenir, pour chaque token, les informations suivantes :

— le token en caractères minuscules ;
— le lemme du token ;
— la catégorie syntaxique ;
— en cohérence avec la catégorie syntaxique, les étiquettes morpho-syntaxiques détaillées telles que le genre, le nombre, le type de numéral (ordinal, cardinal), le type de pronom ou de déterminant (relatif, personnel, démonstratif, article, etc.), le temps, la personne, le mode, la voix, la polarité, etc. ;
— la dépendance syntaxique ;
— la forme du token (caractères alphabétiques remplacés par "x" ou "X" et chiffres remplacés par "d") ;
— la forme détaillée du token, c'est-à-dire s'il est composé : uniquement de caractères alphabétiques (A), uniquement de chiffres (D), de chiffres seuls ou avec ponctuation (NB ; exemple : "115/60"), uniquement de signes de ponctuation (P) ou autres (O ; exemples : "d'", "4H") ;
— si le token est composé de caractères alphabétiques (True) ou non (False) ;
— si le token est ou fait partie d'une entité nommée, le type de l'entité : personne (PER), lieu politique ou géographique (LOC), nom d'organisation gouvernementale ou autre (ORG), entités diverses telles que des produits, des événements, des nationalités, etc. (MISC) ;
— la position B/I/O du token au sein d'une entité nommée.

Pour chaque token, nous avons également ajouté des informations relatives à la présence ou l'absence de préfixe et/ou suffixe. Pour ce faire, nous avons utilisé une liste de préfixes et suffixes du français extraits du TLFi [14] pour identifier si les tokens en étaient composés ou non. Le paramétrage est défini afin de rechercher uniquement les préfixes d'une longueur comprise entre 3 et 8 caractères et les suffixes d'une longueur comprise entre 3 et 9 caractères. Pour chaque token composé de plusieurs affixes de même type (préfixe ou suffixe), celui ayant la longueur la plus importante est conservé

10. https://brat.nlplab.org
11. https://spacy.io/
12. Outil développé par Philipp Koehn and Josh Schroeder (https://github.com/berkmancenter/mediacloud-sentence-splitter).
13. Nous avons utilisé le modèle *fr_core_news_md*.
14. https://hugonlp.wordpress.com/2015/10/22/



	Niv.1
On	O
note	O
une	O
fréquence	O
cardiaque	O
(	O
FC	O
)	O
103	B-VAL
battements	I-VAL
/	I-VAL
minute	I-VAL

	Niv.1	Niv.2
On	O	O
note	O	O
une	O	O
fréquence	O	O
cardiaque	O	O
(	O	O
FC	O	O
)	O	O
103	B-VAL	O
battements	I-VAL	O
/	I-VAL	O
minute	I-VAL	O

	Niv.1	Niv.2/3	Niv.4	Niv.5	Niv.6	Niv.7
On	O	O	O	O	O	O
note	O	O	O	O	O	O
une	O	O	O	O	O	O
fréquence	O	O	O	B-EXAM	O	B-SOSY
cardiaque	O	O	B-ANAT	I-EXAM	O	I-SOSY
(	O	O	O	I-EXAM	O	I-SOSY
FC	O	O	O	I-EXAM	O	I-SOSY
)	O	O	O	I-EXAM	O	I-SOSY
103	B-VAL	O	O	O	O	I-SOSY
battements	I-VAL	O	O	O	O	I-SOSY
/	I-VAL	O	O	O	O	I-SOSY
minute	I-VAL	O	O	O	O	I-SOSY

FIGURE 1 – Illustration du fonctionnement en cascade. Le premier tableau représente les prédictions après l'application du modèle 1, le deuxième après l'application du modèle 2, et le 3ème après l'application de tous les modèles.

(exemple : "héma-" et "hémato-"). Enfin, nous avons ajouté deux informations supplémentaires indiquant, pour chaque token, ses quatre premiers et quatre derniers caractères.

3.2 Cascade de CRF

Pour entraîner des CRF, nous utilisons l'outil Wapiti [15] (Lavergne *et al.*, 2010), avec l'algorithme RPROP (*resilient backpropagation*). Nous avons conservé les paramètres par défaut.

