Traitement Automatique des Langues Naturelles (TALN) [1]

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles.
Atelier DÉfi Fouille de Textes (DEFT)

Cyril Grouin, Natalia Grabar, Gabriel Illouz (Éds.)

Lille, France, 28 juin au 2 juillet 2021

1. https://talnrecital2021.inria.fr/

Avec le soutien de

Soutiens institutionnels

Sponsors industriels

Partenaires « Argent »

Partenaires « Bronze »

Préface

DEFT 2021 : une campagne sur l'identification du profil clinique des patients et l'évaluation automatique de réponses d'étudiants

Les campagnes d'évaluation consistent à proposer, dans un temps limité, un jeu de données et un ensemble de tâches à réaliser sur ces données. Pour les participants, l'intérêt consiste à développer ou adapter des méthodes existantes à une nouvelle problématique de recherche, et à comparer les performances des systèmes par rapport à ceux des autres participants, dans des conditions expérimentales parfaitement identiques. Ces conditions permettent ainsi une reproductibilité des résultats plus aisée, ce qui constitue désormais un enjeu majeur en traitement automatique des langues.

Depuis sa création en 2005, le défi fouille de textes (DEFT) propose des tâches régulièrement renouvellées, sur des corpus textuels produits en français, représentatifs de différents usages de la langue, aussi bien en langue générale (articles de presse, messages d'utilisateurs sur les réseaux sociaux, etc.) que sur des langues de spécialité notamment la langue médicale (cas cliniques).

Depuis plusieurs éditions, nous constatons avec satisfaction l'intérêt des entreprises et centres de recherche privés pour les thématiques abordées par DEFT, permettant des échanges et rencontres entre les milieux industriels et académiques. Cette année, nous observons malicieusement que plusieurs équipes formées d'étudiants en master ou en thèse ont trouvé un intérêt pour les tâches d'évaluation automatique des réponses d'étudiants à des questionnaires en ligne utilisés en cours.

L'édition DEFT 2021 (https://deft.lisn.upsaclay.fr/2021/) a porté, d'une part sur la suite du traitement des cas cliniques rédigés en français, et d'autre part sur la correction automatique de réponses d'étudiants provenant de questionnaires sous Moodle.

Nous avons proposé une tâche d'identification du profil clinique du patient, fondée sur les principaux axes du chapitre C du MeSH, en se fondant sur les pathologies présentes dans un cas clinique. Cette tâche s'inscrivant dans la continuité des éditions DEFT 2019 et 2020 (repérage d'entités nommées fines), les participants ont eu la possibilité de s'appuyer sur les annotations des années passées. Le corpus utilisé reprend les cas déjà traités dans les deux éditions antérieures et les enrichit avec de nouveaux cas cliniques annotés selon les mêmes catégories.

Nous avons proposé deux tâches autour de l'évaluation automatique de réponses d'étudiants, en nous fondant sur un cas d'usage réel : simplifier et accélérer le travail d'évaluation de copies d'étudiants. En premier lieu, faire une évaluation automatique en prenant pour référence la correction produite par l'enseignant. En second lieu, poursuivre l'évaluation en prenant pour référence les réponses déjà corrigées de la question traitée. Le corpus utilisé se compose d'une centaine d'énoncés en informatique (programmation web et bases de données) rassemblant des questions ouvertes et fermées, produit à partir de deux années d'enseignement. Ce corpus a été anonymisé.

L'accès aux données d'entraînement a été possible à partir de février 2021, tandis que la phase de test s'est déroulée du 17 au 23 mai. Quatorze équipes se sont inscrites et onze ont participé jusqu'au bout. Nous comptons quatre équipes industrielles (y compris trois

spécifiques au domaine médical), une équipe hospitalo-universitaire, trois équipes formées d'étudiants uniquement (niveau master ou doctorat), et six équipes académiques.

Ces actes rassemblent la présentation des objectifs de la campagne (corpus, tâches, et évaluation), les résultats obtenus sur les différentes tâches et la description des systèmes participants.

Les organisateurs remercient le comité de programme pour avoir apporté leur soutien et leur expertise à la campagne d'évaluation DEFT 2021.

Cyril Grouin, Natalia Grabar et Gabriel Illouz

Comités

Comité de programme

— Alexandre ALLAUZEN (ESPCI, LAMSADE, Dauphine Université Paris/PSL)

— Patrice BELLOT (Université Aix-Marseille, LSIS-Lab, Marseille)

— Yolaine BOURDA (CentraleSupélec, Université Paris-Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique – LISN, Gif-sur-Yvette)

— Natalia GRABAR (Université de Lille, CNRS, UMR8163 STL – Savoirs Textes Langage)

— Cyril GROUIN (Université Paris-Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique – LISN, Orsay)

— Thierry HAMON (Université Sorbonne Paris-Nord, Université Paris-Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique – LISN, Orsay)

— Gabriel ILLOUZ (Université Paris-Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique – LISN, Orsay)

— Anne-Laure LIGOZAT (ENSIIE, Université Paris-Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique – LISN, Orsay)

— Yue MA (Université Paris-Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique – LISN, Gif-sur-Yvette)

— Fleur MOUGIN (Bordeaux Population Health, Université de Bordeaux)

— Fabrice POPINEAU (CentraleSupélec, Université Paris-Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique – LISN, Gif-sur-Yvette)

Comité d'organisation

— Natalia GRABAR (Université de Lille, CNRS, UMR8163 STL – Savoirs Textes Langage)

— Cyril GROUIN (Université Paris-Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique – LISN, Orsay)

— Gabriel ILLOUZ (Université Paris-Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique – LISN, Orsay)

Table des matières

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Classification de cas cliniques et évaluation automatique de réponses d'étudiants : présentation de la campagne DEFT 2021

Cyril Grouin[1] Natalia Grabar[2] Gabriel Illouz[1]
(1) Université Paris Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique, 91400 Orsay, France
(2) Univ. Lille, CNRS, UMR 8163 - STL - Savoirs Textes Langage, 59000 Lille, France
prenom.nom@lisn.upsaclay.fr, prenom.nom@univ-lille.fr

RÉSUMÉ

Le défi fouille de textes (DEFT) est une campagne d'évaluation annuelle francophone. Nous présentons les corpus et baselines élaborées pour trois tâches : (i) identifier le profil clinique de patients décrits dans des cas cliniques, (ii) évaluer automatiquement les réponses d'étudiants sur des questionnaires en ligne (Moodle) à partir de la correction de l'enseignant, et (iii) poursuivre une évaluation de réponses d'étudiants à partir de réponses déjà évaluées par l'enseignant. Les résultats varient de 0,394 à 0,814 de F-mesure sur la première tâche (7 équipes), de 0,448 à 0,682 de précision sur la deuxième (3 équipes), et de 0,133 à 0,510 de précision sur la dernière (3 équipes).

ABSTRACT

Clinical cases classification and automatic evaluation of student answers : Presentation of the DEFT 2021 Challenge

DEFT is an annual French-speaking text mining challenge. We present the corpora, tasks, and baselines we produced : (i) identify the clinical profile of patients described in clinical cases, (ii) automatically assess student answers from online survey (Moodle) based on the teacher's correction, and (iii) continue to evaluate student answers based on answers already assessed by the teacher. Results ranged from 0.394 to 0.814 F-score on the first task (7 participants), 0.448 to 0.682 accuracy on the second one (3 participants), and 0.133 to 0.510 accuracy on the last one (3 participants).

MOTS-CLÉS : Extraction d'information, cas cliniques, réponses courtes d'étudiants.

KEYWORDS: Information extraction ; Clinical cases ; Short Answer Grading ; Student answers.

1 Introduction

Le défi fouille de textes (DEFT) est une campagne d'évaluation annuelle francophone qui permet à plusieurs équipes de confronter leurs méthodes sur une ou plusieurs tâches régulièrement renouvelées. Pour cette nouvelle édition, nous proposons deux thématiques principales. La première concerne le domaine clinique au travers d'une tâche d'extraction d'information depuis des cas cliniques. La deuxième concerne l'enseignement et le traitement de copies d'étudiants avec deux tâches sur l'évaluation automatique des réponses d'étudiants dans des questionnaires en ligne de type Moodle [1].

