Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Atelier DÉfi Fouille de Textes

Rémi Cardon, Natalia Grabar, Cyril Grouin, Thierry Hamon

6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition) (JEP-TALN-RÉCITAL) [1]

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition).

Atelier DÉfi Fouille de Textes

Rémi Cardon, Natalia Grabar, Cyril Grouin, Thierry Hamon (Éds.)

Nancy, France, 08-19 juin 2020

1. https://jep-taln2020.loria.fr/

Avec le soutien de

Message des présidents de l'AFCP et de l'ATALA

En ce printemps 2020, et les circonstances exceptionnelles qui l'accompagnent, c'est avec une émotion toute particulière que nous vous convions à la 6e édition conjointe des Journées d'Études sur la Parole (JEP), de la Conférence sur le Traitement Automatique des Langues Naturelles (TALN) et des Rencontres des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RECITAL). Après une première édition commune en 2002 (à Nancy, déjà !), et une expérience renouvelée avec succès en 2004, c'est désormais tous les quatre ans (Avignon 2008, Grenoble 2012) que se répète cet événement commun, attendu de pied ferme par les membres des deux communautés scientifiques voisines.

Cette édition 2020 est exceptionnelle, puisque dans le cadre des mesures sanitaires liées à la pandémie mondiale de COVID-19 (confinement strict, puis déconfinement progressif), la conférence ne peut avoir lieu à Nancy comme initialement prévu, mais se déroule à distance, sous forme virtuelle, soutenue par les technologies de l'information et de la communication. Nous remercions ici chaleureusement les organisateurs, Christophe Benzitoun, Chloé Braud, Laurine Huber, David Langlois, Slim Ouni et Sylvain Pogodalla, qui ont dû faire preuve de souplesse, d'inventivité, de détermination, de puissance de travail, et de tant d'autres qualités encore, afin de maintenir la conférence dans ces circonstances, en proposant un format inédit. Grâce aux différentes solutions mises en œuvre dans un délai court, la publication des communications scientifiques est assurée, structurée, et les échanges scientifiques sont favorisés, même à distance.

Bien entendu, nous regrettons tous que cette réunion JEP-TALN-RECITAL ne permette pas, comme ses prédécesseurs, de nouer ou renforcer les liens sociaux entre les différents membres de nos communautés respectives – chercheurs, jeunes et moins jeunes, académiques et industriels, professionnels et étudiants – autour d'une passionnante discussion scientifique ou d'un mémorable événement social... Notre conviction est qu'il est indispensable de maintenir à l'avenir de tels lieux d'échanges dans le domaine francophone, afin bien sûr de permettre aux jeunes diplômés de venir présenter leurs travaux et poser leurs questions sans la barrière de la langue, mais aussi de dynamiser nos communautés, de renforcer les échanges et les collaborations, et d'ouvrir la discussion autour des enjeux d'avenir, qui questionnent plus que jamais la place de la science et des scientifiques dans notre société.

Lors de la précédente édition, nous nous interrogions sur les phénomènes et tendances liés à l'apprentissage profond et sur leurs impacts sur les domaines de la Parole et du TAL. Force est de constater que l'engouement pour ces approches dans nos domaines a permis un retour sur le devant de la scène des domaines liés à l'Intelligence Artificielle, animant parfois un débat tant philosophique que technique sur la place de la machine dans la société, notamment à travers le questionnement sur la vie privée de l'utilisateur. Ces questionnements impactent tant la Parole que le TAL, d'une part sur la place de la gestion des données, d'autre part sur les modèles eux-mêmes. Malgré ces questionnements, nous constatons que les acquis et les expertises perdurent, et les nouvelles approches liées à l'apprentissage profond ont permis un rapprochement des domaines de la Parole et du TAL, sans les dénaturer, à la manière des conférences JEP-TALN-RECITAL qui créent un espace plus grand d'échange et d'enrichissement réciproques.

Nous terminons ces quelques mots d'ouverture en remerciant l'ensemble des personnes qui ont rendu possible cet événement qui restera, nous l'espérons, riche et passionnant, malgré les circonstances. L'ATALA et l'AFCP tiennent tout d'abord à réitérer leurs remerciements aux organisateurs des JEP, de TALN et de RECITAL, qui sont parvenus à maintenir le cap à travers vents et marées. Nos remerciements vont également à l'ensemble des membres des comités de programme, dont le travail et l'implication ont permis de garantir la qualité et la cohérence du programme finalement retenu. Un grand merci aux relecteurs pour le temps et le soin qu'ils ont dédiés à ce travail anonyme et indispensable. Ils se reflètent dans la qualité des soumissions que chacun pourra découvrir sur le site de la conférence.

En conclusion, cette 6e édition conjointe JEP-TALN-RECITAL est exceptionnelle parce qu'elle se tient dans un contexte de crise généralisée — crise sanitaire, économique, voire sociale et politique. Mais nous

formons le vœu qu'elle reste également dans les annales pour la qualité des échanges scientifiques qu'elle aura suscités, et pour le message envoyé à nos communautés scientifiques et à la société dans son ensemble, un message de détermination et de confiance en l'avenir, où la science et les nouvelles technologies restent au service de l'humain.

Véronique Delvaux, présidente de l'Association Francophone de la Communication Parlée
Christophe Servan, président de l'Association pour le Traitement Automatique des LAngues

Table des matières

Présentation de la campagne d'évaluation DEFT 2020 : similarité textuelle en domaine ouvert et extraction d'information précise dans des cas cliniques **1**
Rémi Cardon, Natalia Grabar, Cyril Grouin, Thierry Hamon

Calcul de similarité entre phrases : quelles mesures et quels descripteurs ? **14**
Davide Buscaldi, Ghazi Felhi, Dhaou Ghoul, Joseph Le Roux, Gaël Lejeune, Xudong Zhang

Participation d'EDF R&D à DEFT 2020 **26**
Danrun Cao, Alexandra Benamar, Manel Boumghar, Meryl Bothua, Lydia Ould Ouali, Philippe Suignard

Contextualized French Language Models for Biomedical Named Entity Recognition **36**
Jenny Copara, Julien Knafou, Nona Naderi, Claudia Moro, Patrick Ruch, Douglas Teodoro

Approche supervisée de calcul de similarité sémantique entre paires de phrases **49**
Khadim Dramé, Gorgoumack Sambe, Ibrahima Diop, Lamine Faty

DEFT 2020 - Extraction d'information fine dans les données cliniques : terminologies spécialisées et graphes de connaissance **55**
Thomas Lemaitre, Camille Gosset, Mathieu Lafourcade, Namrata Patel, Guilhem Mayoral

DOING@DEFT : cascade de CRF pour l'annotation d'entités cliniques imbriquées **66**
Anne-Lyse Minard, Andréane Roques, Nicolas Hiot, Mirian Halfeld Ferrari Alves, Agata Savary

Extraction d'information de spécialité avec un système commercial générique **79**
Clothilde Royan, Jean-Marc Langé, Zied Abidi

DEFT 2020 : détection de similarité entre phrases et extraction d'information **91**
Mike Tapi Nzali

Similarité sémantique entre phrases : apprentissage par transfert interlingue **97**
Charles Teissèdre, Thiziri Belkacem, Maxime Arens

Participation de l'équipe du LIMICS à DEFT 2020 **108**
Perceval Wajsbürt, Yoann Taillé, Guillaume Lainé, Xavier Tannier

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e 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

Présentation de la campagne d'évaluation DEFT 2020 : similarité textuelle en domaine ouvert et extraction d'information précise dans des cas cliniques

Rémi Cardon[1, 2] Natalia Grabar[1, 2] Cyril Grouin[3] Thierry Hamon[3, 4]
(1) CNRS, UMR 8163, F-59000 Lille, France
(2) Univ. Lille, UMR 8163 – STL – Savoirs Textes Langage, F-59000 Lille, France
(3) Université Paris-Saclay, CNRS, LIMSI, F-91400 Orsay, France
(4) Université Paris 13, Sorbonne Paris Cité, F-93430, Villetaneuse, France
prenom.nom@univ-lille.fr, prenom.nom@limsi.fr

RÉSUMÉ

L'édition 2020 du défi fouille de texte (DEFT) a proposé deux tâches autour de la similarité textuelle et une tâche d'extraction d'information. La première tâche vise à identifier le degré de similarité entre paires de phrases sur une échelle de 0 (le moins similaire) à 5 (le plus similaire). Les résultats varient de 0,65 à 0,82 d'EDRM. La deuxième tâche consiste à déterminer la phrase la plus proche d'une phrase source parmi trois phrases cibles fournies, avec des résultats très élevés, variant de 0,94 à 0,99 de précision. Ces deux tâches reposent sur un corpus du domaine général et de santé. La troisième tâche propose d'extraire dix catégories d'informations du domaine médical depuis le corpus de cas cliniques de DEFT 2019. Les résultats varient de 0,07 à 0,66 de F-mesure globale pour la sous-tâche des pathologies et signes ou symptômes, et de 0,14 à 0,76 pour la sous-tâche sur huit catégories médicales. Les méthodes utilisées reposent sur des CRF et des réseaux de neurones.

ABSTRACT

Presentation of the DEFT 2020 Challenge : open domain textual similarity and precise information extraction from clinical cases

The 2020 edition of the French Text Mining Challenge proposed two tasks about textual similarity and one information extraction task. The first task aims at identifying the degree of similarity between pairs of sentences, from 0 (the less similar) to 5 (the most similar). The results range from 0.65 to 0.82 (EDRM). The second task consists in identifying the closest sentence from a source sentence among three given sentences. The results are very high, ranging from 0.94 to 0.99 (precision). Both tasks rely on a corpus from the general and health domains. The third task proposes to extract ten categories of information from the medical domain, from a corpus of clinical cases used during the last competition. The results range from 0.07 to 0.66 (F-measure) on the sub-task identifying pathologies and signs or symptoms, and from 0.14 to 0.76 for the sub-task concerning eight medical categories. Methods used rely on CRF and neural networks.

MOTS-CLÉS : Cas cliniques, extraction d'information, similarité textuelle.

KEYWORDS: Clinical Cases ; Information Extraction ; Textual Similarity.

Atelier DÉfi Fouille de Textes, pages 1–13.
HAL : hal-02784737.

1 Introduction

Cet article présente la campagne d'évaluation 2020 du défi fouille de texte (DEFT). Cette nouvelle édition se compose de trois tâches, dont deux nouvelles portant sur la similarité entre phrases et l'identification de phrases parallèles en domaine général et domaine de spécialité, tandis que la troisième tâche reprend le corpus de cas cliniques utilisé pour DEFT 2019 (Grabar *et al.*, 2019) et propose de travailler sur des catégories d'informations plus fines.

La similarité textuelle offre plusieurs possibilités d'utilisation, de la détection du plagiat jusqu'à la réécriture dans un but de simplification. Dans cette campagne, nous avons souhaité étudier les niveaux de similarité qu'il est possible d'inférer automatiquement, entre phrases portant soit sur le même sujet, soit des sujets fondamentalement différents.

L'extraction d'informations précises contenues dans des documents de spécialité, tels que les cas cliniques, constitue une première étape pour l'accès à l'information, la recherche de cas similaires, ou encore le peuplement de bases de données. A l'image de la campagne d'évaluation américaine i2b2/VA 2010, nous avons proposé pour le français une tâche d'extraction d'informations pour le domaine médical.

Nous avons lancé le défi le 27 janvier. Treize équipes se sont inscrites. Dix équipes ont participé à la phase de test, qui s'est déroulée du 28 au 30 avril. Parmi les équipes étant allé au terme de la campagne, nous retenons la présence d'une équipe académique africaine (Université Assane Seck Ziguinchor, Sénégal) et de cinq équipes industrielles (EDF R&D, Palaiseau ; Reezocar, Boulogne-Billancourt ; Synapse, Toulouse) ou mixtes industrielles et académiques (IBM France, Bois-Colombes et Université Paris 7, Paris ; LIRMM et ONAOS, Montpellier). Quatre équipes académiques complètent la liste des participants (BiTeM, Haute Ecole Spécialisée, Genève ; DOING, Université d'Orléans ; LIMICS, Sorbonne Université, Paris ; LIPN, Sorbonne Paris-Nord, Villetaneuse). La diffusion des corpus d'entraînement s'est faite à partir du 3 février pour la première tâche, à partir du 13 février pour la troisième tâche et du 17 février pour la deuxième tâche, pour des raisons de finalisation des corpus. Enfin, une version améliorée des annotations de la troisième tâche a été distribuée le 20 mars.

2 Présentation

Le corpus utilisé pour les deux premières tâches se compose de pages Wikipedia et Vikidia [1] relatives à différents sujets (par exemple, les pages Almaty, Apiculture, Biberon, Boris Godounov, etc.) ainsi que du contenu en santé tel que des notices de médicaments (Bromazepam, Buprénorphine, etc.) et des résumés Cochrane (Grabar & Cardon, 2018).

2.1 Tâche 1 – Degré de similarité entre paires de phrases

La première tâche consiste à identifier le degré de similarité entre deux phrases, sur une échelle de valeurs comprises entre 0 (le moins similaire) et 5 (le plus similaire), sans que la sémantique associée à chaque valeur de cette échelle n'ait été définie. Cinq personnes ont annoté les paires de phrases du corpus, chacune ayant son interprétation personnelle du type de contenu associé à chaque degré dans

1. Vikidia est la Wikipedia destinée aux 8–13 ans, https://fr.vikidia.org/wiki/Vikidia:Accueil

la mesure où nous n'avons pas souhaité fournir de définition des degrés de similarité. Pour constituer les valeurs de référence, nous avons retenu la valeur issue du vote majoritaire (Cardon & Grabar, 2020). Le corpus d'entraînement intègre 600 paires de phrases tandis que le corpus d'évaluation contient 410 paires. Le tableau 1 fournit quelques exemples de paires de phrases source et cible avec le degré de similarité issu de l'annotation humaine.

Phrases source et cible	Degré
Il commence par s'intéresser à la résistance à la faim, la soif et à la fatigue en 1951.	0
Pour prouver qu'on pouvait vivre sans eau ni nourriture, il traversa en solitaire l'Atlantique sans autres ressources que les poissons, le plancton, l'eau de pluie et de petites quantités d'eau de mer durant 65 jours.	
En cas de survenue d'une hypotension importante, le patient doit être mis en décubitus dorsal, et recevoir, si nécessaire, une perfusion iv de chlorure de sodium.	1
Si une hypotension importante se produit, elle peut être combattue en allongeant le patient jambes relevées.	
Deux essais (106 participants) comparaient l'héparine de bas poids moléculaire à un placebo ou à l'absence de traitement.	2
Deux essais (259 participants) comparaient l'héparine à l'absence de traitement.	
La vieille ville est entourée de remparts, érigés au XIIIe siècle, très appréciés par les promeneurs.	3
La ville haute, ceinte de remparts, est très pittoresque.	
Le biberon, (du latin bibere, « boire ») ou bouteille en Suisse, est un ustensile utilisé pour l'allaitement artificiel.	4
Un biberon (une bouteille, en Suisse), est un outil permettant d'allaiter un bébé artificiellement, ou naturellement, si la mère a tiré son lait.	
Le médecin spécialisé pratiquant la neurologie s'appelle le neurologue.	5
Le médecin qui pratique la neurologie est le neurologue.	

TABLE 1 – Degré de similarité pour quelques paires de phrases source et cible de la tâche 1

Le tableau 2 présente le nombre et le pourcentage d'annotations pour chaque degré de similarité dans les corpus d'entraînement et d'évaluation. On observe que le degré de similarité le plus faible (0) est celui qui contient le plus d'annotations dans les deux corpus (plus du tiers du nombre total d'annotations), suivi de l'avant-dernier degré (4), couvrant déjà plus de la moitié des paires du corpus.

Corpus	Degrés de similarité											
	0		1		2		3		4		5	
Entraînement (600 paires)	216	36,0%	56	9,3%	29	4,8%	66	11,0%	136	22,7%	97	16,2%
Evaluation (410 paires)	147	35,9%	37	9,0%	28	6,8%	44	10,7%	90	22,0%	64	15,6%

TABLE 2 – Nombre et pourcentage d'annotations par degré de similarité dans les corpus de la tâche 1

Bien qu'aucune définition n'ait été fournie, les annotateurs humains ont sensiblement convergé vers

les observations suivantes : degré 5 pour une similarité quasi parfaite, degré 4 si l'une des deux phrases apporte une information de plus, degré 3 si une information importante est manquante, degrés 2 ou 1 en fonction du niveau de reformulation, et degré 0 pour une absence de similarité ou trop complexe. Nous observons que les participants du défi ont eu une interprétation similaire de ces degrés de similarité, comme celle indiquée par Cao *et al.* (2020) pour l'équipe EDF R&D.

2.2 Tâche 2 – Identification des phrases parallèles

La deuxième tâche vise à identifier, parmi trois phrases cibles, celle qui correspond le mieux à la phrase source en terme de phrase parallèle. Une réponse parmi les trois phrases cibles fournies est toujours attendue, les participants ont donc l'obligation d'identifier une phrase parallèle pour chaque ensemble de phrases source et cibles. Le parallélisme des phrases est lié à la relation simple-compliqué : la phrase source correspond au contenu compliqué alors que les phrases simples contiennent le contenu simple ou simplifié. L'une des phrases simples est donc dérivée de la phrase compliquée. La tableau 3 fournit des exemples de phrases source et cibles sur différents sujets. On observe ainsi que, dans certains cas, la même phrase a été utilisée comme phrase source et comme l'une des phrases cibles (deuxième exemple), et que dans d'autres cas, des indices numériques tels que les dates ne correspondent pas forcément (troisième exemple). Le corpus d'entraînement comprend 572 ensembles de phrases source et cibles tandis que le corpus d'évaluation en compte 530.

Type	Phrases proposées
Source	Arrivé en France en 1972, ce chat reste méconnu en dehors de son pays d'origine.
Cibles	Les principaux matériaux sont le grès et la latérite.
	Ce chat est apparu dans une portée d'American Shorthairs, en 1966, dans l'État de New-York.
	Bien qu'il soit apparu en France dès 1972, ce chat reste méconnu hors de son État d'origine.
Source	en suède, le taux légal est de 0,2 g par litre de sang
Cibles	en suisse, le taux légal est de 0,5 g/l de sang ou 0,22 mg d'alcool par litre d'air expiré, depuis 2005
	en suède, le taux légal est de 0,2 g par litre de sang
	en belgique, le taux légal est de 0,5 g/l de sang ou 0,22 mg d'alcool par litre d'air expiré
Source	En 1534, il est appelé comme maître d'œuvre par le comte Giangiorgio Trissino pour diriger le chantier de la villa Cricoli.
Cibles	*En 1537, il est appelé par le comte Giangiorgio Trissino pour diriger le chantier de la villa Cricoli.*
	Trissino est un humaniste, poète, philosophe et diplomate au service de la curie romaine (le gouvernement pontifical) ; c'est aussi un passionné d'architecture .
	Le théâtre Olympique, achevé après 1580, est l'œuvre ultime de Palladio, terminée après sa mort par son fils Silla et son disciple Scamozzi.

TABLE 3 – Exemples de phrases sources et cibles pour la tâche 2. La phrase cible la plus parallèle de la phrase source apparaît en italiques

2.3 Tâche 3 – Extraction d'information fine

Présentation Dans la continuité de DEFT 2019, qui portait sur l'analyse de cas cliniques, une sous-partie du corpus utilisé [2] (Grabar *et al.*, 2018) a été annotée avec des catégories d'informations médicales fines autour de quatre domaines [3] : autour des patients *(anatomie)*, de la pratique clinique *(examen, pathologie, signe ou symptôme)*, des traitements médicamenteux et chirurgicaux *(dose, durée, fréquence, mode d'administration, substance, traitement, valeur)*, et autour du temps *(date, moment)* (Grouin *et al.*, 2019). Les annotations de pathologies et de signes ou symptômes couvrent aussi bien des mots isolés que de longues portions textuelles (jusqu'à 33 mots dans une seule portion [4]), alors que les annotations des autres catégories sont généralement plus courtes (entre un et quatre mots). De plus, les annotations d'examens, de pathologies et de signes ou symptômes peuvent englober des annotations d'autres catégories (telles que des parties anatomiques ou des valeurs numériques). Nous avons complété ces annotations avec l'information sur les assertions des pathologies et signes ou symptômes *(présent, absent, possible, hypothétique, non-associé)*, de la norme des valeurs d'examens biologiques et physiques *(normal, haut, bas)* et de la prise des traitements et substances *(arrêt, reprise)*. Ces catégories d'annotation s'inspirent de celles utilisées en 2010 dans la campagne d'évaluation internationale i2b2/VA (Uzuner *et al.*, 2011). Seules dix catégories [5] d'annotations ont été utilisées dans cette campagne. Nous présentons un extrait de corpus annoté sur la figure 1.

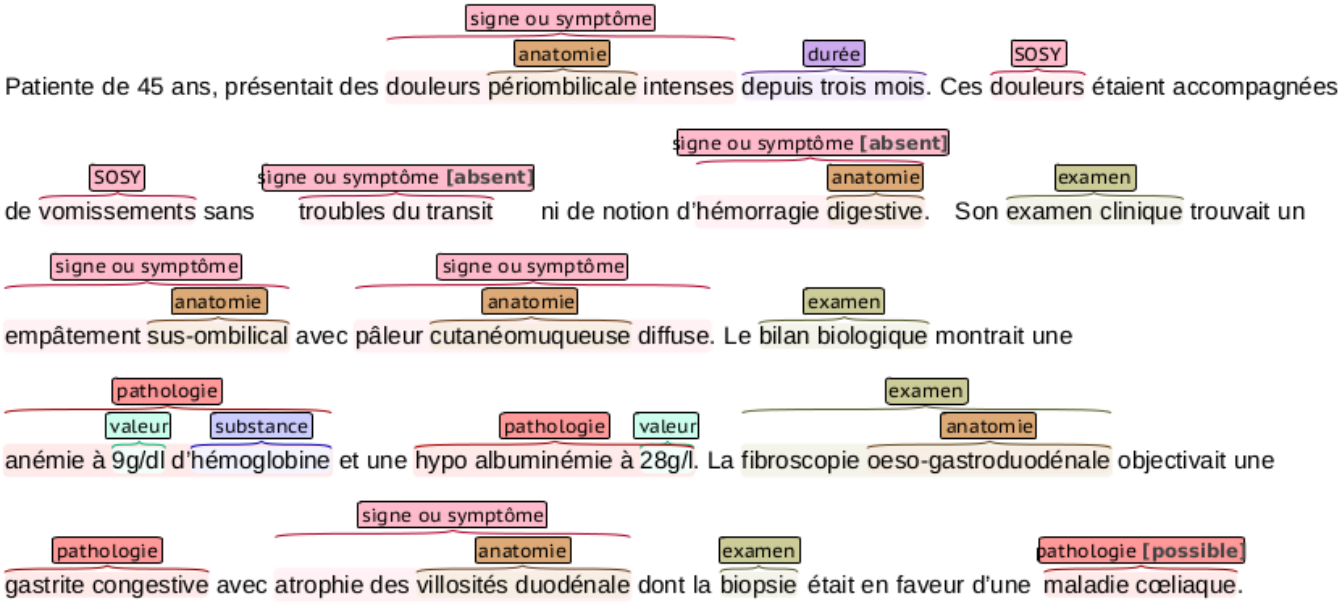


FIGURE 1 – Extrait de corpus annoté. Les boîtes renvoient aux catégories, les éléments entre crochets aux informations complémentaires

Le corpus d'entraînement comprend 100 fichiers (soit 7844 annotations sur les dix catégories) tandis que le corpus d'évaluation compte 67 fichiers (4738 annotations). Le tableau 4 renseigne du nombre et du pourcentage d'annotations par catégorie dans les corpus d'entraînement et d'évaluation, pour

2. Nous avons aléatoirement extrait 167 fichiers parmi les 717 fichiers du corpus utilisé l'année dernière.
3. Le guide d'annotation est accessible à l'adresse : https://deft.limsi.fr/2020/guide-deft.html
4. La portion suivante a intégralement été annotée comme un signe ou symptôme : « *volumineuse masse pelvienne de 170 sur 150 x 120 mm, de contours polylobés, de densité spontanément hétérogène, se rehaussant de façon intense et hétérogène après injection, avec des zones d'hypodensité, quasi liquidiennes* » car renvoyant à une description complète d'une observation.
5. Anatomie, dose, examen, mode, moment, pathologie, signe ou symptôme, substance, traitement, valeur.

les catégories utilisées dans cette édition de la campagne DEFT.

Catégories utilisées	Corpus			
	Entraînement (100 fichiers)		Evaluation (67 fichiers)	
Anatomie	1454	18,5%	1121	23,7%
Dose	380	4,9%	52	1,1%
Examen	1219	15,5%	817	17,2%
Mode	225	2,9%	89	1,9%
Moment	453	5,8%	165	3,5%
Pathologie	368	4,7%	166	3,5%
Signe ou symptôme	1799	22,9%	1279	27,0%
Substance	1016	13,0%	313	6,6%
Traitement	383	4,9%	304	6,4%
Valeur	547	7,0%	432	9,1%

TABLE 4 – Nombre et pourcentages d'annotations par catégorie dans les corpus de la tâche 3

Annotation humaine Deux annotateurs ont réalisé ce travail en utilisant l'interface d'annotation BRAT (Stenetorp *et al.*, 2012). Compte tenu de l'ampleur du travail d'annotation à réaliser (au total, 8615 entités seront annotées dans le corpus d'entraînement et 4933 dans le corpus d'évaluation), nous avons opté pour une annotation séquentielle du corpus d'entraînement. D'abord composé de 70 fichiers, ce corpus a été annoté par un premier annotateur puis corrigé par le deuxième annotateur. Trente fichiers supplémentaires ont été annotés par le deuxième annotateur, puis corrigés par le premier annotateur. En conséquence, il n'est pas possible de calculer d'accord inter-annotateur sur les fichiers du corpus d'entraînement. Néanmoins, nous avons mesuré l'écart avant et après correction, et les résultats obtenus [6] nous ont confortés dans la possibilité de poursuivre avec les choix effectués, en ajustant les règles du guide d'annotation au besoin.

L'annotation des 67 fichiers du corpus d'évaluation s'est faite sur la base d'une pré-annotation automatique, réalisée grâce à un modèle CRF entraîné sur les cent fichiers du corpus d'entraînement. Les annotateurs humains ont corrigé cette pré-annotation de manière indépendante, puis réalisé une phase d'adjudication. Malgré la pré-annotation automatique, les accords inter-annotateur calculés sur ces fichiers se montent à 0,67 de F-mesure en évaluation stricte et 0,80 en évaluation souple [7] sur les treize catégories annotées par les humains. Les catégories pour lesquelles les humains ont obtenu de moins bonnes performances sont les pathologies, les signes ou symptômes, ainsi que les fréquences et modes d'administration. Les différences observées concernent des choix différents de catégories ainsi que des oublis. Nous observons également que les deux annotateurs ont pu choisir des mots différents pour annoter le même concept [8]. Enfin, l'absence de formation médicale des annotateurs peut également présenter un obstacle dans la qualité du travail d'annotation. Par exemple, la différence entre pathologie et signe ou symptôme reste complexe. Les annotateurs ont notamment

6. Nous calculons une F-mesure stricte globale de 0,8183 et une F-mesure souple de 0,8946 sur les 70 premiers fichiers, et une F-mesure stricte de 0,9565 et une F-mesure souple de 0,9781 sur les trente derniers fichiers.

7. L'évaluation stricte repose sur un appariement exact entre frontières et étiquette. Une évaluation souple accepte quelques caractères d'écart au niveau des frontières.

8. Pour le mode d'administration, un annotateur aura annoté le mot *injecte* alors que l'autre aura annoté le mot *seringue* de la même phrase : « L'infirmière anesthésiste prépare la seringue de morphine et la remet à l'anesthésiste qui l'injecte. »

considéré que les tumeurs bénignes sont des signes ou symptômes, alors que les tumeurs malignes sont des pathologies. Certains mots ou suffixes sont également des indices importants pour déterminer la catégorie [9]. Concernant la taille des portions, les annotateurs ont établi qu'une portion doit contenir le maximum d'informations se rapportant à la même idée. Ce choix explique la taille de certaines portions annotées en signes ou symptômes.

Une fois ce travail d'annotation terminé, nous sommes revenus sur les annotations du corpus d'entraînement pour le rendre plus homogène [10] avec le corpus d'évaluation. Cette deuxième version du corpus d'entraînement a été distribuée aux participants le 20 mars.

3 Evaluation

Tâche 1 Les degrés de similarité de la tâche 1 renvoient à une graduation sur une échelle de six valeurs. Nous avons donc retenu comme mesure principale la distance relative moyenne à la solution, calculée en micro-moyenne. A chacune des six valeurs possibles pour la donnée de référence r_i, correspond une valeur de distance maximale possible entre la réponse du système et cette donnée $dmax(h_i, r_i)$. Ainsi, si le degré de similarité attendu est 5, la distance maximale possible est alors maximale ($5 - 0 = 5$), mais si le degré attendu est 2, la distance maximale possible sera de 3 ($5 - 2 = 3$ alors que $2 - 0 = 3$). L'exactitude en distance relative à la solution moyenne (EDRM) se calcule en micro-moyenne comme indiqué dans l'équation 1.

$$\text{EDRM} = \frac{1}{n}\sum_{i=1}^{n}\left(1 - \frac{d(h_i, r_i)}{dmax(h_i, r_i)}\right) \tag{1}$$

Tâche 2 Sur cette tâche, il est possible d'évaluer, soit la meilleure réponse ramenée par le système (la phrase cible la plus similaire à la phrase source), soit le classement ou l'ordre des trois phrases cibles proposées, qui auront été ramenées de la plus similaire à la moins similaire. Pour évaluer un classement, la moyenne des précisions non interpolées $P(I_i^j)$ calculées à chaque position, dans la liste des hypothèses, d'une des n_i réponses correctes I_i^j pour la phrase source S_i, est alors une mesure pertinente. Sur l'ensemble des phrases source et cible, nous utilisons alors la moyenne de cette précision moyenne (MAP, voir formule 2). Dans le cas où un système ne classerait pas toutes les phrases cibles fournies, et notamment la phrase cible attendue, sa précision moyenne est alors nulle, comme si elle avait été classée à l'infini par le système.

$$\text{MAP} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{n_i}\sum_{j=1}^{n_i}P(I_i^j) \tag{2}$$

Si une seule phrase cible est renvoyée pour chaque phrase source, une précision classique est alors employée.

9. Le suffixe *-mégalie* qui renvoie à un gonflement *(hépatomégalie, splénomégalie)* permet d'annoter un signe ou symptôme, tandis que les *(adéno)carcinomes* et *adénopathies* seront annotés en pathologie.

10. Cette nouvelle version s'accompagne d'un nombre plus élevé d'annotations dans les catégories *anatomie* (de 1454 à 1608), *signe ou symptôme* (de 1799 à 1831) et *valeurs* (de 547 à 588) pour les plus fortes hausses. Dans le même temps, certaines annotations de la catégorie *traitement* ont été supprimées (passant de 383 à 374 annotations).

Tâche 3 La dernière tâche est une tâche de repérage d'entités nommées. Les mesures habituelles de précision, rappel et F-mesure, calculées sur chaque catégorie et au niveau global, constituent le mode d'évaluation le plus pertinent et le mieux compris du point de vue des valeurs calculées.

4 Résultats

4.1 Tâche 1 – Degré de similarité entre paires de phrases

Le tableau 5 présente les résultats en EDRM (mesure officielle) ainsi que la corrélation de Spearman et la p-value sur les prédictions de la première tâche. Sur l'ensemble des soumissions, la moyenne est de 0,7617 et la médiane se situe à 0,7947.

Soumission	EDRM	Corrélation de Spearman	p-value
EDF R&D, 1	0,8198	0,7305	1,3963e-69
EDF R&D, 2	0,8018	0,7105	2,9216e-64
EDF R&D, 3	0,8069	0,7164	9,2243e-66
Reezocar, 1	0,7919	0,7060	4,1583e-63
Reezocar, 2	0,8105	0,7352	6,6003e-71
Reezocar, 3	0,8022	0,7080	1,2883e-63
Sorbonne, 1	0,7092	0,7485	8,4089e-75
Sorbonne, 2	0,6734	0,7321	5,2500e-70
Sorbonne, 3	0,8147	0,7479	1,2845e-74
Synapse, 1	0,6533	0,7499	3,1295e-75
Synapse, 2	0,6663	0,7421	7,0960e-73
Synapse, 3	0,6838	0,7679	6,1899e-81
UASZ, 1	0,7947	0,7528	4,3371e-76
UASZ, 2	**0,8217**	0,7691	2,3769e-81
UASZ, 3	0,7755	0,7769	5,5766e-84

TABLE 5 – Evaluation des prédictions en EDRM. Le meilleur résultat est en gras

Nous avons utilisé le test de Student pour évaluer la significativité statistique des résultats des participants à la tâche 1 (voir figure 2). De manière globale, nous observons que les résultats des équipes EDF R&D, Sorbonne et REEZOCAR n'ont pas de différence significative entre eux, tandis que ceux de l'équipe Synapse diffèrent statistiquement de ceux de tous les autres participants. Aussi, les résultats de l'équie UASZ sont significativement différents de la deuxième soumission de l'équipe Sorbonne et des soumissions de l'équipe REEZOCAR. Enfin, la troisième soumission de l'équipe UASZ est éloignée de l'ensemble des soumissions. De plus, nous pouvons remarquer plusieurs spécificités des soumissions. La meilleure soumission (UASZ_2) n'est pas significativement différente des résultats des équipes EDF R&D, Sorbonne (à l'exception de la deuxième soumission), et de la troisième soumission de l'équipe REEZOCAR. De même, il n'y a pas de différences significatives entre les soumissions d'une même équipe, à l'exception de celles de Synapse. Ceci peut s'expliquer par le fait que les soumissions sont toutes basées sur la même méthode et ne diffèrent qu'à travers des paramétrages particuliers. En revanche, les variations significatives dans les soumissions (1 et 3 vs. 2) de Synapse peuvent certainement s'expliquer par l'utilisation des modèles BERT ou MUSE.

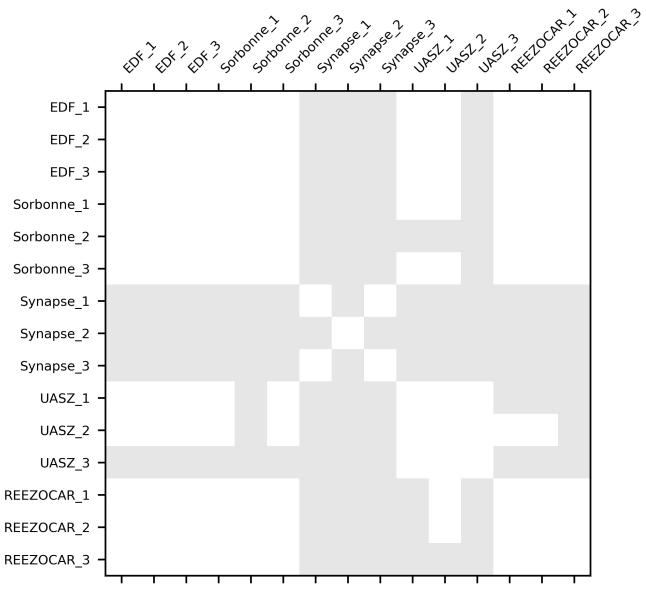


FIGURE 2 – Significativité statistique des soumissions de la tâche 1. Les zones grisées représentent une différence significative entre deux soumissions.

Méthodes Sur cette tâche, les méthodes utilisées par les participants sont variées. Plusieurs reposent sur des calculs de distance entre phrases (distances euclidiennes, Jaccard, Manhattan), avec des représentations de type TF×IDF, notamment par l'équipe Sorbonne (Buscaldi *et al.*, 2020) qui a mesuré l'intérêt et l'absence d'intérêt de cette représentation, dont ces distances peuvent ensuite servir de traits pour des algorithmes d'apprentissage (régression logistique, forêt d'arbres, graphes sémantiques, etc.), approche qui a été suivie par les équipes EDF R&D (Cao *et al.*, 2020), Reezocar (Tapi Nzali, 2020) et UASZ (Drame *et al.*, 2020). Cette dernière équipe obtient les meilleures performances de la tâche avec sa deuxième soumission fondée sur un perceptron multi-couche. D'autres participants ont utilisés des modèles de plongements lexicaux multilingues dérivés de BERT, en particulier Sentence M-BERT et MUSE par Synapse (Belkacem *et al.*, 2020).

4.2 Tâche 2 – Identification des phrases parallèles

Le tableau 6 présente les résultats sur la deuxième tâche, évalués avec une précision classique. Sur l'ensemble des soumissions, la moyenne est de 0,9822 et la médiane se situe à 0,9868. Les résultats ne présentent pas de différence significative lorsqu'on utilise un test de Student.