Les prédictions sur les données non annotées sont faites au fur et à mesure de l'apprentissage des modèles. Ainsi, par exemple, pour construire le modèle du niveau 4 ($mode$ et $anatomie$), nous pouvons utiliser les prédictions faites par le modèle 3 comme nouveaux traits. Dans la figure 1, nous donnons un aperçu du fonctionnement en cascade. Nous n'avons pas représenté les caractéristiques associées aux tokens, mais juste les dernières colonnes contenant les étiquettes au format BIO.

Toutes les prédictions faites pour chaque niveau sont conservées dans le fichier de sortie du système.

3.3 Traits

Nous décrivons dans cette section les traits utilisés pour les différents modèles appris. Ces traits correspondent aux informations obtenues lors du pré-traitement. Les combinaisons de traits et les choix des fenêtres ont été obtenus à partir d'expérimentations en validation croisée à 10 plis.

Les traits utilisés sont de quatre types [16] : sémantiques, morphologiques, morpho-syntaxiques et de surface.

Descripteurs sémantiques :

— le token ;

— le type de l'entité si le token est ou fait partie d'une entité nommée ;

15. https://wapiti.limsi.fr

16. Cette classification des descripteurs n'est pas absolue dans la mesure où certains traits, par exemple ceux liés aux préfixes et suffixes, pourraient être considérés comme des descripteurs sémantiques et morphologiques.

— la position B/I/O du token dans une entité nommée.

Le trait relatif au token a été ajouté dans tous les templates, avec une fenêtre de [-2,2]. Les traits concernant les entités nommées sont utilisés uniquement dans le template pour $mode$ et $anatomie$.

Descripteurs morphologiques :

— la présence (le cas échéant, lequel) ou l'absence d'un préfixe ;
— la présence (le cas échéant, lequel) ou l'absence d'un suffixe.

Les traits concernant les préfixes et suffixes ont été utilisés dans la quasi-totalité des templates mais plus particulièrement dans celui de $mode$ et $anatomie$ (fenêtre élargie : [-2,2]).

Descripteurs morpho-syntaxiques (variables selon les tokens) :

— le lemme du token ;
— la catégorie syntaxique ;
— en cohérence avec la catégorie syntaxique, les étiquettes morpho-syntaxiques détaillées : genre, nombre, type de numéral, type de pronom ou de déterminant, article défini/indéfini, temps du verbe, forme verbale, personne, mode ;
— la dépendance syntaxique.

Avec une fenêtre plus ou moins grande, certains traits ont été utilisés dans tous les templates, tels que ceux relatifs au lemme, à la catégorie syntaxique, au genre, au nombre. D'autres sont davantage spécifiques à certaines entités. Par exemple, pour les entités $dose$, $valeur$, $duree$, $frequence$, $date$ et $moment$, le type de numéral pour le token courant a été pris en compte. Les traits relatifs au temps, à la forme verbale, à la personne et au mode ont principalement été utilisés pour définir le contexte précédant l'entité. En effet, ces entités tendaient à être précédées d'un verbe.

Descripteurs de surface :

— la forme du token ;
— la forme détaillée du token ;
— la présence de caractères alphabétiques ou non ;
— les quatre premiers caractères du token ;
— les quatre derniers caractères du token.

En ce qui concerne les traits liés à la forme et la composition du token, la fenêtre définie est majoritairement [-1,1] dans tous les templates.

Dans tous les templates, nous avons également ajouté un "B" (pour bigramme) qui permet de prendre en compte l'enchaînement des étiquettes choisies par le système. Cette option permet au format BIO d'être correctement appris par le système, à savoir qu'un I- ne peut pas suivre un O mais seulement un B- ou un autre I-, etc.

3.3.1 Traits issus de la cascade de CRF

Afin d'utiliser les annotations des niveaux précédents, nous avons défini une fenêtre de [-3,3] pour les étiquettes de chaque niveau. Pour chaque niveau, des traits relatifs à toutes les étapes précédentes ont été ajoutés, excepté pour le niveau 4 $mode$ et $anatomie$ qui n'utilise pas les traits relatifs aux annotations de $date$ et $moment$. Par exemple le niveau 2 $duree$ et $frequence$ utilise les annotations en $dose$ et $valeur$ et le dernier niveau $sosy$ et $pathologie$ se sert des annotations de tous les autres niveaux. Dans l'exemple de la figure 1, le modèle 5 pourra utiliser comme trait le fait que "cardiaque" a été annoté "B-ANAT", et "103 battements/minute" a été annoté avec les étiquettes "B-VAL" et "I-VAL".