Les tâches d'extraction d'information constituent une première étape d'accès aux informations pré-

1. https://moodle.org

Atelier DÉfi Fouille de Textes (DEFT), pages 1–13.

sentes dans des documents, pour des objectifs plus généraux (recherche de cas similaires, résumé automatique, etc.). A l'image des campagnes de repérage d'entités nommées, nous proposons de repérer les maladies, signes et symptômes des patients décrits dans des cas cliniques, dans la perspective de dresser le profil clinique des patients. Cette tâche fait suite aux précédentes éditions sur l'identification d'informations démographiques et cliniques (Grabar *et al.*, 2019), et l'extraction d'information fine autour des patients, de la pratique clinique, des traitements et du temps (Cardon *et al.*, 2020). L'évaluation automatique de réponses d'étudiants constitue une tâche originale qui n'a jamais été abordée dans les campagnes DEFT. Elle voit son utilité dans l'assistance automatique lors de la correction des copies et dans la comparaison qualitative entre une copie et une référence.

Organisation de la compétition. Les participants ont pu s'inscrire à la compétition et accéder aux données d'entraînement à partir du 12 février 2021. Ils ont eu accès aux scripts d'évaluation officiels le 26 avril. La phase de test s'est déroulée entre le 17 et le 23 mai, sur une période de trois jours choisie par chaque équipe de participants. L'atelier de clôture s'est déroulé le 28 juin 2021. Quatorze équipes se sont inscrites, toutes françaises, parmi lesquelles quatre entreprises (y compris trois spécifiques au domaine médical), une équipe hospitalo-universitaire, trois équipes formées d'étudiants uniquement (niveau master ou doctorat), et six équipes académiques. Trois équipes ont abandonné la compétition avant la phase de test et une équipe a abandonné la compétition pendant la phase de test.

Dans le contexte du règlement général européen sur la protection des données [2] (RGPD), nous relevons l'intérêt des acteurs du domaine clinique pour accéder à des données de type clinique. Nous observons que la majorité des participants s'intéresse aux cas cliniques (cinq équipes académiques : DOING, Orléans ; ISME, Grenoble ; LIRMM, Montpellier ; QUEER, Paris ; Team Stel, Paris, et deux entreprises : BL.Santé, Toulouse ; Everteam Lab, Lyon). Trois équipes se sont consacrées à l'évaluation des réponses d'étudiants, avec un industriel (EDF Lab, Palaiseau) et deux équipes d'étudiants en master (Nantalco, Univ. Nanterre et INaLCO) ou en thèse (Proofreaders, Univ. Nantes).

2 Corpus

2.1 Domaine clinique

Concernant le domaine clinique, nous reprenons le corpus de cas cliniques (Grabar *et al.*, 2018; Grouin *et al.*, 2019) que nous avons proposé aux participants les années passées. Le corpus d'entraînement se compose des données d'entraînement et de test de DEFT 2020 (soit 167 cas cliniques), tandis que le corpus de test rassemble 108 nouveaux cas. Les annotations des campagnes de 2019 et 2020 sont mises à disposition comme informations complémentaires mais leur utilisation reste facultative.

Afin de préparer les données de référence de 2021, nous avons annoté toutes les maladies, signes ou symptômes, correspondant à un descripteur français du MeSH (Medical Subject Headings) (NLM, 2001), en prenant comme label l'un des vingt-six axes de l'arborescence du chapitre C uniquement [3]. Cependant, vingt-trois axes sont présents et annotés dans le corpus (voir tableau 1).

Le corpus se compose d'un total de 275 cas cliniques annotés sous BRAT (Stenetorp *et al.*, 2012)

2. https://www.cnil.fr/fr/reglement-europeen-protection-donnees
3. Le chapitre C renvoie aux maladies, voir http://mesh.inserm.fr/FrenchMesh/view/index.jsp

C01	Infections bactériennes et mycoses	C13	Maladies de l'appareil urogénital féminin et complications de la grossesse
C02	Maladies virales	C14	Maladies cardiovasculaires
C03	Maladies parasitaires	C15	Hémopathies et maladies lymphatiques
C04	Tumeurs	C16	Malformations et maladies congénitales, héréditaires et néonatales
C05	Maladies ostéomusculaires	C17	Maladies de la peau et du tissu conjonctif
C06	Maladies de l'appareil digestif	C18	Maladies métaboliques et nutritionnelles
C07	Maladies du système stomatognathique	C19	Maladies endocriniennes
C08	Maladies de l'appareil respiratoire	C20	Maladies du système immunitaire
C09	Maladies oto-rhino-laryngologiques	C23	États, signes et symptômes pathologiques
C10	Maladies du système nerveux	C25	Troubles dus à des produits chimiques
C11	Maladies de l'œil	C26	Plaies et blessures
C12	Maladies urogénitales de l'homme		

TABLE 1 – Les axes du MESH et les types de maladies, signes et symptômes annotés en 2021

autour de trois dimensions : les informations démographiques et cliniques [4] (DEFT 2019), les informations cliniques fines [5] (DEFT 2020), et les types de maladies (DEFT 2021). Dans nos annotations et dans la référence, les axes sont désignés au moyen d'un mot-clé représentatif (« infections » pour l'axe C01, « homme » en C12, « femme » en C13, « hemopathies » en C15, « peau » en C17, etc.).

Nous présentons dans le tableau 2 un extrait de cas clinique (fichier filepdf-144-2-cas) avec l'ensemble des axes du MeSH à identifier pour ce fichier. Les pathologies permettant d'identifier les différents axes sont mises en italiques dans le texte (par exemple, les problèmes immunitaires, hémopathiques, et de tumeur de ce patient sont identifiables par la mention d'un myélome dans le texte).

Texte	Axes associés aux pathologies identifiées		
Mr. E., âgé de 70 ans, était suivi depuis cinq ans pour un *myélome*, une *polyarthrite rhumatoïde*, et une *amylose rectale et rénale*. Alors qu'il n'avait pas de troubles mictionnels anciens, il a présenté un *épisode de rétention aiguë d'urine*.			
	immunitaire	hemopathies	tumeur
	peau	osteomusculaires	
	nutritionnelles	digestif	
	homme		

TABLE 2 – Extrait de cas clinique avec tous les axes du MeSH à identifier pour ce fichier (les axes sont inscrits en vis à vis de la pathologie présente dans le texte)

Le tableau 3 présente le nombre d'annotations par classe dans le corpus total de 275 cas cliniques.

4. Soit 4 types d'information : âge ; genre ; origine / motif de la consultation ou de l'hospitalisation ; issue du traitement

5. Soit 12 classes parmi quatre domaines : (i) anatomie ; (ii) examen, pathologie, signe ou symptôme ; (iii) substance, dose, durée, fréquence, mode d'administration, traitement (chirurgical ou médical) ; et (iv) date, moment

DEFT 2019	âge	271	genre	276	origine	178	issue	180
DEFT 2020	anatomie	4780	examen	3355	pathologie	764	sosy	5240
	substance	2009	dose	562	durée	375	fréquence	383
	mode	484	traitement	1311	date	250	moment	970
DEFT 2021	blessures	13	cardiovasc.	103	chimiques	89	digestif	91
	endocrinien	30	état/sosy	546	femme	134	génétique	33
	hémopathies	75	homme	199	immunitaire	34	infections	53
	nerveux	109	nutrition	43	œil	21	ORL	11
	ostéomuscul.	44	parasitaire	11	peau	84	respiratoire	70
	stomato.	12	tumeur	276	virales	14		
inutilisées	fonction	21	organisme	20	poids	5	taille	3
	température	9	valeur	1743				

TABLE 3 – Nombre d'annotations par classe dans le corpus DEFT 2021 (275 cas cliniques)

2.2 Réponses d'étudiants

Des évaluations pour les réponses courtes d'étudiants existent en langue anglaise (Mohler & Mihalcea, 2009). Le lecteur intéressé trouvera un état de l'art dans Burrows *et al.* (2015). Des campagnes d'évaluation ont été menées sur des données en anglais, telles que les compétitions Kaggle [6] ou certaines éditions de SemEval (Dzikovska *et al.*, 2013). Les dispositifs d'aide à la correction ont aussi donné lieu à des expériences. Le but, contrairement à mettre une note en ayant la correction, consiste à trouver comment corriger par regroupement de réponses (Basu *et al.*, 2013; Horbach *et al.*, 2014).