Soumission	EDF R&D		Reezocar			Sorbonne			Synapse		
	1	2	1	2	3	1	2	3	1	2	3
Précision	0,9830	0,9868	0,9868	0,9811	0,9849	0,9887	0,9887	0,9887	**0,9906**	0,9849	0,9396

TABLE 6 – Evaluation des prédictions en précision. Le meilleur résultat est en gras

Méthodes Sur cette tâche, des coefficients de similarité ont été employés, tel que le coefficient de Dice par EDF R&D (Cao *et al.*, 2020) ou de plusieurs distances (euclidienne, Manhattan, Minkovski)

par Buscaldi *et al.* (2020) pour l'équipe Sorbonne. Des approches à base de représentations vectorielles plus lourdes ont également été essayées, telle que USE (Universal Sentence Encoder) par Cao *et al.* (2020), ou DRMM (Deep Relevance Matching Model) et MUSE (Multilingual Universal Sentence Encoder) entraînés sur des corpus disponibles en interne et ensuite fournis dans des classifieurs à base de descente stochastique de gradient (SGD) ou de gradient boosting extrême (XGB) par les équipes Reezocar (Tapi Nzali, 2020) et Synapse (Belkacem *et al.*, 2020).

4.3 Tâche 3 – Extraction d'information fine

L'évaluation de la tâche d'extraction d'information est divisée en deux sous-tâches, en fonction de la taille des portions à traiter : une première sous-tâche pour les signes ou symptômes et pathologies (tableau 7), en raison de leur complexité, et une deuxième sous-tâche pour les huit autres catégories (tableau 8). Les résultats ont été calculés au moyen de l'outil BRATeval en évaluation stricte. Sur l'ensemble des soumissions (sauf le premier run de l'équipe Lirmm/Onaos qui ne contenait volontairement aucune prédiction pour cette sous-tâche), la moyenne est de 0,4618 et la médiane se monte à 0,4706 pour la première sous-tâche, tandis que pour la deuxième sous-tâche, la moyenne est de 0,6012 et la médiane s'élève à 0,6151. Notons que, sur ce corpus d'évaluation, les accords inter-annotateur calculés en F-mesure stricte entre les deux annotateurs humains se sont élevés à 0,460 sur les pathologies et 0,470 sur les signes ou symptômes.

Soumission	GLOBAL			Pathologie			Sosy		
	P	R	F	P	R	F	P	R	F
Doing, 1	0,568	0,484	0,523	0,571	0,361	0,443	0,568	0,500	0,532
Doing, 2	0,577	0,493	0,531	0,505	0,337	0,404	0,584	0,513	0,546
Doing, 3	0,532	0,446	0,486	0,434	0,319	0,368	0,543	0,463	0,500
EDF R&D, 1	0,137	0,042	0,065	0,137	0,368	0,199	0,000	0,000	0,000
HESGE, 1	0,576	0,439	0,498	**0,613**	0,295	0,398	0,573	0,458	0,509
HESGE, 2	0,609	0,622	0,615	0,492	0,584	0,534	0,627	0,627	0,627
HESGE, 3	**0,702**	**0,624**	**0,660**	0,575	0,554	0,564	**0,720**	**0,633**	**0,673**
IBM, 1	0,495	0,376	0,427	0,429	0,398	0,413	0,506	0,373	0,430
IBM, 2	0,448	0,419	0,433	0,345	0,416	0,377	0,466	0,419	0,441
Limics, 1	0,422	0,305	0,354	0,264	0,277	0,271	0,454	0,308	0,367
Limics, 2	0,609	0,576	0,592	0,428	**0,645**	0,514	0,649	0,567	0,605
Limics, 3	0,660	0,574	0,614	0,512	0,633	**0,566**	0,689	0,567	0,623
Lirmm/Onaos, 1	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
Lirmm/Onaos, 2	0,342	0,294	0,316	0,199	0,416	0,270	0,397	0,278	0,327
Reezocar, 1	0,525	0,372	0,436	0,489	0,386	0,431	0,531	0,371	0,437
Reezocar, 2	0,456	0,331	0,383	0,426	0,259	0,322	0,459	0,340	0,391
Reezocar, 3	0,536	0,397	0,456	0,480	0,440	0,459	0,545	0,391	0,455

TABLE 7 – Evaluation stricte globale et par catégorie (pathologie, signe ou symptôme) en précision, rappel et F-mesure sur la sous-tâche des longues portions. Les meilleurs résultats sont en gras

Méthodes Pour cette tâche, les champs aléatoires conditionnels (CRF de chaîne linéaire) ont majoritairement été employés, notamment par les équipes Doing (Minard *et al.*, 2020) avec des

Soumission	GLOBAL			Anat	Dose	Exam	Mode	Momt	Subs	Trait	Val
	P	R	F	F	F	F	F	F	F	F	F
Doing, 1	0,831	0,606	0,701	0,732	0,347	0,723	0,531	0,726	0,625	0,496	0,802
Doing, 2	**0,839**	0,613	0,708	0,736	0,347	0,731	0,540	0,726	0,643	0,520	0,802
Doing, 3	0,785	0,585	0,670	0,688	0,317	0,701	0,588	0,686	0,580	0,519	0,769
EDF R&D, 1	0,415	0,314	0,358	0,251	0,286	0,465	0,442	0,263	0,432	0,275	0,468
HESGE, 1	0,781	0,507	0,615	0,556	0,368	0,679	0,342	0,627	0,578	0,476	0,797
HESGE, 2	0,737	0,737	0,737	0,798	0,412	0,715	0,639	0,758	0,670	0,557	0,839
HESGE, 3	0,788	0,725	0,755	**0,807**	0,522	0,733	**0,649**	**0,787**	0,638	0,608	**0,856**
IBM, 1	0,743	0,339	0,466	0,152	0,164	0,670	0,527	0,529	0,534	0,510	0,526
IBM, 2	0,695	0,573	0,628	0,662	0,154	0,649	0,587	0,563	0,563	0,438	0,752
Limics, 1	0,659	0,567	0,610	0,694	0,296	0,619	0,453	0,520	0,428	0,406	0,699
Limics, 2	0,764	**0,749**	0,756	0,758	**0,547**	0,802	0,561	0,688	0,769	0,649	0,819
Limics, 3	0,795	0,733	**0,763**	0,763	0,539	**0,805**	0,569	0,701	**0,786**	**0,659**	0,815
Lirmm/Onaos, 1	0,414	0,081	0,135	0,000	0,274	0,000	0,286	0,345	0,000	0,000	0,442
Lirmm/Onaos, 2	0,627	0,508	0,561	0,616	0,245	0,575	0,518	0,000	0,579	0,417	0,671
Reezocar, 1	0,679	0,511	0,583	0,664	0,275	0,470	0,584	0,586	0,599	0,378	0,750
Reezocar, 2	0,681	0,505	0,580	0,653	0,192	0,477	0,561	0,552	0,588	0,380	0,761
Reezocar, 3	0,678	0,530	0,595	0,678	0,260	0,469	0,592	0,633	0,595	0,390	0,768

TABLE 8 – Evaluation stricte globale en précision, rappel et F-mesure, et par catégorie (anatomie, dose, examen, mode, moment, substance, traitement, valeur) en F-mesure uniquement sur la sous-tâche des courtes portions. Les meilleurs résultats sont en gras

modèles CRF pour chaque catégorie, les équipes HESGE (Copara *et al.*, 2020) et Reezocar (Tapi Nzali, 2020). Les modèles de reconnaissance d'entités nommées de l'outil SpaCy fondés sur des réseaux neuronaux convolutifs (CNN) ont également été testés par les équipes Doing et EDF R&D. Les modèles de langue de l'anglais (BERT, BioBERT, RoBERTa), et ceux adaptés au français tels que CamemBERT, dont une version pré-entraînée sur des données biomédicales issues de PubMed, ont également été utilisés par plusieurs équipes dont HESGE et le LIMICS (Wajsbürt *et al.*, 2020) qui a utilisé des versions entraînées sur des corpus multilingues de données web (OSCAR et CCNET) utilisés dans un bi-LSTM avec une couche finale de CRF.

Nous relevons que certaines équipes industrielles ont mis à profit la campagne d'évaluation DEFT pour appliquer sur ces corpus des ressources créées à partir de données métier avec l'outil SpaCy pour l'équipe EDF R&D (Cao *et al.*, 2020), ou d'outils internes tel que WKS (Watson Knowledge Studio) de la suite IBM Watson, algorithme fondé sur l'entropie maximale, pour l'équipe IBM France (Royan *et al.*, 2020). Enfin, l'équipe Lirmm/Onaos (Lemaitre *et al.*, 2020) a utilisé un système à base de règles fondé sur des ressources de la base JeuxDeMots ou du domaine médical. Plusieurs équipes ont témoigné de la difficulté à distinguer les catégories pathologie et signe ou symptôme d'une part, ce que nous reconnaissons pour avoir eut des difficultés à les annoter lors de la préparation des corpus, et à gérer les imbrications d'entités d'autre part.

La complexité des annotations fournies a également poussé l'équipe IBM France a réannoter le corpus en incluant notamment des relations entre entités (localisation, temporalité, précision, etc.) pour exclure les imbrications d'entités. Des règles de post-traitement sont ensuite utilisées pour faire correspondre ces annotations à celles attendues dans la campagne DEFT.

5 Conclusion

L'édition 2020 du défi fouille de texte a proposé trois tâches. Malgré sa complexité, la tâche d'extraction d'informations fine a permis l'obtention des résultats corrects sur les deux catégories d'information les plus difficiles (pathologies et signes ou symptômes), alors même que ces deux catégories ont posé problème lors du travail d'annotation humaine. De plus, les résultats ne dépassaient que rarement les 0,8 de F-mesure sur des catégories a priori plus simples (doses, moments, valeurs). La diversité des contenus dans chacune des catégories semble expliquer ces résultats.

La tâche d'identification de la phrase la plus similaire d'une phrase source parmi trois cibles fournies a permis à l'ensemble des participants d'obtenir d'excellents résultats, qui varient de 0,94 à 0,99 de précision. Le corpus fourni pour cette deuxième tâche semble manifestement facile à traiter de manière automatique. Par contre, lorsqu'il s'agit d'attribuer un degré de similarité sur une échelle à six valeurs, la tâche paraît plus complexe. L'absence volontaire de définition du contenu de chaque degré de similarité et le nombre relativement élevé de degrés disponibles (six degrés de 0 à 5) réduisent les chances de succès.

Références

BELKACEM T., TEISSEDRE C. & ARENS M. (2020). Similarité Sémantique entre Phrases : Apprentissage par Transfert Interlingue. In *Actes de DEFT*, Nancy, France.

BUSCALDI D., FELHI G., GHOUL D., LE ROUX J., LEJEUNE G. & ZHANG X. (2020). Calcul de similarité entre phrases : quelles mesures et quels descripteurs ? In *Actes de DEFT*, Nancy, France.

CAO D., BENAMAR A., BOUMGHAR M., BOTHUA M., OULD-OUALI L. & SUIGNARD P. (2020). Participation d'EDF R&D à DEFT 2020. In *Actes de DEFT*, Nancy, France.

CARDON R. & GRABAR N. (2020). A French corpus for semantic similarity. In *LREC 2020*, p. 1–12.

COPARA J., KNAFOU J., MORO C., RUCH P. & TEODORO D. (2020). Contextualized French Language Models for Biomedical Named Entity Recognition. In *Actes de DEFT*, Nancy, France.

DRAME K., SAMBE G., DIOP I. & FATY L. (2020). Approche supervisée de calcul de similarité sémantique entre paires de phrases. In *Actes de DEFT*, Nancy, France.

GRABAR N. & CARDON R. (2018). CLEAR-Simple Corpus for Medical French. In *Proc of ATA*, Tilburg, The Netherlands. HAL : halshs-01968355.

GRABAR N., CLAVEAU V. & DALLOUX C. (2018). CAS : French Corpus with Clinical Cases. In *Proc of LOUHI*, p. 122–128, Brussels, Belgium. DOI : 10.18653/v1/W18-5614.

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. HAL : hal-02280852.

GROUIN C., GRABAR N., HAMON T. & CLAVEAU V. (2019). Clinical Case Reports for NLP. In *Proc of BioNLP*, Florence, Italy. DOI : 10.18653/v1/W19-5029.

LEMAITRE T., GOSSET C., LAFOURCADE M., PATEL N. & MAYORAL G. (2020). DEFT 2020 – Extraction d'information fine dans les données cliniques : terminologies spécialisées et graphes de connaissance. In *Actes de DEFT*, Nancy, France.

MINARD A.-L., ROQUES A., HIOT N., ALVES M. H. F. & SAVARY A. (2020). DOING@DEFT : cascade de CRF pour l'annotation d'entités cliniques imbriquées. In *Actes de DEFT*, Nancy, France.

ROYAN C., LANGÉ J.-M. & ABIDI Z. (2020). Extraction d'information de cas cliniques avec un système commercial générique. In *Actes de DEFT*, Nancy, France.

STENETORP P., PYYSALO S., TOPIĆ G., OHTA T., ANANIADOU S. & TSUJII J. (2012). brat : a Web-based Tool for NLP-Assisted Text Annotation. In *Proc of EACL*, p. 102–107, Avignon, France.

TAPI NZALI M. (2020). DEFT 2020 : détection de similarité entre phrases et extraction d'information. In *Actes de DEFT*, Nancy, France.

UZUNER O., SOUTH B. R., SHEN S. & DUVALL S. L. (2011). 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *J Am Med Inform Assoc*, **18**(5), 552–556. DOI : 10.1136/amiajnl-2011-000203.

WAJSBÜRT P., TAILLÉ Y., LAINÉ G. & TANNIER X. (2020). Participation de l'équipe du LIMICS à DEFT 2020. In *Actes de DEFT*, Nancy, France.

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e 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

Calcul de similarité entre phrases : quelles mesures et quels descripteurs ?

Davide Buscaldi[1], Ghazi Felhi[1], Dhaou Ghoul[2],
Joseph Le Roux[1], Gaël Lejeune[2] Xudong Zhang[1]
(1) Sorbonne Paris Nord, LIPN, 99 Avenue Jean Baptiste Clément, 93430 Villetaneuse
(2) Sorbonne Université, 1 rue Victor Cousin, 75005 Paris, France
(1)prenom.nom@lipn.univ-paris13.fr,
(2) prenom.nom@sorbonne-universite.fr

RÉSUMÉ

Cet article présente notre participation à l'édition 2020 du Défi Fouille de Textes DEFT 2020 et plus précisément aux deux tâches ayant trait à la similarité entre phrases. Dans notre travail nous nous sommes intéressé à deux questions : celle du choix de la mesure du similarité d'une part et celle du choix des opérandes sur lesquelles se porte la mesure de similarité. Nous avons notamment étudié la question de savoir s'il fallait utiliser des mots ou des chaînes de caractères (mots ou non-mots). Nous montrons d'une part que la similarité de Bray-Curtis peut être plus efficace et surtout plus stable que la similarité cosinus et d'autre part que le calcul de similarité sur des chaînes de caractères est plus efficace que le même calcul sur des mots.

ABSTRACT

Sentence Similarity : a study on similarity metrics with words and character strings

This article details the participation of the Sorbonne team, composed of researchers from Sorbonne Paris Nord (LIPN lab) and Sorbonne University (STIH lab) to the 2020 Deft challenge. We participated in the two tasks involving similarity measurement. We have been interested in two questions : first of all choosing the appropriate similarity measure and secondly choosing the appropriate features to construct the vectors. We show that (I) the Bray-Curtis similarity can be more efficient and more stable than a classical cosine distance and (II) that character n-grams tend to be more efficient for similarity tasks without needing fine-tuning or data description (lemmatization ...).

MOTS-CLÉS : similarité, n-grammes de caractères, distance euclidienne, distance de Bray-Curtis.

KEYWORDS: similarity, character n-grams, euclidean distance, Bray-Curtis distance.

1 Introduction

Cette édition 2020 du défi Fouille de Textes était principalement consacrée aux données médicales et comprenait trois tâches : (I) identification du degré de similarité entre paires de phrases (parallèles et non parallèles), (II) identification des phrases parallèles possible pour une phrase source dans le domaine médical et (III) extraction d'information sur des cas cliniques dans des textes biomédicaux. Les détails sur le processus de collecte et d'annotation des données dans l'article introducteur du défi (Cardon *et al.*, 2020).

Atelier DÉfi Fouille de Textes, pages 14–25.
HAL : hal-02784738.

Notre travail s'est concentré sur les tâches 1 et 2 que nous avons traité sous l'angle des mesures de similarité. Nous avons conçu une architecture extrêmement simple, que l'on peut sans doute qualifier de *baseline* améliorée, exploitant des mesures de similarité sur des vecteurs. Notre contribution est de réfléchir d'une part aux bonnes manières de construire ces vecteurs, quelles caractéristiques ou dit autrement quelles opérandes, et d'autre part sur les meilleures manières de comparer ces représentations, la recherche en quelque sorte des bons opérateurs.

Dans la Section 2 nous présenterons quelques grandes lignes des approches possibles en calcul de similarité puis dans la Section 3 nous exposerons la méthode que nous avons développé pour ce défi et enfin dans la Section 4 nous présenterons les résultats obtenus et quelques éléments de discussion.

2 Approches en calcul de similarité

Le calcul de similarité est une tâche essentielle du Traitement Automatique de données, textuelles ou non, et a toujours reçu une attention particulière de la communauté scientifique. Si l'on se restreint aux données textuelles, le besoin est avant tout venu de besoin pour la recherche d'information. Il s'agit en effet de pouvoir d'une part de mesurer le degré de proximité entre des documents et d'autre part de pouvoir identifier, et ordonner, la liste des documents les plus pertinents à offrir en réponse à une requête dans un moteur de recherches.

Assez classiquement, le défi posé ici est d'identifier les observables permettant de représenter numériquement des documents et de choisir les mesures appropriées pour en déduire la meilleure mesure de similarité pour une tâche donnée. Les données textuelles sont représentées informatiquement parlant comme des séquences de caractères, sans représentation linguistique autre l'ordre des caractères dans la séquence. Dès lors pouvoir comparer deux chaînes autrement que par une opération de recherche d'identité stricte impose d'identifier des caractéristiques internes de ces chaînes qui vont permettre d'identifier des opérandes sur lesquelles la comparaison pourra porter, ce choix n'étant pas sans impact sur les résultats (Mehdad & Tetreault, 2016). Les opérandes peuvent être de deux grandes catégories : (I) les sous-chaînes de caractères elles mêmes, les formes brutes, dont les mots graphiques sont un sous-ensemble, et (II) les redescriptions, généralement calculées à partir des mots, par exemple la racinisation ou la lemmatisation. Le type de redescription utilisé affectera le qualificatif que l'on va donner à la similarité calculée : si la redescription encode des propriétés syntaxiques on parlera plus facilement de similarité syntaxique, si elle encode des propriétés sémantiques alors on parlera de similarité sémantique… Ensuite, pour pouvoir appliquer des opérateurs de comparaison on aura deux grands types d'approches pour la similarité : d'une part la recherche de similarités séquentielles calculées par comparaison, sous-chaînes communes ou encore distance d'édition, et d'autre part la vectorisation qui permet de se placer dans un cadre méthodologique bien adapté au calcul automatique.

3 Quelle mesures de similarité et quels descripteurs ?

Notre approche était fondée sur une représentation assez simple du problème : que pouvaient donner des mesures de similarité très simples appliquées sur la tâche 2, tâche la plus simple puisqu'elle consistait à extraire la phrase la plus proche parmi trois candidates. L'approche la plus immédiate, en tout cas celle qui nous a paru comme telle, a été de calculer une vectorisation en mots et d'appliquer

une simple mesure de similarité cosinus pour classer les phrases candidates. Les premiers résultats étaient très élevés (au-delà de 93% de bons appariements) ce qui laissait à penser que la tâche était assez aisée. Nous avions envisagé d'exploiter des méthodes sophistiquées à base de plongements de mots, spécialisés ou non sur le domaine médical, mais il nous a semblé qu'il serait intéressant de partir de cette *baseline* plutôt convaincante. Plusieurs auteurs se sont intéressés à cette question de la capacité des *baseline*, pour peu qu'on leur porte suffisamment d'attention, à avoir des résultats équivalents (Moreno & Dias, 2014) voire supérieurs (Rendle *et al.*, 2019) à des approches état de l'art. C'est aussi une question qui s'est posé régulièrement dans la communauté DEFT, par exemple lorsque la *baseline* conçue par les organisateurs du Défi 2019 s'est avérée plus performante que des approches plus complexes implantées par les participants du défi (Grabar *et al.*, 2019). La question scientifique est tout à fait intéressante puisqu'il s'agit aussi de mesurer la valeur ajoutée apportée par des méthodes sophistiquées qui sont souvent plus gourmandes en ressources : taille des jeux de données d'entraînement, disponibilité de données linguistiques (lexiques, plongements de mots ...) pour une langue et/ou un domaine donné ou encore tout simplement coût en temps de calcul (Strubell *et al.*, 2019).

Dès lors, nos investigations se sont portées sur deux objectifs : d'une part chercher dans la méthode elle même quelles pouvaient être les points d'amélioration et d'autre part regarder si cette méthode pouvait être adaptée pour traiter également la tâche 1. Nous avons donc exploré la question des opérations appliquées, les mesures de similarités (Section 3.1) d'une part et les opérandes sur lesquelles les mesures allaient porter, mots ou chaînes de caractères, d'autre part (Section 3.2.

3.1 Choix des mesures de similarité

Différentes mesures de similarité peuvent être utilisées pour rapprocher des documents représentés sous forme vectorielle. Nous avons cherché à comparer différentes mesures de similarité s'appuyant sur des vecteurs. Dans tous ces cas, on va comparer des phrases représentées sous la forme de deux vecteurs V et W, V représentant la phrase à apparier et W représentant chacun des candidats tout à tour. Assez naturellement, on va se tourner vers la distance euclidienne mais il peut être intéressant de se tourner vers des variantes telles que la distance de Manhattan ou la distance de Minkovski. Pour rappel, on peut réécrire la distance de Manhattan pour la faire passer de la forme :

$DistManh = \sum_{i=1}^{n} |x_i - y_i|$

à la forme : $DistManh = \sqrt[1]{\sum_{i=1}^{n} (x_i - y_i)^2}$

De sorte que la filiation avec la distance de Minkovski et la distance euclidienne devienne plus évidente :

$DistEucl = \sqrt[2]{\sum_{i=1}^{n} (x_i - y_i)^2}$ et $DistMink = \sqrt[3+]{\sum_{i=1}^{n} (x_i - y_i)^3+}$

Avec les mesures précitées, les segments les plus proches sont ceux qui minimisent la distance. Nous intégrons ensuite des mesures de similarité plus classiques en TAL dont la similarité cosinus, l'indice de Jaccard et le coefficient de Dice qui sont souvent considérés comme une des mesures de référence quand il est question de similarité textuelle (Huang, 2008). Nous avons également intégré la dissimilarité de Bray-Curtis (Bray & Curtis, 1957).

— $Cosinus = \frac{V.W}{||V||.||W||}$
— $Jaccard = \frac{|set(V) \cap set(W)|}{|set(V) \cup set(W)|}$

— $Dice = \frac{2*|set(V) \cap set(W)|}{|set(V) \cup set(W)|}$

— $Bray - Curtis = \frac{2\sum_{i=1}^{n} min(V[i], W[i])}{\sum_{i=1}^{n} (V[i]+W[i])}$

Nous pouvons donc d'ores et déjà tester une première *baseline* qui va exploiter ces distances et similarités en travaillant simplement sur les effectifs des mots graphiques, sans pré-traitement ni élimination de *stop-words*. La tokenisation est effectuée par un simple découpage sur les espaces. Nous avons appliqué cette *baseline* sur les 572 instances du jeu d'apprentissage de la tâche 2 (Cardon *et al.*, 2020). Parmi les trois candidats proposés on choisit celui qui présente la similarité la plus grande (ou la distance la plus faible pour les trois premières mesures décrites ci-dessus) sans seuil d'aucune sorte [1].

Les résultats sont présentés dans le tableau 1, nous pouvons voir deux choses : d'une part la tâche est assez facile et d'autre le choix de la mesure de similarité peut avoir son importance.

	Bons résultats	MAP
Distance Euclidienne	534/572	0,9336
Distance de Minkowski	534/572	0,9336
Distance de Manhattan	536/572	0,9371
Similarité Cosinus	553/572	0,9668
Coefficient de Dice	553/572	0,9668
Similarité de Bray-Curtis	557/572	0,9738
Indice de Jaccard	559/572	0,9772

TABLE 1 – Résultats de l'application des mesures de similarité, au grain mot sans pré-taritement ni pondération, sur la tâche 2 triés par ordre croissant de MAP

Afin d'enrichir légèrement et à faible coût la représentation, nous montrons dans le tableau 2 les résultats obtenus avec des représentations en n-grammes de mots en prenant différents intervalles de valeur de N de 1 à 4. Les résultats en haut à gauche de chaque sous-tableau ($N_{min} = N_{max} = 1$) correspondent donc aux valeurs de la *baseline* du tableau 1. Nous pouvons voir que tenir compte des bi-grammes de mots permet d'améliorer les résultats, même si pris individuellement les bi-grammes offrent une représentation moins dense et moins efficace que les unigrammes. Par contre dans cette configuration l'impact des tri-grammes n'est pas significativementpositif. Nous proposns ensuite dans le tableau 3 les résultats avec une pondération tf-idf, où l'Idf est calculé en prenant compte de l'ensemble du corpus d'apprentissage. Nous avons laissé les résultats avec l'indice de Jaccard à titre d'illustration puisqu'ils ne semble pas pertinent de combiner tf-idf et indice de Jaccard. Nous pouvons voir que la pondération tf-idf ne permet pas d'améliorer les résultats. Il y a certainement plusieurs raisons derrière cela : le fait que le nombre de doucments dans le corpus est peut être trop petit pour que le Tf-Idf puisse offrir une plus-value, peut être que l'utilisation d'Okapi BM-25 améliorerait les résultats comme cela avait pu être montré par (Claveau, 2012) mais ce n'était pas le cas ici. On peut penser aussi que la tâche étant simple, cette *baseline* atteignait tout simplement un plafond de verre. Nous montrerons dans la section suivante qu'il n'en est rien, en travaillant sur des n-grammes de caractères nous arrivons à améliorer encore un peu les résultats.

1. Pas de seuil de score similarité minimale ou d'écart minimal de similarité entre deux candidats

	max=1	max=2	max=3	max=4
min=1	0.9773	**0.979**	0.9773	0.9738
min=2		0.958	0.9545	0.9563
min=3			0.9126	0.9108
min=4				0.8374

(a) Indice de Jaccard

	max=1	max=2	max=3	max=4
min=1	0.9668	**0.972**	0.965	0.965
min=2		0.9545	0.9545	0.9545
min=3			0.9126	0.9108
min=4				0.8374

(b) Coefficient de Dice

	max=1	max=2	max=3	max=4
min=1	0.9668	**0.972**	0.9668	0.9633
min=2		0.9545	0.9545	0.951
min=3			0.9126	0.9108
min=4				0.8374

(c) Similarité cosinus

	max=1	max=2	max=3	max=4
min=1	0.9738	**0.9773**	0.9755	0.972
min=2		0.958	0.9563	0.9563
min=3			0.9126	0.9108
min=4				0.8374

(d) Similarité de Bray-Curtis

TABLE 2 – Résultats en MAP, sans pondération, sur les données d'apprentissage avec des n-grammes de mots et différents intervalles de longueur de 1 à 4

	max=1	max=2	max=3	max=4
min=1	0.4073	0.4091	0.4021	0.4056
min=2		0.4353	0.4283	0.4161
min=3			**0.4371**	0.4196
min=4				0.4266

(a) Indice de Jaccard

	max=1	max=2	max=3	max=4
min=1	0.9633	**0.9668**	0.9633	0.9615
min=2		0.9563	0.9528	0.9493
min=3			0.9091	0.9091
min=4				0.8374

(b) Coefficient de Dice

	max=1	max=2	max=3	max=4
min=1	0.9685	**0.972**	0.9668	0.9668
min=2		0.9545	0.9563	0.9563
min=3			0.9143	0.9108
min=4				0.8374

(c) Similarité cosinus

	max=1	max=2	max=3	max=4
min=1	0.9738	**0.9773**	0.972	0.972
min=2		0.958	0.9563	0.9563
min=3			0.9143	0.9143
min=4				0.8392

(d) Similarité de Bray-Curtis

TABLE 3 – Résultats en MAP, avec pondération Tf-Idf, sur les données d'apprentissage avec des n-grammes de mots et différents intervalles de longueur de 1 à 4 (NB : résultats avec l'indice de Jaccard, pour information)

3.2 Choix des descripteurs : mots ou chaînes de caractères

La dimension que nous avons souhaité examiner ensuite a donc été le choix des des descripteurs, en d'autres termes le choix des opérandes sur lesquelles allait porter le calcul de similarité. Une piste naturelle, et employée par d'autres participants du défi, était certainement d'avoir recours à des représentation plus riches des mots de manière notamment à mieux encoder la synonymie et plus généralement la proximité sémantique. Ceci pouvait prendre la forme d'une racinisation, d'une lemmatisation ou de l'exploitation de plongement de mots. Ici nous avons choisi une méthode qui se rapproche dans une certaine mesure de la racinisation mais qui permet aussi d'encoder des relations séquentielles entre les mots : l'utilisation de n-grammes de caractères.

L'idée, que l'on peut résumer sous la forme « Tout ce que nous savons faire avec des mots, nous

devrions pouvoir le faire avec des chaînes de caractères »[2] (Umemura & Church, 2009), est double : d'une part rechercher les limites des analyses au grain mot d'un point de vue efficacité et d'autre part d'un point de vue plus épistémologique interroger la pertinence de chercher systématiquement à travailler à un grain d'analyse "interprétable" tel que le mot alors que l'utilisation des chaînes de caractères est plus naturelle pour la machine et sachant que la tokenisation n'est pas une tâche de TAL réglée à l'heure actuelle dans tous les contextes. Ce qui va amener à chercher à standardiser les textes comme on le fait souvent pour les *tweets*(Nebhi *et al.*, 2015), les textes bruités (issus d'océrisation par exemple) ou encore les textes anciens (Gabay *et al.*, 2019).

Pour examiner cela nous avons simplement utilisé les mêmes mesures de similarité mais en les appliquant cette fois sur des vecteurs de N-grammes de caractères (Figure 1). Nous pouvons voir que les résultat sont systématiquement supérieurs à ceux obtenus au grain mot dès lors que la représentation inclut les 2-grammes de caractères et ceci reste vrai quel que soit le calcul de similarité utilisé. On peut observer que les résultats avec l'indice de Jaccard augmentent très peu lorsque l'on augmente la taille des n-grammes de caractères. Cela ne semble pas très étonnant puisque le fait de binariser les valeurs des dimensions (0 ou 1) va lisser très fortement l'effet de redondance amené par les n-grammes de caractères. Par exemple pour la chaîne `tototo`, les 4-grammes `toto` et `otot` vont tout deux être représentés de la même manière que le 5-gramme `totot`. Les résultats obtenus avec le coefficient de Dice sont notablement moins stables mais on observe un pic, supérieur à ce que l'on voit avec l'indice de Jaccard, avec $N_{min} = 4$ et $6 \leq N_{max} \leq 8$. En moyenne les résultats avec la similarité Cosinus sont supérieurs mais le pic est moins élevé. Enfin, les résultats obtenus avec la distance de Bray-Curtis, s'ils ne montraient pas le même pic que le coefficient de Dice, offraient selon nous le meilleur compromis entre stabilité et efficacité puisque l'on pouvait s'abstenir de définir un seuil minimal.

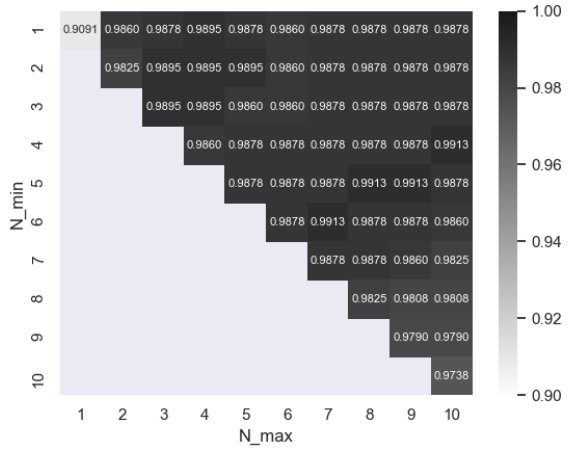
3.3 Passage de la tâche 2 à la tâche 1 et choix des run

Pour passer de la tâche 2 à la tâche 1, nous avons choisi tout simplement de retravailler les scores de similarités nativement normalisés[3] : Jaccard, Dice, Cosinus et Bray-Curtis). Nous transformons le score de similarité en un vote de la façon suivante : $vote = int(Sim * 5)$. Il est évident que des optimisations étaient possibles mais nous avons souhaité conserver la simplicité de l'approche *baseline*.

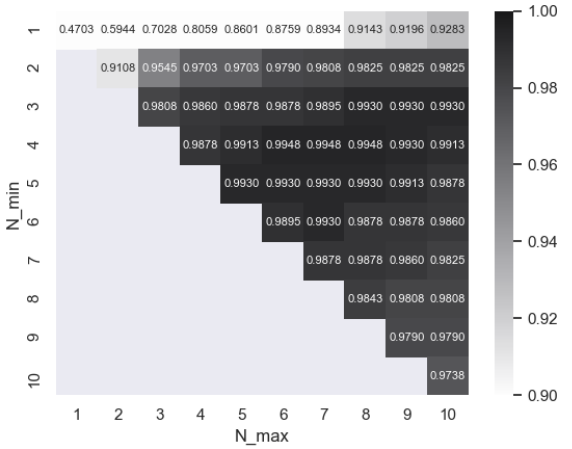
Nous pouvons voir dans la Figure 2 les résultats obtenus avec des n-grammes de caractères sur la tâche 1. La distance de Bray Curtis offre là encore des résultats plus stables même si le meilleur résultat est obtenu avec la distance cosinus. Nous avons tout de même choisi de conserver les configurations optimales que nous avions identifié pour la tâche 2 afin de limiter l'aspect *fine tuning* des valeurs N_{min} et N_{max}. Les configurations choisies sont présentées dans le tableau 4.

Le `run3` est un classifieur SVM à noyau RBF qui utilise comme caractéristiques pour chaque phrase à apparier les résultats de tous les systèmes utilisant les distances Bray Curtis et Cosinus, systèmes qui étaient apparus comme les plus complémentaires. Ce système de vote a obtenu des résultats suivants en validation croisée (10 strates) sur le jeu d'entraînement : 0,99 de MAP en moyenne sur la tâche 2, et 0,73 d'EDRM en moyenne sur la tâche 1.