3.3.2 Connaissances externes : MedDRA

MedDRA© [17] ou Dictionnaire Médical des Affaires Réglementaires (Brown *et al.*, 1999) est un dictionnaire international de terminologie médicale standardisé destiné à être utilisé dans les affaires réglementaires. Il regroupe ainsi l'ensemble des termes médicaux représentant aussi bien des symptômes que des examens ou encore des traitements. Il constitue une base riche pour l'identification de ces termes. L'extraction des termes est réalisée à l'aide de l'outil d'étiquetage de texte fourni par le moteur de recherche SolR (Apache Software Foundation, 2006). Ce dernier utilise des n-grammes afin de calculer la similarité entre les lexèmes et une suite de termes que nous appelons lexique (Kim & Shawe-Taylor, 1994). Dans le corpus d'entraînement de DEFT, 2320 entités ont ainsi été extraites.

MedDRA a une structure hiérarchique en 5 niveaux. Le niveau le plus bas est le terme et le plus haut est une classification par discipline médicale. Cette classification est composée de 26 classes et regroupe des termes par étiologie, site de manifestation, etc. Les classes sont par exemple *Affections vasculaires*, *Affections du rein et des voies urinaires*, *Affections du système immunitaire*, etc.

Pour les modèles 4, 5, 6 et 7 (c'est-à-dire les modèles pour des entités cliniques), des traits supplémentaires sont utilisés pour indiquer si un token fait partie d'une entité MedDRA et pour indiquer la classe associée. Selon les modèles, la fenêtre utilisée est différente, allant de 1 (c'est-à-dire juste le token courant) à 7 (c'est-à-dire un intervalle de [-3 ;3]).

3.3.3 Combinaison de plusieurs algorithmes pour l'extraction d'entités : pipeline Ennov

Dans un effort de proposer des outils plus intelligents à destination de ses clients, l'entreprise française Ennov [18], éditeur de logiciel à destination du secteur médical, travaille sur l'implémentation d'outils permettant d'intégrer et de combiner diverses approches pour l'extraction d'information. Ce projet intègre aujourd'hui l'analyse syntaxique, l'extraction d'entités nommées (basée sur des grammaires, des lexiques ou des approches statistiques) ainsi que l'enrichissement des entités par des déclencheurs. C'est un outil modulaire qui permet de définir un pipeline où chaque composant est minimal, déplaçable et interchangeable ce qui permet une grande flexibilité.

Une version du pipeline a spécifiquement été construit pour la tâche DEFT. Elle constitue un travail préliminaire et aucun des paramètres n'a été modifié pour améliorer le résultat de l'extraction. Ce dernier génère donc de l'erreur qui impacte le résultat final. Il est cependant intéressant de chercher à optimiser ce pipeline pour obtenir de meilleurs résultats. Le pipeline est construit de la façon suivante : (1) analyse syntaxique avec SpaCy, (2) utilisation d'un CRF classique (c'est-à-dire sans entités imbriquées), (3) étiquetage avec le dictionnaire MedDRA, (4) extraction de données structurées (dates, e-mails, ...) au travers de grammaires locales, (5) utilisation du CNN (Réseau Neuronal Convolutif) de SpaCy pour la reconnaissance d'entités et (6) fusion des entités recouvrantes.

L'annotation produite par ce pipeline nous a permis d'ajouter les traits suivants :

1. étiquette au format BIO associée au token indiquant le type de l'entité (B-TIME, I-TIME, B-ANATOMIE, I-ANATOMIE, etc.)
2. étiquette au format BIO indiquant les entités identifiées sans leur type (B-ent, I-ent ou O)

17. La marque MedDRA© est enregistrée par l'IFPMA au nom du CIH. MedDRA© est développé par le Conseil International d'Harmonisation des exigences techniques pour l'enregistrement des médicaments à usage humain (CIH).
18. https://fr.ennov.com

Certains modèles (*date* et *moment*, *anatomie* et *mode*, *substance*) utilisent uniquement le trait 1 dans une fenêtre de 5, et les autres utilisent les deux traits dans des fenêtres de 5 également ([-2,2]).