Nous avons constitué un corpus d'une centaine d'énoncés produits par des étudiants en informatique pendant les contrôles de programmation web et de bases de données. Les identités des étudiants ont été anonymisées. Ces énoncés sont composés de questions ouvertes et fermées. Nous distinguons les questions qui impliquent l'écriture du code informatique (« *Modifiez le code XML ci-dessous pour le rendre valide* ») de celles qui appellent une réponse en langue naturelle (« *À quoi sert l'attribut alt de la balise <img> ?* »). Nous avons réparti ces deux types de questions entre les corpus d'entraînement et de test. Les énoncés sont accompagnés de la correction de l'enseignant et des réponses produites par une cinquantaine d'étudiants en moyenne par question. Les énoncés ont été collectés sur deux années d'enseignement. Comme indiqué, ces énoncés correspondent aux réponses formulées sur des questionnaires en ligne. Ils intègrent des balises XML de mise en forme pour l'affichage.

Le tableau 4 présente un extrait du fichier de questions du corpus d'entraînement [7], avec une question en langue naturelle (numéro 1001) et une question de code (numéro 2045). Pour chaque question, une ou plusieurs corrections de l'enseignant sont associées. Un commentaire de l'enseignant peut également être présent pour pénaliser certaines réponses (sur la question 2045, l'enseignant attribue 0,5 point si la réponse fournie est partiellement exacte).

Le tableau 5 présente un ensemble de réponses faites par les étudiants, sans correction orthographique, aux questions du tableau 4. En cas d'absence de réponse, la mention « NO_ANS » est indiquée.

Toutes les notes attribuées aux étudiants ont été normalisées sur un point avec une décimale conservée.

6. https://www.kaggle.com/spscientist/students-performance-in-exams et https://www.kaggle.com/c/asap-sas

7. Le corpus distribué aux participants contient cinq colonnes : l'identifiant de la question, la note maximale d'origine, le numéro de question (similaire à l'identifiant), et la correction de l'enseignant.

Id	Question	Correction ou commentaire de l'enseignant
1001	Qu'est-ce que le World Wide Web ?	système hypertexte fonctionnant sur internet = une des applications d'internet, comme courrier électronique, messagerie instantanée...
2045	Pourquoi le code HTML suivant ne respecte-t-il pas les principes d'accessibilité de WCAG ? <code> Site de la RATP </code>	car la légende de l'image ne lui est pas associée (avec un figcaption par exemple) .5 pour ceux qui ont dit que le texte alternatif n'était pas suffisamment précis

TABLE 4 – Fichier de questions avec correction et commentaire de l'enseignant

Id	Note	Etudiant	Réponse de l'étudiant
1001	0.5	student101	Ce sont les pages web accessible par tout navigateur.\n
1001	0	student108	Un réseau mondial \n
1001	1	student3	C'est le systeme hypertexte qui sert à consulter des documents et des pages hébergés sur le réseau internet\n
1001	0	student95	NO_ANS
2045	0	student101	Les mal-voyant ne peuvent pas y accéder.
2045	1	student109	L'image n'a pas de légende. Les malvoyants ne pourront pas savoir qu'il y a une image.\nL'utilisation de la balise <figcaption>logo RATP</figcaption> aurait permit de respecter le principe d'accessibilité.\nComme alt peut etre lu par les lecteur d'ecran, changer "RATP" en "Logo RATP" pour plus de comprehension.
2045	0.2	student42	Il n'y a pas une description qui décrit l'image
2045	0.8	student70	Le texte par défaut de l'image ne décrit pas l'image précisément.

TABLE 5 – Fichier de réponses des étudiants avec note associée

Afin de comprendre la valeur de certaines notes (0.2 ou 0.8 sur la question 2045), la note maximale d'origine est indiquée dans le fichier de questions (généralement comprise entre 1 et 2,5).

3 Description des tâches

3.1 Tâche 1 : identification du profil clinique du patient

La tâche vise à identifier le profil clinique du patient décrit dans chaque cas. Cela revient à normaliser les pathologies décrites, en identifiant l'axe correspondant du MeSH pour le chapitre C (voir tableau 1). Si une pathologie renvoie à plusieurs axes, tous devront être identifiés. Le choix entre « maladies de l'appareil urogénital féminin et complications de la grossesse » (C13) et « maladies urogénitales de l'homme » (C12) dépend du genre de la personne dont le cas est décrit. Nous fournissons les annotations des éditions DEFT 2019 et 2020 (voir tableau 3) comme aide facultative.

3.2 Tâches autour des réponses d'étudiants

Nous proposons deux tâches autour de l'évaluation de réponses d'étudiants en considérant un enseignant qui souhaiterait améliorer la qualité de son évaluation tout en économisant le temps passé à cette activité. Nous considérons deux situations : (i) celle où l'enseignant dispose déjà des corrections et souhaite développer un système d'évaluation automatique, et (ii) celle où il n'existe pas encore de correction, mais où les premières réponses évaluées permettent déjà de se faire une idée des réponses et des notes à leur associer. La première situation vise à fournir une base, l'enseignant vérifiant la pertinence de l'évaluation automatique. La deuxième évite à l'enseignant de perdre du temps à évaluer les réponses proches, en associant à ces réponses la note des réponses proches déjà corrigées.

3.2.1 Tâche 2 : évaluation automatique de copies d'après une référence existante

À partir d'une liste de questions avec corrections et commentaires de l'enseignant (tableau 4) et d'une liste de réponses d'étudiants à ces questions (tableau 5), l'objectif consiste à évaluer et à noter (sur un point) les réponses des étudiants, en se fondant sur la correction de l'enseignant.

3.2.2 Tâche 3 : poursuite de l'évaluation de réponses à partir de premières évaluations

À partir d'une liste de questions sans aucune correction de l'enseignant (la dernière colonne du tableau 4 est systématiquement indiquée « NO_CORR ») et d'une liste de réponses d'étudiants à ces questions avec de premières notes fournies, l'objectif consiste à évaluer et à noter (sur un point) les réponses des étudiants qui n'ont pas encore été corrigées, à partir des réponses déjà évaluées (la deuxième colonne du tableau 5 comprend, pour une minorité de réponses, la note de l'enseignant, et pour une majorité de réponses la mention « A_CORRIGER »). Le tableau 6 fournit le nombre de questions et de réponses proposées dans chaque tâche dans les corpus d'entraînement et de test.

Corpus	Tâche 2		Tâche 3		Total
	train	test	train	test	
Questions	50	21	11	6	87
Réponses	3820	1644	769	387	6620

TABLE 6 – Nombre de questions et de réponses par corpus pour chaque tâche

Pour le corpus de test de la troisième tâche, nous fournissons 5 % des réponses déjà corrigées pour trois questions (5005, 5011, 5012), et 10 % des réponses déjà corrigées pour les trois autres (5001, 5009, 2012). Les réponses déjà corrigées ont été aléatoirement choisies. Pour que les participants puissent se mettre dans les conditions du test, nous avons fourni le corpus d'entraînement en deux versions : une version avec toutes les notes, et une version avec 5 ou 10 % des réponses déjà corrigées.