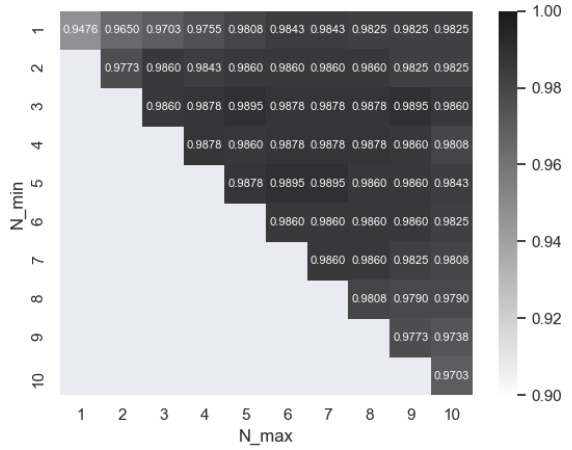
2. *Anything we can do with words we ought to be able to do with substrings"*
3. Nous n'avons a pas exploré la normalisation des autres distances du fait que leurs résultats sur la tâche 2 étaient significativement moins bons



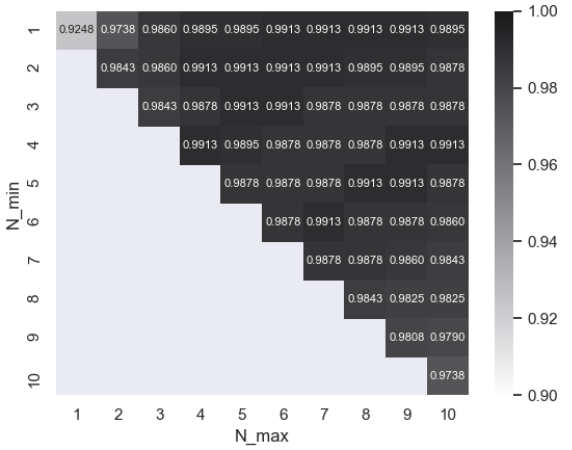
(a) Indice de Jaccard



(b) Coefficient de Dice



(c) Distance cosinus

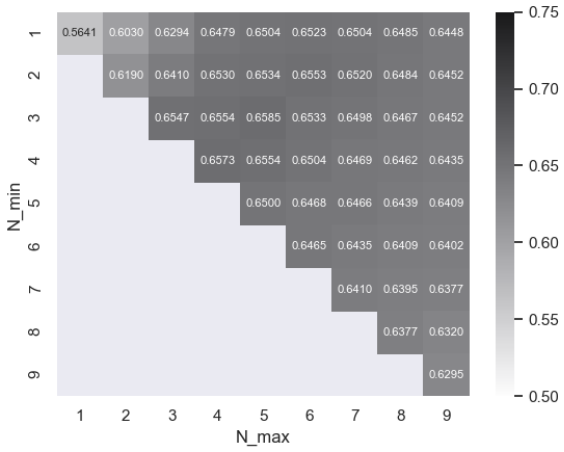


(d) Distance de Bray-Curtis

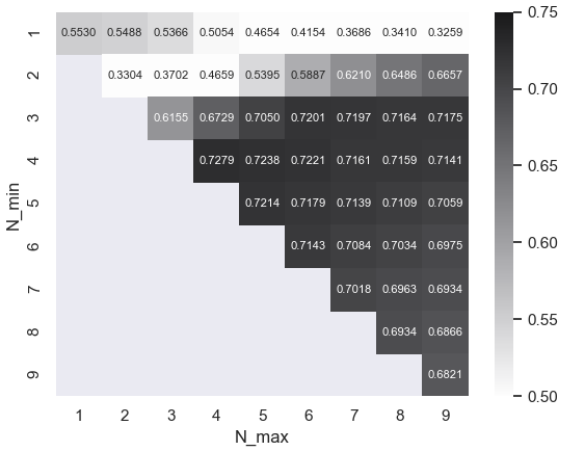
FIGURE 1 – Tâche 2 : résultats (MAP) avec des n-grammes de caractères sur le jeu d'apprentissage

	Distance utilisée	Opérandes utilisées	Tâche1	Tâche 2
run1	Cosinus	n-grams de 3 à 5	min(dist)	int(5*Sim)
run2	Bray-Curtis	n-grams de 1 à 10	min(dist)	int(5*Sim)
run3	Cosinus+Bray-Curtis(*bagging*)	Toutes	SVM radial	SVM radial

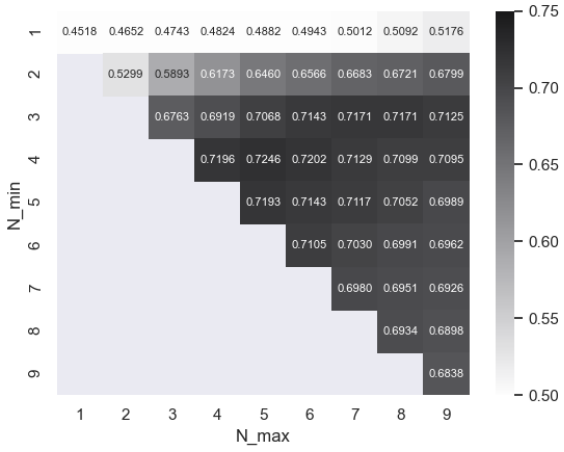
TABLE 4 – Configuration des runs soumis



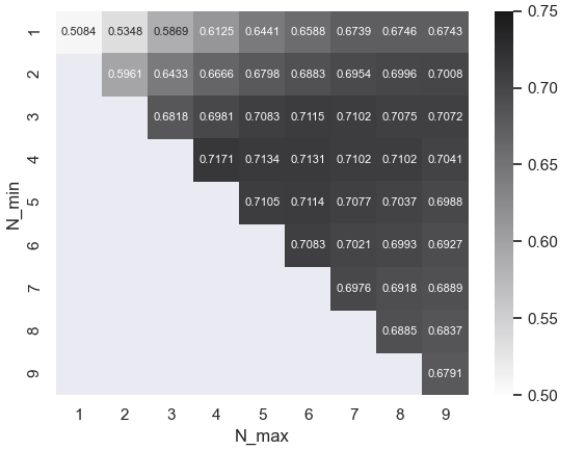
(a) Indice de Jaccard



(b) Coefficient de Dice



(c) Distance cosinus



(d) Distance de Bray-Curtis

FIGURE 2 – Tâche 1 : résultats (EDRM) avec des n-grammes de caractères sur le jeu d'apprentissage

4 Résultats et Discussion

4.1 Résultats officiels

Les tableaux 5 et 6 présentent nos résultats officiels sur les tâches 1 et 2. Les méthodes que nous avons proposé ont été moins performantes sur la tâche 1, avec un `run1` et un `run2` en dessous ou au niveau de la moyenne. Mais, le système de vote (`run3`) a offert une valeur ajoutée très importante à nos résultats sur cette tâche, $+10pp.$ par rapport au `run1`. Ce gain est plus grande que celui que nous avions observé sur le jeu de données d'entraînement. Sur la tâche 2, nos résultats sont globalement meilleurs, entre la médiane et le maximum des résultats soumis. Il est à noter que nos trois `run` ont donné un score strictement égal bien que les fichiers de résultats soient différents.

minimum : 0,653	médiane : 0,795	maximum : 0,822
run1 : 0,709	run2 : 0,673	run3 : 0,815

TABLE 5 – Évaluation officielle de nos trois runs sur la tâche 1 (EDRM) et comparaison avec le minimum, le maximum et la médiane (moyenne des soumissions : 0,762)

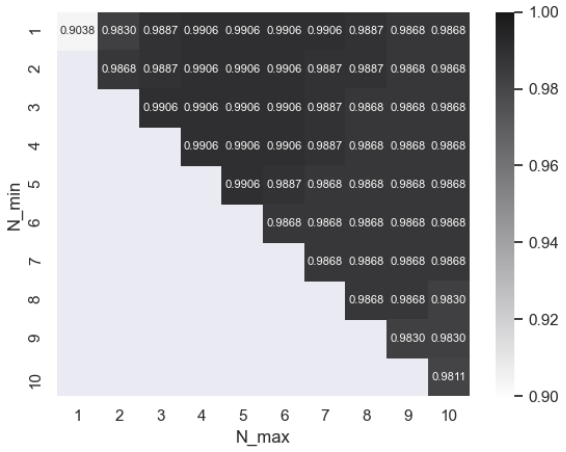
minimum : 0,9396	médiane : 0,9868	maximum : 0,9906
`run1` : 0,9887	`run2` : 0,9887	`run3` : 0,9887

TABLE 6 – Évaluation officielle de nos trois runs sur la tâche 1 (MAP) et comparaison avec le minimum, le maximum et la médiane (moyenne des soumissions : 0,9822)

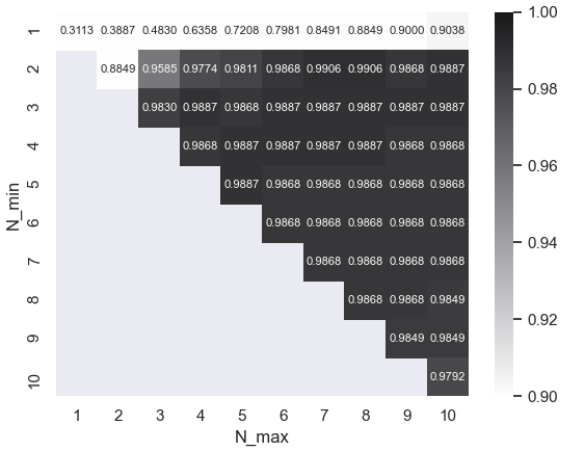
4.2 Variation des résultats sur les données de test

La figure 3 montre les résultats obtenus sur le jeu de test de la tâche 2 en faisant varier les mesures de similarité ainsi que la taille des n-grammes de caractères. Nous pouvons observer que les résultats sont très stables. Le score obtenu par nos 3 `runs` (0,9887) étant trouvé avec de nombreuses configurations ce qui correspond à 524 bons résultats sur 530. Plusieurs configurations amènent un résultat de 0,9906 ce qui correspond au meilleur système répertorié (1 instance bien appariée de plus que ce que nous avons soumis). Enfin, nous avons plusieurs cas avec la distance de Bray Curtis où nous obtenons un score meilleur de 0,9925 ce qui correspond à 526 bons résultats.

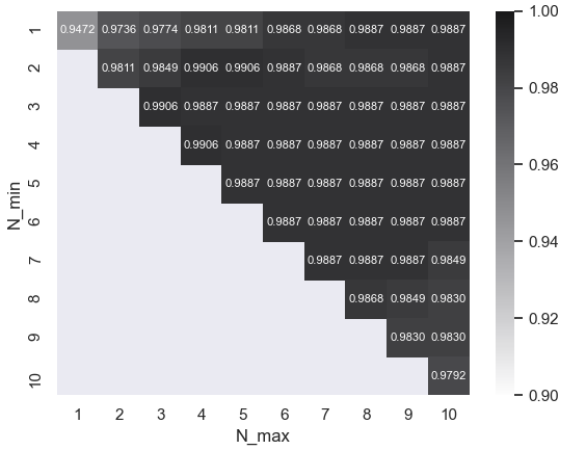
Sur la tâche 1, les méthodes `run1` et `run2` se sont avérées éloignées des meilleurs résultats. De fait, les variations sur les mesures de similarité ou la taille des n-grammes de caractères n'apportaient qu'un bénéfice somme toute relatif comme nous le montrons dans la figure 4. Les résultats plafonnent autour de 0,72 d'EDRM avec une pointe à 0,7294 avec le coefficient de Dice et des N-grammes de taille 4 à 5. Il apparaît que ces différentes variations étaient assez complémentaires ce qui explique pourquoi le système de *bagging* utilisé pour le `run3` a pu apporter 10 points de pourcentage de mieux que le meilleur système soumis (`run1` et 8 points de mieux que le meilleur résultat enregistré (coefficient de Dice et des N-grammes de taille 4 à 5).



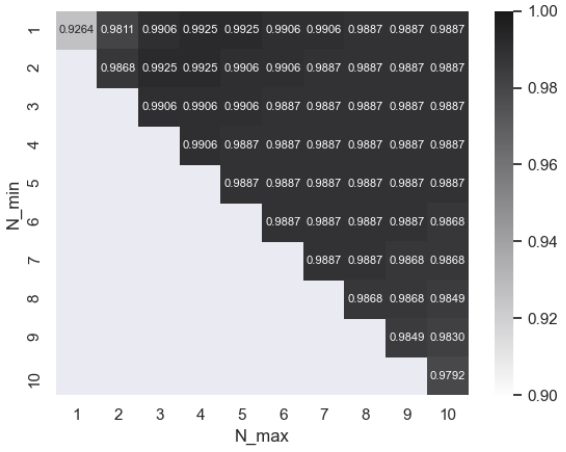
(a) Indice de Jaccard



(b) Coefficient de Dice

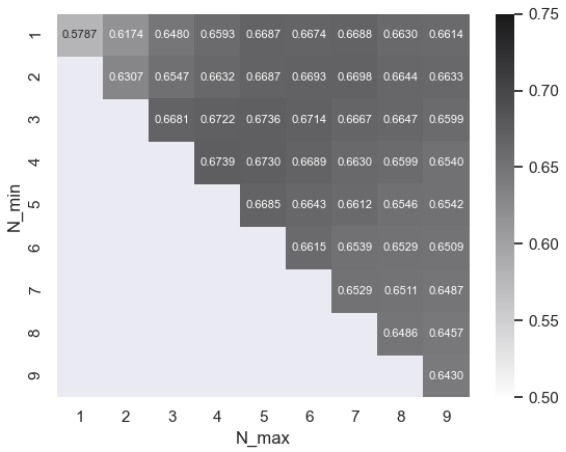


(c) Distance cosinus

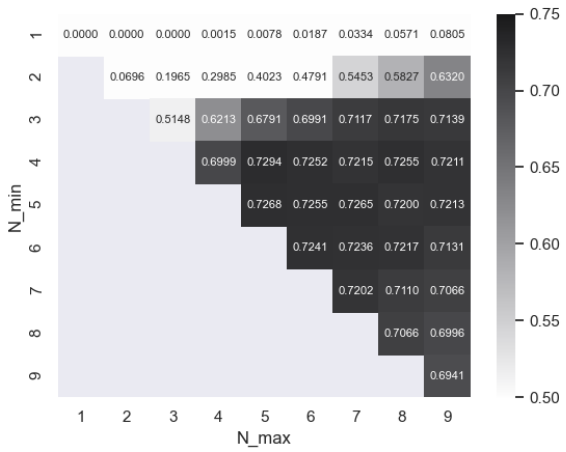


(d) Distance de Bray-Curtis

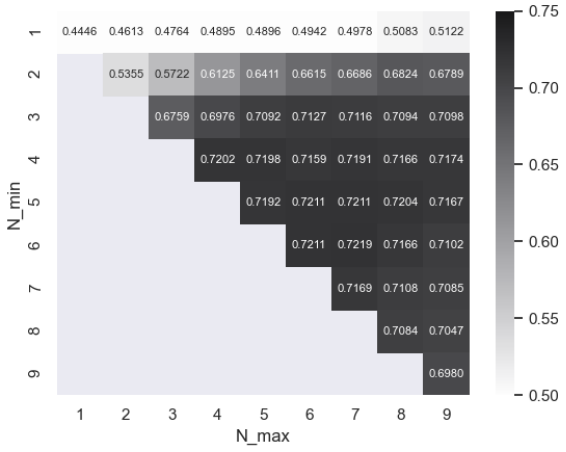
FIGURE 3 – Tâche 2 : résultats (MAP) avec des n-grammes de caractères sur le jeu de test



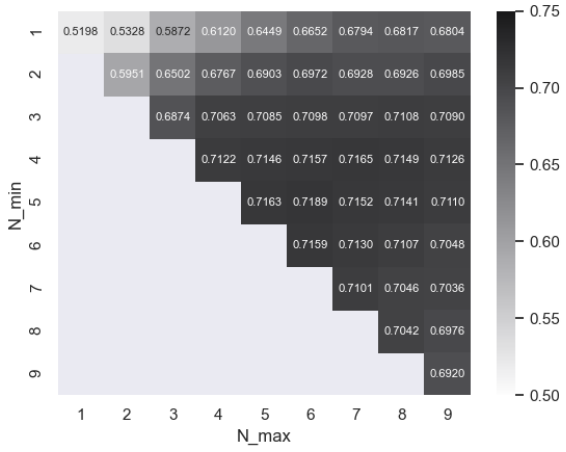
(a) Indice de Jaccard



(b) Coefficient de Dice



(c) Distance cosinus



(d) Distance de Bray-Curtis

FIGURE 4 – Tâche 1 : résultats (EDRM) avec des n-grammes de caractères sur le jeu de test

4.3 Discussion

Les résultats que nous avons présenté montrent l'intérêt de s'intéresser à optimiser des solutions simples de type *baseline*. eN effet, il peut suffire de modifier des paramètres simples pour faire progresser les résultats jusqu'à des niveaux qui apparaissent comme compétitifs vis-à-vis d'approches plus complexes. En particulier, l'utilisation de représentations en n-grammes de caractères plutôt qu'en mots présente l'avantage de diversifier, si l'on ne souhaite pas utiliser le terme « enrichir », à moindre coût la représentation et donc de la rendre plus à même de modéliser finement les relations entre les segments comparés. Ceci est d'autant plus important que les segments sont courts. Il semble évident que dans le cas de ces deux tâches de similarité, la relative stabilité du vocabulaire utilisé était sans doute favorable à une approche en chaînes de caractères. Cette approche permet notamment de capturer des racines et donc de détecter des familles de mots par exemple. Au contraire, quand des éléments censés être proches sémantiquement parlant ne partagent pas une grande proximité formelle, l'approche serait sans doute trop frustre.

Références

BRAY J. R. & CURTIS J. T. (1957). An ordination of the upland forest communities of southern wisconsin. *Ecological Monographs*, **27**(4), 325–349.

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 2020 (TALN 2020)*, p. 3–14.

CLAVEAU V. (2012). Vectorisation, Okapi et calcul de similarité pour le TAL : pour oublier enfin le TF-IDF. In *TALN - Traitement Automatique des Langues Naturelles*, p.=-, Grenoble, France.

GABAY S., RIGUET M. & BARRAULT L. (2019). A Workflow For On The Fly Normalisation Of 17th c. French. In *DH2019*, Utrecht, Netherlands : ADHO.

GRABAR N., GROUIN C., HAMON T. & CLAVEAU V. (2019). Information Retrieval and Information Extraction from Clinical Cases. Presentation of the DEFT 2019 Challenge. In *DEFT 2019 - Défi fouille de texte*, p. 1–10, Toulouse, France.

HUANG A. (2008). Similarity measures for text document clustering. In *Proceedings of the Sixth New Zealand Computer Science Research Student Conference (NZCSRSC2008), Christchurch, New Zealand*, p. 49–56.

MEHDAD Y. & TETREAULT J. (2016). Do characters abuse more than words ? In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, p. 299–303, Los Angeles : Association for Computational Linguistics.

MORENO J. G. & DIAS G. (2014). Easy Web Search Results Clustering : When Baselines Can Reach State-of-the-Art Algorithms. In *14th Conference of the European Chapter of the Association for Computational Linguistics*, Gotenburg, Sweden.

NEBHI K., BONTCHEVA K. & GORRELL G. (2015). Restoring capitalization in #tweets. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, p. 1111–1115, New York, NY, USA : Association for Computing Machinery.

RENDLE S., ZHANG L. & KOREN Y. (2019). On the difficulty of evaluating baselines : A study on recommender systems. arXiv preprint : 1905.01395.

STRUBELL E., GANESH A. & MCCALLUM A. (2019). Energy and policy considerations for deep learning in NLP. *CoRR*, **abs/1906.02243**.

UMEMURA K. & CHURCH K. (2009). Substring statistics. In *Proceedings of the 10th International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing '09, p. 53–71, Berlin, Heidelberg : Springer-Verlag.

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e 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

Participation d'EDF R&D à DEFT 2020

Danrun Cao[1], Alexandra Benamar, Manel Boumghar, Meryl Bothua, Lydia Ould-Ouali, Philippe Suignard
EDF R&D, 7 boulevard Gaspard Monge, 91120 Palaiseau
prenom.nom@edf.fr

RESUME

Ce papier décrit la participation d'EDF R&D à la campagne d'évaluation DEFT 2020. Notre équipe a participé aux trois tâchés proposées : deux tâches sur le calcul de similarité sémantique entre phrases et une tâche sur l'extraction d'information fine autour d'une douzaine de catégories. Aucune donnée supplémentaire, autre que les données d'apprentissage, n'a été utilisée. Notre équipe obtient des scores au-dessus de la moyenne pour les tâches 1 et 2 et se classe 2e sur la tâche 1. Les méthodes proposées sont facilement transposables à d'autres cas d'application de détection de similarité qui peuvent concerner plusieurs entités du groupe EDF. Notre participation à la tâche 3 nous a permis de tester les avantages et limites de l'outil SpaCy sur l'extraction d'information.

ABSTRACT

This paper describes the participation of EDF R&D at DEFT 2020 evaluation campaign. Our team participated in the three proposed tasks: two of them on Semantic Similarity Detection and one on Information Extraction in Clinical Cases. No additional data other than the training data was used. Our team gets above average results for the first and the second task and got the second place on the second task. The proposed methods are easily transferable to other Semantic Similarity Detection use cases and may interest several entities of the EDF group.

MOTS-CLES : données cliniques, détection de similarité sémantique, Word2Vec, graphes sémantiques, extraction d'information.

KEYWORDS: clinical data, Semantic Similarity Detection, Information Extraction, Word2Vec, semantic graphs.

1 Introduction

Dans la continuité de DEFT 2019, l'édition 2020 du défi fouille de textes (Cardon, 2020) continue d'explorer les cas cliniques rédigés en français. Cette nouvelle édition porte sur l'extraction

[1] Contribution égale de tous auteurs, listés par ordre alphabétique

Atelier DÉfi Fouille de Textes, pages 26–35.
HAL : hal-02784739.

d'information fine autour d'une douzaine de catégories. En dehors du domaine clinique, deux tâches sont proposées sur le calcul de similarité sémantique entre phrases. Plusieurs éléments nous ont motivés à participer à cette édition du défi. Participer à DEFT est l'occasion :

- de tester plusieurs méthodes de calcul de similarité dont les résultats contribuent directement à EDF Commerce et à d'autres entités du groupe EDF (tâches 1 et 2).

- d'évaluer des outils d'extraction d'entités nommées, comme SpaCy, même si ce travail est réalisé sur des données différentes des données EDF, comme ici avec des données médicales (tâche 3).

2 Description des tâches et méthodes utilisées

2.1 Tâche 1 : « Identifier le degré de similarité entre paires de phrases parallèles et non-parallèles sur plusieurs domaines »

2.1.1 Présentation

La tâche 1 consiste à déterminer le niveau de similarité entre paires de phrases, sur une échelle allant de 0 à 5 :

- 5 : Les deux phrases sont complètement équivalentes, car elles veulent dire la même chose ;
- 4 : Les deux phrases sont pour la plupart équivalentes, mais certains détails sans importance diffèrent ;
- 3 : Les deux phrases sont à peu près équivalentes, mais certaines informations importantes diffèrent ou manquent ;
- 2 : Les deux phrases ne sont pas équivalentes, mais partagent quelques détails ;
- 1 : Les deux phrases ne sont pas équivalentes, mais portent sur le même sujet ;
- 0 : Les deux phrases sont complètement différentes.

Exemple, avec une similarité de 4 entre les deux phrases suivantes :

- En l'absence d'amélioration comme en cas de persistance des symptômes, prendre un avis médical.
- En l'absence d'amélioration comme en cas de persistance des symptômes au-delà de 7 jours de traitement, prenez un avis médical.

La difficulté de la tâche consiste à affecter une note de manière absolue. Pour résoudre cette difficulté, les méthodes que nous proposons reposent toutes sur un apprentissage, puisqu'un jeu de données de 600 paires de phrases était fourni et annoté. Nos méthodes consistent à extraire des *features* pour ces différentes phrases, à calculer des similarités ou distances entre les paires de phrases, puis à entraîner un classifieur.

2.1.2 Run 1 : Dice + RL

Les différentes étapes sont les suivantes, pour une paire de phrases donnée :

- Suppression de la ponctuation et des mots faisant partie d'une stop-liste ;
- Troncature des mots à 6 caractères : « biberon » et « biberons » sont transformés en « bibero » ;
- Pour chaque phrase, ne sont gardés que les mots uniques ;
- Calcul des « features » ou descripteurs suivants :

1. Sorensen-Dice (Dice, 1945) : 2 fois le nombre de mots en commun divisé par la somme des nombres de mots de chaque phrase.

$$sim = \frac{2 * |X \cap Y|}{|X| + |Y|}$$

2. Le nombre de mots en commun entre les deux phrases
3. Le nombre de mots de la phrase 1
4. Le nombre de mots de la phrase 2
5. L'écart = nombre de mots de la phrase 1 - nombre de mots de la phrase 2
6. La valeur absolue de l'écart précédent
7. La distance de Levenshtein entre les deux phrases

Les descripteurs ont été calculés pour chaque paire du corpus d'apprentissage puis convertis au format ARFF pour être utilisées au sein du logiciel WEKA (Hall, 2009). Deux classifieurs ont été testés : « Régression Logistique » et « Random Forest ». La régression logistique obtenait les meilleurs résultats sur le corpus d'apprentissage, c'est donc cette méthode qui a été utilisée pour la phase de test.

2.1.3 Run 2 : Graphes sémantiques + RL

Dans cette méthode, nous nous appuyons sur des graphes sémantiques obtenus à partir d'arbres syntaxiques dont nous éliminons les informations jugées non pertinentes. Ensuite pour chaque paire de phrases, nous alignons les graphes sémantiques et extrayons des *features*. Ces *features* constituent notre entrée pour l'entraînement d'un classifieur, en l'occurrence une Régression Logistique.

2.1.3.1 Graphes sémantiques

L'analyse syntaxique est effectuée avec le schéma d'annotation UD (Nivre et al.,2016) du logiciel Talismane (Urieli, 2013). Les arbres syntaxiques sont décrits sous le format CoNLL. Ces arbres sont ensuite simplifiés afin d'éliminer les éléments grammaticaux. À l'issu de cette étape de simplification, nous obtenons un graphe sémantique. Les critères de simplifications des arbres syntaxiques sont :

- Suppression de la ponctuation.
- Suppression des mots présents dans une « stop-liste ».
- Suppression de certaines catégories de mots : *DET* (déterminant), *ADP* (préposition), *PRON* (pronom), *CCONJ* (conjonction de coordination), *ADP+DET* (déterminant composé), *PART* et *X* (particule et partie de locution qui ne correspond à aucune catégorie traditionnelle).
- Suppression des nœuds ayant comme dépendance : *det* (déterminant), *mark* (mot introduisant une proposition de subordination), *punct* (ponctuation), *cc* (marqueur de

conjonction de coordination), *case* (préposition), *fixed* (structuré figée non grammaticale), *reparandum* (reformulation), *discourse* (mot de discours), *cop* (verbe copule), *aux* (verbe auxiliaire).

Si les nœuds supprimés ont des descendants, ces derniers sont rattachés à la racine du nœud supprimé.

2.1.3.2 Alignement de graphes

L'alignement des graphes se fait au niveau des nœuds afin de trouver les termes similaires entre les phrases sources et les phrases cibles : nous produisons une matrice de distance d'édition entre les mots au sein de deux phrases et nous appliquons l'algorithme hongrois pour trouver le meilleur alignement entre les arbres syntaxiques simplifiés. La distance d'édition entre les mots est définie par leur distance cosinus, calculée avec un modèle word2vec pré-entraîné par Jean-Philippe Fauconnier[2]. En cas de mots hors vocabulaire, nous comparons les lemmes : s'il s'agit du même lemme, la distance minimum (=0) est attribuée, sinon la distance maximum (=2). Comme les deux phrases n'ont souvent pas la même longueur, nous insérons des nœuds vides pour assurer que chaque mot ait un match. Tout match avec un nœud vide reçoit la distance maximum (=2). Concernant les types de match entre nœuds, il en existe trois : le *match exact* où il s'agit du même mot, le *match loose* où deux mots différents sont alignés et le *match dummy* où l'un des nœuds alignés est un nœud vide.

Les arêtes sont représentées par les nœuds qu'elles relient. Pour chaque arête source, nous cherchons les nœuds cibles qui correspondent aux nœuds sources en question. Le match de l'arête source est donc le chemin le plus court qui relie les nœuds cibles trouvés. Ce match ne correspond donc pas forcément à une arête réelle dans le graphe, il peut en nécessiter plusieurs. La distance d'édition de ce match est calculée ainsi :

$$DE[(a1, b1) \rightarrow (a2, b2)] = (|DE[a1 \rightarrow b1]| + |DE[a2 \rightarrow b2]|) \times NbArêtes$$

Il existe également trois types de match entre arêtes : le *match exact* où une seule arête cible est nécessaire, le *match loose* où plusieurs arêtes cibles sont nécessaires et le *match dummy* où l'un des nœuds cibles est un nœud vide.

2.1.3.3 Extraction de features et entraînement

Nous avons défini trois grandes familles de *features* ci-dessous que nous avons extraites :

1. Au niveau de phrase entière :
   - Différence de nombre de mots,
   - Distance Damerau-Levenshtein de deux phrases au niveau du caractère,
   - Distance Damerau-Levenshtein de deux phrases au niveau du mot.

2. Au niveau de mot/nœud de graphe :
   - Distance d'édition (DE) absolue : la somme d'DE des matches de nœuds
   - Distance d'édition relative : DE absolue/DE maximum (toutes les DE à maximum)

[2] http://fauconnier.github.io/

- Nombre et pourcentage de *matches exact*
- Nombre et pourcentage de *matches loose*
- Nombre et pourcentage de *matches dummy*

3. Au niveau dépendance/arête de graphes :
   - Différence entre le nombre d'arêtes
   - Nombre et pourcentage de *matches exact*
   - Nombre et pourcentage de *matches loose*
   - Nombre et pourcentage de *matches dummy*
   - Distance d'édition absolue : la somme d'DE des matches d'arêtes
   - Distance d'édition relative : DE absolue/DE maximum (toutes les DE à maximum)

Nous obtenons donc un total de 20 *features*. Plusieurs classifieurs ont été testés également : Régression Logistique et Random forest. Si les deux classifieurs renvoient des résultats similaires, la régression logistique est plus stable. Nous avons donc retenu cette dernière.

2.1.4 Run 3 : Features + RL

Les différentes étapes pour ce run pour une paire de phrase donnée, sont les suivantes :

- Suppression de la ponctuation, des espaces multiples et des mots faisant partie d'une stop-liste.
- Calcul des *features* ou descripteurs suivants (Cardon, 2018) :

1. Le nombre de mots en commun entre les deux phrases ;
2. Le pourcentage de mots de la phrase 1 inclus dans la phrase 2 ;
3. La différence de la longueur des deux phrases ;
4. La différence de la longueur moyenne des mots entre les deux phrases ;
5. Le nombre des *n-grams* en commun entre les deux phrases. Nous avons calculé les *bi-grams*, *tri-grams* et *n-grams* de longueur 4 ;
6. La distance de Levenshtein entre les deux phrases ;
7. La distance de Jaccard entre les deux phrases ;
8. La distance sémantique (Word Mover Distance) entre les deux phrases.

Les descripteurs ont été calculés pour chaque paire de phrases au niveau du mot sauf la distance de Levenshtein qui a été calculée au niveau des caractères et au niveau des mots.

Deux classifieurs ont été testés à savoir « Régression Logistique » et « Random Forest ». La régression logistique obtenait les meilleurs résultats sur le corpus d'apprentissage car plus stable, c'est donc cette méthode qui a été retenue pour la phase de test.

2.1.5 Résultats

Méthode	Evaluation en EDRM
Maximum	*0,8217*
Run 1 : Dice + RL	***0,8198***
Run 3 : Features + RL	***0,8069***
Run 2 : Graphes sémantiques + RL	***0,8018***
Médiane	*0,7947*
Moyenne	*0,7617*
Minimum	*0,6533*

Table 1 : résultats de la tâche 1

Les résultats de nos trois *run* se trouvent tous au-dessus de la moyenne. Notre premier *run* est très proche du meilleur résultat obtenu (0,0019 d'écart). Nous avons observé dans les sorties que les différences entre les valeurs 0 et 1 ainsi qu'entre les valeurs 4 et 5 sont parfois difficiles à appréhender, même pour un humain. En voici un exemple :

- « Boris Godounov meurt, subitement, le 13 avril 1605 à Moscou : on parla alors d'empoisonnement ou de suicide. »
- « Boris Godounov est un monarque russe qui régna de 1598 jusqu'à sa mort en 1605 sur la Russie. »

La note à prévoir ici était 0 (note pour deux phrases complètement différentes), alors qu'elles parlent de Boris Godounov. La note aurait pu être 1 (note pour deux phrases ne sont pas équivalentes, mais qui portent sur le même sujet), le sujet étant la mort de B. Godounov.

2.2 Tâche 2 : « identifier les phrases parallèles possible pour une phrase source »

2.2.1 Présentation

La tâche 2 est une tâche d'appariement entre une phrase source et trois phrases cibles. L'objectif consiste à trouver quelle phrase, parmi les trois phrases cibles, est la plus proche de la phrase source. Exemple : la phrase la plus proche de la phrase source est la phrase cible n°2 :

- Source : « compte tenu des données disponibles, l'utilisation chez la femme enceinte ou qui allaite est possible ponctuellement » ;
- Cible 1 : « ce médicament est un laxatif utilisé par voie orale » ;
- Cible 2 : « ce médicament, dans les conditions normales d'utilisation, peut être utilisé ponctuellement pendant la grossesse et l'allaitement » ;
- Cible 3 : « boîte de 1 flacon de 250 ml ou 500 ml ».

Nous proposons deux méthodes différentes, l'une basée sur un calcul de similarité de type Dice et l'autre sur des embeddings de documents.

2.2.2 Run 1 : Dice

La première méthode est similaire à la méthode 1 de la tâche 1. Elle consiste à éliminer les mots faisant partie d'une stop-liste, à garder les mots uniques puis à calculer le coefficient dice-sorensen entre la phrase source et les trois phrases cibles, puis à choisir la phrase qui maximise cette similarité.

2.2.3 Run 2 : USE

Cette méthode est basée sur une représentation vectorielle des phrases calculée à l'aide de USE (Universal Sentence Encoder) (CER *et al.*, 2018), puis à calculer une similarité cosinus entre la phrase source et les 3 phrases cibles puis à choisir la phrase qui maximise cette similarité. Pour cette tâche, aucun prétraitement n'a été effectué. Le modèle encodeur de USE utilisé est « Universal Sentence Encoder Multilingual module », proposé par Google en 2019, appris sur 16 langues dont le français.

2.2.4 Résultats

Méthode	Evaluation en MAP	Nb d'erreurs
Maximum	*0,9906*	*5*
Médiane	*0,9868*	
Run 2 : USE	**0,9867**	**7**
Run 1 : Dice + RL	**0,9830**	**9**
Moyenne	*0,9822*	
Minimum	*0,9396*	*32*

Table 2 : résultats de la tâche 2

Les résultats de nos deux *run* se trouvent au-dessus de la moyenne. Notre deuxième *run* est très proche du meilleur résultat obtenu (0,0039d'écart) ainsi que le premier (0,0076). Le gagnant a commis 5 erreurs, notre *run 2* 7 et notre *run 1* 9 erreurs. Ces différences sont facilement explicables. Par exemple, la méthode du *run 1* se trompe sur la phrase « les stéroïdes oraux sont le traitement standard » en lui affectant la cible « les effets indésirables les plus courants associés aux stéroïdes oraux sont la prise de poids et l'augmentation de la pression artérielle » au lieu de « les corticostéroïdes par voie orale sont le traitement le plus courant ». Cela s'explique par les différences entre les mots employés : « corticostéroïde »s au lieu de « stéroïdes », voie « orale » au lieu de « oraux » et « courant » au lieu de « standard ». La méthode du *run 1* se trompe s'il y a plusieurs synonymes ou expressions légèrement différentes entre la source et la bonne cible.

2.3 Tâche 3 : « extraction d'information »

La tâche 3 est une tâche d'extraction d'information. Elle consiste à repérer, dans les cas cliniques, les informations fines autour d'une dizaine de catégories. Quatre domaines sont couverts :

- autour des patients : anatomies ;
- autour de la pratique clinique : examen, pathologie, signe ou symptôme ;

- autour des traitements médicamenteux et chirurgicaux : substance, dose, durée, fréquence, mode d'administration, traitement (chirurgical ou médical), valeur ;
- autour du temps : date, moment.

2.3.1 Run 1 : Spacy

Nous avons adopté des approches symboliques pour extraire les informations recherchées. Nous avons ainsi utilisé la bibliothèque python SpaCy[3], et plus précisément ses composants de reconnaissance d'entités nommées personnalisées. Bien que l'outil propose de nombreuses fonctionnalités intéressantes, une contrainte importante est que chaque *token* ne peut correspondre qu'à un seul type d'entité nommée. Ceci a été problématique pour les syntagmes complexes, notamment pour les catégories « pathologie » et « sosy ». Nous avons tout de même essayé de repérer des « pathologies », mais aucune prédiction n'a été faite pour « sosy ». Concernant les catégories restantes, nous décrivons ci-dessous les règles développées.

2.3.1.1 Règles lexicales

Il s'agit des règles basées sur des lexiques spécifiques. Ces règles permettent de récupérer les termes n'ayant pas ou peu de variations. Nous avons utilisé ce type de règles pour les catégories « substance », « état », « prise », « changement », « norme », « assertion ». Les lexiques sont la plupart extraits du corpus d'entraînement, à l'exception de « substance » pour lequel nous avons introduit une liste de spécialités pharmaceutiques et leurs compositions provenant du site officiel de l'ANSM[4].

2.3.1.2 Règles à la base des patterns morphologiques

Certaines catégories, telles que « date », « dose », « mode » et « fréquence », se caractérisent par un nombre limité de structures récurrentes. Grâce à SpaCy, nous avons pu implémenter les règles permettant de reconnaître ces structures, et ce sous plusieurs formes. Il s'agit des patrons morphosyntaxiques, des expressions régulières, des règles décrivant les caractéristiques morphologiques de *token*, et finalement des combinaisons de *token* et de sous-*token*.

2.3.1.3 Règles à la base de l'analyse syntaxique

La plupart des expressions à repérer sont des syntagmes nominaux. Dans un arbre syntaxique, un tel syntagme se présente sous forme du sous arbre d'un mot clé. Nous élaborons donc une liste de mots clés à partir du corpus d'entraînement. L'analyse syntaxique se fait avec SpaCy et nous récupérons les sous arbres contenant ces mots clés. Ces sous arbres sont ensuite nettoyés afin d'éliminer les branches redondantes. Ce nettoyage est effectué à l'aide des filtres lexicaux et morphosyntaxiques. Finalement, ces arbres nettoyés constituent nos prédictions. Le plus grand avantage de ce type de règle est sa flexibilité. Ceci est surtout utile pour les catégories qui n'ont pas ou peu de contraintes morphologiques. Au lieu de récupérer un nombre défini de modifieurs d'un tel mot clé, nous pouvons

[3] https://spacy.io/
[4] http://agence-prd.ansm.sante.fr

prendre tous les modifieurs qui y sont attachés, puis enlever ceux qui ne sont pas pertinents. Ainsi nous évitons d'élaborer les règles trop précises seulement pour repérer quelques structures rares.

2.3.2 Résultats

	TP	FP	FN	Precision	Recall	F1
pathologie	61	386	105	0,1365	0,3675	0,1990
sosy	0	0	1279	0,0000	0,0000	0,0000
Overall	61	386	1384	0,1365	0,0422	0,0645

Table 3: résultats de la sous tâche 1

	TP	FP	FN	Precision	Recall	F1
anatomie	238	537	883	0,3071	0,2123	0,2511
dose	19	62	33	0,2346	0,3654	0,2857
examen	368	397	449	0,4810	0,4504	0,4652
mode	34	31	55	0,5231	0,3820	0,4416
moment	36	73	129	0,3303	0,2182	0,2628
substance	93	25	220	0,7881	0,2971	0,4316
traitement	88	247	216	0,2627	0,2895	0,2754
valeur	159	89	273	0,6411	0,3681	0,4676
Overall	1035	1461	2258	0,4147	0,3143	0,3576

Table 4 : résultats de la sous tâche 2

Les erreurs proviennent principalement des catégories « anatomie » et « examen », ce qui correspond à la contrainte de non chevauchement d'entités de SpaCy. Il est attendu que les termes anatomiques soient présents dans les syntagmes « examen » « traitement » et « pathologie », alors que ce genre d'annotations est difficile à gérer avec SpaCy.