4 Résultats

Nous avons soumis 3 runs à DEFT. Pour le premier run, nous avons utilisé le système décrit précédemment sans les descripteurs provenant de MedDRA et du pipeline Ennov. Pour le deuxième run, nous avons ajouté les descripteurs provenant de MedDRA et pour le troisième, les annotations produites par le pipeline. En résumé, les 3 runs sont :

— **run 1 :** cascade de CRF
— **run 2 :** cascade de CRF + traits MedDRA
— **run 3 :** cascade de CRF + traits MedDRA + traits pipeline Ennov

Dans le tableau 3, nous présentons les résultats pour toutes les catégories confondues (sous-tâches 1 et 2).

	TP	FP	FN	Précision	Rappel	F1
Run1	2695	937	2042	0,7420	0,5689	0,6440
Run2	2729	911	2008	0,7497	0,5761	**0,6515**
Run3	2570	1095	2167	0,7012	0,5425	0,6117

TABLE 3 – Résultats obtenus pour toutes les catégories (10 catégories). La meilleure F1 obtenue à la compétition est de 0,72.

Les résultats détaillés obtenus aux deux sous-tâches sont présentés dans les tableaux 4 et 5. Nous pouvons noter que les meilleurs résultats sont obtenus avec le système utilisant la ressource externe de MedDRA. Le gain de l'utilisation de cette ressource est relativement faible, avec une amélioration de 0,75 points de F-mesure par rapport au système ne l'utilisant pas (run 1). Ce gain est cependant intéressant pour des catégories particulières comme *substance* et *traitement*, où il atteint 1,79 et 2,47 respectivement. L'utilisation des traits provenant du pipeline Ennov fait diminuer les performances du système, excepté pour les entités *mode* et *traitement*. Une analyse fine des erreurs permettrait de mieux comprendre l'impact de l'utilisation de ces traits et d'expliquer la baisse des performances.

Nous observons que les types souvent imbriqués (*anatomie*, *valeur*, etc.) sont particulièrement bien reconnus. Ceci s'explique en partie par le fait que ces entités sont composées en moyenne de 1,5 à 3 tokens, ce qui facilite leur identification. En revanche, les performances du système pour les entités *sosy* et *pathologie* formées en moyenne de respectivement 4,9 et 3,1 tokens sont plus faibles. Les résultats assez bas pour *pathologie* peuvent être expliqués également par la forte ambiguïté avec le type *sosy* et sa faible représentativité dans le corpus (369 entités contre 1831 pour *sosy*).

Notre système n'obtient pas de bons résultats pour l'extraction des dosages, nous n'avons pour le moment pas d'explication à donner à cela. En validation croisée de 10 plis sur le corpus d'entraînement, nous obtenions une F1 de 0,66 pour cette classe. Selon le guide d'annotation, les dosages sont associés à une entité *substance*. Nous n'avons pas exploité cette caractéristique puisque nous avons fait le choix de placer l'extraction de *dose* avant *substance* dans l'architecture en cascade de notre système, ce qui pourrait expliquer les faibles résultats.

La dernière observation que nous pouvons faire concerne les entités *mode*. Les résultats ne sont pas très élevés pour une classe à la variation assez faible. En effet dans le corpus d'entraînement, nous

		TP	FP	FN	Précision	Rappel	F1
pathologie	run1	60	45	106	0,5714	0,3614	**0,4428**
	run2	56	55	110	0,5045	0,3373	0,4043
	run3	53	69	113	0,4344	0,3193	0,3681
sosy	run1	640	487	639	0,5679	0,5004	0,5320
	run2	656	468	623	0,5836	0,5129	**0,5460**
	run3	592	498	687	0,5431	0,4629	0,4998
Overall	run1	700	532	745	0,5682	0,4844	0,5230
	run2	712	523	733	0,5765	0,4927	**0,5313**
	run3	645	567	800	0,5322	0,4464	0,4855