4 Baselines

Tâche 1. Cette baseline repose sur les annotations du corpus d'entraînement dont nous avons extrait une centaine de concepts représentatifs, jusqu'à 21 concepts par axe. Certains concepts sont communs

à plusieurs axes. Les concepts peuvent renvoyer à des pathologies (*Pott's Puffy tumor, asthme, brugada, dermatite, pneumonie*), des symptômes (*agitation, fatigue, rash, toux*), des adjectifs ou noms de parties anatomiques (*hépatique, poumon, pulmonaire, rectal, rein, rénal*), ou aux informations de genre (*femme, fille, patiente, vagin, homme, pénis*). Nous présentons quelques axes et leurs concepts :

- infections (C01) : Pott's Puffy tumor, tuberculose
- ostéomusculaire (C05) : fatigue, ostéolyse, ostéosarcome, Pott's Puffy tumor
- œil (C11) : mydriase, myosis, Pott's Puffy tumor
- immunitaire (C20) : allergie, urticaire, VIH
- chimique (C25) : anticholinergique, cirrhose, datura, intoxication

Pour chaque concept identifié dans un document, nous conservons l'axe ou les axes associés. Sur le corpus d'entraînement, nous obtenons une F-mesure globale de 0,568 (rappel de 0,468 et précision de 0,723) et pour le test, une F-mesure globale de 0,546 (rappel de 0,416 et précision de 0,796).

Tâche 2. Cette baseline consiste à compter le nombre de mots en commun entre (1) les mots de la réponse de l'étudiant et (2) les mots de la question et de la réponse de l'enseignant. Cette comparaison repose sur des mots mis en minuscules, sans chiffre ni ponctuation, d'au-moins quatre caractères. Ce décompte est divisé par le nombre de mots conservés dans la question et la réponse de l'enseignant pour produire un score, multiplié par deux (pour compenser le nombre réduit de mots et les fautes d'orthographe). Ce score est normalisé : valeur finale de 1 si le score est supérieur ou égal à 0,5 ; 0,5 si supérieur ou égal à 0,4 ; les autres valeurs sont conservées. Par exemple :

- Question 2032 normalisée : quel est intérêt utiliser du code ajax
- Réponse et commentaire enseignant normalisés : permet échange de données avec le serveur sans mise à jour complète de la page ok pour permet de màj une partie de la page sans avoir à la recharger complètement.
- Réponse normalisée (student7) : utiliser du code ajax permet de mettre à jour certaines parties une page web sans recharger toute la page.
- Soit 5 mots en communs (*jour, page, permet, recharger, sans*) sur 17 mots conservés dans la question et réponse enseignant. Score de $(5/17) \times 2 = 0,59$ normalisé à 1 (score final).

Sur le corpus d'entraînement, nous obtenons une précision de 0,484 (1 847 évaluations correctes pour 1 973 incorrectes) et sur le test, une précision de 0,477 (785 évaluations correctes et 859 incorrectes).

Tâche 3. Nous avons produit deux baselines pour la tâche 3. La première s'inspire de celle utilisée pour la tâche 2, en comptant le nombre de mots communs entre les réponses déjà corrigées et celles restant à évaluer, pour attribuer la note de la réponse corrigée la plus proche. Sur le corpus d'entraînement, nous obtenons une précision de 0,439 (corrélation de 0,60) et une précision de 0,397 sur le test (corrélation de 0,41). La deuxième baseline est une implémentation des k plus proches voisins. Sur le corpus d'entraînement, nous obtenons une précision de 0,561 (corrélation de 0,62) et une précision de 0,462 sur le test (corrélation de 0,58).

5 Résultats

Les performances des systèmes ont été évaluées en termes de rappel, précision et F-mesure :

$$\text{Rappel} = \frac{\text{prédictions correctes}}{\text{prédictions attendues}} \qquad \text{Précision} = \frac{\text{prédictions correctes}}{\text{prédictions réalisées}} \qquad \text{F-mesure} = \frac{2 \times \text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}}$$

Tâche 1. Le tableau 7 présente les résultats (rappel, précision, F-mesure) des participants sur la tâche 1, classés par F-mesure décroissante, ainsi que le rang dans le classement final. Sept équipes ont participé à cette tâche, pour une F-mesure moyenne de 0,636 et une médiane de 0,700.

Equipe	Run	R	P	F	Rang
DOING (Hiot *et al.*, 2021)	2	0,750	**0,888**	**0,814**	1
	1	0,713	0,873	0,785	–
QUEER (Dupont *et al.*, 2021)	1	0,734	0,838	0,782	2
Team Stel (Gérardin *et al.*, 2021)	1	0,874	0,696	0,775	3
	2	**0,895**	0,677	0,771	–
	3	0,872	0,689	0,770	–
QUEER	2	0,684	0,843	0,755	–
	3	0,677	0,819	0,741	–
DOING	3	0,769	0,686	0,725	–
Advanse LIRMM	2	0,637	0,783	0,703	4
	1	0,627	0,784	0,697	–
BL.Santé (Billami *et al.*, 2021)	3	0,730	0,558	0,633	5
	1	0,677	0,570	0,619	–
	2	0,471	0,786	0,589	–
Everteam Lab (Bailly *et al.*, 2021)	1	0,683	0,370	0,480	6
ISME (Mannion *et al.*, 2021)	2	0,423	0,496	0,457	7
	3	0,398	0,439	0,417	–
	1	0,390	0,444	0,416	–
Everteam Lab	3	0,637	0,298	0,406	–
	2	0,651	0,283	0,394	–

TABLE 7 – Résultats et classement sur la tâche 1. Les meilleurs résultats sont en gras

La figure 1 présente les valeurs moyennes de rappel, précision, et F-mesure, calculées sur l'ensemble des soumissions, pour chacun des axes du chapitre C du MeSH, classées par F-mesure croissante. Les axes les mieux identifiés par les participants, en termes de F-mesure moyenne, sont : signes ou symptômes (F=0,955) ; maladies parasitaires (F=0,821) ; maladies urogénitales de l'homme (F=0,765) ; tumeurs (F=0,756). Cette réussite s'explique par le nombre élevé d'exemples en corpus (l'axe signes ou symptômes est le plus représenté) et leur relative régularité (le mot *tumeur* ou les parties anatomiques masculines). A l'opposé, les axes les plus compliqués sont : maladies virales (F=0,319) ; malformations et maladies congénitales, héréditaires et néonatales (F=0,351) ; plaies et blessures (F=0,372) ; et maladies de la peau et du tissu conjonctif (F=0,393). Ces axes sont peu représentés dans les corpus, renvoient à des entités plus complexes à identifier et nécessitent de réelles connaissances médicales (telles que les maladies congénitales).

Les méthodes employées par les participants sont des méthodes de classification multi-labels supervisées, éventuellement complétées par des plongements lexicaux tels que CamemBERT par Bailly *et al.* (2021) et Gérardin *et al.* (2021), ou en réentraînant des plongements avec Word2Vec comme réalisé par Billami *et al.* (2021). Une autre approche, suivie par Dupont *et al.* (2021), repose sur l'utilisation du MeSH, complété de termes du corpus, pour indexer le contenu des documents. Les meilleurs résultats, obtenus par Hiot *et al.* (2021), reposent sur des transducteurs à états finis et des listes de mots-clés issus du MeSH et de MedDRA. Les auteurs soulignent la simplicité de cette méthode et