Par ailleurs, un type d'erreurs fréquent est la frontière des syntagmes récupérés avec les règles syntaxiques. Elles permettent de trouver les syntagmes les plus longs, cependant ce n'est pas toujours ce qui est demandé. Par exemple, les adjectifs suivants un mot clé sont gardés, mais avec peu de contrôle du vocabulaire. Il est possible que les adjectifs « non médicaux » soient pris en compte. De plus, nous utilisons des filtres lexicaux et morphosyntaxiques pour nettoyer les sous arbres. Les erreurs de parsing morphosyntaxique peuvent donc également conduire à un mauvais découpage de syntagme.

3 Conclusion

Participer à la campagne DEFT 2020, nous a permis de tester plusieurs méthodes de calcul de similarité dont les résultats prometteurs pourront être utilisés directement à EDF Commerce et à d'autres entités du groupe EDF (tâches 1 et 2). Nous travaillons également régulièrement sur l'évaluation des outils d'extraction d'entités nommées, comme SpaCy. Les données médicales du défi, bien que différentes de nos données métiers, sont pour nous l'occasion d'évaluer ces outils, de connaître leurs avantages mais également leurs limites.

Références

CARDON R, GRABAR N. Identification of parallel sentences in comparable monolingual corpora from different registers. In: *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*. 2018. p. 83-93

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.

CER, D, YANG, YINFEI, KONG, SHENG-YI, ET AL. Universal sentence encoder. arXiv preprint arXiv:1803.11175, 2018.

DICE, L. R. (1945). Measures of the amount of ecologic association between species. Ecology, 26(3), 297-302.

HALL, M., FRANK, E., HOLMES, G., PFAHRINGER, B., REUTEMANN, P., & WITTEN, I. H. (2009). The WEKA data mining software: an update. ACM SIGKDD explorations newsletter, 11(1), 10-18.

NIVRE, J., DE MARNEFFE, M. C., GINTER, F., GOLDBERG, Y., HAJIC, J., MANNING, C. D., ... & TSARFATY, R. (2016, May). Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)* (pp. 1659-1666).

URIELI, A. (2013). *Robust French syntax analysis: reconciling statistical methods and linguistic knowledge in the Talismane toolkit* (Doctoral dissertation).

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e 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

Contextualized French Language Models for Biomedical Named Entity Recognition

Jenny Copara*[1,2,3] Julien Knafou*[1,2] Nona Naderi[1,2] Claudia Moro[4] Patrick Ruch[1,2] Douglas Teodoro[1,2]

(1) University of Applied Sciences and Arts of Western Switzerland, Rue de la Tambourine 17, 1227, Geneva, Switzerland
(2) Swiss Institute of Bioinformatics, Rue Michel-Servet 1, Geneva, Switzerland
(3) University of Geneva, Rue du Général-Dufour 24, 1211, Geneva, Switzerland
(4) Pontifical Catholic University of Paraná, Rua Imaculada Conceição 1155, 80215-901, Curitiba, Brazil
{jenny.copara,julien.knafou}@hesge.ch,
{nona.naderi,patrick.ruch,douglas.teodoro}@hesge.ch, c.moro@pucpr.br

RÉSUMÉ

Modèles contextualisés en langue française pour la reconnaissance des entités nommées dans le domaine biomédical

La reconnaissance des entités nommées (NER) est essentielle pour les applications biomédicales car elle permet la découverte de connaissances dans des données en texte libre. Comme les entités sont des phrases sémantiques, leur signification est conditionnée par le contexte pour éviter toute ambiguïté. Dans ce travail, nous explorons les modèles de langage contextualisés pour la NER dans les textes biomédicaux français dans le cadre du Défi Fouille de Textes. Notre meilleure approche a obtenu une mesure F1 de 66% pour les symptômes et les signes, et les catégories de pathologie, en étant dans le top 1 pour la sous-tâche 1. Pour les catégories anatomie, dose, examen, mode, moment, substance, traitement et valeur, elle a obtenu une mesure F1 de 75% (sous-tâche 2). Si l'on considère toutes les catégories, notre modèle a obtenu le meilleur résultat dans le cadre de ce défi, avec une mesure F1 de 72%. L'utilisation d'un ensemble de modèles de langages neuronaux s'est révélée très efficace, améliorant une base de référence du CRF de 28% et un modèle de langage spécialisé unique de 4%.

ABSTRACT

Named entity recognition (NER) is key for biomedical applications as it allows knowledge discovery in free text data. As entities are semantic phrases, their meaning is conditioned to the context to avoid ambiguity. In this work, we explore contextualized language models for NER in French biomedical text as part of the Défi Fouille de Textes challenge. Our best approach achieved an F_1-measure of 66% for symptoms and signs, and pathology categories, being top 1 for subtask 1. For anatomy, dose, exam, mode, moment, substance, treatment, and value categories, it achieved an F_1-measure of 75% (subtask 2). If considered all categories, our model achieved the best result in the challenge, with an F_1-measure of 72%. The use of an ensemble of neural language models proved to be very effective, improving a CRF baseline by up to 28% and a single specialised language model by 4%.

MOTS-CLÉS : reconnaissance d'entités nommées, encapsulation de mots contextualisés, CRF,

∗. Authors JC and JK contributed equally to this work.

Atelier DÉfi Fouille de Textes, pages 36–48.
HAL : hal-02784740.

BERT, CamemBERT.

KEYWORDS: named entity recognition, contextualized word embeddings, CRF, BERT, CamemBERT.

1 Introduction

The large amount of raw text data available in the biomedical domain enables to leverage the wealth of the content. Combined with manually curated data, it allows the development of automatic techniques to unlock the value of the raw resources for supporting healthcare and advance science. In particular, information extraction methods enable the extraction of specific data types from text data (e.g., entities). Information extraction fosters several applications from tracking of technologies (Teodoro *et al.*, 2010) in patents to clinical decision support (Liu *et al.*, 2016), biocuration assistance (Liu *et al.*, 2016; Teodoro *et al.*, 2020), and healthcare-associated infections detection (Tvardik *et al.*, 2018) in the biomedical domain. It has also important challenges associated with the application domain and the language in which the text is available. Indeed, information extraction systems for recognizing entities are mostly focused on English. However, it is widely recognised that it is crucial that research expands to other languages in the same scale (Dupont, 2017; Grabar *et al.*, 2019).

Named entity recognition (NER) is a key part in information extraction systems. Named entities are phrases that contain names of persons, organizations, locations (Tjong Kim Sang & De Meulder, 2003) to name but a few examples. There are many studies for NER in French language, for instance in i) journalistic data (Dupont, 2017; Martin *et al.*, 2020) with a set of entities like person, organization, company, location, point of interest, fiction character and product ; and ii) recognizing entities in tweets (Sileo *et al.*, 2017), including person, music artist, organisation, product and media, among others. In the biomedical domain, recognizing entities is mainly focused on semantic groups and concepts from Unified Medical Language System (UMLS) on The Quaero French Medical Corpus (Névéol *et al.*, 2014). The Quaero corpus contains annotated French Medline (titles and abstracts) and European Medicines Agency (EMEA) documents (drug labels).

Community challenges, such as CLEF eHealth, have been evaluating specific information extraction tasks for the clinical domain (Sankhavara & Majumder, 2019). Erasmus MC, one of the CLEF eHealth top scorers, is a dictionary-based NER for French UMLS and translations for non-French terms, achieving 74.9% of F_1-measure for EMEA and 69.8% of F_1-measure for Medline corpus in semantic groups annotation (Van Mulligen *et al.*, 2016). In Erasmus MC, semantics could be missed by the presence of compounded semantic groups or UMLS concepts. Similarly, SIFR annotator is a semantic annotator for French clinical narratives (Tchechmedjiev *et al.*, 2018). It relies on Mgrep, a concept recognizer based on label matching and heuristics, and achieves 62.6% of F_1-measure in EMEA and 58.9% of F_1-measure in Medline for semantic groups. The tool is limited due to the lack of word disambiguator and scarce French ontologies concerning English ontologies.

Deep neural language models have been recently leveraged to improve NER methods (Lee *et al.*, 2019). Deep neural language models are self-supervised models that take advantage of free text to learn word representations using their context (Turian *et al.*, 2010). With the advent of low-dimensional representation models supported by deep neural networks, such as word2vec (Mikolov *et al.*, 2013), the importance of word representations has become more evident. Further research has been taken to find out more accurate representations of words, such as in Global Vectors (GloVe) (Pennington *et al.*, 2014), and to the more recent contextualized representations, like ELMo (Peters

et al., 2018), UMLFiT (Howard & Ruder, 2018), and BERT (Devlin *et al.*, 2019). In particular, BERT is based on the transformers architecture, which uses an attention mechanism, via bidirectional pre-training from unlabeled text, conditioned in left and right contexts in all layers (Devlin *et al.*, 2019). BioBERT (Lee *et al.*, 2019), a BERT-based model trained on large-scale biomedical corpora has shown significant improvements in downstream tasks in the biomedical domain, including NER. Similarly, CamemBERT [1] is a contextualized language model trained and optimized specifically for French language (Martin *et al.*, 2020; Devlin *et al.*, 2019) based on RoBERTa model (Liu *et al.*, 2019).

In this paper, we investigate contextualized language models for French NER in clinical texts in the context of the Information Extraction task of the Défi Fouille de Textes (DEFT) challenge (Cardon *et al.*, 2020). This task is divided in two subtasks, which aim to identify *anatomie* (anatomy), *dose*, *examen*, *mode*, *moment*, *pathologie* (pathology), *sosy* (symptoms and signs), *substance*, *traitement* and *valeur* (value) entities in clinical narratives. Inasmuch as each language has its own peculiarities, our hypothesis is that it is worth designing a specific language model for French clinical corpora. Thus, we explore a CamemBERT-based model pre-trained on a biomedical corpus and fine-tuned on the DEFT information extraction task data. We compare its performance with multilingual BERT, CamemBERT and an ensemble of language models. In the following sections, we describe the design and results of the experiments.

2 Methods

In this work, we explore two perspectives for NER : as information extraction and as word representation. For the first, named entities are considered as a sequence classification problem, for which we propose a baseline method using the conditional random fields (CRF) framework. For the latter, our methodology is based on different deep neural language models derived from the BERT architecture. These methods were used to extract named entities in subtask 1 - symptoms and signs, and pathology - and subtask 2 - anatomy, dose, exam, mode, moment, substance, treatment, and value of the DEFT Information extraction task.

2.1 Conditional Random Fields

We used a linear chain CRF sequence classifier as a baseline and relied on the implementation of *CRFSuite* [2]. This probabilistic graphical model considers correlations between the neighborhood of words in a sentence and its features, jointly with the corresponding labels. Such correlation allows this model to learn the labels in a sequence (Lafferty *et al.*, 2001). In fact, linear chain CRF estimates the conditional probability of a label given a word sequence (Sutton, 2012). As shown in Table 1, our model relies on a set of NER standard features defined over a window of ± 2 tokens (Guo *et al.*, 2014; Copara *et al.*, 2016), including the word itself, lower-cased word, capitalization pattern, prefixes, suffixes, among others. Additionally, we used language-based features, such as lower-casing the words in the text, checking if the current token is a measure unit and whether the current token contains a French character. It is worth noting that we have not used gazetteers extensively, just a short list of units.

1. https://camembert-model.fr/
2. http://www.chokkan.org/software/crfsuite/

	Feature					
word	Une	première	dose	de	20	mg
lowercase word	une	première	dose	de	20	mg
capitalization pattern	ULL	LLLLLLLL	LLLL	LL	DD	LL
type	InitUpper	AllLetter	AllLetter	AllLetter	AllDigit	AllLetter
prefixes	u, un, une	p, pr, pre	d, do, dos	d, de	2, 20	m, mg
sufixes	e, ne, une	e, re, ère	e, se, ose	e, de	0, 20	g, mg
unit	no	no	no	no	no	yes
french char	no	yes	no	no	no	no

TABLE 1 – Example of features for the sentence *"Une première dose de 20 mg"*. *U → uppercase ; L → lowercase ; D → digit.

In our CRF model each entity is associated with one label (as usually in NER) and when there are nested entities, we keep entities that encompass other and dismiss nested entities.

2.2 Transformers with a token classification on top

For this experiment, we selected five BERT-based language models. The first, bert-base-multilingual-cased (Devlin *et al.*, 2019), is used as our transformer baseline as it was not trained specifically on a French corpus. The second and the third models, camembert-base and camembert-large, respectively, are based on the RoBERTa architecture (Liu *et al.*, 2019), a BERT-based model with some changes (tokenizer, training task, optimization, etc.) and trained on a large French corpus (Martin *et al.*, 2020). Models 4 and 5, so called, camembert-bio-base and camembert-bio-large, respectively, are CamemBERT-based models pre-trained on a french biomedical corpus containing 31k+ scientific publications extracted from PubMed. To further pre-train these models, we took CamemBERT weights as a starting point. Then, using an Adam optimizer (Kingma & Ba, 2014), we minimized a masked-language modeling loss. We trained it using 512 tokens during 5 epochs with a learning rate of 5e-5 and batch size of 24 [3].

As RoBERTa models are based on the BERT architecture, all our base and large models share hyper-parameters. For the base models, we have 12 layers (L), with 768 hidden units (H) and 12 attention heads (A). For the large architecture versions, we have L=24, H=1024 and A=16. The multilingual BERT model [4] uses WordPiece [5] as a tokenizer whereas the CamemBERT-based models use SentencePiece (Kudo & Richardson, 2018). [6] The tokenizer's choice was driven by the original model's tokenizer. Indeed, as we were fine-tuning BERT or CamemBERT models, we had to reuse the whole pipeline which includes the tokenizer (makes the link between a token and its trained representation possible). As explained in their paper (Martin *et al.*, 2020), SentencePiece does not require pre-tokenization which makes it a non-language specific tokenizer. Table 2 summarizes these architectural differences.

3. For the large model, as each step was too big for our 4 GPUs machine, we used gradient accumulation (i.e., the accumulation of 2 batches of 12 in order to get a batch of 24)

4. The multilingual BERT model uses a vocabulary size of 30K.

5. Comparison between WordPience and SentencePiece tokenizers : https://github.com/google/sentencepiece

6. CamemBERT uses a vocabulary size of 32K.

Tokenizer	WordPiece	SentencePiece			
Model	bert-base-multilingual-cased	camembert-base	camembert-bio-base	camembert-large	camembert-bio-large
layer (L)	12			24	
hidden (H)	768			1024	
heads (A)	12			16	

TABLE 2 – Architectural differences of BERT-based models.

For the fine-tuning of the NER models, we used the hugging face [7] framework, which basically standardizes the process for all the transformers. Each NER model is a BERT module with a fully connected layer on top of the hidden states of each token. As entities could overlap, we decided to use a binary or one-vs-all approach per entity instead of using a softmax which does not allow multi-labelling. All previously presented language models were fine-tuned on the DEFT task 3 dataset for 20 epochs, with a sequence length of maximum 256 tokens, a learning rate of 4e-5 and a warmup proportion of 0.1. As for the CRF baseline, we use one label for each entity, discarding nested entities.

2.3 Dataset

In DEFT task 3 - information extraction - there are two subtasks. Subtask 1 is focused on the *pathologie* and *sosy* (symptoms and signs) entities. Subtask 2 concerns the identification of *anatomie*, *dose*, *examen*, *mode*, *moment*, *substance*, *traitement*, and *valeur* entities. For assessing these subtasks, the challenge organisers provided a training dataset composed of 100 French clinical documents manually annotated with the 8098 entities (Grabar *et al.*, 2018). The annotated data include all the entities mentioned for each subtask, in addition to informational entities (e.g. *date*, *durée*, *frequence*) that have not been considered in our models. An example of annotation is shown in Figure 1. As we can notice, nested entities appear often in the annotations, sometimes within the same subtask and sometimes across subtasks.

7. https://huggingface.co/transformers/

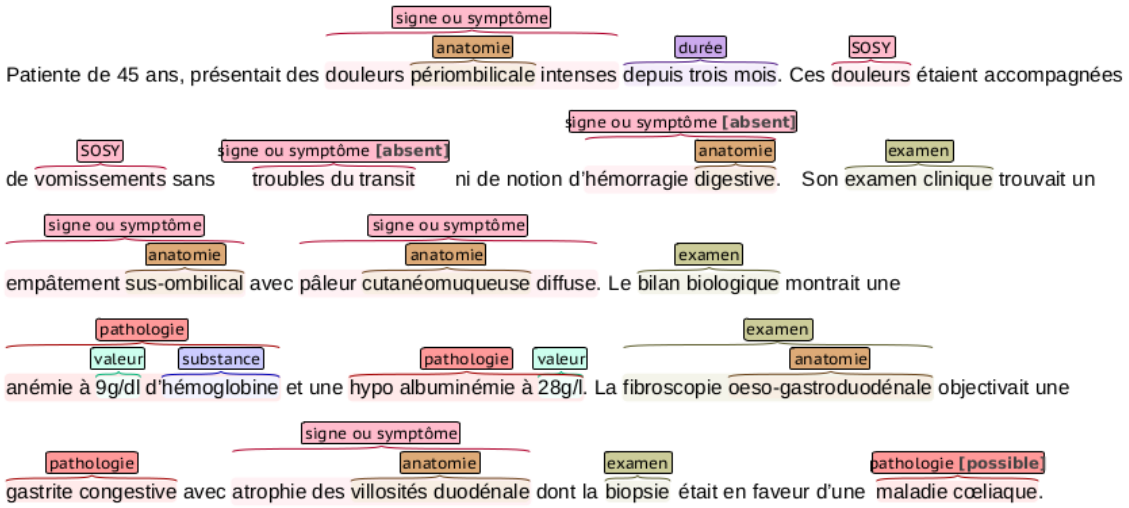


FIGURE 1 – An example of clinical narrative with entity annotations for subtasks 1 and 2. The annotations are color coded.

Table 3 shows the distribution of annotations among the entities in the training data. The majority of annotations come from the *sosy*, *anatomie* and *examen* entities, which compose together 54% of the training data. On the other hand, *mode*, *dose* and *moment* represent 13% of the dataset. To train and validate our models in the training phase, this dataset was split into train (80%), dev (10%) and test (10%) sets. The hyper-parameters of the models were selected for the test phase based on their performance on the dev set.

Entity	Train (count / %)	Dev (count / %)	Test (count / %)	All (count / %)
anatomie	1241/ 19	57 / 6	174 / 26	1472 / 18
dose	302 / 5	40 / 4	5 / 1	347 / 4
examen	962 / 15	119 / 12	137 / 20	1218 / 15
mode	214 / 3	24 / 2	11 / 2	249 / 3
moment	363 / 6	77 / 8	54 / 8	494 / 6
pathologie	260 / 4	91 / 9	184 / 27	535 / 7
sosy	1451 / 23	196 / 20	33 / 5	1680 / 21
substance	883 / 14	85 / 8	22 / 3	990 / 12
traitement	301 / 5	193 / 19	52 / 8	546 / 7
valeur	443 / 7	119 / 12	5 / 1	567 / 7
Total	6420 / 100	1001 / 100	677 / 100	8098 / 100

TABLE 3 – Entity distribution for the training phase collection.

3 Results and Discussion

In this section, we present the results of our models for the training and test phases for both subtasks. During the training phase, we used only the training collection provided in the challenge in order to

develop and tune our models. During the test phase, we evaluated over the test collection with the parameters identified in the training phase.

3.1 Training phase

Table 4 shows the results of our models in the training phase. The baseline CRF model achieved 0.4641 of overall micro F_1-measure, having a highest F_1-measure for the *valeur* entity (0.7708) and the lowest for the *pathologie* entity (0.1967). For the transformer-based models, the camembert-bio-base model outperforms both base models (BERT and CamemBERT) for the overall micro and macro F_1-measures, demonstrating the effectiveness of the specific biomedical corpus for the clinical NER task. For the large transformer models, CamemBERT achieves the highest micro F_1-measure (0.6826) and camembert-bio achieves the highest macro F_1-measure (0.5790). All contextualized language models outperform the baseline CRF model significantly, showing the outstanding performance of these architectures for NER in biomedical French texts.

As described in Section 2.2, we used a one-vs-all approach to predict the overlapped (nested) entities for the transformer models. This approach was not as effective for the CRF baseline, reducing the overall (micro) F_1-measure performance to 0.4464. Using this approach, no *dose* entity was correctly recognised and F_1-measure of *traitement* decreased to 0.1538. Nonetheless, the performance for recognising *anatomie* improved to 0.5300 of F_1-measure. These results suggest that in order to predict *dose* or *traitement* correctly, it is necessary to observe near entities, given the nature of the CRF learning model. We believe that the *anatomie* entity increased its performance mainly due to the fact that it appears usually nested in the *sosy*, *pathologie* or *examen* entities. However, in a one-vs-all approach, this entity will not be nested.

F_1-measure	baseline	bert-base-multilingual-cased	camembert-base	camembert-large	camembert-bio-base	camembert-bio-large
anatomie	0.3673	0.7170	0.7675	**0.8022**	0.7921	0.7751
dose	0.2500	0.1111	**0.4286**	0.1538	0.2857	0.1538
examen	0.5727	0.6618	0.6957	0.6667	0.6926	**0.7011**
mode	0.2857	0.2857	0.3333	0.2857	**0.4444**	0.2500
moment	0.4000	0.6957	**0.7273**	0.6364	0.6667	**0.7273**
pathologie	0.1967	0.3725	0.4956	**0.5714**	0.4248	0.5474
sosy	0.4356	0.6139	0.5838	**0.6961**	0.6563	0.6772
substance	0.4878	0.4400	0.3902	0.5854	0.5909	**0.6341**
traitement	0.4255	0.4590	0.3810	0.4848	0.3582	**0.5161**
valeur	0.7708	0.7961	0.7885	**0.8302**	0.8182	0.8077
Overall (micro)	0.4641	0.6135	0.6311	**0.6826**	0.6569	0.6791
Overall (macro)	0.4192	0.5153	0.5592	0.5713	0.5730	**0.5790**

TABLE 4 – Evaluation of different models in the training phase.

We also assessed a voting strategy, or ensemble, between the transformers models, where all 5 BERT

models vote with their predictions. For example, when the voting threshold $v = 1$, we use the set of positive predictions coming from all the models, whereas when $v = 3$, we only use the positive predictions when the majority of the models agree on an annotation. As shown in Figure 2, the precision increases proportionally to the voting threshold, whereas the recall decreases. This clearly points out the fact that the predictions of those models are different, otherwise recall would keep constant as we increase the number of votes to validate a positive prediction. For the test phase, we used the ensemble threshold $v = 3$, which resulted in the best overall (micro) F_1-score in the training phase.

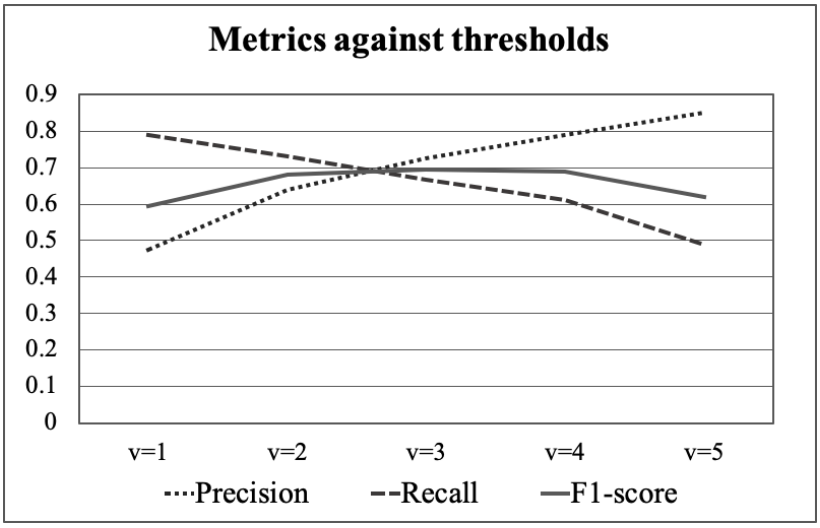


FIGURE 2 – Validation of the voting strategy.

3.2 Test phase

In the test phase, we evaluated 3 runs for a dataset of 67 clinical narratives. For run 1, we used the baseline model based on CRF. For run 2, we used the camembert-bio-large model. Finally, for run 3, we used an ensemble based on a voting threshold of 3. The performance of our models is summarised in Table 5. The ensemble model achieves 0.7262 of overall micro F_1-measure, surpassing in 2.6% the camembert-bio-large and in 14.8% the baseline. Taking into account camembert-bio-large and baseline models, the former is better by 12.18%. This clearly shows that transformer methods in biomedical French NER reach outstanding performance by only leveraging wealth in unstructured data and without the necessity to design handcrafted features. Concerning the ensemble model, it achieved the best overall F_1-measure for both subtasks among our models, being the highest score for subtask 1 and for the "all categories" evaluations among all models in the competition.

Similarly to the training phase, the highest F_1-measure in the test phase is achieved for the *valeur* entity (0.8561). This entity represents 9% of the annotations in the test collection, while in the whole data collection it represents 16%. Thus, it seems that the training data is sufficiently characterized to learn this entity automatically. The lowest performance for the ensemble method is found for *dose* entity, as well we can confirm the lowest performance for this entity in Table 4 (during training phase). This can be due to the variety of values in the annotated data, combining numbers and words (e.g. *de 0,5 à 0,75 litre*), measure units (e.g. *1mg/kg/j*) or simply words that easily could be associated with a non-entity word (e.g. *'24 paquets/année'* or *'02'*). *Mode* entities mostly are words without

abbreviations neither numbers (e.g. *'voie parasternale droite'* or *'voie centrale intraveineuse'*) i.e. it contains less variety in the kind of values, this could come with an easier way to learn patterns and make predictions.

Task 3	F_1-measure	baseline	bert-base-multilingual-cased*	camembert-large*	camembert-bio-large	ensemble (t=3)
Subtask 1	pathologie	0.3984	0.3628	0.5617	0.5344	0.5644
	sosy	0.5091	0.5574	0.6318	0.6268	0.6733
	Overall	0.4984	0.5303	0.6225	0.6153	**0.6603**
Subtask 2	anatomie	0.5561	0.7646	0.8024	0.7978	0.8069
	dose	0.3684	0.3604	0.5385	0.4118	0.5217
	examen	0.6787	0.6842	0.7169	0.7149	0.7333
	mode	0.3423	0.5935	0.6543	0.6386	0.6486
	moment	0.6273	0.6748	0.7219	0.7576	0.7869
	substance	0.578	0.5586	0.6667	0.6702	0.6379
	traitement	0.4756	0.4598	0.5724	0.5573	0.6076
	valeur	0.7969	0.8160	0.8637	0.8393	0.8561
	Overall	0.6151	0.6894	0.7441	0.7370	**0.7547**
All categories	**Overall** (micro)	0.5778	0.6380	0.7073	0.6996	**0.7262**
	Overall (macro)	0.5331	0.5832	0.6730	0.6549	**0.6837**

TABLE 5 – Test phase results. *Non-official runs.

Table 6 shows the statistics of the official results for all participants in the challenge. For subtask 1, our voting approach resulted in a F_1-score of 0.6603, which is the max reported for the overall result of the competition. In subtask 2, our best model was 1% lower than the top score, which achieved a F_1-score of 0.7626 (against our 0.7547). Considering both tasks across all categories, our voting model achieved the highest score in the competition. This results is shown in the "Non official" of Table 6 provided by the challenge organisers. The "Non official" results takes into account even entities that were counted as informational (e.g. *date*, *durée*, *frequence*). As we did not predict any of those informational entities, our F_1-measure for those are 0.0000 and we end up with a diminished overall F_1-measure of 0.7152, which is the max reported in the non official row. However, without taking into account those entities, our overall F_1-measure is 0.7262 as reported in Table 5.

Task 3	Min	Max	Median	Mean
Subtask 1	0.0645	0.6603	0.4557	0.4347
Subtask 2	0.1352	0.7626	0.6151	0.6012
Non official	0.1297	0.7152	0.5679	0.5533

TABLE 6 – Official summarize results over DEFT task 3.

3.3 CamemBERT *vs.* CamemBERT bio

If we compare the results of the camembert-base model against the camembert-bio-base model, we notice a significant improvement in performance for the latter. However, this result is unexpectedly not translated to the large version of the camembert model, as locally trained models tend to have superior performance (Lee *et al.*, 2019). We believe that this is due to the size of the biomedical corpus (31k French abstracts from PubMed) used to pre-train the CamemBERT models, which is relatively small compared to the size of the original CamemBERT corpus. For a comparison, BioBERT (Lee *et al.*, 2019) was pre-trained on 18B words corpora extracted from PubMed and PMC ; while Clinical BERT (Alsentzer *et al.*, 2019) was trained on clinical text from approximately 2 million notes in the MIMIC-III v1.4 database. While the biomedical French corpus works well for the smaller model, it was limited to improve the original camembert-large model weights for the specificities of the biomedical language as this model contains much more parameters than the base version (335M vs. 110M parameters).

Nevertheless, what makes our approach powerful is the dissimilarities of the respective model predictions. Indeed, if camembert-bio and CamemBERT models were to predict the same entities for a given text, the voting would not have made any sense. Our hypothesis is that, by creating different models, we were able to start our fine-tuning with a language model that has different perspectives. Then, by allowing each model to vote, we were able to outperform the camembert-large model by two basis points. To verify this hypothesis, it would be interesting to see if fine-tuning the same model 5 times (number of models we used for voting) would have improved its performances. For example, would the camembert-large have improved if we had fine-tuned it 5 times and used those 5 models as an ensemble ? The only randomness in such experiment would be the order of the documents during the training phase.

3.4 Language specific *vs.* multi-language model

Bert-base-multilingual-cased model was trained in 104 languages, including French, however camembert-base model was trained and optimized specifically for French language. The camembert-base model (not part of the official evaluation) shows slightly better performance for most entities (*pathologie*, *sosy*, *anatomie*, *dose*, *examen*, *moment*, *traitement*, *valeur*) compared to the bert-base-multilingual-cased model.

For subtask 1, camembert-base achieves 0.5802 of overall F_1-measure *vs* 0.5303 of bert-base-multilingual-cased, i.e. almost 5% of improvement. In subtask 2, an overall F_1-measure of 0.7081 of camembert-base *vs* 0.6894 of overall F_1measure of bert-base-multilingual-cased, makes camembert-base 1.87% better. These improvements in camembert-base highlight a direct relationship between language and performance. In addition, these differences show that subtask 1 is more dependent on the language than subtask 2.

4 Conclusion

Among the experiments we have done, we can conclude that recognizing entities in biomedical domain is not a straightforward task and adding the complexity of language makes this task more difficult. For DEFT task 3 challenge, we proposed mainly two families of learning methods: a baseline

based on CRF and a set of transformers language models. We focused on exploring the performance of our NER models with the contextualized language models enriched with local text. Our best results were given by the ensemble method based on a voting strategy between the BERT based models, including CamemBERT (trained specifically for French) and CamemBERT-bio (trained on French biomedical texts), achieving 66% F_1-measure in subtask 1 and 75% F_1-measure in subtask 2. The ensemble of deep neural language models proved to be the most effective method for biomedical information extraction in French texts. As next steps, we will investigate whether fine-tuning the same model a number of times would improve performance. We are also interested to investigate nested entities approaches over DEFT task 3 data.

Références

ALSENTZER E., MURPHY J., BOAG W., WENG W.-H., JINDI D., NAUMANN T. & MCDERMOTT M. (2019). Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, p. 72–78, Minneapolis, Minnesota, USA : Association for Computational Linguistics. DOI : 10.18653/v1/W19-1909.

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 *DEFT 2020 - Défi fouille de texte*, France.

COPARA J., OCHOA LUNA J. E., THORNE C. & GLAVAŠ G. (2016). Spanish NER with word representations and conditional random fields. In *Proceedings of the Sixth Named Entity Workshop*, p. 34–40, Berlin, Germany : Association for Computational Linguistics. DOI : 10.18653/v1/W16-2705.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 4171–4186, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1423.

DIAS G., Éd. (2015). *Actes de TALN 2015 (Traitement automatique des langues naturelles)*, Caen. ATALA, HULTECH.

DUPONT Y. (2017). Exploration de traits pour la reconnaissance d'entités nommées du Français par apprentissage automatique. In *TALN 2017*, Orléans, France. HAL : hal-02448614.

GRABAR N., CLAVEAU V. & DALLOUX C. (2018). CAS : French corpus with clinical cases. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, p. 122–128, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/W18-5614.

GRABAR N., CLAVEAU V. & DALLOUX C. (2019). **28**(01), 218–222. DOI : 10.1055/s-0039-1677937.

GUO J., CHE W., WANG H. & LIU T. (2014). Revisiting embedding features for simple semi-supervised learning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 110–120, Doha, Qatar : Association for Computational Linguistics. DOI : 10.3115/v1/D14-1012.

HOWARD J. & RUDER S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume*

1 : Long Papers), p. 328–339, Melbourne, Australia : Association for Computational Linguistics. DOI : 10.18653/v1/P18-1031.

KINGMA D. P. & BA J. (2014). Adam : A method for stochastic optimization. *arXiv preprint arXiv :1412.6980*.

KUDO T. & RICHARDSON J. (2018). SentencePiece : A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing : System Demonstrations* : Association for Computational Linguistics. DOI : 10.18653/v1/d18-2012.

LAFFERTY J. D., MCCALLUM A. & PEREIRA F. C. N. (2001). Conditional random fields : Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, p. 282–289, San Francisco, CA, USA : Morgan Kaufmann Publishers Inc.

LEE J., YOON W., KIM S., KIM D., KIM S., SO C. H. & KANG J. (2019). BioBERT : a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*. DOI : 10.1093/bioinformatics/btz682.

LIU F., CHEN J., JAGANNATHA A. & YU H. (2016). Learning for biomedical information extraction : Methodological review of recent advances. *CoRR*, **abs/1606.07993**.

LIU Y., OTT M., GOYAL N., DU J., JOSHI M., CHEN D., LEVY O., LEWIS M., ZETTLEMOYER L. & STOYANOV V. (2019). Roberta : A robustly optimized BERT pretraining approach. *CoRR*, **abs/1907.11692**.

MARTIN L., MULLER B., ORTIZ SUÁREZ P. J., DUPONT Y., ROMARY L., VILLEMONTE DE LA CLERGERIE É. & SAGOT B. (2020). CamemBERT : a Tasty French Language Model. In *The 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, Seattle, Washington, United States.

MIKOLOV T., SUTSKEVER I., CHEN K., CORRADO G. & DEAN J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, p. 3111–3119, Red Hook, NY, USA : Curran Associates Inc.

NÉVÉOL A., GROUIN C., LEIXA J., ROSSET S. & ZWEIGENBAUM P. (2014). The QUAERO French medical corpus : A ressource for medical entity recognition and normalization. In *Proc of BioTextMining Work*, p. 24–30.

PENNINGTON J., SOCHER R. & MANNING C. (2014). Glove : Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* : Association for Computational Linguistics. DOI : 10.3115/v1/d14-1162.

PETERS M., NEUMANN M., IYYER M., GARDNER M., CLARK C., LEE K. & ZETTLEMOYER L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)* : Association for Computational Linguistics. DOI : 10.18653/v1/n18-1202.

SANKHAVARA J. & MAJUMDER P. (2019). Advances in biomedical entity identification : A survey. In *Biotechnology and Biological Sciences*, p. 114–120. CRC Press. DOI : 10.1201/9781003001614-19.

SILEO D., PRADEL C., MULLER P. & DE CRUYS T. V. (2017). Synapse at cap 2017 NER challenge : Fasttext CRF. *CoRR*, **abs/1709.04820**.

SUTTON C. (2012). An introduction to conditional random fields. *Foundations and Trends® in Machine Learning*, **4**(4), 267–373. DOI : 10.1561/2200000013.

TCHECHMEDJIEV A., ABDAOUI A., EMONET V., ZEVIO S. & JONQUET C. (2018). SIFR annotator : ontology-based semantic annotation of french biomedical text and clinical notes. *BMC Bioinformatics*, **19**(1). DOI : 10.1186/s12859-018-2429-2.

TEODORO D., GOBEILL J., PASCHE E., RUCH P., VISHNYAKOVA D. & LOVIS C. (2010). Automatic ipc encoding and novelty tracking for effective patent mining. In *The 8th NTCIR Workshop Meeting on Evaluation of Information Access Technologies : Information Retrieval, Question Answering, and Cross-Lingual Information Access*, p. 309–317, Tokyo, Japan.

TEODORO D., KNAFOU J., NADERI N., PASCHE E., GOBEILL J., ARIGHI C. N. & RUCH P. (2020). UPCLASS : a deep learning-based classifier for UniProtKB entry publications. *Database*, **2020**. DOI : 10.1093/database/baaa026.

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.

TURIAN J., RATINOV L.-A. & BENGIO Y. (2010). Word representations : A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, p. 384–394, Uppsala, Sweden : Association for Computational Linguistics.

TVARDIK N., KERGOURLAY I., BITTAR A., SEGOND F., DARMONI S. & METZGER M.-H. (2018). Accuracy of using natural language processing methods for identifying healthcare-associated infections. *International Journal of Medical Informatics*, **117**, 96–102. DOI : 10.1016/j.ijmedinf.2018.06.002.