TABLE 4 – Résultats obtenus à la sous-tâche 1. La meilleur F1 obtenue lors de la compétition est de 0,66. (TP = vrais positifs, FP = faux positifs, FN = faux négatifs)

		TP	FP	FN	Précision	Rappel	F1
anatomie	run1	744	168	376	0,8158	0,6643	0,7323
	run2	743	156	377	0,8265	0,6634	**0,7360**
	run3	691	199	429	0,7764	0,6170	0,6876
dose	run1	13	10	39	0,5652	0,2500	**0,3467**
	run2	13	10	39	0,5652	0,2500	**0,3467**
	run3	13	17	39	0,4333	0,2500	0,3171
examen	run1	516	94	301	0,8459	0,6316	0,7232
	run2	527	97	290	0,8446	0,6450	**0,7314**
	run3	511	131	306	0,7960	0,6255	0,7005
mode	run1	34	5	55	0,8718	0,3820	0,5313
	run2	34	3	55	0,9189	0,3820	0,5397
	run3	40	7	49	0,8511	0,4494	**0,5882**
moment	run1	106	21	59	0,8346	0,6424	**0,7260**
	run2	106	21	59	0,8346	0,6424	**0,7260**
	run3	97	21	68	0,8220	0,5879	0,6855
substance	run1	154	26	159	0,8556	0,4920	0,6247
	run2	160	25	153	0,8649	0,5112	**0,6426**
	run3	148	49	165	0,7513	0,4728	0,5804
traitement	run1	115	45	189	0,7188	0,3783	0,4957
	run2	121	40	183	0,7516	0,3980	**0,5204**
	run3	128	61	176	0,6772	0,4211	0,5193
valeur	run1	313	36	119	0,8968	0,7245	**0,8015**
	run2	313	36	119	0,8968	0,7245	**0,8015**
	run3	297	43	135	0,8735	0,6875	0,7694
Overall	run1	1995	405	1297	0,8313	0,6060	0,7010
	run2	2017	388	1275	0,8387	0,6127	**0,7081**
	run3	1925	528	1367	0,7848	0,5848	0,6701

TABLE 5 – Résultats obtenus à la sous-tâche 2. La meilleure F1 obtenue à la compétition est de 0,76.

relevons 89 termes différents annotés comme $mode$: *voie orale*, *voie intraveineuse*, *perfusion*, etc. Nous pouvons imaginer dans ce cas qu'une approche symbolique aurait pu nous apporter des meilleurs résultats. De plus dans le guide d'annotation, il est indiqué « Pas d'annotation isolée en "mode" dans une phrase s'il n'y a pas également un "traitement" ou une "substance". » Cette contrainte pourrait être ajoutée dans une phase de post-traitement ou comme un trait supplémentaire.

Notre système nous permet également d'extraire les entités $date$, $duree$ et $frequence$ même si cela ne faisait pas partie de la tâche de DEFT. Nous présentons dans le tableau 6 les résultats obtenus pour ces 3 entités. Pour les runs 1 et 2, nous obtenons les mêmes résultats car nous n'utilisions pas de traits MedDRA pour ces entités. Nous remarquons que la F1 pour les dates est très élevée, ce qui s'explique par le peu de variation dans cette catégorie (formats des dates les plus fréquents : année, ou mois année ou jour mois année).

		TP	FP	FN	Précision	Rappel	F1
date	run1 et run2	46	0	7	1,0000	0,8679	0,9293
	run3	46	0	7	1,0000	0,8679	0,9293
duree	run1 et run2	41	4	18	0,9111	0,6949	0,7885
	run3	45	7	14	0,8654	0,7627	0,8108
frequence	run1 et run2	12	4	16	0,7500	0,4286	0,5455
	run3	11	3	17	0,7857	0,3929	0,5238

TABLE 6 – Résultats non-officiels obtenus pour les entités "date", "duree" et "frequence".