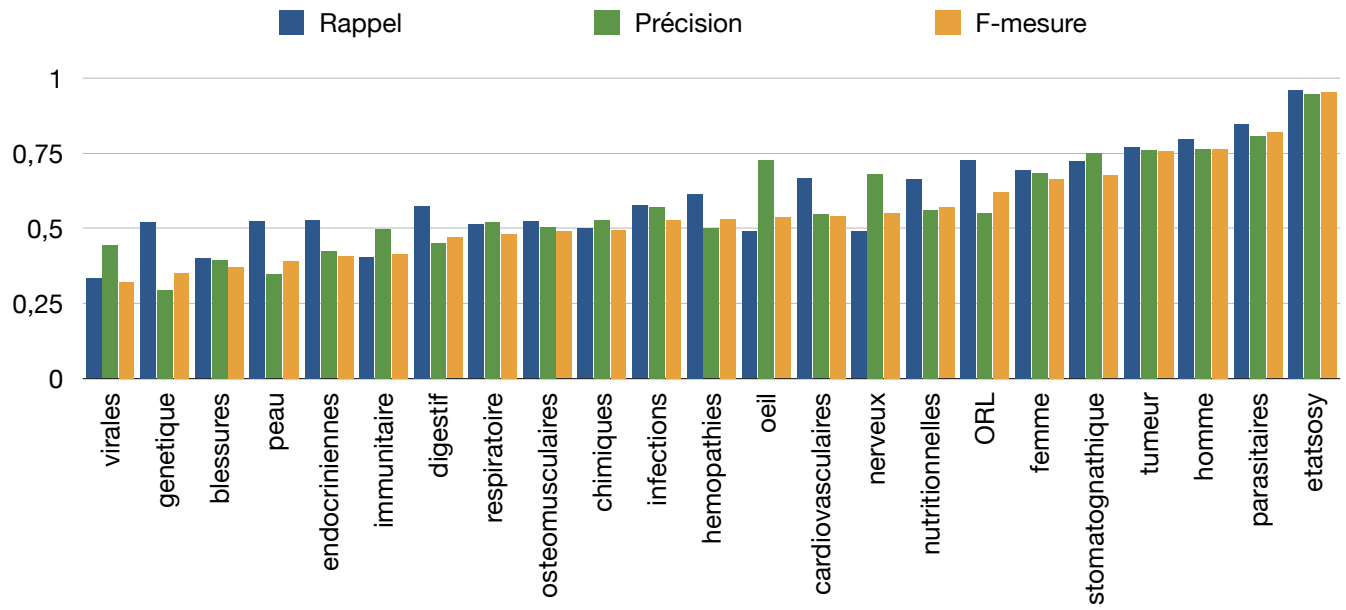

FIGURE 1 – Valeurs moyennes de rappel (bleu), précision (vert), et F-mesure (jaune), pour chaque axe, sur l'ensemble des soumissions des participants, par F-mesure croissante

son coût environnemental faible puisque ne nécessitant aucun entraînement.

Les pré-traitements de tokénisation, lemmatisation, et suppression des mots outils ont été appliqués par l'ensemble des participants pour réduire le nombre de descripteurs. Dupont *et al.* (2021) ont également travaillé sur la détection des phrases négatives et hypothétiques tandis que Hiot *et al.* (2021) ont identifié les négations et les informations de genre.

Tâche 2. Le tableau 8 présente les résultats (précision pour l'ensemble et moyenne par question des corrélations de Pearson) et le classement sur la tâche 2 (RP=rang précision, classement officiel, RC=rang corrélation), par précision décroissante. Trois équipes ont participé, pour une précision moyenne de 0,607 et une médiane de 0,627. Une partie du fichier de la deuxième soumission de l'équipe EDF Lab étant compromise, une évaluation du fichier corrigé a été réalisée hors-délais, sans remettre en cause le classement.

Les meilleurs résultats ont été obtenus avec des algorithmes classiques de classification, tel que Random Forest sous WEKA utilisé par Suignard *et al.* (2021), ou en calculant une similarité entre vecteurs de la question et de la réponse tel que réalisé par Wang *et al.* (2021) au moyen d'un SVM et des informations contextuelles, ou par Dupont *et al.* (2021) qui ont comparé plusieurs coefficients de similarité. Nous observons que les approches utilisant CamemBERT et SentenceBERT (autres soumissions EDF Lab et Nantalco) ont obtenu de moins bons résultats.

Tâche 3. Pour l'évaluation des résultats de la troisième tâche, nous avons également utilisé la corrélation de Pearson (formule 1, avec Cov(X,Y) la covariance des variables X et Y, et σX et σY les écarts-types de ces variables) en complément de la précision.

$$r = \frac{Cov(X,Y)}{\sigma X \sigma Y} \quad (1)$$

Equipe	Run	Précision	RP	Corrélation	RC
EDF Lab (Suignard *et al.*, 2021)	1	0,682	1	0,57 (1 N/A)	1
	*2**	*0,589**	–	*	–
	3	0,638	–	0,57	–
QUEER (Dupont *et al.*, 2021)	1	0,448	–	0,46	–
	2	0,624	–	0,52	2
	3	0,630	3	0,47 (2 N/A)	–
Nantalco (Wang *et al.*, 2021)	1	0,639	2	0,52 (5 N/A)	3
	2	0,580	–	0,38 (2 N/A)	–
	3	0,627	–	0,40 (2 N/A)	–

TABLE 8 – Résultats et classement des équipes participantes à la tâche 2 (RP=rang précision, classement officiel, RC=rang corrélation ; *évaluation hors-délais, N/A : nombre de question où on ne peut calculer la corrélation)

Le tableau 9 présente les résultats (précision pour l'ensemble et moyenne par question des corrélations de Pearson) et le classement sur la tâche 3 (RP=rang précision, classement officiel, RC=rang corrélation), par précision décroissante. Avec trois équipes, la moyenne est de 0,264 et la médiane de 0,241.

Equipe	Run	Précision	RP	Corrélation	RC
EDF Lab (Suignard *et al.*, 2021)	1	0,510	1	0,45	–
	2	0,382	–	0,40	–
	3	0,292	–	0,47	2
QUEER (Dupont *et al.*, 2021)	1	0,278	2	0,00	–
	2	0,241	–	-0,01	–
	3	0,212	–	-0,04	–
Proofreaders (Poulain & Connes, 2021)	1	0,170	3	0,54	1
	2	0,159	–	0,49	–
	3	0,133	–	0,51	–

TABLE 9 – Résultats et classement des équipes participantes à la tâche 3 (RP=rang précision, classement officiel, RC=rang corrélation)

Alors que les résultats des participants à la deuxième tâche sont parfois très proches entre soumissions de deux participants, nous observons que les résultats obtenus sur la troisième tâche conservent les soumissions groupées par participant, témoignant à la fois de la difficulté de la tâche, et des différences méthodologiques, qui ne permettent pas à un participant de dépasser un autre participant.

Sur cette tâche, les meilleurs résultats ont été obtenus avec une simple similarité fondée sur des trigrammes de caractères, utilisée par Suignard *et al.* (2021), qui obtient de bien meilleurs résultats que SentenceBERT (autres soumissions EDF Lab), ou que les réseaux de neurones LSTM utilisés par Dupont *et al.* (2021). Enfin, Poulain & Connes (2021) ont travaillé sur une approche d'extraction de traits lexicaux en combinant les corpus du défi avec un sous-corpus de textes pédagogiques issus de Wikilivres. Nous observons que ces participants ont également comparé les mesures de similarité disponibles pour comparer des vecteurs.

Les questions appellent des réponses soit sous la forme d'un résultat attendu (une réponse qui peut

être discrétisée), soit en langue formelle (du code informatique), soit en langue naturelle. Pour les réponses sous la forme d'un résultat attendu, certains étudiants produisent des réponses en langue naturelle pour justifier leur réponse. Les résultats pour le meilleur run et en prenant pour chaque question le système qui répond le mieux sont donnés dans le tableau 10. On note pour la tâche 2, que les précisions suivent l'ordre attendu de facilité à noter les réponses (Résultat, Formelle, puis Naturelle). Pour la tâche 3, malgré le peu de questions (6), la même tendance est observée (il n'existe aucune réponse de type résultat attendu sur la tâche 3).

Type de réponse	Tâche 2		Tâche 3	
	MRun(=EDF1)	MSPQ	MRun	MSPQ
Naturelle	0,66	0,71	0,16(EDF-2)	0,18
Formelle	0,78	0,81	0,76(EDF-1)	0,76
Résultat	0,80	0,82	—	—

TABLE 10 – Précisions par type de questions (MRun : meilleur run ; MSPQ : Meilleur système par question)

Nous remercions les participants à la tâche 3 qui était assez expérimentale, et pour lequelle nous espérons à long terme l'intégration d'outils d'évaluation dans Moodle, permettant un gain de temps et de qualité lors des évaluations. A l'instar des campagnes SemEval, nous envisageons de limiter le nombre de notes (deux ou trois niveaux). Une autre direction d'évaluation serait d'introduire une évaluation par rubrique (évaluation des réponses des étudiants selon plusieurs dimensions : syntaxe, sémantique, etc.) comme abordée par Mizumoto *et al.* (2019).