VAN MULLIGEN E. M., AFZAL Z., AKHONDI S., VO D. & KORS J. (2016). Erasmus mc at clef ehealth 2016 : Concept recognition and coding in french texts. In *CEUR Workshop Proceedings*, p. 171–178.

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e 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

Approche supervisée de calcul de similarité sémantique entre paires de phrases

Khadim Dramé[1, 2] Gorgoumack Sambe[1, 2] Ibrahima Diop[1, 2] Lamine Faty[1, 2]
(1) Université Assane Seck de Ziguinchor, Diabir, Ziguinchor, Sénégal
(2) Laboratoire d'Informatique et d'Ingénierie pour l'Innovation, Ziguinchor, Sénégal
khadim.drame@univ-zig.sn, gsambe@univ-zig.sn,
ibrahima.diop@univ-zig.sn, lamine.faty@univ-zig.sn

RÉSUMÉ

Ce papier décrit les méthodes que nous avons développées pour participer aux tâches 1 et 2 de l'édition 2020 du défi fouille de textes (DEFT 2020). Pour la première tâche, qui s'intéresse au calcul de scores de similarité sémantique entre paires de phrases, sur une échelle de 0 à 5, une approche supervisée où chaque paire de phrases est représentée par un ensemble d'attributs a été proposée. Des algorithmes classiques d'apprentissage automatique sont ensuite utilisés pour entrainer les modèles. Différentes mesures de similarité textuelle sont explorées et les plus pertinentes sont combinées pour supporter nos méthodes. Différentes combinaisons ont été testées et évaluées sur les données de test du DEFT 2020. Notre meilleur système qui s'appuie sur un modèle Random Forest a obtenu les meilleures performances sur la première tâche avec une EDRM de 0,8216.

ABSTRACT

Supervised approach to compute semantic similarity between sentence pairs.

In this paper, we present the methods that we developed to participate in tasks 1 and 2 of the 2020 edition of the french text mining challenge (DEFT 2020). For the first task, which focuses on semantic similarity computation between sentence pairs, a supervised approach where sentence pairs are first represented by a set of features has been proposed. Classical machine learning algorithms are then used to train the models. Different measures of textual similarity are explored and the most relevant are combined to support our methods. Different combinations were tested and evaluated on test data of the DEFT 2020. Our best system based on a Random Forest model performed best on the first task with an EDRM of 0,8216.

MOTS-CLÉS : similarité sémantique, phrases parallèles, méthodes supervisés, apprentissage automatique, forêts aléatoires, perceptron multicouche.

KEYWORDS: semantic similarity, parallel sentences, supervised methods, machine learning, Random Forest, Multilayer Perceptron.

1 Introduction

Le défi fouille de textes (DEFT) est une campagne d'évaluation visant à promouvoir le développement de méthodes et d'applications dans le domaine du traitement automatique de langues naturelles (TALN). Dans son édition de 2020, il continue à s'intéresser à l'analyse de cas

Atelier DÉfi Fouille de Textes, pages 49–54.
HAL : hal-02784741.

cliniques rédigés en Français; les tâches 1 et 2 portent sur la similarité sémantique entre phrases tandis que la troisième tâche s'intéresse à l'extraction d'information fine à partir de textes biomédicaux (Cardon et al., 2020).

La tâche 1 du défi consiste à déterminer le degré de similarité entre paires de phrases, sur une échelle de 0 à 5. La tâche 2, quant à elle, consiste, pour une phrase source donnée, à identifier à partir de phrases cibles fournies celle qui est parallèle à cette dernière. Ces questions se rapportent au calcul de similarité sémantique entre phrases.

Dans la littérature, ce problème a été largement exploré et différentes mesures de similarité sont proposées (Agirre et al., Agirre et al.,2015; 2016; Cer et al., 2017). Certaines approches couramment utilisées exploitent la structure syntaxique des phrases ; le nombre de tokens ou n-grams en commun entre la phrase source et la phrase cible est généralement calculé. D'autres tentent de prendre en compte les problèmes de synonymie et la sémantique des phrases en exploitant des ressources sémantiques ou des méthodes statistiques (Chen et al., 2020). Dans les dernières campagnes d'évaluation, les méthodes supervisées se sont montrées performantes pour mesurer la similarité sémantique entre phrases (Cer et al., 2017; Mojarad et al., 2018).

Les méthodes que nous proposons s'inscrivent dans cette dernière approche et utilisent les algorithmes classiques d'apprentissage automatique pour déterminer les scores de similarité entre les paires de phrases. Ce papier décrit les méthodes développées pour participer aux deux premières tâches du DEFT 2020. Le reste du papier est structuré comme suit : les méthodes de calcul de similarité sémantique sont présentées dans la section 2; les résultats obtenus sont décrits et discutés respectivement dans les sections 3 et 4.

2 Méthodes de calcul de similarité entre phrases

Dans cette section, nous décrivons les trois méthodes que nous avons développées pour le calcul de similarité entre paires de phrases.

Nous proposons une approche supervisée où chaque paire de phrases est représentée par un ensemble d'attributs. Différentes mesures de similarité sémantique sont explorées : les mesures de similarité basées sur les tokens (mesure de Dice (Dice, 1945), mesure de Ochiai (OCHIAI, 1957), mesure de Jaccard (Jaccard, 1912)), les mesures utilisant les séquences de caractères (Q-gram (Ukkonen, 1992)), la distance d'édition de Levenshtein (Levenshtein, 1966), les mesures basées sur la représentation vectorielle (TF.IDF (Jones, 2004) et les plongements lexicaux (Mikolov et al., 2013) combinées avec le cosinus).

Tout d'abord, chaque paire de phrases, qui est une instance, est représentée par un ensemble d'attributs, constitués par les scores des mesures de similarité citées ci-dessus. Ensuite, des algorithmes classiques d'apprentissage automatique sont utilisés pour entrainer les modèles, qui sont ensuite utilisés pour déterminer la similarité entre des paires de phrases non annotées.

Différents algorithmes d'apprentissage sont expérimentés mais seuls les résultats des modèles Random Forest (RF) et MultiLayer Perceptron (MLP) qui ont donné les meilleures performances sur les jeux de données d'entrainement sont soumis. De plus, nous avons développé un modèle de régression linéaire (LR) qui prend en entrée les scores de similarité de ces deux modèles et le score moyen des différentes mesures de similarité.

Une validation croisée est effectuée sur les jeux de données d'entrainement pour sélectionner les attributs les plus pertinents mais aussi pour déterminer les meilleures combinaisons. La combinaison de quatre mesures de similarité sémantique (Dice, Ochiai, Q-gram, Levenshtein) a donné les meilleures performances.

3 Evaluation

Dans cette section, nous allons d'abord présenter les jeux de données et les métriques utilisées pour évaluer les systèmes participants au DEFT 2019. Ensuite, les résultats de nos méthodes seront analysés et discutés.

3.1 Jeux de données

Pour chaque tâche, les organisateurs du DEFT 2020 ont fourni des jeux de données annotées (Grabar, Claveau & Dalloux, 2018; Grabar & Cardon, 2018). Pour la première tâche, un corpus d'entrainement constitué de 600 paires de phrases a été fourni avec, pour chaque paire, son score de similarité. Chaque paire de phrases est annotée manuellement avec un score indiquant le degré de similarité des phrases. Les données sont annotées indépendamment par deux experts qui attribuent des scores de similarité entre les paires de phrases allant de 0 (complètement différentes) à 5 (sémantiquement équivalentes). Ensuite, les annotations de référence font l'objet d'un accord entre les deux annotateurs. Le corpus de test est quant à lui constitué de 410 paires de phrases.

3.2 Mesures d'évaluation

L'exactitude en distance relative à la solution moyenne (EDRM) et la corrélation de Spearman sont utilisées pour mesurer les performances des systèmes participants à la tâche 1. Pour la deuxième tâche, la MAP (Mean Average Precision) est utilisée pour évaluer les résultats.

3.3 Résultats

Les résultats de nos différents systèmes participants à la tâche 1 sur les jeux de données de test officiels sont présentés dans TABLE 1. Nous remarquons que le système *uasz-run2*, utilisant le modèle perceptron multicouche (MLP), a obtenu des résultats largement meilleurs selon l'exactitude en distance relative à la solution moyenne. Notons également que le système *uasz-run1*, qui utilise le modèle Random Forest (RF), est plus performant que *uasz-run3*, qui lui combine les scores de similarité de ces deux modèles dans un modèle de régression linéaire. Nous avons également expérimenté une approche supervisée combinant les différentes mesures de similarité avec différents classifieurs (Naive Bayes, MultiLayer Perceptron, Support Vector Machine et Random Forest) mais les systèmes présentés ont obtenu les meilleurs résultats sur les jeux de données d'entrainement.

Systèmes	EDRM	Corrélation de Spearman
uasz-run1	0,7946	0,7527
uasz-run2	**0,8216**	0,7691
uasz-run3	0,7755	**0,7768**

TABLE 1 : Résultats de nos systèmes participants à la tâche 1 du DEFT 2020 sur les jeux de données de test officiels

Comparé aux différents systèmes participants à la tâche 1, *uasz-run2*, notre meilleur système, a obtenu les meilleurs résultats (avec une EDRM de 0,8216). Nous notons aussi que tous nos systèmes ont obtenu une EDRM dépassant la moyenne (0,762). Nos deux meilleurs systèmes, *uasz-run2* et *uasz-run1* ont également obtenu une EDRM supérieure (*uasz-run2*) ou égale (*uasz-run1*) à la médiane (0,795).

4 Discussion

L'évaluation de nos différentes méthodes de calcul de similarité sémantique sur les jeux de données du DEFT 2020 montre la bonne performance des algorithmes classiques d'apprentissage automatique pour cette tâche. Les résultats montrent également la pertinence des mesures de similarité utilisées pour capturer la similarité sémantique entre phrases. Les méthodes développées permettent toutes d'estimer correctement la similarité de la plupart des paires de phrases du corpus de test. Une analyse des résultats a permis de relever des limites de ces méthodes pour la prédiction des scores de similarités pour certaines paires de phrases. Les mesures de similarité utilisées (Dice, Ochiai, Q-gram, Levenshtein) ne prennent pas en compte la dimension sémantique des phrases ; par conséquent, nos méthodes peinent à prédire correctement les scores de similarité des phrases qui ont des structures syntaxiques similaires mais sont sémantiquement différentes. Par exemple, pour la paire de phrases 22 (id=22) du corpus de test, toutes les trois méthodes ont estimé les deux phrases similaires avec un score de similarité de 4 tandis que le degré de similarité fourni par les experts est de 1. De manière analogue, nos méthodes sont limitées pour la prédiction de la similarité des phrases sémantiquement proches mais utilisant des mots différents. Par exemple, les phrases de la paire 52 (id=52) sont considérées comme différentes avec un score de similarité de 0 tandis qu'elles sont similaires selon les experts (avec un degré de similarité de 4).

Nous remarquons également que les méthodes proposées, et particulièrement *uasz-run1* et *uasz-run2* utilisant respectivement le modèle Random Forest et le perceptron multicouche, peinent à prédire les classes les moins représentatives (1 et 2) dans le corpus d'entrainement. Dans le jeu de données officiel de test, les classes 1 et 2 sont respectivement 37 et 28. *uasz-run1* ne prédit aucune valeur de ces deux classes tandis que *uasz-run2* prédit seulement 9 valeurs dans la classe 1.

5 Conclusion

Dans ce papier, nous avons présenté les méthodes que notre équipe a développées pour participer aux tâches 1 et 2 du DEFT 2020. Trois méthodes de calcul de similarité sémantique entre phrases ont été proposées : une méthode utilisant le modèle Random Forest (RF), une autre utilisant le perceptron multicouche (MLP) et une dernière combinant les résultats de ces deux modèles. Les résultats officiels du DEFT 2020 montrent que notre méthode basée sur le perceptron multicouche a

obtenu les meilleures performances sur la tâche 1. Comparés aux différents systèmes participants, les deux autres méthodes ont donné des résultats encourageants. Nous envisageons d'exploiter d'autres mesures de similarité notamment celles permettant de capturer la sémantique des phrases afin d'améliorer les performances ; une première expérimentation avec les plongements lexicaux sur un corpus moyen n'a pas permis d'améliorer les résultats. Leur utilisation sur un corpus plus conséquent pourrait permettre d'augmenter les performances des systèmes.

Remerciements

Nous remercions les organisateurs du DEFT 2020.

Références

AGIRRE E., BANEA C., CARDIE C., CER D., DIAB M., AGIRRE A. G. & GUO W. (2015). SemEval-2015 Task 2: Semantic Textual Similarity, English, Spanish and Pilot on Interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 252–263. Denver, Colorado: Association for Computational Linguistics. https://doi.org/10.18653/v1/S15-2045.

AGIRRE E., BANEA C., CER D., DIAB M., AGIRRE A. G., MIHALCEA R., RIGAU G. & WIEBE J. (2016). SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 497–511. San Diego, California: Association for Computational Linguistics. https://doi.org/10.18653/v1/S16-1081.

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*.

CER D., DIAB M., AGIRRE E., GAZPIO I. L. & SPECIA L. (2017). SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 1–14. Vancouver, Canada: Association for Computational Linguistics. https://doi.org/10.18653/v1/S17-2001.

CHEN Q., DU J., KIM S., WILBUR W. J. & LU Z. (2020). Deep Learning with Sentence Embeddings Pre-Trained on Biomedical Corpora Improves the Performance of Finding Similar Sentences in Electronic Medical Records. *BMC Medical Informatics and Decision Making* 20 (1): 73. https://doi.org/10.1186/s12911-020-1044-0.

DICE L. R. (1945). "Measures of the Amount of Ecologic Association Between Species." *Ecology* 26 (3): 297–302. https://doi.org/10.2307/1932409.

GRABAR N. & CARDON R. (2018). CLEAR – Simple Corpus for Medical French. In *Proceedings of the 1st Workshop on Automatic Text Adaptation (ATA)*, 3–9. Tilburg, the Netherlands: Association for Computational Linguistics. https://doi.org/10.18653/v1/W18-7002.

GRABAR N., CLAVEAU V. & DALLOUX C. (2018). CAS: French Corpus with Clinical Cases. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, 122–128. Brussels, Belgium: Association for Computational Linguistics. https://doi.org/10.18653/v1/W18-5614.

JACCARD P. (1912). The Distribution of the Flora in the Alpine Zone.1. *New Phytologist* 11 (2): 37–50. https://doi.org/10.1111/j.1469-8137.1912.tb05611.x.

LEVENSHTEIN V. I. (1966). Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *SPhD* 10 (February): 707.

MIKOLOV T., SUTSKEVER I., CHEN K., CORRADO G. & DEAN J. (2013). Distributed Representations of Words and Phrases and Their Compositionality. *ArXiv:1310.4546 [Cs, Stat]*, October. http://arxiv.org/abs/1310.4546.
OCHIAI A. (1957). Zoogeographical Studies on the Soleoid Fishes Found in Japan and Its Neighhouring Regions-II. *NIPPON SUISAN GAKKAISHI* 22 (9): 526–30. https://doi.org/10.2331/suisan.22.526.
MOJARAD M. R., LIU S., WANG Y., AFZAL N., WANG L., SHEN F., FU S. & LIU H. (2018). BioCreative/OHNLP Challenge 2018. In *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, 575. BCB '18. Washington, DC, USA: Association for Computing Machinery. https://doi.org/10.1145/3233547.3233672.
JONES K. S. (2004). A Statistical Interpretation of Term Specificity and Its Application in Retrieval. *Journal of Documentation* 60 (5): 493–502. https://doi.org/10.1108/00220410410560573.
UKKONEN E. (1992). Approximate String-Matching with q-Grams and Maximal Matches. *Theoretical Computer Science* 92 (1): 191–211. https://doi.org/10.1016/0304-3975(92)90143-4.

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e 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

DEFT 2020 - Extraction d'information fine dans les données cliniques : terminologies spécialisées et graphes de connaissance

Thomas Lemaitre[1], Camille Gosset[1], Mathieu Lafourcade[1], Namrata Patel[2,3], Guilhem Mayoral[3]
(1) LIRMM, Université Montpellier, Montpellier, France
(2) AMIS, Université Paul-Valéry Montpellier 3, Montpellier, France
(3) Onaos, Montpellier, France
(1) prenom.nom@lirmm.fr
(2) prenom.nom@univ-montp3.fr
(3) prenom.nom@onaos.com

RÉSUMÉ

Nous présentons dans cet article notre approche à base de règles conçue pour répondre à la tâche 3 de la campagne d'évaluation DEFT 2020. Selon le type d'information à extraire, nous construisons (1) une terminologie spécialisée à partir de ressources médicales et (2) un graphe orienté basé sur les informations extraites de la base de connaissances généraliste et de grande taille - JeuxDeMots.

ABSTRACT

Fine-grained Information Extraction in Clinical Data : Dedicated Terminologies and Knowledge Graphs

This paper presents our rule-based approach for fine-grained information extraction in clinical data, submitted in reponse to Task 3 at the DEFT 2020 evaluation campaign. We design (1) a dedicated medical terminology from existing medical references and (2) a knowledge graph based on the semantically rich knowlege base - JeuxDeMots.

MOTS-CLÉS : données cliniques, extraction d'information fine, graphes de connaissance.

KEYWORDS: clinical data, fine-grained information extraction, knowledge graphs.

1 Introduction

Nous présentons dans cet article les méthodes que nous avons conçues pour répondre à la tâche 3 de la campagne d'évaluation DEFT 2020 (Cardon *et al.*, 2020). Cette tâche porte sur l'extraction d'informations dans un corpus de cas cliniques (Grabar *et al.*, 2018). Notre système est conçu à base de règles et s'appuie sur deux sources de données différentes :

— une base de connaissances JeuxDeMots qui est généraliste et de grande taille (Lafourcade, 2015)
— Une terminologie médicale dédiée construite pour la détection de chacune des catégories d'informations à extraire.

Un ensemble de règles est ensuite appliqué pour détecter les différentes catégories d'information recherchées. Ce travail a été réalisé en collaboration avec un expert du domaine médical, qui a été principalement impliqué dans la constitution de notre terminologie spécialisée ainsi que la

Atelier DÉfi Fouille de Textes, pages 55–65.
HAL : hal-02784742.

définition de règles d'extraction d'information ayant un fondement médical. Notre choix d'approche symbolique nous a permis également de faire une analyse fine des résultats produits au fur et à mesure de l'évolution du système, afin d'en améliorer les lexiques et les règles.

2 Méthodologie

Afin de répondre à la tâche d'extraction d'informations fines dans des dossiers cliniques, nous adoptons des méthodes à base de règles s'appuyant sur deux sources de données différentes, selon le type d'information à extraire.

Pour les catégories spécialisées autour du patient et des pratiques cliniques, nous avons construit une terminologie dédiée à partir de ressources existantes, organisée et complétée selon les besoins de la tâche. Un ensemble de règles est ensuite appliqué pour détecter les différentes catégories d'information recherchées. Nous appliquons les pré-traitements classiques liées aux approches symboliques à base de règles (tokénisation, lemmatisation, étiquetage en parties du discours, etc.) afin de générer nos résultats.

Pour les catégories autour du temps et des traitements médicaux, nous explorons une nouvelle approche exploitant la base de connaissances JeuxDeMots, basée sur le parcours d'un graphe orienté.

2.1 Extraction d'informations autour des patients et des pratiques cliniques

L'extraction automatique des informations de cette catégorie de données, en particulier la classe *signe ou symptôme* qui selon le DEFT définit à la fois des symptômes cliniques mais aussi les observations ou résultats d'examens complémentaires, est essentielle à l'établissement d'un diagnostic médical précis. L'association de ces éléments permet en effet, de définir pour un patient donné, un état pathologique particulier.

Même selon une approche automatisée, il est indispensable pour un médecin utilisateur d'un système informatique d'accéder aux explications et de pouvoir critiquer la catégorisation des entités trouvées automatiquement dans le dossier clinique de son patient. Pour cette raison nous avons, en collaboration avec un médecin, choisi une approche symbolique d'extraction d'information, permettant une transparence complète sur les résultats du système.

L'algorithme d'extraction d'information a été défini selon la démarche suivante :

- constitution d'une terminologie spécialisée cohérente aux annotations de référence
- définition de règles linguistiques permettant de compléter la détection de termes issus de la terminologie
- définition de règles médicales pour les cas complexes, tels que *signe ou symptôme* ou *pathologie*
- amélioration récursive de la terminologie et des règles par analyse des faux positifs (FP) et faux négatifs (FN) générés à chaque itération.

2.1.1 Constitution d'une terminologie spécialisée

Ressource principale Les données issues des ressources "Logical Observation Identifiers Names and Codes" (LOINC) et "Unified Medical Language System" (UMLS) ont servi de support à la constitution d'une base de données spécialisée en terminologie médicale. Chacune des entrées UMLS étant classées par un « semantic type », nous avons sélectionné celles qui correspondent aux catégories définies pour la tâche 3 du DEFT 2020, notamment les informations liées aux *sites anatomiques*, *pathologies* et *signes ou symptômes*. Ainsi, les « semantic type »issus de l'UMLS ont été rapprochées des catégories d'entités recherchées, en accord avec le guide d'annotation du DEFT 2020. Nous détaillons dans ce qui suit notre sélection de classes UMLS :

Anatomie :
- ANAT|Anatomy|T017|Anatomical Structure
- ANAT|Anatomy|T029|Body Location or Region
- ANAT|Anatomy|T023|Body Part, Organ, or Organ Component
- ANAT|Anatomy|T030|Body Space or Junction ...

Pathologie :
- DISO|Disorders|T049|Cell or Molecular Dysfunction
- DISO|Disorders|T047|Disease or Syndrome
- DISO|Disorders|T191|Neoplastic Process

Signes ou symptômes :
- DISO|Disorders|T184|Sign or Symptom
- DISO|Disorders|T033|Finding
- DISO|Disorders|T046|Pathologic Function

Examen :
- PROC|Procedures|T060|Diagnostic Procedure
- PROC|Procedures|T059|Laboratory Procedure
- PHEN|Phenomena|T034|Laboratory or Test Result

Traitement :
- PROC|Procedures|T061|Therapeutic or Preventive Procedure

Substance :
- CHEM|Chemicals and Drugs|T131|Hazardous or Poisonous Substance
- CHEM|Chemicals and Drugs|T125|Hormone
- CHEM|Chemicals and Drugs|T121|Pharmacologic Substance
- CHEM|Chemicals and Drugs|T127|Vitamin
- CHEM|Chemicals and Drugs|T195|Antibiotic
- CHEM|Chemicals and Drugs|T200|Clinical Drug
- LIVB|Living Beings|T002|Plant

Ressources supplémentaires Nos terminologies spécialisées ont été complétées selon les besoins de la tâche en fonction de termes manquants des ressources principales, celles-ci étant riches en anglais, mais pas entièrement traduites en français.

2.1.2 Écriture de règles linguistiques et médicales avec la terminologie spécialisée

Grâce aux étapes de prétraitement des approches classiques d'extraction d'information (tokenisation, etc.), nous avons élaboré des règles linguistiques et médicales visant à raffiner la détection des entités issues des terminologies.

Les entités plus complexes telles que *signes ou symptômes* ou *pathologie* ont nécessité la définition de règles complexes, exploitant à la fois (1) des informations linguistiques (étiquettes des parties du discours, dépendances morphologiques, etc.) et (2) des informations médicales combinant souvent plusieurs entités médicales, elles-mêmes extraites par le système.

Par exemple, le terme « sténose » issu du lexique *signe ou symptome* a permis au système de détecter l'entité complète suivante :

> « *sténose complète de l'uretère gauche au niveau de la jonction iliopelvienne étendue environ sur 5 cm* »

par application d'une règle qui détecte le groupe nominal associé à « sténose », contenant des entités *anatomie* (uretère gauche, jonction iliopelvienne) et *valeur* (5 cm).

2.2 Extraction d'informations autour du temps et des traitements médicaux

Les entrées du système sont des textes tirés de dossiers patient. Le but est de détecter des entités nommées. Une entité nommée peut représenter un « moment » ou une « durée », par exemple. Nous souhaitons utiliser une bonne structure de données capable de regrouper un maximum d'informations sur les éléments du texte. Notre choix de structure de données s'est orienté vers une structure de graphe. Cette structure de graphe est couplée à une base de connaissances JeuxDeMots qui est généraliste et de grande taille (Lafourcade *et al.*, 2015). L'intérêt de notre structure est de pouvoir regrouper toutes les informations récupérées de JeuxDeMots pour chaque élément du texte. Cela permet ensuite d'obtenir une trace concrète d'inférences faites par notre système. JeuxDeMots est l'outil d'un programme de recherche en Traitement Automatique du Langage Naturel conçu par Mathieu Lafourcade au sein de l'équipe TEXTE du LIRMM (Lafourcade, 2007). JeuxdeMots est un jeu sérieux appartenant à la famille des jeux avec buts, et permet d'enrichir un réseau lexico-sémantique représentant les connaissances avec des relations orientés, typées et pondérées.

2.2.1 Représentation des connaissances

Comme pour le réseau de JeuxDeMots, notre structure de graphe utilise des relations orientés, typées et pondérées. Pour chaque mot de notre texte, nous créons un nœud dans lequel nous pouvons récupérer la chaîne de caractères associée, l'index du début de la chaîne de caractères dans le texte et l'index de fin . L'ordre des mots est représenté par la relation `r_succ` entre un nœud et son nœud successeur.

Une fois notre graphe de base obtenu, nous utilisons JeuxDeMots pour effectuer une lemmatisation et l'étiquetage des mots (*Postagging*). La lemmatisation consiste à déterminer la racine d'un mot. Par exemple, le mot « mangera » a pour forme lemmatisée « manger ». L'étiquetage des mots, quant à lui, consiste à assigner à un mot une ou plusieurs classes grammaticales possibles. Par exemple, le mot « attaque », hors contexte, a les classes grammaticales de nom commun et de verbe. Pour réaliser ces

deux opérations nous utilisons respectivement les relations `r_lemma` et `r_pos` qui relient un nœud donné avec ses formes lemmatisées et ses étiquettes.

Ensuite, il est primordial de ne pas perdre le sens d'un mot en le décortiquant. Lorsque le texte est mis sous forme de graphe, chaque mot est décortiqué un à un. Cependant, le mot peut être dit composé. Un mot composé est une succession de plusieurs mots qui donne un sens précis. Par exemple, le mot composé « mettre les pieds dans le plat » obtient un sens propre à cette expression différent des mots successifs un à un. Afin de retrouver ces mots composés, nous utilisons un dictionnaire de mots composés bien fourni extrait depuis JeuxDeMots. Nous parcourons notre graphe en prenant une suite de mots et vérifions l'existence d'un tel mot composé dans le dictionnaire.

Finalement, nous utilisons un ensemble de règles générales. Ces règles permettent de réaliser une analyse syntaxique du texte. L'analyse syntaxique permet de s'intéresser à la structure du texte en établissant des relations entre les mots basés sur des motifs. Grâce à cet ensemble de règles, le système va créer de nouveaux nœuds. En s'appuyant sur l'étiquetage des mots, il va former de nouveaux groupes. Par exemple, on obtiendra un groupe nominal à partir d'un nom et un adjectif. Ces deux mots sont naturellement successeurs. Cette phase permet également de retrouver des mots composés constitués de plusieurs mots consécutifs qui ne se réfèrent pas forcément au dictionnaire précédemment utilisé.

Lorsqu'une règle est appliquée, celle-ci créée le nœud du groupe de mot. Puis, elle relie ce nœud nouvellement créé au reste du graphe, afind de créer un chemin alternatif dans le graphe. Grâce à la relation `r_isa` de JeuxDeMots, il est possible de vérifier si certains mots représentent des mois ou des moments. Cela permet de regrouper une suite de mots et d'améliorer la détection des mots qui doivent être annotés avec l'utilisation d'un second ensemble de règles spécialisées pour le DEFT.

2.2.2 Utilisation des règles

Précédemment, nous avons expliqué que nous utilisions un système de règles afin de regrouper des groupes de mots. Nous avons, donc, rédigé un ensemble de règles. Ce système de règles ainsi que les règles elles-mêmes ont été rempli par nos soins. Le but de cette analyse syntaxique est n'utiliser que peu de règles afin de ne pas exploser en complexité computationnelle.

Chaque règle est constituée de deux parties : une partie de conditions et une partie d'application. La partie de conditions peut être composée du symbole $\&$ et/ou du symbole $\|$. Cette partie contient un ensemble de conditions non limité. Lorsque des conditions sont entourées par le symbole $\&$ alors cet ensemble se doit de respecter toutes les conditions. Lorsque des conditions sont entourées par le symbole $\|$, alors seulement une condition se doit d'être respectée. Le système réalise une évaluation paresseuse. Si la partie de conditions est validée alors la partie d'application pourra être lancée sur les nœuds sélectionnés. La partie d'application peut contenir plusieurs actions. Chacune des actions pourra être appliquée sur les nœuds sélectionnés préalablement. Une condition ou une application est représentée par un triplet.

Deux type de triplets sont possibles pour la partie condition. Le premier type est *triplet link* qui permet de vérifier l'existence d'une relation entre deux nœuds variables. Un nœud variable peut être remplacé par un nœud constante. Par exemple, on cherchera à sélectionner les liens de type $\$x\ r\_pos\ Nom$:. Le deuxième type est *triplet equals* et permet d'indiquer si une chaîne de caractère d'un nœud est bien égale à un autre nœud ou à une valeur donnée.

Lorsque l'on lance la phase de conditions, on récupère un ensemble de nœud. Dans le cas où cet ensemble n'est pas vide, la partie application pourra lancer les triplets sur ces nœuds sélectionnés. Deux types de triplet d'application sont possibles. Le premier consiste à créer un nouveau lien et se nomme *triplet link*. Il s'utilise de la même manière que celui de la partie condition. Le second consiste à créer un nouveau nœud représentant la composition de plusieurs nœuds ou bien du label de la constante indiquée (*triplet makeNode*).

Basé sur ce même système règles, l'utilisateur peut poser des questions sur le graphe ou sur des informations liées au graphe et présentes dans JeuxDeMots. Notre système de règles est extensible et réutilisable. Nous l'avons donc facilement adapté afin de répondre aux questions. Une requête est de la forme $condition \rightarrow Return : \ elements\ a\ retourner$. Le but est de savoir quels nœuds correspondent à la condition de la condition pour ensuite retourner les nœuds sélectionnés.

2.2.3 Génération des résultats

Finalement, lorsque l'ensemble des règles est appliqué nous parcourons une dernière fois l'ensemble des nœuds du graphe dans le but de récupérer les nœuds représentants les entités détectées. Pour chaque nœud nous recherchons des relations indiquant des annotations possibles. Chaque annotation possède un nœud dans le graphe et les nœuds détectés comme une entité nommée possèdent une relation `-r_annoted->` vers les annotations. Par exemple : "durant 3 mois" `r_annoted` « Durée ». Ainsi, pour chaque nœud d'entités détectées, nous rajoutons dans le fichier de sortie, la chaîne de caractères associée avec son annotation, et ses index de début et de fin.

3 Résultats

Nous présentons ci-dessous les résultats issus des deux approches décrites. Notre choix d'approche symbolique nous permettant de réaliser une analyse fine des résultats obtenus, nous distinguons deux manières de calculer les mesures d'évaluation : (1) les mesures strictes (officielles) et (2) des mesures plus souples qui prennent en compte les annotations partiellement correctes.

La mise en perspective du différentiel de performance entre ces deux mesures permet d'isoler les entités détectées de manière « aberrante » (FP) et entièrement non-détectées par le système (FN). Cette distinction nous permet de discuter finement les divergences obtenues entre notre système et les annotations de référence.

3.1 Autour du patient et des pratiques cliniques

Mesures officielles	TP	FP	FN	Précision	Rappel	F1
anatomie	631	297	489	0,6800	0,5634	0,6162
examen	417	217	400	0,6577	0,5104	0,5748
substance	171	107	142	0,6151	0,5463	0,5787
traitement	125	170	179	0,4237	0,4112	0,4174
signe ou symptôme	356	540	923	0,3973	0,2783	0,3274
pathologie	69	277	97	0,1994	0,4157	0,2695

Mesures souples	TP	FP	FN	Précision	Rappel	F1
anatomie	818	110	317	0,8815	0,7207	0,7930
examen	563	71	265	0,888	0,68	0,7702
substance	199	79	114	0,7158	0,6358	0,6734
traitement	210	85	101	0,7119	0,6752	0,6931
signe ou symptôme	753	143	530	0,8404	0,5869	0,6911
pathologie	116	230	50	0,3353	0,6988	0,4531

Notre analyse des résultats du système face aux annotations de référence permet d'identifier les typologies principales d'erreurs suivantes :

— mauvaises étiquettes des étapes de prétraitement
— terminologie spécialisée incomplète/divergente
— règles linguistiques/médicales non exhaustives
— règles médicales en accord avec le guide d'annotation mais en désaccord avec l'annotation de référence

Nous discutons dans ce qui suit, les détails de cette analyse par catégorie d'entité, accompagnée des exemples d'erreurs les plus notables.

Anatomie :

FN (35% partiellement corrects) : Presque 30% des entités entièrement non détectées sont représentés par des termes relatifs à des *liquides biologiques* (e.g. : "sang", "urines") et par des termes qui ne sont pas en rapport avec une partie anatomique (e.g. : "anatomopathologique", "psychomoteurs"), non présents dans notre terminologie spécialisée *anatomie*.

FP (63% partiellement corrects) : La plupart des FP restants sont des erreurs dues aux étiquettes de prétraitement. Il en existe un faible nombre (13) qui sont des oublis d'annotation de la référence.

Examen :

FN (34% partiellement corrects) : Presque 25% des entités entièrement non détectés par notre système concernent des termes génériques et mettent en évidence *la limite d'utiliser des terminologies hyperspécialisées pour répondre à la tâche*. Ainsi, notre terminologie issue de l'UMLS, intégrait des « procédures médicales diagnostiques », mais n'intégrait pas *des termes d'ordre plus général employés seuls*, tels que : "analyse", "bilan", "consultation", "examen", "exploration", "interrogatoire", "investigation", "surveillance".

FP (67% partiellement corrects) : Nous avons intégré dans notre terminologie d'examens certains noms de substances qui sont normalement mesurées dans des examens biologiques : ces termes détectés seuls ont produit des FP.

Substance :

FN (20% partiellement corrects) : Près de 30% des entités non détectés par notre système sont dues à une construction incomplète de notre terminologie spécialisée. Il s'agit de termes génériques ne renvoyant pas à un nom de substance médicamenteuse précis, mais, soit à une *classe médicamenteuse* ("anti-androgène", "antiarythmique", "traitement anti-bacillaire"), soit à des *types de traitement* ("traitement de rattrapage", "traitement local d'appoint", "traitement

prophylactique", "transfusions", "antibiothérapie", "analgésie"). Notre terminologie de référence était majoritairement constituée de noms de molécules et ne pouvait permettre à notre système de détecter ce type de termes.

FP (26% partiellement corrects) : Les FP générés par notre système ne présentent pas de cas aberrants, il s'agit d'erreurs dues au problèmes de prétraitement.

Traitement :

FN (44% partiellement corrects) : La difficulté principale avec cette catégorie a été la détection de cas complexes telles que :

> « fond du néo-vagin avait été créé en suturant par un surjet résorbable le moignon colique droit ».

FP (50% partiellement corrects) : Dans cette catégorie, les FN et les FP sont expliqués par la même difficulté.

Signe ou symptôme :

FN (43% partiellement corrects) : Pour cette catégorie on note un différentiel important de performance entre les mesures strictes et souples. Ceci s'explique par la complexité inhérente de l'information représentée. Le guide d'annotation précise qu'il s'agit *« des résultats d'observations cliniques ou à un examen avec son résultat »*. Ceci peut constituer des groupes nominaux complexes qui doivent faire appel à des règles linguistiques complexes, par exemple :

> Gold : « masse de 1,5 x 1,0 cm au niveau du pôle inférieur du rein gauche, se rehaussant après injection du produit de contraste »
> Système : « masse de 1,5 x 1,0 cm au niveau du pôle inférieur du rein gauche »

FP (73% partiellement corrects) : Le nombre aussi élevé d'annotations partiellement correctes s'explique par une discordance relevée entre nos résultats et ceux de référence en rapport avec les termes qualifiant *l'absence ou la présence* d'un *signe ou symptôme*, par exemple :

> Gold : « zone hypoéchogène centro-rénale gauche »
> Système : « présence d'une zone hypoéchogène centro-rénale gauche »

Pathologie :

FN (48% partiellement corrects) : Le nombre élevé de FN pour cette catégorie s'explique par *le manque d'exhaustivité des règles expertes* qui ne peuvent couvrir l'ensemble des cas de figure rencontrés en médecine. Ainsi que par des *divergences entre règles expertes et annotation de référence*, cette dernière annote ces termes comme *pathologie* tels que : "adénopathies", "adénopathies périphériques", "cystalgies", "oedème réactionnel important de tout le fourreau du pénis", "mydriase bilatérale aréactive", alors qu'il s'agit de *signe ou symptôme*.

FP (17% partiellement corrects) : Parmi toutes les catégories d'entités traitées, c'est la seule pour laquelle le rappel est nettement meilleur que la précision. Ceci s'explique par le grand nombre de FP générés par le système. On peut l'expliquer par une divergence entre la classe d'entités UMLS *disease or syndrome* que nous avions rapproché de *pathologie* et qui comporte en fait d'authentiques *signes ou symptômes* tels que : "hématurie", "thrombopénie", "anémie".