5 Conclusion

Nous avons présenté dans cet article notre participation à la tâche 3 de la campagne d'évaluation DEFT 2020. Cette tâche s'intéressait à l'extraction d'information fine dans le domaine médical. Nous avons proposé un modèle basé sur une cascade de CRF, qui nous a permis de gérer l'extraction d'entités imbriquées. Nous avons obtenu des résultats supérieurs à la moyenne et à la médiane de DEFT. Nous avons déjà des pistes pour l'amélioration de notre système.

Ainsi, comme évoqué précédemment, la prise en compte de la présence ou de l'absence d'une entité $substance$ dans un contexte donné pourrait être ajoutée comme trait pour améliorer l'identification des entités $dose$. De plus, pour ces entités, l'utilisation d'une liste d'unités de mesure, de pression, etc. pourrait également être bénéfique. De même, en ce qui concerne l'amélioration des résultats pour les entités $mode$, nous pourrions ajouter des informations lors du pré-traitement afin d'indiquer, par exemple, la présence ou l'absence d'entités $traitement$ et/ou $substance$ dans la même phrase. Quant aux entités $date$ et $moment$, qui apparaissent fréquemment en début de phrase ou en apposition, un trait relatif à la position du token dans la phrase pourrait être défini. Par ailleurs, nous pouvons constater que les performances de notre système sont meilleures en termes de précision qu'en termes de rappel. Nous pensons que cela est en partie dû au fait que les traits définis sont trop nombreux et/ou spécifiques à certains contextes. Afin d'améliorer les résultats de rappel et de F-mesure, nous envisageons de réduire le nombre de traits et de les généraliser davantage. Bien que les résultats de précision risquent d'être moins bons, cela permettra d'équilibrer l'ensemble des résultats.

La prochaine étape de ce travail dans le cadre du groupe de travail DOING sera de travailler sur l'extraction de relations entre des entités cliniques.

Références

ALEX B., HADDOW B. & GROVER C. (2007). Recognising nested named entities in biomedical text. In *Biological, translational, and clinical language processing*, p. 65–72, Prague, Czech Republic : Association for Computational Linguistics.

APACHE SOFTWARE FOUNDATION (2006). Apache SolR ©.

BROWN E. G., WOOD L. & WOOD S. (1999). The Medical Dictionary for Regulatory Activities (MedDRA). *Drug Safety*, **20**(2), 109–117. DOI : 10/czv6mb.

CAMPOS D., MATOS S. & OLIVEIRA J. L. (2012). Biomedical named entity recognition : A survey of machine-learning tools. In S. SAKURAI, Éd., *Theory and Applications for Advanced Text Mining*, chapitre 8. Rijeka : IntechOpen. DOI : 10.5772/51066.

CARDON R., GRABAR N., GROUIN C. & HAMON T. (2020). Présentation de la campagne d'évaluation deft 2020 : similarité textuelle en domaine ouvert et extraction d'information précise dans des cas cliniques. In *Actes de DEFT*.

FINKEL J. R. & MANNING C. D. (2009). Nested named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, p. 141–150, Singapore : Association for Computational Linguistics.

HONNIBAL M. & MONTANI I. (2017). spacy 2 : Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, **7**(1).

JURAFSKY D. & MARTIN J. H. (2009). *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*. Upper Saddle River, N.J. : Pearson Prentice Hall.

KIM J. Y. & SHAWE-TAYLOR J. (1994). Fast string matching using an n-gram algorithm. *Software : Practice and Experience*, **24**(1), 79–88.

LAFFERTY J. (2001). Conditional random fields : Probabilistic models for segmenting and labeling sequence data. p. 282–289 : Morgan Kaufmann.

LAVERGNE T., CAPPÉ O. & YVON F. (2010). Practical very large scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, p. 504–513 : Association for Computational Linguistics.

TJONG KIM SANG E. F. (2002). Introduction to the CoNLL-2002 shared task : Language-independent named entity recognition. In *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*, COLING-02, p. 1–4, Stroudsburg, PA, USA : Association for Computational Linguistics. DOI : 10.3115/1118853.1118877.

TJONG KIM SANG E. F. & DE MEULDER F. (2003). Introduction to the CoNLL-2003 shared task : Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, p. 142–147.

YADAV V. & BETHARD S. (2018). A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, p. 2145–2158, Santa Fe, New Mexico, USA : Association for Computational Linguistics.