6 Conclusion

L'édition 2021 du défi fouille de texte (DEFT) a traité le domaine clinique dans la continuité de DEFT 2019 et DEFT 2020 d'une part, et pour la première fois dans DEFT le domaine des réponses d'étudiants à des questionnaires en ligne de type réponses courtes (dans Moodle) d'autre part.

La première tâche a rassemblé sept équipes et visait à établir le profil clinique des patients décrits dans un corpus de 275 cas cliniques, en normalisant les pathologies, signes ou symptômes par rapport à l'un des 23 axes du chapitre C du MeSH. Les F-mesures obtenues par les participants sur le corpus de test varient de 0,394 à 0,814, avec une moyenne de 0,636 et une médiane de 0,700. Notre baseline (identification de concepts représentatifs de chaque axe) obtient une F-mesure de 0,546. L'utilisation de transducteurs à états finis avec des listes de mots-clés s'est révélée la plus efficace.

La deuxième tâche a rassemblé trois équipes et portait sur l'évaluation automatique de réponses d'étudiants à des questionnaires en ligne, en prenant pour référence la correction de l'enseignant. Les précisions obtenues par les participants varient de 0,448 à 0,682, avec une moyenne de 0,607 et une médiane de 0,627. Notre baseline (nombre de mots en commun entre réponse et question/correction) obtient une précision de 0,477. La classification avec Random Forest a permis l'obtention des meilleurs résultats.

Enfin, la dernière tâche a également rassemblé trois équipes et concernait la poursuite de l'évaluation de réponses d'étudiants, en prenant pour référence les réponses déjà évaluées par l'enseignant (entre 5 et 10 % des réponses à chaque question étaient déjà évaluées). Les précisions obtenues varient de

0,133 à 0,510, avec une moyenne de 0,264 et une médiane de 0,241, et les corrélations de Pearson varient de 0,02 à 0,65. Nos baselines obtiennent une précision de 0,397 (recherche des réponses évaluées les plus similaires) et de 0,561 (k plus proches voisins). Comme pour la précédente tâche, la méthode la plus simple a obtenu les meilleurs résultats, grâce à une similarité de trigrammes de caractères.

Cette nouvelle édition du défi fouille de texte se termine avec une variété de méthodes testées sur chacune des tâches proposées, et un constat que les méthodes les plus simples continuent, pour l'instant, de surpasser les approches à base de plongements lexicaux.

Références

BAILLY A., BLANC C. & GUILLOTIN T. (2021). Classification multi-label de cas cliniques avec CamemBERT. In *Actes de DEFT*, Lille, France.

BASU S., JACOBS C. & VANDERWENDE L. (2013). Powergrading : a clustering approach to amplify human effort for short answer grading. *Transactions of the Association for Computational Linguistics*, **1**, 391–402.

BILLAMI M. B., NICOLAIEFF L., GOSSET C. & BORTOLASO C. (2021). Participation de Berger-Levrault (BL.Research) à DEFT 2021 : de l'apprentissage des seuils de validation à la classification multi-labels de documents. In *Actes de DEFT*, Lille, France.

BURROWS S., GUREVYCH I. & STEIN B. (2015). The eras and trends of automatic short answer grading. *International Journal of Artificial Intelligence in Education*, **25**(1), 60–117.

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*, p. 1–13, Nancy, France : ATALA. HAL : hal-02784737.

DUPONT Y., GONZÁLEZ-GALLARDO C.-E., LEJEUNE G., MILLOUR A. & TANGUY J.-B. (2021). QUEER@DEFT2021 : Identification du profil clinique de patients et notations automatique de copies d'étudiants. In *Actes de DEFT*, Lille, France.

DZIKOVSKA M. O., NIELSEN R. D., BREW C., LEACOCK C., GIAMPICCOLO D., BENTIVOGLI L., CLARK P., DAGAN I. & DANG H. T. (2013). Semeval-2013 task 7 : The joint student response analysis and 8th recognizing textual entailment challenge. In *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, p. 263–274, Atlanta, Georgia.

GÉRARDIN C., VAILLANT P., WAJSBÜRT P., GILAVERT C., BELLAMINE A., KEMPF E. & TANNIER X. (2021). Classification multilabel de concepts médicaux pour l'identification du profil clinique du patient. In *Actes de DEFT*, Lille, France.

GRABAR N., CLAVEAU V. & DALLOUX C. (2018). CAS : French corpus with clinical cases. In *Proc of LOUHI*, p. 122–128, Brussels, Belgium : Association for Computational Linguistics. 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 DEFT*, p. 1–10, Toulouse, France : ATALA. HAL : hal-02280852.

GROUIN C., GRABAR N., HAMON T. & CLAVEAU V. (2019). Clinical case reports for NLP. In *Proc of BioNLP*, p. 273–282, Florence, Italy : Association for Computational Linguistics. DOI : 10.18653/v1/W19-5029.

HIOT N., MINARD A.-L. & BADIN F. (2021). DOING@DEFT : utilisation de lexiques pour une classification efficace de cas cliniques. In *Actes de DEFT*, Lille, France.

HORBACH A., PALMER A. & WOLSKA M. (2014). Finding a tradeoff between accuracy and rater's workload in grading clustered short answers. In *LREC*, p. 588–595 : Citeseer.

MANNION A., CHEVALIER T., SCHWAB D. & GOEURIOT L. (2021). Identification de profil clinique du patient : Une approche de classification de séquences utilisant des modèles de langage français contextualisés. In *Actes de DEFT*, Lille, France.

MIZUMOTO T., OUCHI H., ISOBE Y., REISERT P., NAGATA R., SEKINE S. & INUI K. (2019). Analytic score prediction and justification identification in automated short answer scoring. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, p. 316–325.

MOHLER M. & MIHALCEA R. (2009). Text-to-text semantic similarity for automatic short answer grading. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, p. 567–575.

NLM (2001). *Medical Subject Headings*. National Library of Medicine, Bethesda, Maryland. https://www.nlm.nih.gov/mesh/meshhome.html.

POULAIN T. & CONNES V. (2021). DEFT 2021 : Évaluation automatique de réponses courtes, une approche basée sur la sélection de traits lexicaux et augmentation de données. In *Actes de DEFT*, Lille, 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 *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, p. 102–107, Avignon, France : Association for Computational Linguistics.

SUIGNARD P., BENAMAR A., MESSOUS N., CHRISTOPHE C., JUBAULT M. & BOTHUA M. (2021). Participation d'EDF R&D à DEFT 2021. In *Actes de DEFT*, Lille, France.

WANG X., LIU X. & YUE Y. (2021). Mesure de similarité textuelle pour une évaluation automatique de copies d'étudiants. In *Actes de DEFT*, Lille, France.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Classification multi-label de cas cliniques avec CamemBERT

Alexandre Bailly[1, *] Corentin Blanc[1, *] Thierry Guillotin [1]
(1) Everteam Software, 17 quai Joseph Gillet, 69004 Lyon, France
(*) Contributions égales
a.bailly@everteam.com, c.blanc@everteam.com, t.guillotin@everteam.com

RÉSUMÉ

La quantité de documents textuels médicaux allant grandissant, la nécessité d'en extraire automatiquement des informations concernant des patients devient de plus en plus grande. La prédiction du profil clinique permet de gagner du temps pour le praticien tout en extrayant l'essentiel de l'information concernant un patient. Avec l'explosion du nombre de documents (médicaux ou non), des modèles pré-entraînés tels que BERT pour l'anglais ou CamemBERT pour le français ont émergé. L'utilisation de ces modèles permet d'encoder contextuellement du texte afin de l'utiliser dans des réseaux neuronaux pour notamment prédire des profils cliniques. Cet article vise à comparer différentes méthodes de prédiction de profil clinique en se basant sur l'utilisation de CamemBERT. Dans un premier temps, uniquement du texte provenant de documents médicaux a été utilisé. Dans un second temps, des entités nommées ont été injectées en plus du texte par concaténation ou par sommation pondérée. Les résultats ont montré un succès limité et dépendant de la prévalence des chapitres à prédire dans le corpus ainsi qu'une dégradation des performances lors de l'ajout des entités nommées.