3.2 Autour du temps et des traitements médicaux

Mesures officielles	TP	FP	FN	Précision	Rappel	F1
dose	10	11	42	0,4762	0,1923	0,2740
mode	16	7	73	0,6957	0,1798	0,2857
moment	85	243	80	0,2591	0,5152	0,3448
valeur	155	115	277	0,5741	0,3588	0,4416

Mesures souples	TP	FP	FN	Précision	Rappel	F1
dose	12	09	40	0,5714	0,2308	0,3288
mode	20	3	69	0,8696	0,2247	0,3571
moment	156	172	28	0,4756	0,8478	0,6094
valeur	225	45	216	0,8333	0,5102	0,6329

En regardant les résultats obtenus dans ces catégories, nous pouvons observer une précision plus importante par rapport au rappel . Ceci est dû à notre approche qui priorise la précision au rappel, pour la majorité des catégories traités. Ainsi dans les catégories avec un faible rappel, les règles déjà présentent sont efficaces mais il manque des règles pour récupérer les éléments manquants. Dans le cas de la catégorie moment nous pouvons remarquer que la précision est plus faible que le rappel, ce qui montre que certaines sont trop générales et nécessitent d'être affinées.

3.3 Observations générales d'ordre médical

Concernant la détection des classes d'entités liées au patient et aux pratiques cliniques, les *pathologies* et *signe ou symptôme* sont beaucoup plus dépendantes des règles médicales que les autres classes, pour lesquelles la bonne construction des terminologies et des règles linguistiques apportent déjà de bonnes performances de détection. Or, médicalement, la distinction *pathologie* versus *signe ou symptôme* est difficile pour plusieurs raisons :

Consensus entre experts : Intrinsèquement certaines observations cliniques ou paramédicales assimilables donc, à des *signes ou symptômes*, sont aussi d'authentiques *pathologies*. Ainsi, dans les annotations de référence, clairement identifier ce qui relève de la pathologie, de ce qui relève d'observations d'examens cliniques ou paracliniques peut ne pas être consensuel, même entre experts médecins. On peut citer le cas des anomalies morphologiques évocatrices d'une affection congénitale telles que *« rein en fer à cheval »* ou *« hypospadia »*, toutes deux classées en *signes ou symptômes* par la référence du DEFT mais qui pourraient également être reclassées en *pathologie* (congénitale).

La question des « syndromes » : Selon la définition du Larousse, un syndrome est *« un ensemble de plusieurs symptômes ou signes en rapport avec un état pathologique donné et permettant, par leur groupement, d'orienter le diagnostic »* et un syndrome biologique est un *« ensemble des modifications biochimiques, physiques, sérologiques, bactériologiques caractérisant un état pathologique donné »*. Il semblerait donc que nosologiquement les syndromes puissent être assimilés à des *signes ou symptômes*, or certains syndromes sont utilisés pour qualifier d'authentiques *états pathologiques* : *« syndrome néphrotique »* par exemple.

Ambiguïté d'une règle d'annotation : Il faut relever l'ambiguïté de la règle d'annotation du DEFT suivante, concernant les tumeurs : *« Les tumeurs malignes sont annotées* pathologie *tandis que les tumeurs bénignes seront annotées* signe ou symptôme *»*. Le degré de malignité

semblait donc constituer la frontière entre *pathologie* et *signe ou symptôme*. Or une tumeur même bénigne peut constituer une pathologie, citons « l'adénome prostatique » qui est une *pathologie* bien qu'une tumeur bénigne. Le fondement médical de cette distinction reste à préciser.

Importance du contexte pour désambiguïser des termes : la désambiguïsation fait aussi appel au contexte de la phrase dans lequel apparaît le terme. Ainsi dans la phrase suivante : *« à l'examen clinique, le patient présente une hypertension artérielle »*, « *hypertension artérielle »* est à classer en *signe ou symptôme*. A l'inverse, dans l'exemple suivant issu du texte 140-2 des données de test : *« Mme R.S, âgée de 60 ans, suivie depuis 10 ans pour hypertension artérielle, a été admise pour une douleur lombaire gauche évoluant depuis 1 mois »*, « *hypertension artérielle »* pourtant annotée en *signe ou symptôme* par la référence, est sans aucun doute une authentique *pathologie*.

4 Conclusion

Dans cet article nous avons présenté les méthodes que nous avons conçues pour répondre à la tâche 3 de la campagne d'évaluation DEFT 2020. Pour les catégories liées aux patients et aux pratiques cliniques, nous avons adopté une approche classique d'extraction d'informations à base de règles et de lexiques spécialisés. Pour les catégories autour du temps et des traitements médicaux, nous avons développé une approche basée sur les graphes de connaissances, exploitant le réseau sémantique JeuxDeMots.

Nous avons constaté que la performance de notre système a été souvent bonne en précision, grâce aux règles linguistiques et médicales complexes, mais ne pouvait couvrir de manière exhaustive l'ensemble des cas de figure rencontrés en médecine. Cela nécessiterait pour les experts médicaux une charge de travail trop importante et ceci nous interroge sur la possibilité de maintenir une approche uniquement basée sur les règles dans le domaine médical.

En perspectives, nous allons explorer des approches hybrides qui permettraient d'exploiter à la fois (1) les performances des approches par apprentissage, qui sont efficaces pour le traitement de l'exhaustivité, et (2) les connaissances métier apportées par les experts médicaux.

Plus précisément, nous avons identifié lors de la détection d'entités complexes, des cas où nos règles médicales ont été prises en défaut par la non détection d'entités simples (*anatomie, valeur, examen*), servant de point d'ancrage dans le texte à l'application de ces règles. Afin de viser une meilleure couverture des cas de figure rencontrés en médecine, nous pourrions nous appuyer sur des approches par apprentissage automatique pour la détection de ces entités simples, tout en conservant nos règles d'expertise liées aux entités plus complexes telles que *signes ou symptômes* ou *pathologie*.

Références

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*.

GRABAR N., CLAVEAU V. & DALLOUX C. (2018). CAS : French corpus with clinical cases. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, p.

122–128, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/W18-5614.

LAFOURCADE M. (2007). Making people play for lexical acquisition.

LAFOURCADE M. (2015). Jeux de mots, pour quoi faire ?

LAFOURCADE M., LE BRUN N. & JOUBERT A. (2015). Jeux et intelligence collective - résolution de problèmes et acquisition de données sur le web. collection science cognitive et management des connaissances.

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e 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-

Atelier DÉfi Fouille de Textes, pages 66–78.
HAL : hal-02784743.

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

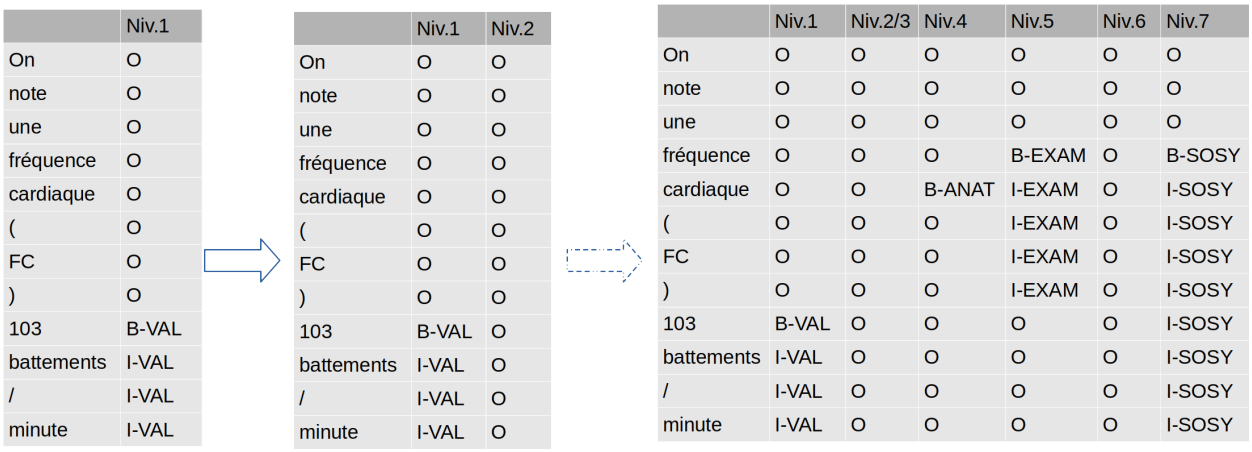
Pour le pré-traitement des fichiers, nous avons principalement utilisé SpaCy [11] (Honnibal & Montani, 2017), excepté pour le découpage en phrases car les performances du module pour le français ne nous satisfaisaient pas. En effet, il considère qu'un tiret entre deux mots indique une fin de phrase, 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.

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e 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

Extraction d'information de spécialité avec un système commercial générique

Clothilde Royan[1,2], Jean-Marc Langé [2], Zied Abidi [2]
(1) Université de Paris 6, Master Ingéniérie des Systèmes Intelligents
(2) IBM France, 17 Avenue de l'Europe, 92275 Bois-Colombes
clothilde.Royan@ibm.com, jmlange@fr.ibm.com,
zied.abidi@fr.ibm.com

RÉSUMÉ

Nous avons participé à la tâche 3 du Défi Fouille de texte 2020, dédiée à l'extraction d'information de spécialité, dans le but de tester notre produit commercial d'extraction d'information, Watson Knowledge Studio (WKS), face à des équipes académiques et industrielles.

Outre la quantité réduite de données d'apprentissage, la nature des annotations des corpus de référence posait des problèmes d'adaptation à notre produit. Aussi avons-nous dû modifier le schéma d'annotation du corpus d'apprentissage, exécuter l'apprentissage, puis appliquer des règles aux résultats obtenus afin d'obtenir des annotations conformes au schéma initial.

Nous avons également appliqué des dictionnaires de spécialité (anatomie, pathologie, etc.) pour injecter de la connaissance du domaine et renforcer les modèles d'apprentissage automatique.

Au final, nos résultats lors de la phase de test se situent dans la moyenne de l'ensemble des équipes, avec des F-mesures de 0,43 pour la sous-tâche 1 et 0,63 pour la sous-tâche 2.

ABSTRACT

Extracting Medical Information with an Off-the-shelf Software Product

We participated in the DEFT 2020 challenge, task 3, to benchmark our software product IBM Watson Knowledge Studio against academic and industry teams, in a demanding information extraction task based on clinical reports.

The data and annotation scheme was challenging for our software, so we change the original DEFT annotation scheme in order to simplify it to avoid embedded annotations and lengthy annotation spans. We apply rules to recombine the results from the ML model into annotations conformant with the original scheme.

We also use medical dictionaries to boost the ML models.

Our final results are very close to the mean values of all participating teams: F1=0,43 on subtask 1, F1=0,63 on subtask 2.

Atelier DÉfi Fouille de Textes, pages 79–90.
HAL : hal-02784744.

MOTS-CLÉS : extraction d'information, données cliniques, Watson Knowledge Studio
KEYWORDS : information extraction, clinical data, Watson Knowledge Studio

1 Introduction

La gamme IBM Watson propose un ensemble de services prêts à l'emploi pour différentes applications de traitement du langage ou de l'image basées sur l'intelligence artificielle : extraction d'information textuelle générale ou spécialisée, recherche sémantique, reconnaissance et synthèse vocale, reconnaissance d'image. Le Défi Fouille de texte 2020 (Cardon et al., 2020), et en particulier la tâche 3, était une bonne occasion de tester notre produit commercial d'extraction d'information, Watson Knowledge Studio (WKS), dans une compétition transparente entre des équipes académiques et industrielles.

2 Description des données

Les données d'entraînement pour la tâche 3 sont un ensemble de 100 cas cliniques ; pour chaque cas, on dispose d'un document contenant le texte (.txt), et d'un document au format Brat (.ann) contenant les annotations identifiées dans ce texte.

Les documents sont de longueur variable, entre 76 et 1407 mots (moyenne 361), avec une majorité de documents relativement courts (autour de 300 mots). Les textes sont spécialisés dans le domaine médical (comme prévu), et plus particulièrement en urologie, avec un vocabulaire très spécialisé.

Les données d'annotation concernent différents types d'information pour les besoins de la tâche 3 : signe ou symptôme (abrégé sosy) et pathologie pour la sous-tâche 1, anatomie, dose, examen, mode, moment, substance, traitement, valeur pour la sous-tâche 2. Dans la suite de ce document, nous mentionnerons ces catégories en majuscules (e.g. PATHOLOGIE). Dans le corpus d'entraînement, les catégories sont diversement représentées, entre 1831 instances de SOSY et 243 instances de MODE.

Les annotations de la sous-tâche 1, catégories SOSY et PATHOLOGIE, sont réputées constituer des « portions assez vastes » (DEFT 2020, 2020), qu'en est-il exactement ? La Figure 3 montre que les SOSY comprennent un nombre significatif d'instances de plus de 5 mots. Cela aura son importance lors du choix de la méthodologie à appliquer, au regard des bonnes pratiques recommandées dans l'utilisation de l'outillage Watson Knowledge Studio.

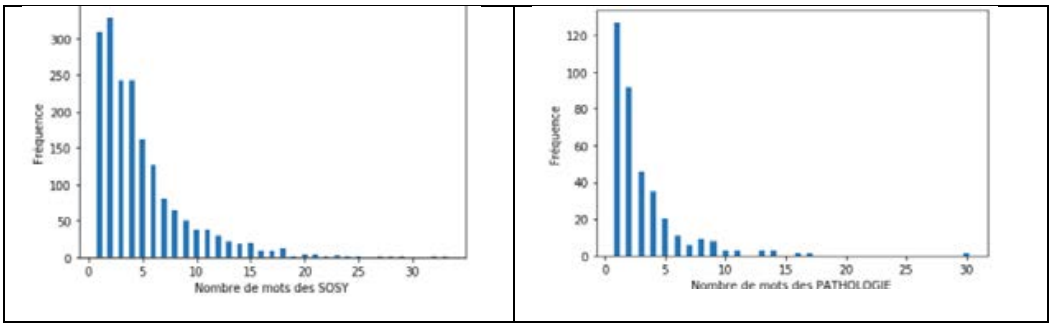


Figure 1 : portée (en mots) des annotations SOSY et PATHOLOGIE

Nous avons analysé un certain nombre d'instances de SOSY, notamment les instances de longue portée. La question qui se posait était alors : comment fait-on pour détecter une annotation d'une portée de plusieurs dizaines de mots ? Il semble facile de repérer où elles commencent, mais détecter où elles s'arrêtent l'est beaucoup moins. Le guide d'annotation (DEFT 2020b) conseille à ce sujet : *Les frontières de ces portions doivent être envisagées sous l'angle d'une seule idée par portion annotée, même si plusieurs informations peuvent venir compléter cette idée.*

3 Méthode

3.1 Aperçu Général

Nous présentons dans cette section l'outillage d'apprentissage et la méthode et l'outillage utilisés pour les deux sous-tâches ; nous aborderons les spécificités des sous-tâches dans les sections suivantes.

3.1.1 Watson Knowledge Studio

L'outil privilégié pour les tâches d'extraction d'information dans le portefeuille IBM d'IA est Watson Knowledge Studio (WKS). Nous avons choisi de participer à la campagne DEFT 2020 avec pour objectif de mesurer la performance de ce système et de le comparer à la « concurrence » sur une tâche *a priori* ardue.

WKS fait partie de la gamme IBM Watson, qui propose un ensemble de services d'IA disponibles dans le cloud. La philosophie de ces services est de fournir une IA prête à l'emploi : un service immédiatement utilisable, libérant les utilisateurs des préoccupations de choix d'algorithme, réglage des paramètres, etc. À la différence de la plupart des services Watson, disponibles sous forme d'interface de programmation (API), WKS propose quant à lui, dans une interface Web unique, de gérer le processus complet de fabrication de modèles d'extraction de connaissances à base d'apprentissage automatique :

- création du modèle de données,
- collecte des documents,
- gestion des corpus d'apprentissage/test
- préannotation automatique et annotation humaine du corpus,

- gestion des équipes d'annotateurs et de l'accord inter-annotateurs,
- préannotation automatique du corpus (avec modèles existants, dictionnaires ou règles)
- création du modèle d'apprentissage
- gestion des tests avec tableaux de bord des résultats
- versionnage des modèles d'apprentissage
- déploiement des modèles d'apprentissage

Le modèle de données (*type system*) est un modèle classique d'entités et relations. WKS permet également de gérer les *coréférences*, liens permettant de lier les mentions d'une même entité, comme pour la reprise anaphorique par des pronoms.

Il est possible -optionnellement- de renforcer l'apprentissage des modèles au moyen de dictionnaires qui contribuent à l'apprentissage sans pour autant oblitérer l'influence des autres attributs calculés à partir du contexte textuel. WKS propose en outre un **moteur de règles** permettant, dans une interface visuelle, de concevoir des modèles d'extraction de connaissances s'appuyant sur des dictionnaires et des règles.

D'un point de vue technique, le module d'apprentissage de WKS s'appuie sur un moteur de classification de séquences basé sur l'entropie maximale, descendant du système décrit dans (Radu et al., 2004).

3.1.2 WKS, adapté au Défi ?

Le fait de proposer une démarche de bout en bout, depuis l'annotation humaine jusqu'au déploiement des modèles, impose un certain nombre de contraintes qui justifient la nécessité d'adapter notre démarche en regard des spécificités des données proposées dans DEFT 2020.

En effet, WKS prescrit en partie son modèle d'annotation ; ainsi la documentation du produit (IBM, 2019) préconise les bonnes pratiques suivantes pour l'annotation d'entités :

- annoter des passages plutôt **courts** (de préférence sur 1 ou 2 mots)
- **éviter** absolument les **imbrications** d'entités. Il est préconisé, pour ce faire, d'utiliser des relations entre deux entités pour former des instances de concepts d'une portée plus étendue.

Nous savions dès le départ, avec la définition de la tâche 3 donnée en page principale du Défi 2020, que nous allions devoir nous adapter pour résoudre cette tâche.

3.1.3 Evolution de la démarche

3.1.3.1 Démarche, premier temps : utilisation du corpus d'apprentissage non modifié

Pour produire une ligne de base, nous avons dans un premier temps écrit un script de conversion du corpus d'apprentissage au format BRAT vers le format d'ingestion de WKS. Les phases classiques de tokenisation et segmentation en phrases sont effectuées

automatiquement par WKS lors de l'ingestion. La segmentation en phrases n'a posé que quelques rares problèmes, où des expressions chiffrées contenant un point sont coupées. Du fait de la rareté de ce phénomène, nous n'avons pas cherché à le rectifier.

Les figures 4 et 5 montrent un extrait du corpus d'apprentissage au format Brat et au format WKS :

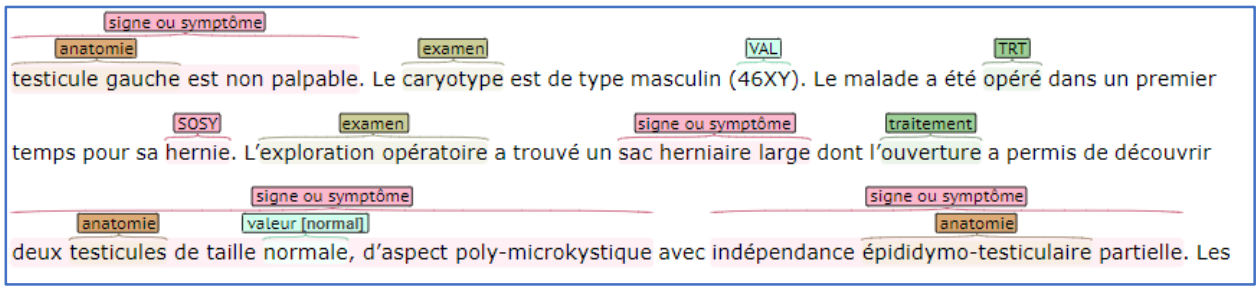


Figure 2 : extrait du corpus affiché dans BRAT

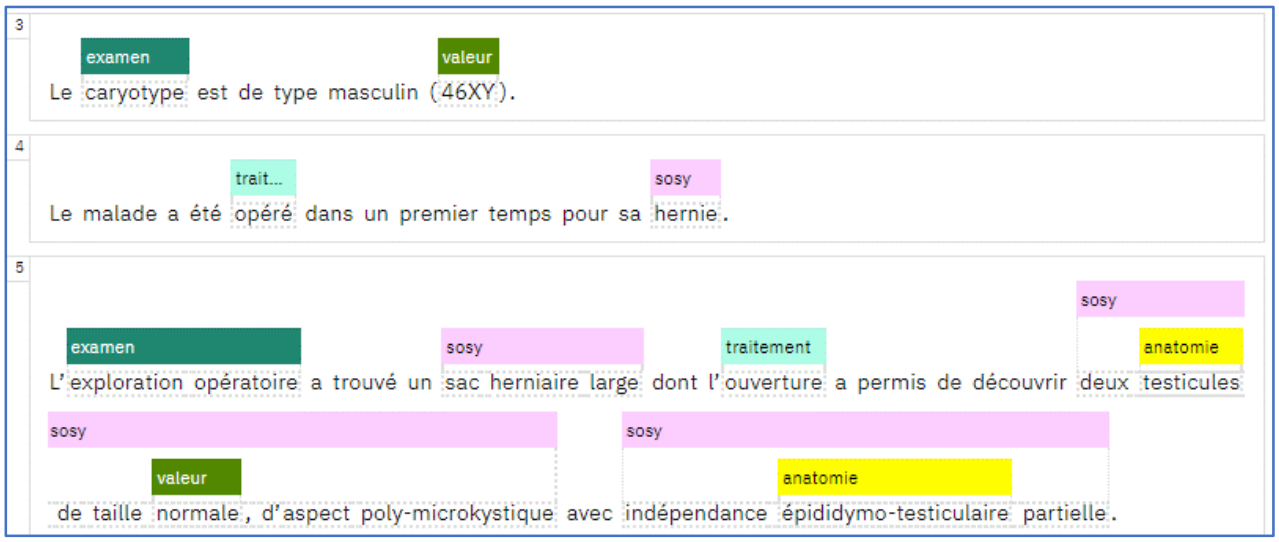


Figure 3 : extrait du corpus affiché dans WKS

Une fois le corpus disponible en format WKS, nous avons produit un premier modèle WKS, que nous avons testé sur une portion du corpus d'entraînement non vue durant l'entraînement.

Les résultats de tests avec cette approche « corpus brut » sont très faibles. Voici par exemple le résultat du modèle entraîné sur la totalité (100 documents) du corpus d'entraînement et testé sur le corpus de test DEFT 2020 :

	Précision	Rappel	F1
pathologie	0.03	0.31	0.05
sosy	0.03	0.29	0.06
anatomie	0.39	0.12	0.18
dose	0.19	0.33	0.24
examen	0.13	0.63	0.22
mode	0.00	0.00	0.00

moment	0.25	0.56	0.35
substance	0.12	0.42	0.18
traitement	0.07	0.47	0.11
valeur	0.10	0.53	0.17

Figure 4 : résultats du modèle "brut"

Pour tenter d'expliquer ce résultat, comparons un passage annoté du corpus de test avec la prédiction du modèle sur ce même passage :

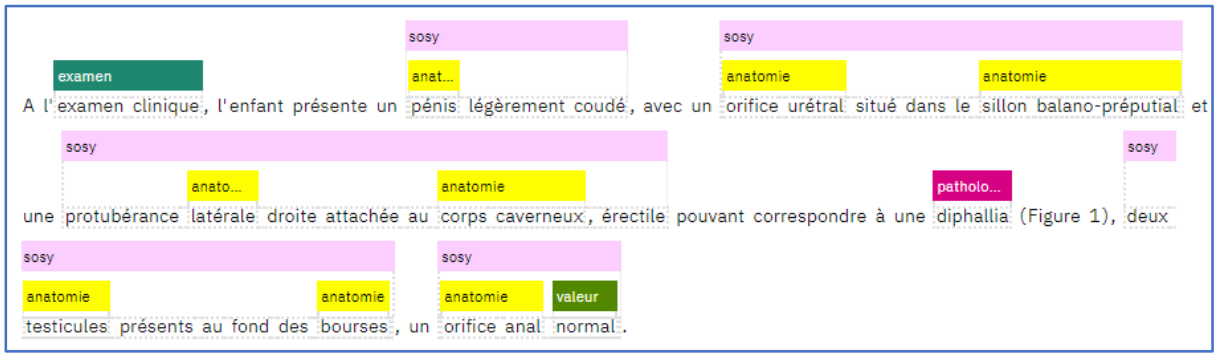


Figure 5 : Annotations du corpus de référence (interface Watson Knowledge Studio)

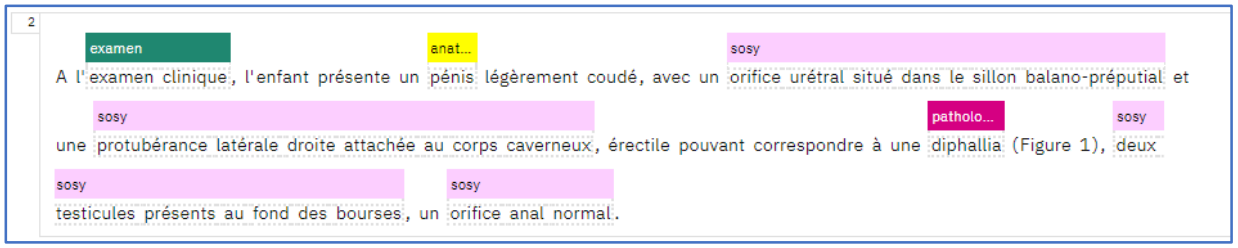


Figure 6 : Annotations prédites par le modèle "brut" (interface Watson Knowledge Studio)

Le problème est très clair : lorsque des annotations sont imbriquées (empilées dans la figure 7 ci-dessus), le modèle WKS ne prédit que la catégorie englobante (tous les SOSY sauf le premier), ou bien la ou les catégories englobées (e.g. *pénis* comme ANATOMIE dans la première phrase). De plus, les annotations longues semblent systématiquement oblitérer les annotations englobées, alors que dans le cas des annotations courtes, ce sont les englobées qui survivent. Or il y a beaucoup plus d'annotations courtes que longues dans le corpus, même pour les SOSY ; voilà qui explique les mauvais scores. Les bonnes pratiques de la documentation de WKS semblent pertinentes.

Nous avons donc dû adapter notre méthode aux contraintes posées par WKS, à savoir non-imbrication et courte portée des annotations. Nous avons donc défini un nouveau processus de traitement avec trois grandes étapes :

- **réannotation** du corpus dans WKS pour dégager des annotations plus courtes (entités au sens WKS), et en remplaçant les imbrications par des relations : cette étape a été en partie automatisée pour certaines imbrications, mais l'essentiel du travail reposait sur des annotateurs humains dans l'interface de WKS ;
- **création d'un modèle** basé sur ces réannotations ;

- **recombinaison** des entités et relations prédites par ce modèle, afin de retrouver des annotations conformes au corpus d'apprentissage DEFT 2020, notamment pour obtenir toute la portée des annotations longues telles que SOSY et PATHOLOGIE ; pour ce faire, nous avons conçu des règles de recombinaison.

Nous détaillons ces différentes étapes dans la section suivante.

3.1.3.2 Démarche, deuxième temps : réannotation du corpus et recomposition

Pour mettre en œuvre la réannotation du corpus, il nous faut au préalable examiner la structure des imbrications d'annotation pour dégager le jeu d'entités et relations, et les pratiques d'annotation qui permettent de restituer au mieux, *in fine*, les annotations originelles.

Etude des annotations longues/imbriquées

Pour étudier les patrons d'imbrication d'annotations, en particulier des annotations longues, afin de dégager le schéma d'annotation alternatif à utiliser dans WKS, nous avons procédé comme suit :

1. récupération de toutes les annotations imbriquées ;
2. suppression des mots-outils ;
3. extraction des mots de tête des annotations « coiffantes » ;
4. remplacement des passages de texte par la catégorie d'annotation ;
5. stockage du vocabulaire des passages de texte restant hors des annotations et remplacement par une marque « MOT ».

Une fois identifiés ces patrons, nous pouvons les visualiser avec des couleurs affectées à certaines entités. En parallèle, nous avons extrait le vocabulaire des passages annotés et des passages inter-annotations avec les fréquences des mots.

Cela nous a permis d'identifier les patrons les plus saillants de combinaisons d'annotation. À titre d'exemple, voici un exemple de visualisation des motifs de SOSY, ainsi qu'un extrait du vocabulaire utilisé dans ANATOMIE :

examen MOT **valeur** (77) **examen** **valeur** (56) **anatomie** MOT (17) **examen** : **valeur** 10)	gauche (31) droit (18) droite (17) rein (13) inférieure (6)	supérieure (6) inférieur (4) supérieur (4) vessie (4) col (4)

Figure 7: Patrons de SOSY et vocabulaire d'ANATOMIE

Cela nous indique clairement le genre de de réannotation que nous devons mener à bien, et les règles de recomposition à appliquer pour obtenir l'annotation finale. Cela nous suggère aussi l'opportunité de créer une nouvelle catégorie, POSITION, (*gauche*, *droit*, *inférieur*, *supérieur*…) pour mieux annoter les ANATOMIE.

Réannotation

Suite à l'étude des patrons d'imbrication, nous avons conclu à la nécessité d'étendre le jeu de catégories DEFT 2020 en y ajoutant :

- deux catégories :
  - POSITION pour rendre compte que les ANATOMIE sont fréquemment composées d'un concept d'anatomie proprement dit et de précisions de localisation : « **raphé** *antérieur* », « *extrémité proximale* du **moignon urétéral** *gauche* »
  - CARACTERE pour rendre compte de différentes précisions affectées aux têtes de SOSY et autres : **obésité** *importante*, **rétention vésicale** *complète*
- des relations entre ces catégories, afin de couvrir des portées de texte se rapprochant de l'annotation originale :

Relation	Entité source	Entité cible
A_LIEU	SOSY, PATHOLOGIE, EXAMEN, TRAITEMENT	ANATOMIE
A_POSITION	ANATOMIE et autres	POSITION
A_VALEUR	EXAMEN, SUBSTANCE	VALEUR
PORTE_SUR	TRAITEMENT, SOSY, autres	PATHOLOGIE, SUBSTANCE, …
A_CARACTÈRE	SOSY, PATHOLOGIE, EXAMEN, TRAITEMENT	CARACTÈRE
A_PRECISIONS	EXAMEN, TRAITEMENT	EXAMEN, TRAITEMENT
A_TEMPORALITE	SOSY, PATHOLOGIE, EXAMEN, TRAITEMENT	MOMENT

Figure 8 : Schéma d'annotation modifié

Nous avons ensuite procédé à modifier le corpus en appliquant ce nouveau schéma d'annotations, sans imbrication d'entités, et en visant des annotations les moins longues possibles grâce au nouveau modèle d'entités-relations. Le travail a été partagé par deux annotateurs humains, sans contrôle inter-annotateurs par manque de temps. La figure suivante montre un extrait d'un document réannoté :

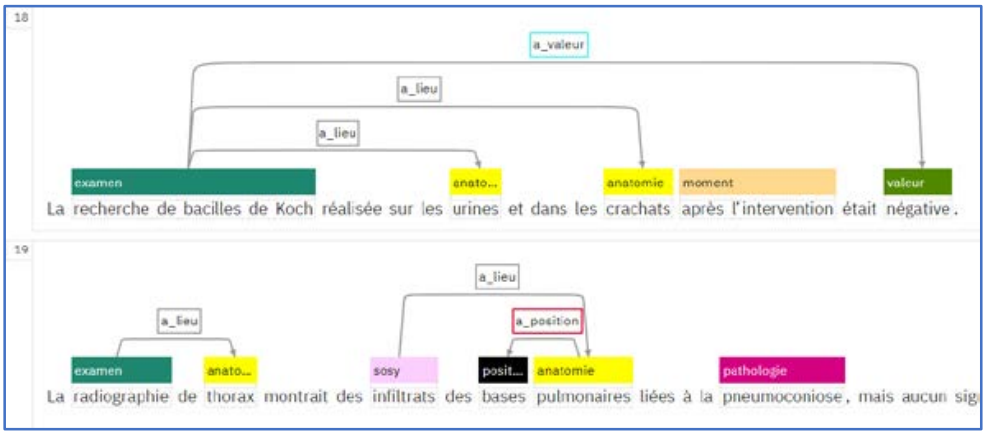


Figure 9 : Exemple de réannotation

Règles de recombinaison des entités-relations

Une fois le corpus d'entraînement réannoté, nous pouvons produire des modèles, mais les résultats de ces modèles utilisent notre schéma d'annotations modifié. Il nous faut donc recombiner, au moyen de règles, les entités et relations obtenues, afin de retrouver

des annotations conformes au corpus d'apprentissage Golden DEFT 2020, notamment pour obtenir toute la portée des annotations longues telles que SOSY et PATHOLOGIE.

Après application du modèle WKS sur le corpus de test, nous convertissons le résultat au format BRAT (.ann), puis nous appliquons les règles comme suit :

1. Reconstruction des annotations le long des relations A_CARACTERE et A_POSITION : les entités CARACTERE et POSITION constituent une brique de base dans plusieurs annotations, notamment SOSY, et il faut donc les réincorporer aux annotations de tête, auxquelles elles sont liées par les relations.

2. Reconstruction des autres relations (sauf EXAMEN <A_VALEUR> VALEUR qui fait l'objet d'une règle particulière, cf plus bas): les différentes relations qui lient les entités annotées permettent de reconstituer les annotations longues. Une fois les deux entités trouvées, nous fusionnons la mention en prenant les bornes les plus larges puis actualisons l'entité correspondant à la tête de l'annotation avec les nouvelles valeurs, avant de supprimer l'entité correspondant à la fin de l'annotation.

3. Pour les relations de type « EXAMEN <A_VALEUR> VALEUR », ce type de relation doit être annoté systématiquement comme SOSY, tout en conservant les entités EXAMEN et VALEUR dans les annotations. Nous lions d'abord des entités voisines (1 avant et 1 après) puis explorons un contexte jusqu'à 20 entités avant et 20 après l'entité qui nous intéresse. Finalement, nous cherchons des ponctuations (point, virgule, saut de ligne) qui invaliderait la qualité d'annotation.

4. Reconstruction « FREQUENCE – X – FREQUENCE » : la visualisation des patrons de combinaison d'annotations (cf supra) nous a appris que le patron « FREQUENCE – X – FREQUENCE » revenait souvent (par ex. « *4 épisodes* de vomissement *par jour* »). Dans le corpus d'entraînement, cette combinaison forme une seule annotation FREQUENCE. Pour reconstruire l'annotation canonique, nous cherchons les entités FREQUENCE puis dans leur voisinage (à 2 entités près) une autre entité FREQUENCE. Nous vérifions que la portée résultant de la fusion ne comporte pas de ponctuation ou de saut de ligne qui invaliderait le patron.

5. Reconstruction de la juxtaposition des SOSY : pour reconstruire des SOSY longs qui seraient une juxtaposition de SOSY plus courts (par exemple : *une masse tumorale, de 6 cm de diamètre avec une forme* …), nous vérifions si la position de début d'une annotation est distante de moins de *n* caractères de la valeur de fin d'une autre annotation. Si tel est le cas, nous fusionnons les annotations en prenant les bornes les plus larges puis actualisons ces valeurs dans l'entité correspondant à la tête de l'annotation.

6. Vérification des imbrications (éviter SOSY dupliqués) : Nous avons remarqué que certaines annotations étaient présentes deux fois, une fois en tant que « petite » annotation et une autre fois au sein d'une « grande » annotation. Nous vérifions toutes les annotations pour éviter cette incohérence.

7. Nettoyage des entités CARACTERE et POSITION sans relation : nous supprimons ces entités si elles sont orphelines.

Résultats de l'approche « réannotation »

La version la plus aboutie du système de règles de recomposition nous permet d'atteindre les scores suivants (ces scores sont les valeurs finales pour la tâche et tiennent compte de l'amélioration du modèle WKS avec l'apport de dictionnaires que nous verrons plus loin) :

3.1	Précision	Rappel	F1
pathologie	0.35	0.42	0.38
sosy	0.47	0.42	0.44
Overall	**0.45**	**0.42**	**0.43**

3.2	Précision	Rappel	F1
anatomie	0.75	0.59	0.66
dose	0.23	0.12	0.15
examen	0.66	0.64	0.65
mode	0.72	0.49	0.59
moment	0.72	0.46	0.56
substance	0.60	0.53	0.56
traitement	0.52	0.38	0.44
valeur	0.83	0.69	0.75
Overall	**0.70**	**0.57**	**0.63**

Figure 10 : Évaluation après application des règles de recomposition

Cette approche permet donc d'améliorer nettement les résultats par rapport à l'approche « corpus brut », mais elle reste à l'évidence limitée par les erreurs dans la détection des « petites » annotations et par les imperfections du jeu de règles. Nous n'avons pas eu suffisamment de ressources pour explorer plus avant les règles de recombinaison. En revanche, nous avons cherché à améliorer la détection des « petites » entités grâce à l'apport des dictionnaires, ce que nous verrons dans la section suivante.

3.2 Travail spécifique sur la sous-tâche 1

3.2.1 Apport des dictionnaires

WKS offrant la possibilité d'augmenter les attributs du modèle d'apprentissage au moyen de dictionnaires. Lors de l'import d'un dictionnaire, on précise l'entité à laquelle il contribue : en effet, WKS prend en compte les dictionnaires comme un attribut de plus dans la modélisation du contexte d'une entité, sans leur donner la priorité.

Nous avons utilisé des dictionnaires pour « fortifier » les annotations SOSY et PATHOLOGIE, ainsi que celles qui sont souvent imbriquées dans ces dernières, telles qu'ANATOMIE, POSITION (*supérieur, antérieur, basal, apical*…), ou EXAMEN. Ces dictionnaires sont constitués par conversion de ressources terminologiques fournis par les classifications médicales internationales telles que SNOMED ou CIM10, ou récupérés de façon ad hoc sur certains sites Web ou Wikipedia.

Le travail de collecte, nettoyage et enrichissement des dictionnaires (collecte, triage, lemmatisation, génération des formes alternatives, vérification de non-recouvrement

entre deux dictionnaires) est une tâche très chronophage et nous n'avons pas pu constituer des dictionnaires très complets et très propres dans le temps imparti, mais ils contribuent néanmoins à améliorer les résultats. À titre d'exemple, la figure ci-dessous indique l'apport (en différence de précision, rappel et F-mesure par rapport à un modèle calculé sans dictionnaire, appliqué au corpus de test), d'un dictionnaire de symptômes lié aux SOSY (à gauche), et d'anatomie, lié à ANATOMIE, à droite :

cat	diff_Pre	diff_Rec	diff_F1
anatomie	0.0047	0.0018	0.0031
dose	0.0192	0.0192	0.0216
examen	0.0035	0.0025	0.0029
mode	0	0	0
moment	0.0026	0.006	0.0052
pathologie	-0.0053	-0.006	-0.0057
sosy	-0.0013	0.0078	0.0039
substance	0.0054	-0.0032	0.0005
traitement	0.0031	-0.0098	-0.0055
valeur	0.0005	0.0023	0.0016
Overall	0.0013	0.0027	0.0021

cat	diff_Pre	diff_Rec	diff_F1
anatomie	0.0453	0.0732	0.0603
dose	-0.01	0	-0.0021
examen	-0.0096	-0.0171	-0.0136
mode	0.0038	0.0112	0.0093
moment	-0.0121	-0.0122	-0.0127
pathologie	-0.0107	-0.006	-0.0088
sosy	0.0047	0.0055	0.0051
substance	-0.0031	0.0032	0.0005
traitement	0.0005	-0.0065	-0.0042
valeur	0	0	0
Overall	0.0105	0.0152	0.0132

Figure 11 : Apport d'un dictionnaire de symptômes d'un dictionnaire d'anatomie

Ces visualisations permettent de jauger l'impact du dictionnaire, minime avec le dictionnaire de symptômes, qui contient des termes de 1 ou 2 mots, alors que les annotations SOSY couvrent typiquement plusieurs mots, voire dizaines de mots. Le dictionnaire d'anatomie a un impact non seulement sur l'entité à laquelle il est lié, mais il affecte également -en mieux ou en pire- les scores d'autres entités. Nos différents tests montrent que l'on peut gagner 3 points de F-mesure avec des dictionnaires rapidement préparés.

3.3 Travail spécifique sur la sous-tâche 2

Nous n'avons consacré que peu de ressources à cette deuxième sous-tâche, mais avons décidé de concourir car nous pouvions produire des résultats par simple apprentissage. Nous avons cependant amélioré, par un travail sur les règles ou par ajout de dictionnaires, les catégories qui contribuent aux SOSY comme EXAMEN, VALEUR, ou ANATOMIE.

4 Résultats et discussion

Au final, nos résultats (F-mesure 0,43 pour la sous-tâche 3.1 et de 0,64 pour la sous-tâche 3.2) sont très proches de la moyenne de l'ensemble des équipes. Cela valide en partie notre approche, qui a consisté à compenser les contraintes liées à notre logiciel en réannotant le corpus d'entraînement avec un schéma d'annotation modifié, puis en appliquant des règles pour recombiner les annotations obtenues afin de reconstituer les annotations d'origine. Le cas d'école est la séparation de ANATOMIE en deux entités (anatomie, position) et une relation entre les deux, qui permet à l'apprentissage automatique, en ciblant mieux ses clients, d'obtenir de meilleurs résultats. Ce principe de réannotation est parfois utilisé -de façon peut-être moins appuyée-, dans nos projets

clients avec WKS, où le travail selon des méthodes agiles induit fréquemment des ajustements du schéma d'annotation pour prendre en compte de nouveaux motifs textuels qui émergent au cours du projet.

Nous avons également amélioré les résultats en utilisant des dictionnaires de domaine.

Nous voyons plusieurs axes d'amélioration:

– Plus de données d'entraînement : notre expérience avec WKS nous indique que l'apport de données supplémentaires permettrait d'atteindre des résultats bien meilleurs pour la sous-tâche 3.2 et partiellement pour la 3.1.
– Automatisation de la réannotation : nous avons réalisé par programmation les réannotations simples, il est parfaitement envisageable de le faire de façon extensive.
– Utilisation d'un vrai moteur de règles : nos règles sont codées, mais cela est difficilement maintenable. WKS dispose d'un moteur de règles qui permettra bientôt d'appliquer les règles sur des informations extraites par le modèle d'apprentissage.
– Analyse syntaxique : extraire des annotations longues reste une tâche ardue, et il faut chercher une autre approche pour la résoudre. Une plus grande familiarité avec le corpus nous indique que les segments longs de SOSY et PATHOLOGIE suivent pour la plupart des motifs grammaticaux classiques : liens de complémentation, subordination et coordination. Les exemples abondent, comme par exemple :

kyste rénal / avec une paroi épaisse / se rehaussant / après injection / de produit de contraste.

Cela suggère l'utilisation d'un analyseur grammatical de dépendance (basé sur des règles ou sur l'apprentissage automatique) pour dérouler la pelote à partir de la tête de l'annotation jusqu'au terme de celle-ci.

Références

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

DEFT. (2020a). Défi fouille de texte 2020. https://deft.limsi.fr/2020/

DEFT. (2020b). DEFT 2020: Guide d'annotation. https://deft.limsi.fr/2020/guide-deft.html

FLORIAN R., HASSAN H., ITTYCHERIAH A., JING H., KAMBHATLA N., LUO X., NICOLOV N. & ROUKOS S. (2004). A Statistical Model for Multilingual Entity Detection and Tracking. In *Proceedings of HLT-NAACL 2004*: Boston, Mass., USA

IBM Corp. (2019). Watson Knowledge Studio User's Guide. https://cloud.ibm.com/docs/watson-knowledge-studio?topic=watson-knowledge-studio-user-guide

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e 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

DEFT 2020 : détection de similarité entre phrases et extraction d'information

Mike Tapi Nzali[1]
(1) Reezocar, 20 Rue d'issy 92100 Boulogne-Billancourt, France
mike@reezocar.com

RÉSUMÉ

Ce papier décrit la participation de Reezocar à la campagne d'évaluation DEFT 2020. Cette seizième édition du challenge a porté sur le calcul de similarité entre phrases et l'extraction d'information fine autour d'une douzaine de catégories dans des textes rédigés en Français. Le challenge propose trois tâches : (i) la première concerne l'identification du degré de similarité entre paires de phrases ; (ii) la deuxième concerne l'identification des phrases parallèles possibles pour une phrase source et (iii) la troisième concerne l'extraction d'information. Nous avons utilisé des méthodes d'apprentissage automatique pour effectuer ces tâches et avons obtenu des résultats satisfaisants sur l'ensemble des tâches.

ABSTRACT

DEFT 2020 : sentence similarity detection and information retrieval

This paper describes Reezocar's participation in the DEFT 2020 evaluation campaign. This sixteenth edition of the challenge focused on the calculation of similarity between sentences and the extraction of fine information around a dozen categories in texts written in French. The challenge proposes three tasks : (i) the first concerns the identification of the degree of similarity between pairs of sentences ; (ii) the second concerns the identification of possible parallel sentences for a source sentence ; and (iii) the third concerns the extraction of information. We used machine learning methods to perform these tasks and obtained satisfactory results on all tasks.

MOTS-CLÉS : détection de similarité sémantique, extraction d'information, apprentissage automatique.

KEYWORDS: semantic similarity detection, information extraction, machine learning.

1 Introduction

L'édition 2020 du DÉfi Fouille de Textes (DEFT) porte sur l'exploration des cas cliniques rédigés en langue française (Cardon *et al.*, 2020). Le challenge de cette édition a pour nature l'extraction d'information fine autour d'une douzaine de catégorie, mais aussi la détection de similarité sémantique entre phrases. Le défi repose sur deux corpus. Le premier corpus CAS est constitué de cas cliniques, de plusieurs pays francophones, concernant diverses spécialités médicales (cardiologie, urologie, oncologie, obstétrique, pulmonaire, gasto-entérologie, etc.) (Grabar *et al.*, 2018). Le deuxième corpus est issu du projet CLEAR, il contient des textes issus de trois sources différentes (articles

Atelier DÉfi Fouille de Textes, pages 91–96.
HAL : hal-02784745.

d'encyclopédie, notices de médicaments, et résumés Cochrane) (Grabar & Cardon, 2018). Chaque source met à disposition des versions techniques et simplifiées sur un sujet donné en français.

Dans cet article, nous présentons les méthodes utilisées lors de l'édition 2020 du DEFT. Nous avons participé aux 3 tâches proposées. La première tâche consiste à identifier le degré de similarité entre paires de phrases parallèles et non-parallèles. Cette tâche a pour objectif de déterminer le niveau de similarité entre paires de phrases sur une échelle de 0 à 5. La deuxième tâche consiste à identifier les phrases parallèles possibles pour une phrase source, son objectif étant, pour une phrase source donnée et plusieurs phrases cibles fournies, d'identifier parmi les phrases cibles celle qui est parallèle. Ces deux tâches s'effectuent sur le corpus issu du projet CLEAR. La troisième tâche est une tâche d'extraction d'information, son objectif étant de repérer dans les cas cliniques les informations fines telles que les *pathologies*, les *signes* ou les *symptômes cliniques*. . .Cette tâche s'effectue sur le corpus CAS.

Le reste de l'article sera organisé comme suit. La section 2 décrit l'approche utilisée et présente les résultats obtenues sur les tâches 1 et 2. La section 2 décrit l'approche utilisée et présente les résultats obtenues sur la tâche 3. Enfin, la section 4 conclut ce travail.

2 Méthodes pour les tâches 1 et 2

2.1 Approches

Les tâches 1 et 2 sont respectivement des tâches de similarité entre deux phrases et d'appariements entre une phrase source et plusieurs phrases cibles. Les méthodes proposées ici sont basées sur les vecteurs de poids tf-idf des mots du corpus et sur les vecteurs de poids des mots selon leur contexte. Le principe général de la méthode est de créer un ensemble de descripteurs. Ces derniers, sont les paramètres du modèle d'apprentissage. Nous avons créé plusieurs descripteurs statistiques à partir des phrases du corpus.

Vecteur de poids tf-idf. Pour chaque phrase du corpus, nous créons son vecteur tf-idf. Ensuite, nous calculons les distances entre les vecteurs des phrases pour créer nos descripteurs. Ici, les distances utilisées sont les suivantes : la *similarité cosinus*, la *distance de Manhattan*, la *distance euclidienne* et la *distance de Jaccard*. À l'issue de cette étape, nous avons donc 4 descripteurs : $cosine_{tfidf}$, $manhattan_{tfidf}$, $euclidienne_{tfidf}$, $jaccard_{tfidf}$.

Vecteur de poids selon le contexte. Nous avons aussi utilisé la méthode de transformeurs proposée dans (Reimers & Gurevych, 2019) pour transformer chaque phrase en vecteur de poids suivant le contexte des mots dans les phrases. Pour ce faire, nous avons utilisé BERT (Devlin *et al.*, 2018) et Roberta (Liu *et al.*, 2019). Ensuite, nous avons utilisé les mêmes distances présentées précédemment pour créer des descripteurs supplémentaires. À l'issue de cette étape, nous avons obtenus les 4 descripteurs suivants : $cosine_{transformer}$, $manhattan_{transformer}$, $euclidienne_{transformer}$, $jaccard_{transformer}$

Descripteurs supplémentaires. À ces 8 descripteurs, nous avons ajouté les descripteurs suivants : la longueur de chaque phrase, la différence de longueur entre les phrases et le rapport de longueurs entre les deux phrases. Nous prenons ensuite les descripteurs obtenues, et les utilisons afin d'entraîner nos modèles. Ainsi, nous allons prédire *la valeur de similarité* pour la tâche 1 et *la cible* pour la tâche 2 avec nos descripteurs.

Choix des classifieurs. Comme méthode d'apprentissage, nous avons utilisé XGBoost (*eXtreme Gradient Boosting*) (Chen & Guestrin, 2016). Il s'agit d'une implémentation open source optimisée de l'algorithme d'arbres de boosting de gradient qui utilise des approximations plus précises pour trouver le meilleur modèle d'arbre. Il utilise un certain nombre d'astuces qui lui confèrent un succès exceptionnel, en particulier avec des données structurées.

2.2 Résultats et discussions

Pour évaluer nos modèles, nous avons utilisé l' EDRM (Exactitude en Distance Relative à la solution Moyenne) (Grouin *et al.*, 2013) pour la tâche 1 et la MAP (*Mean Average Precision*) pour la tâche 2. Par rapport à d'autres participants, nous sommes au-dessus des moyennes et des médianes.

Sur la tâche 1, nous obtenons le meilleur score avec le *run* 2, son score EDRM de 81% (voir table 1). La médiane de l'EDRM de l'ensemble des systèmes soumis est de $79,5\%$. Cependant, le meilleur système du défi sur cette tâche a obtenu une valeur de $82,2\%$.

Sur la tâche 2, notre meilleur score est obtenu par le *run* 1 (voir table 2) avec une MAP de 98.7%, la médiane de la MAP de l'ensemble des systèmes soumis est de $98,68\%$. Cependant, le meilleur système du défi sur cette tâche a obtenu une valeur de $99,06\%$.

	Tâche 1		
	EDRM	Spearman Correlation	p-value
Run 1	0.792	0.706	4.15e-63
Run 2	**0.810**	**0.735**	**6.60e-71**
Run 3	0.802	0.708	1.28e-63

TABLE 1 – Résultats des différents *runs* soumis au défi pour la tâche 1

	Tâche 2
	MAP
Run 1	**0.987**
Run 2	0.981
Run 3	0.985

TABLE 2 – Résultats des différents *runs* soumis au défi pour la tâche 2

On remarque que sur la 2ème tâche, les résultats des différents *runs* 2 sont assez proches, cela s'explique par le fait que la différence entre les runs se situe au niveau du paramétrage des modèles.

3 Tâches 3

3.1 Approches

Pour effectuer cette tâche, nous avons utilisé les champs aléatoires conditionnels (CRF (Lafferty *et al.*, 2001)). Il s'agit de l'une des approches les plus efficaces pour l'étiquetage supervisé de séquences. Ils ont été appliqués avec succès pour des tâches telles que l'étiquetage morphosyntaxique (Lafferty *et al.*, 2001), l'extraction d'entités nommées (McCallum & Li, 2003).

Les CRF sont des modèles probabilistes graphiques non dirigés, conçus pour définir une distribution de probabilités conditionnelles sur des séquences d'étiquettes, étant donnée des séquences observées. Cette nature conditionnelle démarque les CRF des modèles qui nécessitent une hypothèse d'indépendance des variables, tels que les modèles de Markov cachés (HMM) (Blunsom, 2004). En pratique, une qualité des modèles CRF est leur robustesse sur des ensembles de données de petite taille. Nous avons appliqué plusieurs pré-traitements comme effectué dans (Tapi Nzali *et al.*, 2015). Nous avons créé un modèle CRF pour cette tâche et avons extrait plusieurs traits d'ordre morphologique, syntaxique et sémantique pour l'apprentissage. Les différentes descripteurs construits sont les suivantes : *Capitalisation du token*, *Longueur du token*, *Présence d'un chiffre dans le token*, *Présence de ponctuation dans le token*.

3.2 Résultats et discussions

Nous présentons les résultats obtenus dans les tables 3 et 4. Nous remarquons que, sur certaines entités nommées, nous obtenons de bon résultats comparés à d'autres. C'est le cas de *dose* et *traitement* pour lesquelles le modèle est moins performant.

	TP	FP	FN	Precision	Recall	F1
pathologie	73	79	93	0,4803	0,4398	0,4591
sosy	500	418	779	0,5447	0,3909	0,4552
Overall	573	497	872	0,5355	0,3965	0,4557

TABLE 3 – Résultats du *run 3* soumis au défi pour la tâche 3.1

	TP	FP	FN	Precision	Recall	F1
anatomie	684	214	436	0,7617	0,6107	0,6779
dose	10	15	42	0,4000	0,1923	0,2597
examen	342	301	475	0,5319	0,4186	0,4685
mode	42	11	47	0,7925	0,4719	0,5915
moment	93	36	72	0,7209	0,5636	0,6327
substance	172	93	141	0,6491	0,5495	0,5952
traitement	99	105	205	0,4853	0,3257	0,3898
valeur	302	53	130	0,8507	0,6991	0,7675
Overall	1744	828	1548	0,6781	0,5298	0,5948

TABLE 4 – Résultats du *run 3* soumis au défi pour la tâche 3.2

Sur la tâche 3, nous avons fournit 3 *runs* et ne présentons ici que celui avec lequel nous avons le meilleur résultat.

4 Conclusion

Nous avons participé aux 3 tâches proposées dans ce DÉfi Fouille de Textes (DEFT2020). Globalement, nos résultats sont assez satisfaisants, car nous sommes tout proches des meilleurs du challenge. Nous ne sommes malheureusement pas parvenus à tester des approches supplémentaires et performantes sur la tâche 3 du défi DEFT 2020 par manque de temps. Nos résultats proviennent d'approches classiques et conduisent à des résultats sans surprise. Pour améliorer les résultats de la tâche 3, Il aurait été intéressant de combiner les approches LSTM (Long Short-Term Memory) et les CRF (Huang *et al.*, 2015).

Références

BLUNSOM P. (2004). Hidden markov models. *Lecture notes, August*, **15**(18-19), 48.

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 DEFT*.

CHEN T. & GUESTRIN C. (2016). Xgboost : A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, p. 785–794.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2018). Bert : Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv :1810.04805*.

GRABAR N. & CARDON R. (2018). Clear-simple corpus for medical french.

GRABAR N., CLAVEAU V. & DALLOUX C. (2018). Cas : French corpus with clinical cases. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, p. 122–128.

GROUIN C., ZWEIGENBAUM P. & PAROUBEK P. (2013). Deft2013 se met à table : présentation du défi et résultats. *Actes du neuvième DÉfi Fouille de Textes*, p.$\tilde{2}$.

HUANG Z., XU W. & YU K. (2015). Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv :1508.01991*.

LAFFERTY J., MCCALLUM A. & PEREIRA F. C. (2001). Conditional random fields : Probabilistic models for segmenting and labeling sequence data.

LIU Y., OTT M., GOYAL N., DU J., JOSHI M., CHEN D., LEVY O., LEWIS M., ZETTLEMOYER L. & STOYANOV V. (2019). Roberta : A robustly optimized bert pretraining approach. *arXiv preprint arXiv :1907.11692*.

MCCALLUM A. & LI W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, p. 188–191 : Association for Computational Linguistics.

REIMERS N. & GUREVYCH I. (2019). Sentence-bert : Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing* : Association for Computational Linguistics.

TAPI NZALI M. D., TANNIER X. & NÉVÉOL A. (2015). Automatic extraction of time expressions across domains in french narratives.

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e 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

Similarité sémantique entre phrases : apprentissage par transfert interlingue

Charles Teissèdre Thiziri Belkacem Maxime Arens
Synapse Développement, 7 Boulevard de la Gare – 31500 Toulouse
{thiziri.belkacem, maxime.arens, charles.teissedre}@synapse-fr.com

RÉSUMÉ

Dans cet article, nous décrivons une approche exploratoire pour entraîner des modèles de langue et résoudre des tâches d'appariement entre phrases issues de corpus en français et relevant du domaine médical. Nous montrons que, dans un contexte où les données d'entraînement sont en nombre restreint, il peut être intéressant d'opérer un apprentissage par transfert, d'une langue dont nous disposons de plus de ressources pour l'entraînement, vers une langue cible moins dotée de données d'entraînement (le français dans notre cas). Les résultats de nos expérimentations montrent que les modèles de langue multilingues sont capables de transférer des représentations d'une langue à l'autre de façon efficace pour résoudre des tâches de similarité sémantique telles que celles proposées dans le cadre de l'édition 2020 du Défi fouille de texte (DEFT).

ABSTRACT

Semantic Sentence Similarity : Multilingual Transfer Learning

In this paper, we describe an exploratory approach to train language models and solve sentence-matching tasks from French corpora in the medical field. We show that, in a context where training data are limited, it may be interesting to transfer learning from a language with more training resources to a target language with less training data (French in our case). The results of our experiments show that multilingual language models are able to transfer representations from one language to another efficiently to solve semantic similarity, tasks such as those proposed in the 2020 edition of the Text Mining Challenge (DEFT).

MOTS-CLÉS : Similarité Sémantique Textuelle, Modèles Neuronaux Multilingues, Apprentissage par transfert Interlingue.

KEYWORDS: Semantic Textual Similarity, Multilingual Neural Models, Cross-lingual Transfer Learning.

1 Introduction et motivation

Mesurer la similarité entre des textes est une tâche importante dans plusieurs applications du traitement des langues et de la recherche d'information (Baziz *et al.*, 2005; Manning *et al.*, 2010; Guo *et al.*, 2016; Kusner *et al.*, 2015). Dans cet article, nous présentons des travaux expérimentaux comparant différents modèles de calcul de similarité entre phrases, pour résoudre deux tâches proposées dans le cadre de l'édition 2020 du défi fouille de textes, DEFT 2020 [1] (Cardon *et al.*, 2020). La première

1. https://deft.limsi.fr/2020/

Atelier DÉfi Fouille de Textes, pages 97–107.
HAL : hal-02784746.

tâche consiste à mesurer la similarité entre des paires de phrases sur une échelle allant de 0 à 5. Elle renvoie à une tâche d'appariement désormais classique proposée dans les campagnes SemEval de 2012 à 2017. La seconde tâche consiste à sélectionner une phrase cible similaire à une phrase source dans un ensemble de phrases candidates. Les corpus d'entraînement fournis pour ces deux tâches d'appariement entre phrases relèvent du domaine médical.

Les difficultés principales de ces deux tâches consiste dans le fait que l'on souhaite manipuler des représentations de phrases et que l'on dispose de données de faible volumétrie pour l'entraînement, en particulier pour la tâche 1 (le corpus d'entraînement contient 600 paires de phrases uniquement).

L'équipe de Synapse souhaitait à travers ce défi tester la capacité des modèles de langue multilingues, ré-entraînés avec peu de données dans une langue cible, à résoudre des problèmes d'appariement entre phrases, un problème central dans les systèmes de Question-Réponse et de recherche d'information couvrant plusieurs langues, dans un contexte où l'on dispose de peu voire même d'aucune donnée d'entraînement. L'enjeu sur un plan industriel est en effet de favoriser le développement d'applications multilingues, sans qu'il soit nécessaire de disposer à l'initialisation des systèmes, de ressources ou de données dans chacune des langues à traiter. Pour compenser le manque de données d'entraînement en français, nous avons ainsi utilisé un jeu de données de plus grande volumétrie en anglais provenant du benchmark STS [2] (STSBenchmark). Dans nos expérimentations, nous avons testé différents modèles de langue multilingues spécialisés pour générer des représentations de phrases. Nous montrons que ces modèles entraînés à partir des ressources disponibles en anglais sont en mesure de transférer des représentations latentes d'une langue à l'autre, et ainsi d'apprendre à résoudre la tâche en français. Afin d'en évaluer l'intérêt, nous comparons les résultats de cette approche à ceux d'approches concurrentes, obtenus par des méthodes d'apprentissage supervisées et des modèles d'apprentissage monolingues.

2 Les corpus et les modèles testés

2.1 Les corpus d'entraînement

Les corpus d'entraînement fournis pour les tâches 1 et 2 proviennent du projet CLEAR (Grabar & Cardon, 2018) et comprennent [3] des articles d'encyclopédie, des notices de médicaments et des résumés Cochrane, dont le contenu présente de grandes similarités d'un sous corpus à l'autre, ce qui permet de constituer des paires de phrases parallèles. Il s'agit ainsi de corpus relevant du domaine médical. Les annotations de référence ont été normalisées par consensus, après une double annotation indépendante.

La tâche 1 correspond à une transposition, sur des données en français, de la tâche de Similarité Sémantique Textuelle (Semantic Textual Similarity) telle que définie dans les campagnes SemEval (2012-2017) [4]. Étant donnés des couples de phrases, la tâche 1 invite les systèmes participants à retourner un score de similarité sur une échelle de valeurs graduées de 0 (phrases sémantiquement indépendantes) à 5 (phrases équivalentes). L'évaluation des différents systèmes mesure l'écart entre la valeur fournie et la valeur de référence. La figure 1 présente un extrait du corpus fourni pour la tâche 1. Le corpus d'entraînement associé à cette tâche contient 600 paires de phrases.

2. https://ixa2.si.ehu.es/stswiki/index.php/STSbenchmark
3. https://deft.limsi.fr/2020/
4. http://alt.qcri.org/semeval2017/task1/

```
<paire id="2" vote="5">
<source>- En l'absence d'amélioration comme en cas de persistance des
symptômes, prendre un avis médical.
</source>
<cible>En l'absence d'amélioration comme en cas de persistance des
symptômes, prenez un avis médical.
</cible>
</paire>
```

FIGURE 1 – Exemple de données du corpus de la tâche 1.

```
<ensemble id="1" cible="2">
  <source>
    compte tenu des données disponibles , l' utilisation chez la femme
enceinte ou qui allaite est possible ponctuellement
  </source>
  <cible num="1">ce médicament est un laxatif utilisé par voie
orale</cible>
  <cible num="2">ce médicament , dans les conditions normales d'
utilisation , peut être utilisé ponctuellement pendant la grossesse et
l' allaitement</cible>
  <cible num="3">boîte de 1 flacon de 250 ml ou 500 ml</cible>
</ensemble>
```

FIGURE 2 – Exemple de données du corpus de la tâche 2.

La tâche 2 consiste à identifier les phrases parallèles d'une phrase source parmi un ensemble de phrases cibles. Un extrait du corpus associé à cette tâche est présenté dans la figure 2. Dans l'exemple illustré, la phrase cible numéro 2 est une phrase parallèle à la phrase source. Le corpus d'entraînement associé à cette tâche contient un peu plus de 1700 ensembles de phrases, chaque ensemble comprenant une phrase source et trois phrases cibles.

2.2 Les modèles testés

Dans cette section, nous décrivons brièvement les différents modèles utilisés pour résoudre les tâches auxquelles nous participions, ainsi que les motivations qui nous ont conduits à tester ces différents modèles. Pour les deux tâches, nous avons utilisé des méthodes d'apprentissage supervisé devant servir de modèles de référence (baseline), puis testé différents modèles de langue multilingues.

2.2.1 Des approches supervisées comme modèles de référence

Pour la tâche 1, le modèle supervisé utilisé comme méthode de référence est un modèle de similarité proposé par (Guo *et al.*, 2016), basé sur la pertinence (Deep Relevance Matching Model for ad-hoc retrieval ou DRMM). Les auteurs montrent que les méthodes d'apprentissage profond conçues pour l'appariement sémantique ne seraient pas bien adaptées à la recherche ad-hoc. Cette dernière concerne essentiellement l'appariement par pertinence plutôt que l'appariement sémantique. Basé sur cette différence, le modèle DRMM calcule les interactions mot-mot entre les séquences d'entrée représentées dans l'espace distribué (embedding), où chaque mot est représenté par un vecteur. DRMM utilise une similarité cosinus et calcule une matrice M qui est ensuite transformée en histogrammes [5] d'interaction qui sont calculés en utilisant les valeurs de similarité entre tout les termes des séquences de texte en entrées.

Les modèles supervisés qui nous ont servis de modèles de référence pour la tâche 2 proviennent de précédentes expériences sur un corpus maison en français contenant des paires de questions parallèles. Ces modèles présentent de bons résultats sur ce corpus permettant de capter efficacement la similarité sémantique entre des questions, un problème très similaire à celui de la tâche 2, pour laquelle il faut retrouver une phrase parallèle à la phrase source dans un ensemble de phrases cibles. Pour entraîner ces modèles d'apprentissage supervisé, nous avons utilisé un ensemble de caractéristiques (features) liées aux différentes séquences d'entrée. Ces caractéristiques sont ensuite combinées dans un modèle de classification entraîné en utilisant un algorithme de descente de gradient (SGD) (Bottou, 2010) et la méthode de gradient boost (XGB) (Chen & Guestrin, 2016).

2.2.2 Modèles de langue multilingues

Les modèles de langue testés sur la tâche 1 sont des dérivés du modèle BERT (Bidirectional Encoder Representations from Transformers) (Devlin *et al.*, 2018), dont une des principales innovations techniques est l'application aux modèles de langue d'un entraînement bidirectionnel des tranformateurs (Dehghani *et al.*, 2018), modèles neuronaux basés sur le mécanisme d'attention. L'entraînement bidirectionnel de BERT exploite le contexte de gauche à droite et de droite à gauche. Les résultats de BERT sur un grand nombre de tâches classiques du TAL montrent qu'un modèle de langue entraîné de manière bidirectionnelle peut avoir une perception plus profonde du contexte et du flux linguistique, comparé notamment aux modèles de langue basés sur un contexte unidirectionnel (Radford *et al.*, 2018).

BERT multilingue (M-BERT) est le pendant multilingue de BERT (Devlin *et al.*, 2018) pré-entraîné à partir de corpus monolingues dans 104 langues différentes. Dans leurs expérimentations, (Pires *et al.*, 2019) ont montré que M-BERT est efficace dans les applications d'apprentissage par transfert appliquées aux traitements des langues. Ce modèle est notamment performant dans des applications multilingues dites zéro-shot (zero-shot transfer), où seules les représentations spécifiques à une langue sont utilisées pour affiner le modèle dans une autre langue. Pré-entraîné sur de grands corpus de textes non annotés et facilement disponibles, le modèle BERT est affiné pour des tâches spécifiques sur de plus petites quantités de données qualifiées, en s'appuyant sur la structure du modèle induit pour faciliter la généralisation au-delà des données d'entraînement. Dans (Pires *et al.*, 2019), les

5. Cette méthode sert à traduire des vecteurs de différentes dimensions dont les valeurs sont dans l'intervalle $[-1; 1]$ à un ensemble de vecteurs de même dimensions et dont les valeurs sont des entiers, en se basant sur une taille prédéfinie des intervalles de valeurs.

auteurs montrent que le transfert inter-linguistique opère d'autant mieux lorsqu'il y a un important chevauchement lexical entre les langues et qu'elles sont typologiquement proches, par exemple, entre des langues de type sujet-verbe-objet (SVO) telles que l'anglais et le français, l'espagnol ou le bulgare.

Dérivé de M-BERT et de Sentence-BERT (Reimers & Gurevych, 2019), le modèle Sentence Multilingual BERT [6] (Sentence M-BERT) que nous avons testé (101 langues) est un encoder de phrases initialisé avec M-BERT et affiné sur le corpus anglais MultiNLI (Conneau *et al.*, 2018) et sur le corpus de développement multilingue XNLI (Williams *et al.*, 2018). Les représentations de phrases sont des moyennes de vecteurs correspondant aux différents symboles (tokens) des phrases d'entrée.

Multilingual Universal Sentence Encoder [7] (MUSE) (Chidambaram *et al.*, 2018) est un modèle de représentation de phrases basé sur la traduction pour traiter plusieurs langues différentes. La traduction est effectuée sur la base d'une tâche de tri, de sorte que les réponses codées obtiennent des représentations très similaires à celles des questions correspondantes à l'aide d'une fonction de produit. MUSE utilise une seule couche d'encodage et est entraîné dans un cadre multi-tâches, où des couches supplémentaires spécifiques à la tâche en cours de traitement sont utilisées en plus de l'encodeur unique. Dans leur travail (Chidambaram *et al.*, 2018), les auteurs ont proposé en fonction des usages plusieurs modèles multilingues qui couvrent 16 langues (dont l'anglais, le français, l'espagnol et l'allemand) dans un unique espace sémantique. Ces modèles obtiennent des performances compétitives avec l'état de l'art sur différentes tâches de traitement automatique des langues (classification de textes, regroupement de textes, recherche de similarités sémantiques).

Nous avons exploités ces différents modèles multilingues dans nos expérimentations, parce qu'ils paraissaient adaptés aux tâches 1 et 2, où il s'agit de mesurer la similarité entre des couples de phrases et sachant que nous souhaitions opérer un apprentissage par transfert de langue.

3 Expérimentations et résultats

Dans cette section, nous décrivons la démarche expérimentale que nous avons suivie pour résoudre les tâches d'appariement entre phrases, ainsi que les résultats de l'évaluation des différents modèles testés.

3.1 Tâche 1 : Similarité sémantique entre phrases

Conformément à la méthodologie d'évaluation proposée dans SemEval (Cer *et al.*, 2017), nous avons évalués et comparés les modèles testés lors de nos expérimentations au moyen de la mesure de Pearson, qui permet d'établir une corrélation entre les valeurs de similarité correctes (de référence) et celles obtenues de façon automatique par les différents modèles entraînés.

La difficulté principale de la tâche 1 est que les données d'entraînement sont peu nombreuses : le jeu de données fourni par les organisateurs comprend 600 couples de phrases pour 6 classes différentes, de 0 à 5. Les prédictions obtenues à partir du modèle de langue française CamemBERT (Martin *et al.*, 2020) sur les données fournies pour DEFT 2020 ont montré des résultats intéressants (0.77 sur le corpus de développement sans exploiter les données d'entraînement), mais nous souhaitions pouvoir exploiter

6. http://files.deeppavlov.ai/deeppavlov_data/bert/sentence_multi_cased_L-12_H-768_A-12.tar.gz
7. https://tfhub.dev/google/universal-sentence-encoder-multilingual/3

Synthèse des résultats - Tâche 1		
Modèle	Corpus Dev	Corpus Test
DRMM	0.55	-
MUSE	0.77	0.73
M-BERT + STS	0.82	0.74
Sentence M-BERT + STS	0.83	0.76

TABLE 1 – Performances, en termes de corrélation de Pearson, calculée pour les différents modèles concernant les prédictions pour la tâche 1, avec et sans entraînement sur les données STS.

des jeux de données plus volumineux, bien que n'étant pas disponibles en français, en particulier le corpus associé au STSBenchmark. D'où l'idée d'exploiter des modèles de langue multilingues et de les affiner avec ce jeu de données, leur permettant ainsi de générer des représentations de phrases et d'évaluer ensuite leurs performances sur des données en français.

Comme le montrent (Reimers & Gurevych, 2019), les modèles BERT (Devlin *et al.*, 2018) / RoBERTa (Liu *et al.*, 2019) / XLM-RoBERTa (Conneau *et al.*, 2019) ne produisent pas par défaut des représentations de phrases efficaces. Une bonne approche pour résoudre une tâche de similarité sémantique de textes consiste ainsi à affiner ces modèles pré-entraînés sur des jeux de données leur permettant d'améliorer leurs représentations de phrases. Le corpus anglais NLI (Natural Language Inference) (Bowman *et al.*, 2015) et son pendant multilingue MultiNLI (Williams *et al.*, 2018) peuvent être utilisés à cette fin. Ils comprennent des couples de phrases classés selon le type de relation qu'elles entretiennent (neutre, contradiction, inférence). Pour améliorer les modèles de langue, les paires de phrases sont passées à un transformeur qui génère des vecteurs de représentation de taille fixe. Ces représentations de phrases sont alors transmises à un classifieur qui prédit le label décrivant leur relation. Ceci permet de générer des représentations qui peuvent alors servir à d'autres tâches, en particulier, pour ce qui nous retient, une tâche mesurant la similarité sémantique entre des phrases. Le modèle Sentence Multilingual BERT (Sentence M-BERT) est un modèle ré-entraîné avec les corpus NLI. Le corpus STS (Cer *et al.*, 2017) peut être également utilisé dans cette optique, ce qui fait ici pleinement sens, puisque la tâche 1 est une transposition directe de la tâche de SemEval à laquelle le corpus STS est associé. Nous avons ainsi testé des modèles dérivés de BERT avec et sans ré-entraînement avec le corpus STS.

Le tableau 1 montre la synthèse des résultats obtenus par plusieurs des modèles que nous avons testés pour résoudre la tâche 1. STS fait référence à l'utilisation du corpus du STSBenchmark comme données d'entraînement pour affiner les modèles. Un sous ensemble des données du corpus DEFT 2020 associés à la tâche 1 (504 couples de phrases) sont utilisés lors de cette phase de ré-entraînement comme données d'évaluation pour guider le fine-tuning. Le corpus de développement qui a servi à l'évaluation des systèmes (Corpus Dev) comprenait 96 phrases, soit 16 de chaque classe.

Il est à noter que les résultats obtenus sur le corpus de développement issus du corpus d'entraînement ont été calculés à partir de mesures de similarité continues (nombres réels), alors que les résultats sur le corpus de test, à la demande des organisateurs, ont été calculés après discrétisation des mesures (entiers naturels).