ABSTRACT

Multi-label classification of clinical cases with CamemBERT

As quantity of textual medical data is increasing, the necessity to extract automatically information about patients increases accordingly. Predicting the clinical profile of a patient record allows to save time for praticians by exhibiting essential information concerning the patient. Together with the explosion in the number of documents (medical or not), pretrained models such as BERT for english or CamemBERT for french has emerged. Using these models allows to encode contextually a text to this encoded representation in neural networks notably for NLP tasks such as predicting clinical profiles. This article aims to compare different methods of clinical profile prediction based on CamemBERT. In a first time, only the text from medical documents was used. In a second time, named entities were injected in addition to the text by concatenation or pondered sum. Results show a limited success depending on the prevalence of the chapters to predict in the corpus as well as a decrease of performances with the use of named entitie types.

MOTS-CLÉS : Classification multi-label ; Fouille de texte ; CamemBERT.

KEYWORDS: Multi-label classification ; Data mining ; CamemBERT.

1 Introduction

Avec l'augmentation du nombre de consultations médicales, la quantité de documents textuels concernant les patients a considérablement augmenté. Les divers compte-rendus de consultation ou de

Atelier DÉfi Fouille de Textes (DEFT), pages 14–20.

prise en charge hospitalière forment une masse de données importante et riche en informations sur le patient. Ces informations sont très intéressantes pour les différents praticiens et leur récupération est un enjeu important. L'extraction du profil clinique d'un patient (l'ensemble des pathologies associées à son cas) à partir d'un document textuel peut prendre un temps important, au détriment du temps accordé au patient. Cette étape reste néanmoins indispensable et une automatisation de l'extraction est une bonne alternative pour obtenir les informations nécessaires.

Cette explosion du nombre de documents médicaux a poussé la communauté scientifique à créer de nouveaux modèles de langue facilitant leur traitement. Ces modèles pré-entraînés sur des quantités colossales de données permettent d'encoder une phrase ainsi que les mots la constituant en tenant compte de leur contexte. Le plus connu est BERT (Bidirectionnal Encoder Representation from Transformers) qui a permis d'améliorer l'état de l'art sur une grande majorité des tâches de Traitement Automatique du Langage Naturel (TALN) en anglais (Devlin *et al.*, 2019). Suite à ce succès, de nombreux autres modèles dérivés de BERT ont vu le jour comme CamemBERT (Martin *et al.*, 2020) pour le français.

Ce papier vise à étudier la prédiction du profil clinique d'un patient en utilisant à la fois du texte brut provenant de documents médicaux mais aussi différentes entités nommées qui ont été préalablement mises en évidence. Trois approches seront étudiées : le traitement du texte brut par CamemBERT dans un premier temps puis l'injection par concaténation et sommation pondérée des entités nommées au texte brut dans un second temps.

2 Matériel et méthodes

2.1 Données

Ces travaux se situent dans le contexte de la compétition DEFT-21 (Grouin *et al.*, 2021). Le corpus de DEFT 2021 était constitué de 275 cas cliniques répartis en un jeu de d'entraînement (167) et un jeu de test (108). Chaque cas clinique était composé d'un texte brut accompagné d'un certain nombre d'entités nommées préalablement identifiées parmi 19 types distincts comme le montre l'exemple sur la Figure 1.

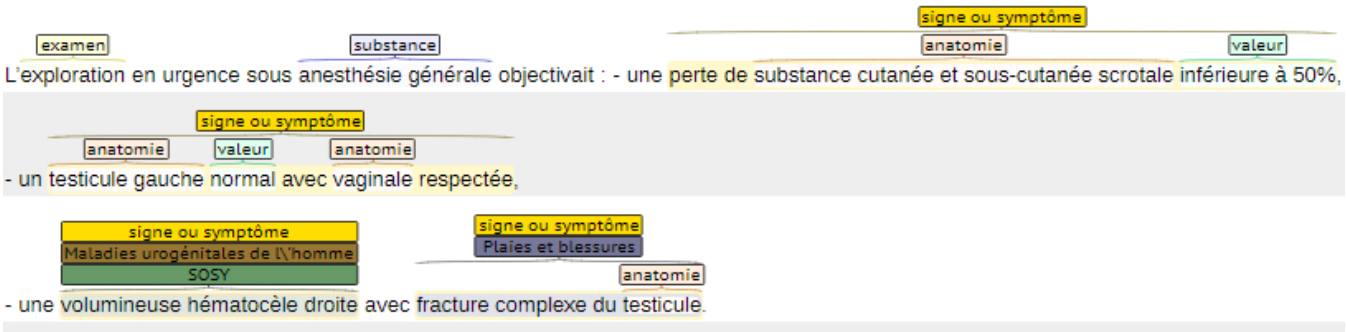

FIGURE 1 – Exemple de texte brut accompagné d'entités nommées d'un cas clinique

Un profil clinique constitué d'au moins un des 23 chapitres du MeSH a été attribué à tous les cas cliniques. Au sein du corpus, la distribution des chapitres était extrêmement déséquilibrée comme suit sur la Table 1.

Chapitre	Entraînement	Test	Total
Stomatognathique	3	3	6
Parasitaires	6	1	7
Virales	8	4	12
ORL	10	3	13
Oeil	11	9	20
Génétique	19	10	29
Immunitaire	20	11	31
Endocriniennes	22	14	36
Blessures	21	19	40
Osteomusculaires	21	22	43
Respiratoire	27	17	44
Peau	31	16	47
Nutritionnelles	29	23	52
Digestif	36	22	58
Hémopathies	34	25	59
Infections	33	27	60
Femme	33	32	65
Cardiovasculaires	39	27	66
Chimiques	45	22	67
Nerveux	41	46	87
Homme	63	36	99
Tumeur	80	51	131
Etatsosy	141	101	242

TABLE 1 – Distribution des chapitres du MeSH dans les données

Dans la suite, certains chapitres trop peu représentés ont été volontairement écartées des possibilités de prédiction afin d'améliorer celles des autres chapitres. Les chapitres écartés sont les suivants : *stomatognathiques* (3), *parasitaires* (6), *virales* (8), *ORL* (10) et *oeil* (11).

2.2 Approches proposées

CamemBERT est un modèle de langue pré-entraîné sur une énorme quantité de données permettant d'encoder le contexte d'une phrase. La meilleure manière d'appliquer un tel modèle à une tâche TALN est de connecter une couche de sortie pour réaliser la prédiction puis de régler finement tous les poids de bout en bout.

Dans la suite, chacune des méthodes introduites utilise le modèle de langue CamemBERT. Pour un cas clinique donné, toutes les phrases du texte brut ont été encodées par CamemBERT puis moyennées afin d'obtenir une unique représentation du cas clinique. Cette représentation a ensuite été utilisée afin d'effectuer la prédiction.

2.2.1 Texte brut

Dans un premier temps, une couche linéaire de 18 neurones avec une sigmoïde comme fonction d'activation a été connectée à CamemBERT afin de calculer une probabilité pour chacun des 18 chapitres précédemment retenus.

2.2.2 Injection des entités nommées au texte brut

Dans un second temps, les entités nommées extraites de chaque document ont été utilisées pour enrichir les entrées du modèle. Pour chaque cas clinique, les types d'entités nommées ont été encodées grâce à un vecteur binaire représentant la présence (1) ou l'absence (0) au sein du texte associé au cas clinique. Afin de les injecter au texte brut, deux méthodes ont été utilisées :

— Concaténation : l'encodage du texte brut était concaténé à celui des entités nommées.
— Somme pondérée : l'encodage du texte brut était sommé à une projection linéaire de l'encodage des entités nommées. La projection tout comme la pondération étaient apprises lors de la phase d'entraînement.