Le modèle MUSE pour Universal Sentence Encoder Multilingue (Yang *et al.*, 2019), sans même être entraîné avec les données de DEFT ni avec celles du STSBenchmark (les données d'entraînement de DEFT n'ont été utilisées que pour l'évaluation) obtient des résultats légèrement inférieurs aux encoders de type BERT affinés pour générer des embeddings de phrases. Ce modèle est donc un

excellent candidat pour des approches multilingues dites zero-shot.

Les modèles qui présentent les meilleurs résultats sont ceux entraînés avec les données du STSBenchmark, ce qui dans un même mouvement les spécialise sur la tâche 1 et leur permet de générer des embeddings de phrases plus riches.

3.2 Tâche 2 : Sélection de phrases parallèles

Pour nos expérimentations, nous avons décomposé la tâche 2 en trois phases : (1) une tâche consistant à mesurer la similarité des phrases cibles avec la phrase source (2) une tâche d'ordonnancement selon la mesure de similarité et (3) une tâche de sélection de la phrase présentant la meilleure similarité.

Pour résoudre cette tâche, nous avons testés différents approches supervisées ainsi qu'une approche entièrement non supervisée, sans ré-entraînement.

Les approches supervisées testées sont deux modèles de classification, le modèle SGD (Bottou, 2010) et le modèle XGB (Chen & Guestrin, 2016). Ils ont été entraînés à partir de l'extraction de différentes caractéristiques des séquences d'entrée.

Les différentes features utilisées sont les suivantes :

— *Features Simples* : elles tiennent compte de la taille des phrases ramenée en nombre de caractères, en nombres de mots, en nombres de mots en commun, ainsi que de la fréquence d'apparition de ces phrases dans la collection.

— *Features Avancées* : elles renvoient à différents ratios qui tiennent compte de la similarité du début et de la fin des phrases, de la taille des phrases à comparer, de la taille de leur partie commune ainsi que des ratios de mots-clés en commun entre les deux phrases.

— *Word Embeddings* sont les représentations distribuées (plongement lexicaux) des mots. Nous avons utilisé le modèle word mover's distance (Kusner *et al.*, 2015) afin de calculer la distance entre les deux phrases en entrée et également calculer différentes distances entre les moyennes de vecteurs de mots de chaque phrase : Cosinus, City Block (Manhattan), Canberra, Euclidienne et Minkowski.

Nous avons combiné deux types de features, *simples* et *avancées* ainsi que les représentations distribuées des mots (word embeddings). Nous avons comparé les résultats de ces modèles au modèle MUSE (Chidambaram *et al.*, 2018), un modèle multilingue de représentation universel des phrases, utilisé pour le calcul des représentations des différentes phrases à l'entrée d'un classifieur. Le tableau 2 montre une synthèse des résultats obtenus par les différents modèles testés sur cette tâche, après ordonnancement des phrases cibles par similarité et extraction de la phrase présentant le plus de similarité avec la phrase source.

Il est à noter qu'à l'issue de la première phase où l'on établit une mesure de similarité entre phrases cibles et phrases source, il est possible de calculer la capacité des modèles à opérer une classification binaire (phrases équivalentes vs phrases indépendantes). Le tableau 3 présente les résultats obtenus par les différents modèles sur cette tâche de classification. Ceci permettrait de traiter les cas où plusieurs phrases cibles seraient parallèles à la phrase source et les cas où aucune d'entre elles ne le seraient, alors que ces informations sont perdues avec la méthode de sélection du meilleur candidat. Le corpus d'entraînement ne semblait toutefois pas contenir de cas de ce type, bien si l'énoncé de la tâche 2 semblait prévoir ces cas.

Synthèse des résultats - Tâche 2		
Méthode	Modèle	P@1
Features simples + TF-IDF	SGD_Clf1	0.975
Embeddings + Features avancées	SGD_Clf2	0.975
Embeddings + Features avancées	XGB_Clf1	0.975
Multilingue (apprentissage par transfert)	MUSE	1.0

TABLE 2 – Comparaison des modèles sur la tâche 2

Synthèse des résultats intermédiaires			
Représentation des entrées	Modèle	Précision Moyenne	Précision Pondérée
Features simples + TF-IDF	SGD_Clf1	0.88	0.88
Word Embed. + Features avancées	SGD_Clf2	0.91	0.91
Word Embed. + Features avancées	XGB_Clf1	0.94	0.94
Multilingue (apprentissage par transfert)	MUSE	0.94	0.94

TABLE 3 – Comparaison des modèles pour la classification en équivalence

3.3 Résultats d'évaluation

Dans cette section, nous présentons les résultats de l'évaluation officielle après soumission des trois meilleurs systèmes (modèles) évalués dans les sections précédentes. Le tableau 4 montre les résultats d'évaluation de chacun des modèles retenus par tâche, en comparaison aux résultats de référence (performances maximales) d'après l'évaluation des différents modèles participants à l'atelier. Nous remarquons que nos systèmes présentent des niveaux de performances opposés pour les deux tâches, comparés aux autres systèmes participants.

Pour la tâche 1, les modèles que nous avons testés sont encore loin des performances des meilleurs modèles concurrents évalués dans cet atelier (Sentence-M-BERT + STS, est près de 17% moins performant par rapport au système le plus performant), mais ils montrent néanmoins qu'il est possible d'obtenir des résultats intéressants dans l'optique de produire des applications multilingues dans un contexte où l'on dispose de peu voire d'aucune donnée d'entraînement pour certaines des langues cibles. Tels quels, les modèles que nous avons expérimentés sont entraînés en utilisant des données génériques et testés sur des données du domaine médical, impliquant ainsi deux niveaux de transfert

Tâche 1	M-BERT + STS	MUSE	Sentence M-BERT + STS	maximum
Sp-Cor	0.7499	0.7421	0.7679	-
p-value	3.1295e-75	7.0960e-73	6.1899e-81	-
EDRM	0.6533	0.6663	0.6838	0.8220
Tâche 2	SGD	MUSE	XGB	maximum
MAP	0.9850	0.9906	0.9396	0.9906

TABLE 4 – Évaluation finale des trois meilleurs modèles par tâche, évalués dans les résultats des tableaux 1 et 2. La colonne *maximum* montre les meilleurs performances des différents modèles participants.

d'apprentissage, la langue et le domaine, pour une tâche assez complexe (prédiction des votes sur six classes (0-5)).

Concernant la tâche 2 où l'objectif était de trouver une phrase parallèle à une phrase source dans un ensemble de phrases cibles, le plus performant des modèles que nous avons expérimentés, le modèle MUSE, obtient les meilleures performances parmi les différents systèmes participants. L'intérêt de ce modèle est qu'il n'a pas du tout été entraîné avec les données d'entraînement fourni par les organisateurs, ce qui en fait un modèle intéressant pour le développement d'applications multilingues sans donnée d'entraînement.

4 Conclusion

Nous avons présenté dans cet article les résultats de notre participation à l'atelier DEFT 2020 avec différents modèles multilingues d'appariement de texte, permettant de résoudre des tâches 1 et 2, auxquelles nous avons participé. Les expérimentations que nous avons menées pour résoudre la tâche 1 montrent que dans un contexte où le volume de données d'entraînement dans une langue est très limité, il est possible d'opérer un transfert d'apprentissage d'une langue à l'autre en recourant à des modèles d'apprentissage profond multilingues, comme MUSE et M-BERT. Cette approche permet d'exploiter conjointement des données d'apprentissage dans une langue pour laquelle on dispose d'un plus grand nombre de données et des données moins nombreuses dans la langue cible. Cette approche obtient cependant des résultats assez loin des performances obtenues par les meilleurs systèmes participants. A l'inverse, nous avons obtenu les meilleurs résultats dans la tâche 2 avec une approche pourtant encore plus restrictive où les données d'entraînement ne sont pas du tout exploitées (modèle MUSE).

Les modèles entraînés durant nos expérimentations sont des modèles de langue généralistes, pré-entraînés sur des textes qui ne relèvent pas du domaine médical. Une piste intéressante pour poursuivre ces travaux serait d'entraîner un modèle de langue multilingue spécialisé sur ce domaine, qui permettrait de disposer de représentations de phrases mieux adaptées aux corpus. A ce jour en effet, à notre connaissance seuls des modèles monolingues de langue anglaise spécialisés dans le domaine médical sont disponibles (Lee *et al.*, 2019). En outre, une extension de ces modèles pour prendre en compte une tâche de multiclassification peut être intéressante afin de déterminer le vote correspondant à un couple de phrases en entrées.

Références

BAZIZ M., BOUGHANEM M., AUSSENAC-GILLES N. & CHRISMENT C. (2005). Semantic cores for representing documents in ir. In *Proceedings of the 2005 ACM Symposium on Applied Computing*, SAC '05, p. 1011–1017, New York, NY, USA : ACM. DOI : 10.1145/1066677.1066911.

BOTTOU L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, p. 177–186. Springer. DOI : 10.1007/978-3-7908-2604-3_16.

BOWMAN S. R., ANGELI G., POTTS C. & MANNING C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)* : Association for Computational Linguistics. arXiv preprint : L-1508.05326.

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.

CER D., DIAB M., AGIRRE E., LOPEZ-GAZPIO I. & SPECIA L. (2017). SemEval-2017 task 1 : Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, p. 1–14, Vancouver, Canada : Association for Computational Linguistics. DOI : 10.18653/v1/S17-2001.

CHEN T. & GUESTRIN C. (2016). Xgboost : A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, p. 785–794. DOI : 10.1145/2939672.2939785.

CHIDAMBARAM M., YANG Y., CER D., YUAN S., SUNG Y., STROPE B. & KURZWEIL R. (2018). Learning cross-lingual sentence representations via a multi-task dual-encoder model. *CoRR*. arXiv preprint : abs/1810.12836.

CONNEAU A., KHANDELWAL K., GOYAL N., CHAUDHARY V., WENZEK G., GUZMÁN F., GRAVE E., OTT M., ZETTLEMOYER L. & STOYANOV V. (2019). Unsupervised cross-lingual representation learning at scale. arXiv preprint : abs/1911.02116.

CONNEAU A., LAMPLE G., RINOTT R., WILLIAMS A., BOWMAN S. R., SCHWENK H. & STOYANOV V. (2018). XNLI : evaluating cross-lingual sentence representations. *CoRR*. arXiv preprint : abs/1809.05053.

DEHGHANI M., GOUWS S., VINYALS O., USZKOREIT J. & KAISER Ł. (2018). Universal transformers. arXiv preprint : L-1807.03819.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2018). Bert : Pre-training of deep bidirectional transformers for language understanding. arXiv preprint : L-1810.04805.

GRABAR N. & CARDON R. (2018). CLEAR – simple corpus for medical French. In *Proceedings of the 1st Workshop on Automatic Text Adaptation (ATA)*, p. 3–9, Tilburg, the Netherlands : Association for Computational Linguistics. DOI : 10.18653/v1/W18-7002.

GUO J., FAN Y., AI Q. & CROFT W. B. (2016). A deep relevance matching model for ad-hoc retrieval. New York, NY, USA : Association for Computing Machinery.

KUSNER M. J., SUN Y., KOLKIN N. I. & WEINBERGER K. Q. (2015). From word embeddings to document distances. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, p. 957–966 : JMLR.org. DOI : 10.5555/3045118.3045221.

LEE J., YOON W., KIM S., KIM D., KIM S., SO C. H. & KANG J. (2019). Biobert : a pre-trained biomedical language representation model for biomedical text mining. *CoRR*. arXiv preprint : abs/1901.08746.

LIU Y., OTT M., GOYAL N., DU J., JOSHI M., CHEN D., LEVY O., LEWIS M., ZETTLEMOYER L. & STOYANOV V. (2019). Roberta : A robustly optimized BERT pretraining approach. *CoRR*. arXiv preprint : abs/1907.11692.

MANNING C., RAGHAVAN P. & SCHÜTZE H. (2010). Introduction to information retrieval. *Natural Language Engineering*, **16**(1), 100–103. DOI : 10.5555/1394399.

MARTIN L., MULLER B., SUÁREZ P. J. O., DUPONT Y., ROMARY L., DE LA CLERGERIE É. V., SEDDAH D. & SAGOT B. (2020). Camembert : a tasty french language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. arXiv preprint : L-1911.03894.

PIRES T., SCHLINGER E. & GARRETTE D. (2019). How multilingual is multilingual bert ? *CoRR*. arXiv preprint : abs/1906.01502.

RADFORD A., NARASIMHAN K., SALIMANS T. & SUTSKEVER I. (2018). Improving language understanding with unsupervised learning. *Technical report, OpenAI*. Technical Report.

REIMERS N. & GUREVYCH I. (2019). Sentence-BERT : Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, p. 3982–3992, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1410.

WILLIAMS A., NANGIA N. & BOWMAN S. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)*, p. 1112–1122 : Association for Computational Linguistics. DOI : 10.18653/v1/N18-1101.

YANG Y., CER D., AHMAD A., GUO M., LAW J., CONSTANT N., ABREGO G. H., YUAN S., TAR C., SUNG Y.-H., STROPE B. & KURZWEIL R. (2019). Multilingual universal sentence encoder for semantic retrieval. arXiv preprint : L-1907.04307.

Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e 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

Participation de l'équipe du LIMICS à DEFT 2020

Perceval Wajsbürt[1], Yoann Taillé[1,2], Guillaume Lainé[1], Xavier Tannier[1]
(1) Sorbonne Université, Inserm, LIMICS, Paris, France
(2) Institut des Sciences du Calcul et des Données, Sorbonne Université, Paris, France

RÉSUMÉ

Nous présentons dans cet article les méthodes conçues et les résultats obtenus lors de notre participation à la tâche 3 de la campagne d'évaluation DEFT 2020, consistant en la reconnaissance d'entités nommées du domaine médical. Nous proposons deux modèles différents permettant de prendre en compte les entités imbriquées, qui représentent une des difficultés du jeu de données proposées, et présentons les résultats obtenus. Notre meilleur run obtient la meilleure performance parmi les participants, sur l'une des deux sous-tâches du défi.

ABSTRACT

Participation of team LIMICS in the DEFT 2020 challenge

In this article, we present the methods developed and the results obtained during our participation in task 3 of the DEFT 2020 evaluation campaign, consisting of the recognition of named entities in the medical field. We propose two different models allowing to take into account the nested entities, which represent one of the difficulties of the proposed data set, and present the results obtained. Our best run obtains the best performance among the participants, on one of the two subtasks of the challenge.

MOTS-CLÉS : Reconnaissance d'entités nommées, apprentissage profond, entités imbriquées.

KEYWORDS: Named entity recognition, deep learning, nested entities.

1 Introduction

Nous présentons dans cet article les méthodes que nous avons conçues pour répondre à la tâche 3 de la campagne d'évaluation DEFT 2020. Cette tâche porte sur la détection d'entités nommées dans des textes de descriptions de cas cliniques (Grabar *et al.*, 2018). Les différents types d'entités sont, d'une part, les *pathologies* et *signes ou symptômes* (tâche 3.1), d'autre part, l'*anatomie*, les *examens*, *substances*, *doses*, *modes d'administration*, *traitements* (chirurgical ou médical), *valeurs*, *moment*.

Le corpus se compose de 100 fichiers pour l'entraînement (8350 annotations) et 67 fichiers pour le test (3800 annotations).

Le jeu d'entraînement fourni par les organisateurs est composé de 100 documents (8350 annotations), tandis que le jeu de test est composé de 67 documents (3800 annotations). Plus de détails sur le défi sont présents dans Cardon *et al.* (2020).

Nous décrivons dans cet articles nos 3 soumissions officielles, dont l'une a obtenu la meilleure performance sur la tâche 3.2.

Atelier DÉfi Fouille de Textes, pages 108–117.
HAL : hal-02784747.

2 Méthode

Nous présentons deux approches visant à traiter le problème des mentions imbriquées, qui sont la particularité principale du jeu de données DEFT et qui sont le sujet d'un regain d'attention dans la communauté TAL depuis 2018. La première difficulté rencontrée est qu'il n'est pas aisé voire impossible, selon la nature des chevauchements, d'encoder sans perte toutes les mentions en une unique séquence de tags.

2.1 CamemBERT + Layered BiLSTM CRF

Ce modèle tente de tirer partie de la combinaison de méthodes actuellement performantes en traitement automatique des langues et en reconnaissance d'entités nommées imbriquées.

Il comprend d'abord une composante CamemBERT (Martin *et al.*, 2020), un modèle BERT pré-entraîné sur des données françaises, afin d'obtenir une représentation des mots de notre corpus qui prenne en compte leur contexte. Une couche dense est ajoutée après cette composante afin de limiter la taille des représentations et d'en faire correspondre les dimensions avec celles des couches suivantes.

Vient ensuite une succession de combinaisons de LSTM bidirectionnels et de CRF, utilisées dans le modèle de Lample *et al.* (2016). Ce modèle stratifié s'inspire de Ju *et al.* (2018). Chacune de ses strates est capable de prédire des entités à partir de séquences de représentations de mots. Si une strate prédit une entité, les représentations de la zone correspondant à cette prédiction sont fusionnées pour obtenir une représentation de l'entité, qui est transmise à la strate suivante. L'enchaînement de strates s'arrête quand aucune entité n'est détectée par la dernière strate, ou quand un nombre défini de strates est atteint.

Comme le montre la Figure 1, les strates détectent d'abord les entités les plus courtes puis les entités plus longues les incluant. La fusion des représentations quand une entité est détectée se fait en moyennant les représentations comprises dans la zone prédite. Les paramètres du LSTM de chaque strate sont partagés entre toutes les strates.

2.1.1 Apprentissage

Une première expérience à été effectuée en entraînant ce modèle de manière multiclasse, mais elle s'est révélée moins performante que plusieurs modèles binaires. Pour chacune des 10 classes, un modèle CamemBERT + Layered BiLSTM-CRF a donc été entraîné. En inspectant le corpus, nous avons fixé le nombre maximal de couches par modèle à 2, correspondant à la profondeur maximale d'imbrication d'entités nommées du même type.

2.1.2 Paramètres du modèle

La composante CamemBERT est initialisée avec les poids de CamemBERT-base, pré-entraîné sur OSCAR (Ortiz Suárez *et al.*, 2019). Seules ses 4 dernières couches sont ré-entraînées sur nos données. Les paramètres des couches suivantes sont initialisés aléatoirement selon la méthode de (Glorot & Bengio, 2010). La couche dense et les LSTM bidirectionnels ont une dimension de 200 unités, et les

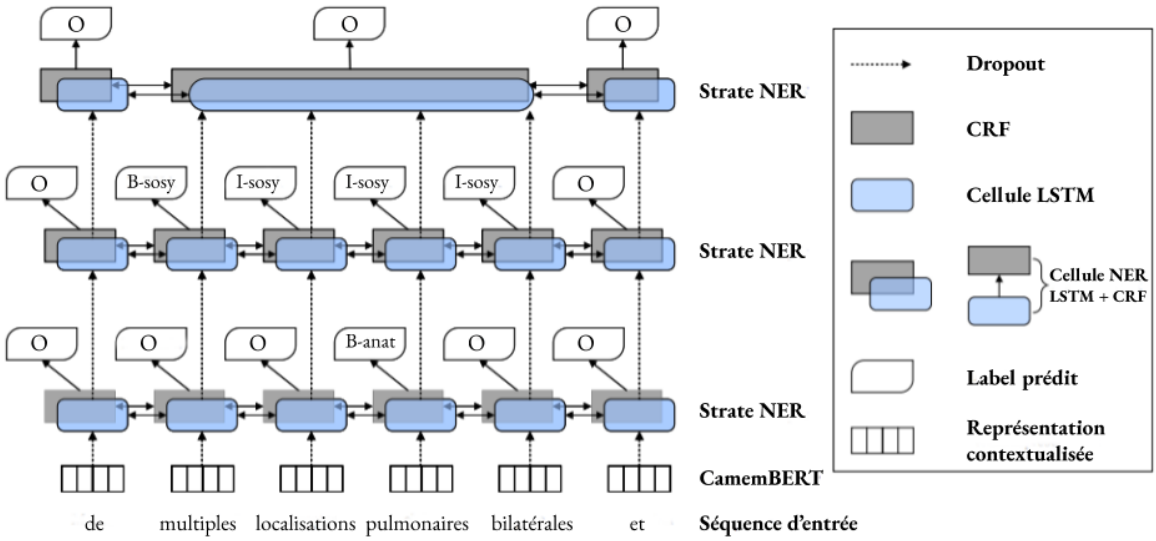


FIGURE 1 – Schéma adapté de Ju *et al.* (2018) de la structure du modèle CamemBERT + Layered BiLSTM-CRF. Dans l'exemple, "pulmonaires" est imbriquée dans "multiples localisations pulmonaires bilatérales".

Classe	sosy	moment	pathologie	mode	substance	examen	anatomie	dose	valeur	traitement
Nombre d'époques	53	42	64	41	49	41	46	45	40	82

TABLE 1 – Nombre d'époques retenu pour l'entraînement de chaque modèle binaire par classe

LSTM comportent deux couches cachées. Du dropout (Srivastava *et al.*, 2014) est appliqué avec une probabilité 0.3 dans les LSTM.

Chaque modèle binaire est optimisé par backpropagation avec Adam (Kingma & Ba, 2015) sans weight decay, avec un pas d'apprentissage de $5 \times 10^{-}5$ pour la composante CamemBERT et $5 \times 10^{-}3$ pour le reste. Le nombre d'époques d'apprentissage pour chaque modèle a été déterminé de la manière suivante : si on n'observe pas d'amélioration du coût sur les données d'apprentissage pendant 5 époques, on arrête l'entraînement. Le nombre d'itérations retenu par classe est présenté dans le tableau 1. Sur une carte graphique NVIDIA Quadro K5200, l'apprentissage sur 100 documents pour toutes les classes prend environ 25 minutes.

2.2 Iterative Greedy NER

2.2.1 Modèle

Nous détaillons ici un nouveau modèle ainsi qu'une procédure permettant d'en apprendre les paramètres. Le modèle est un Transformer (Vaswani *et al.*, 2017) prenant en entrée une séquence de mots ainsi qu'une liste de mentions déjà extraites (liste vide à la première itération), et prédit en sortie une liste de nouvelles mentions. Les mentions prédites à chaque itération ne se chevauchent pas, en revanche l'ensemble des mentions prédites à la fin des itérations peuvent se chevaucher.

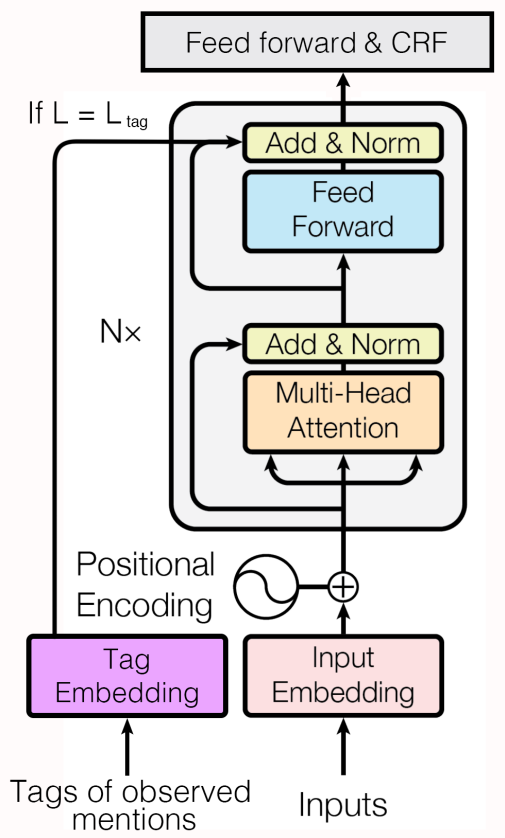


FIGURE 2 – Notre modèle basé sur un Transformer modifié pour intégrer à la fin de la couche L_{tag} les tags des mentions précédemment observées — schéma adapté de Vaswani *et al.* (2017)

Afin d'être manipulables par un réseau de neurones, les mots sont transformés en embeddings. De même, les mentions sont manipulées sous forme de tags au format BIO, chaque tag étant converti en embeddings pour être sommé avec les autres tags à la même position.

Nous calculons les probabilités des séquences de tags en chaînant un transformer CamemBERT (Martin *et al.*, 2020) avec un CRF linéaire (Lafferty *et al.*, 2001). Tandis que les embeddings de tokens sont intégrés par le transformer en amont du modèle, nous intégrons les embeddings de tags à la sortie d'une des couches du modèle (Figures 2, 3).

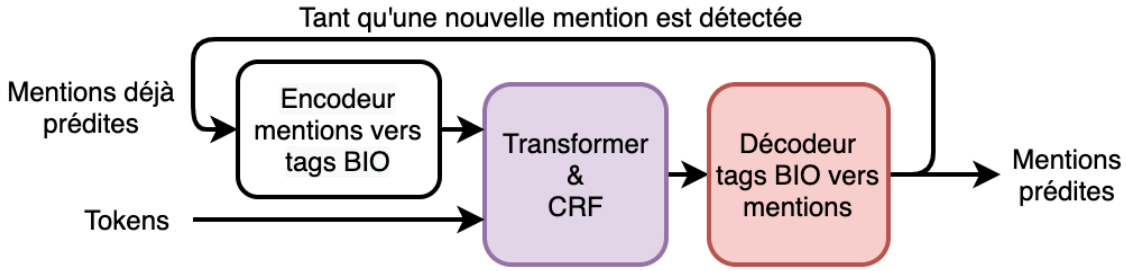


FIGURE 3 – Schéma de prédiction du modèle

2.2.2 Inférence

Pour chaque phrase du corpus à annoter, notre modèle commence par prédire la séquence de mentions la plus probable à partir de la seule séquence de tokens en entrée, puisqu'aucune mention n'a déjà été prédite. Puis, tant qu'une ou plusieurs mentions sont détectées, on les ajoute à la liste des mentions observées, et on ré-itère la prédiction.

2.2.3 Apprentissage

Pendant l'entraînement, on cherche à maximiser les probabilités assignées par le modèle aux listes de mentions à retrouver, avec toujours la difficulté de traiter les mentions chevauchantes.

Comme dans le modèle précédent, on choisit de procéder en plusieurs étapes, et de ne faire prédire par le modèle à chaque exécution que des mentions non imbriquées. Il existe cependant plusieurs combinaisons de mentions de ce type, sans qu'on puisse dire a priori si l'une est meilleure que l'autre. Autrement dit, il existe plusieurs chemins de prédictions possibles menant à une même liste de mentions. Pour deux mentions imbriquées

(annotations que l'on nomme T et H), on peut prédire T en premier, puis H sachant T, ou le contraire, c'est-à-dire choisir d'optimiser entre deux objectifs :

— $P(T, H) = P(T, H|T) \times P(T)$
— $P(T, H) = P(T, H|H) \times P(H)$

Dans cette situation symétrique, une solution qui consisterait à optimiser les deux chemins en même temps, par exemple en sommant plusieurs coûts (*loss*), pourrait conduire à une situation non optimale dans laquelle le modèle est indirectement incité à détecter une combinaison incorrecte des deux mentions, par exemple ces deux mentions non imbriquées mais sous-optimales :

Des modèles comme celui de Ju *et al.* (2018) choisissent une stratégie à l'avance (mentions plus petites d'abord, par exemple), mais le risque est de ne pas profiter de toutes les interdépendances qui font que certaines mentions sont plus simples à trouver quand on connaît les autres. Une autre solution consiste à choisir l'ordre d'extraction menant à la meilleure performance par le modèle, mesurée en F-mesure. On applique ainsi une stratégie gloutonne qui sélectionne, parmi les combinaisons sans chevauchement de mentions non-observées dans ce batch, la plus proche des mentions prédites en terme de recouvrement. Intuitivement, cela signifie qu'on privilégie une combinaison plus facile à prédire pour le modèle.

En pratique, pour choisir à chaque itération cette meilleure combinaison, on définit un graphe de recouvrement $\mathcal{O}$ dans lequel un lien entre deux mentions indique qu'elles se chevauchent. De ce graphe, on extrait les composantes connexes C_i, qui peuvent être vues comme les zones du texte dans lesquelles on observe des recouvrements. On choisit dans chaque composante C_i la mention la plus

proche d'une des mentions prédites, en terme de similarité F1 des intervalles (c'est-à-dire, des tags BIO des tokens qui les composent), puis on réitère en supprimant ses mentions voisines de la liste des candidats.

Enfin, pendant l'entraînement, afin de simuler une extraction en cours, on sélectionne au hasard dans chaque phrase une partie des mentions que l'on définira comme étant déjà extraites par le modèle.

Cette procédure est reprise sous forme de pseudo-code à l'algorithme 1.

Algorithm 1 Algorithme d'apprentissage du modèle

$\mathcal{O}$, le graphe de recouvrement des mentions du corpus
$\mathcal{C}$, les composantes connexes de $\mathcal{O}$
M^{obs}, les mentions observées pour chaque batch
M_i, les mentions de la phrase i
X_i, les tokens de la phrase i
$M^{\text{predicted}}$, les mentions prédites pour chaque batch
M^{target}, les mentions utilisées pour calculer la loss
f, le modèle
for t = 1, . . ., T **do**
    on échantillonne un mini-batch $I_t \subset \{1, \ldots, n\}$ de taille b
    $M^{\text{obs}} \leftarrow \{\}$
    for phrase $i \in I_t$ **do**
        on échantillonne un sous-ensemble P des mentions de la phrase i
        $M^{\text{obs}} \leftarrow M^{\text{obs}} \bigcup P$
    end for
    $M^{\text{predicted}} \leftarrow \arg\max_M P(M|X_i, M^{\text{obs}})$
    $M^{\text{target}} \leftarrow \{\}$
    for phrase $i \in I_t$ **do**
        for composante $j \in$ phrase i **do**
            $m^{closest} \in C_j \setminus M^{obs}$ la mention la plus similaire à une des mentions de $M^{\text{predicted}}$
            $M^{\text{target}} \leftarrow M^{\text{target}} \bigcup \{m^{\text{closest}}\}$
        end for
        $l \leftarrow -\log f(M^{\text{target}}|X_i, M^{\text{obs}})$
        backprop(l)
    end for
end for

2.2.4 Paramètres du modèle

Modèle de base Le Transformer est initialisé avec les poids de CamemBERT-large, pré-entrainé sur CCNET (Wenzek *et al.*, 2019). Les paramètres restants sont initialisés aléatoirement selon la méthode de (Glorot & Bengio, 2010). Du dropout (Srivastava *et al.*, 2014) est appliqué avec une probabilité 0.2 dans le Transformer et 0.4 dans les couches qui le suivent. Les 19 premières couches du transformer sont figées à leur valeur initiale. On optimise les paramètres par backpropagation sur 60 epochs avec Adam (Kingma & Ba, 2015) sans weight decay. Deux pas d'apprentissage sont utilisés : un pour la partie transformer, initialisé à 5×10^{-5}, et un pour le reste du modèle, initalisé à 1×10^{-2}. On divise les pas d'apprentissage par 4 à l'epoch 25, 16 à l'epoch 40 et 64 à l'epoch 50.

		Précision	Rappel	F1
LIMICS run 1	pathologie	0,2644	0,2771	0,2706
	sosy	0,4539	0,3081	0,3670
	Total	0,4223	0,3045	0,3538
LIMICS run 2	pathologie	0,4280	0,6446	0,5144
	sosy	0,6491	0,5668	0,6052
	Total	0,6086	0,5758	0,5917
LIMICS run 3	pathologie	0,5122	0,6325	0,5660
	sosy	0,6885	0,5668	0,6218
	Total	0,6598	0,5744	0,6141
Meilleur DEFT	**Total**			**0,6603**
Médiane DEFT	**Total**			0,4557

TABLE 2 – Résultat de la tâche 3.1 (*pathologie* et *signe ou symptôme*)

Les embeddings de tags sont introduits dans le transformer à la couche $L_{tag} = 18$. Sur une carte graphique GeForce GTX 1080, l'apprentissage sur 100 documents prend environ 20 minutes.

Vote mono-classifieur. Ayant constaté les fortes variations de performance entre différents modèles identiques entraînés avec des amorces aléatoires différentes (variations dans chaque classe mais moyennes finales similaires), nous avons également soumis un dernier run résultant de trois entraînements du modèle Iterative Greedy NER, avec des amorces différentes, puis un vote majoritaire.

3 Résultats et discussion

Les résultats de nos systèmes sont présentés dans les Tables 2 pour la tâche 3.1 et 3 pour la tâche 3.2. Le run 1 correspond au modèle Layered BiLSTM, les run 2 et 3 correspondent au modèle Iterative Greedy NER, le dernier étant la version ensembliste.

Le modèle Iterative Greedy NER obtient de meilleurs résultats que le modèle Layered BiLSTM, sur toutes les classes. Sur la tâche 3.1, il obtient un score F1 de 0.59 (avec délimitation exacte des mentions), et sa version ensembliste un score de 0.61. Sur la tâche 3.2, il obtient le score 0.756 et sa version ensembliste 0.763, meilleur résultat parmi les participants de la compétition.

Notons que pour certaines classes rares mais présentant de faibles variations linguistiques, hybrider notre système avec des règles serait probablement bénéfique.

4 Conclusion

Notre participation au défi fouille de texte de cette année s'est soldée par de bons résultats sur la tâche 3. Une analyse plus détaillée devra nous permettre de comprendre, notamment, pourquoi les types d'entités de la tâche 3.1 se sont moins bien prêtés à nos deux modèles, avec une F-mesure 5 points sous celle du meilleur participant, alors que nous obtenons les meilleurs résultats sur la tâche 3.2. De plus amples expérimentations seront également nécessaires pour estimer l'apport de

		Précision	Rappel	F1
LIMICS run 1	anatomie	0,6819	0,7074	0,6944
	dose	0,3611	0,2500	0,2955
	examen	0,6819	0,5667	0,6190
	mode	0,6458	0,3483	0,4526
	moment	0,6574	0,4303	0,5201
	substance	0,5134	0,3674	0,4283
	traitement	0,4798	0,3520	0,4061
	valeur	0,7746	0,6366	0,6989
	Total	0,6587	0,5673	0,6096
LIMICS run 2	anatomie	0,7674	0,7482	0,7577
	dose	0,6047	0,5000	0,5474
	examen	0,7947	0,8103	0,8024
	mode	0,6471	0,4944	0,5605
	moment	0,6629	0,7152	0,6880
	substance	0,8092	0,7316	0,7685
	traitement	0,6471	0,6513	0,6492
	valeur	0,8224	0,8148	0,8186
	Total	0,7635	0,7494	0,7564
LIMICS run 3	anatomie	0,8056	0,7250	0,7632
	dose	0,6486	0,4615	0,5393
	examen	0,8130	0,7980	0,8054
	mode	0,7455	0,4607	0,5694
	moment	0,6923	0,7091	0,7006
	substance	0,8375	0,7412	0,7864
	traitement	0,6960	0,6250	0,6586
	valeur	0,8313	0,7986	0,8146
	Total	0,7948	0,7330	**0,7626**
Meilleur DEFT	**Total**			**0,7626**
Médiane DEFT	**Total**			0,6151

TABLE 3 – Résultat de la tâche 3.2 (autres classes)

l'idée principale du modèle Iterative Greedy NER, c'est-à-dire la liberté laissée au modèle de trouver les meilleures combinaisons de mentions, indépendamment de leur niveau d'imbrication.

Remerciements

Nous remercions les organisateurs pour la création du corpus ainsi que pour l'organisation du défi.

Références

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*.

GLOROT X. & BENGIO Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, p. 249–256.

GRABAR N., CLAVEAU V. & DALLOUX C. (2018). CAS : French corpus with clinical cases. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, p. 122–128, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/W18-5614.

JU M., MIWA M. & ANANIADOU S. (2018). A neural layered model for nested named entity recognition. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)*, p. 1446–1459, New Orleans, Louisiana : Association for Computational Linguistics. DOI : 10.18653/v1/N18-1131.

KINGMA D. P. & BA J. L. (2015). Adam : A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

LAFFERTY J., MCCALLUM A. & PEREIRA F. C. N. (2001). Conditional random fields : Probabilistic models for segmenting and labeling sequence data. *ICML '01 Proceedings of the Eighteenth International Conference on Machine Learning*, **8**(June), 282–289.

LAMPLE G., BALLESTEROS M., SUBRAMANIAN S., KAWAKAMI K. & DYER C. (2016). Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 260–270, San Diego, California : Association for Computational Linguistics. DOI : 10.18653/v1/N16-1030.

MARTIN L., MULLER B., SUÁREZ P. J. O., DUPONT Y., ROMARY L., DE LA CLERGERIE É. V., SEDDAH D. & SAGOT B. (2020). Camembert : a tasty french language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

ORTIZ SUÁREZ P. J., SAGOT B. & ROMARY L. (2019). Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures. In P. BAŃSKI, A. BARBARESI, H. BIBER, E. BREITENEDER, S. CLEMATIDE, M. KUPIETZ, H. LÜNGEN & C. ILIADI, Éds., *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*, Cardiff, United Kingdom : Leibniz-Institut für Deutsche Sprache. HAL : hal-02148693.

SRIVASTAVA N., HINTON G., KRIZHEVSKY A., SUTSKEVER I. & SALAKHUTDINOV R. (2014). Dropout : A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, **15**(56), 1929–1958.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER Ł. & POLOSUKHIN I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, **2017-Decem**, 5999–6009.

WENZEK G., LACHAUX M.-A., CONNEAU A., CHAUDHARY V., GUZMAN F., JOULIN A. & GRAVE E. (2019). Ccnet : Extracting high quality monolingual datasets from web crawl data. *arXiv preprint arXiv :1911.00359*.