Ces deux méthodes ont permis de construire de nouvelles représentations du cas clinique qui ont ensuite été utilisées dans un couche linéaire de 18 neurones, couplée à une fonction sigmoïde, afin de calculer une probabilité pour chacun des chapitres retenus.

2.3 Paramètres d'entraînement

Pour toutes les approches, le nombre d'epochs a été fixé à 5 au vue de la convergence de la Binary Cross-Entropy Loss. L'optimisateur AdamW (Loshchilov & Hutter, 2019) a été utilisé avec un taux d'apprentissage fixé à 5e-3 sur la première epoch puis diminuant linéairement. Pour chacune des trois méthodes proposées, un seuil par chapitre a été recherché afin d'optimiser les résultats.

2.4 Évaluation

La recherche des paramètres d'entraînement a été effectuée par une méthode de bootstrap (Efron, 1979) à partir du corpus d'entraînement. Une fois les paramètres d'entraînement sélectionnés, les modèles ont été entraînés sur la globalité du corpus d'entraînement. Trois métriques ont été utilisées pour évaluer les performances sur chaque chapitre et de manière globale : le rappel, la précision et le f1-score. Les résultats présentés ci-après sont ceux obtenus sur le jeux de test.

3 Résultats

Évaluation par chapitre Pour chacun des différents modèles qui ont été entraînés, les performances obtenues pour les prédictions dépendent des chapitres et ne différent pas grandement d'une méthode à l'autre. En effet, comme il est visible dans la table 2, le f1-score pour les différents chapitres se situe entre 0.105 et 0.967. Les chapitres les moins représentés dans le corpus sont ceux qui présentent

	Rappel			Précision			F1		
	M1†	M2*	M3$	M1†	M2*	M3$	M1†	M2*	M3$
Stomatognathique	—	—	—	—	—	—	—	—	—
Parasitaires	—	—	—	—	—	—	—	—	—
Virales	—	—	—	—	—	—	—	—	—
ORL	—	—	—	—	—	—	—	—	—
Oeil	—	—	—	—	—	—	—	—	—
Génétique	**1.000**	0.700	0.400	**0.118**	0.091	0.103	**0.211**	0.161	0.163
Immunitaire	0.091	**0.364**	0.091	**0.250**	0.154	0.067	0.133	**0.216**	0.077
Endocriniennes	**1.000**	0.857	0.857	**0.140**	0.126	0.121	**0.246**	0.220	0.212
Blessures	0.158	**0.684**	0.158	0.188	0.171	**0.214**	0.171	**0.274**	0.182
Osteomusculaires	0.000	0.318	**0.409**	0.000	**0.250**	0.191	0.000	**0.280**	0.261
Respiratoire	0.235	**0.471**	0.176	**0.286**	0.167	0.075	**0.258**	0.246	0.105
Peau	0.312	0.688	**0.750**	**0.217**	0.125	0.124	**0.256**	0.212	0.212
Nutritionnelles	**1.000**	0.652	0.696	0.213	0.217	**0.229**	**0.351**	0.326	0.344
Digestif	**0.773**	0.591	0.545	**0.250**	0.197	0.200	**0.378**	0.295	0.293
Hémopathies	**0.520**	0.320	0.120	**0.371**	0.205	0.150	**0.433**	0.250	0.133
Infections	0.778	0.741	**0.926**	**0.292**	0.267	0.255	**0.424**	0.392	0.400
Femme	**0.844**	0.719	0.719	**0.386**	0.303	0.307	**0.529**	0.426	0.430
Cardiovasculaires	**1.000**	**1.000**	**1.000**	0.250	0.250	**0.260**	0.400	0.400	**0.412**
Chimiques	0.182	**0.273**	0.091	**0.571**	0.182	0.065	**0.276**	0.218	0.075
Nerveux	**0.457**	0.304	0.348	**0.636**	0.359	0.457	**0.532**	0.329	0.395
Homme	0.472	**0.722**	0.694	**0.500**	0.342	0.321	**0.486**	0.464	0.439
Tumeur	**0.941**	0.608	0.745	**0.623**	0.397	0.458	**0.750**	0.481	0.567
Etatsosy	**1.000**	0.931	**1.000**	**0.935**	0.931	**0.935**	0.935	0.931	**0.967**
Global	**0.683**	0.651	0.637	**0.370**	0.283	0.298	**0.480**	0.394	0.406

†Modèle textuel uniquement - * Modèle avec concaténation - $ Modèle avec pondération

TABLE 2 – Résultat obtenu pour chaque chapitre du MeSH

les f1-scores les plus faibles, et ce quel que soit le modèle. Le f1-score pour le chapitre *génétique*, qui est le moins représenté, est de 0.161 pour le modèle avec concaténation, de 0.163 pour celui avec pondération et de 0.211 pour le modèle n'utilisant que le texte brut. Au contraire, les chapitres les plus représentés dans le corpus présentent de bon résultats, notamment pour *etatsosy*, avec des f1-score de 0.935, 0.931 et 0.967 respectivement pour le modèle utilisant seulement le texte, le modèle avec la concaténation et le modèle avec la pondération. Le chapitre *chimiques* fait ici figure d'exception, avec des f1-scores de 0.276, 0.218 et seulement 0.075 respectivement, alors qu'il fait partie des chapitres avec les plus grandes prévalences. Les faibles performances en terme de f1-score pour les chapitres sous-représentés sont dues à une faible précision. En effet, pour le chapitre *endocriniennes* par exemple, le modèle avec pondération a obtenu un rappel de 0.857 mais une précision de seulement 0.121, ce qui explique alors le f1-score de 0.212.

Comparaison des modèles L'évaluation globale des modèles montre que quelque soit la métrique observée, le modèle n'utilisant que le texte obtient de meilleurs performances. En terme de f1-score, le modèle utilisant seulement le texte atteint 0.480 alors que les modèles utilisant la concaténation et

la pondération n'atteignent respectivement que 0.394 et 0.406. La comparaison de ces deux dernières valeurs semble indiquer que la pondération conduit à de meilleurs performances que la concaténation.

4 Discussion

La prédiction des différents chapitres est plus ou moins bonne selon leur prévalence dans le corpus d'entraînement. En effet, les chapitres les moins représentés ont tendance à être moins bien prédits. La faible prévalence de certains chapitres semble donc être un frein considérable pour tous les modèles comparés lors de l'apprentissage.

La moyenne des représentations des phrases d'un texte obtenues grâce à CamemBERT permet dans une certaine mesure d'attribuer les chapitres associés à ce même texte. Néanmoins, les performances de ce modèle restent limitées. Cela peut être notamment dû à la quantité limités de cas cliniques disponibles pour l'entraînement. En effet, l'utilisation de CamemBERT implique un grand nombre de poids à entraîner et donc nécessite beaucoup de données pour l'entraînement. Une autre explication pourrait être le fait que l'ensemble du texte a été considéré pour la prédiction, alors que l'information recherchée peut n'être présente que dans une partie seulement. Certaines phrases pourrait donc être à l'origine de bruit dans les données.

L'utilisation des entités nommées pour enrichir le texte était supposée apporter davantage d'informations et permettre d'améliorer la prédiction. Cependant, les modèles les incluant ont vu leurs performances se dégrader et ce peu importe la façon dont elles ont été injectées. L'information de la seule présence des entités dans le texte ne semble donc ne pas être suffisante pour améliorer les prédictions.

5 Conclusion

Dans une certaine mesure, l'utilisation du modèle pré-entraîné CamemBERT a permis de retrouver les chapitres du MeSH associés à différents cas cliniques à partir du texte. En revanche, la quantité de données présente n'a pas permis d'atteindre de bonne performances avec cette méthode. L'ajout de la présence de certaines entités nommées a eu pour seul effet de dégrader légèrement les performances initiales.

Références

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). Bert : Pre-training of deep bidirectional transformers for language understanding.

EFRON B. (1979). Bootstrap Methods : Another Look at the Jackknife. *Annals of Statistics*, **7**, 1–26.

GROUIN C., GRABAR N. & ILLOUZ G., Éds. (2021). *Actes de TALN 2021 (Traitement automatique des langues naturelles)*, Lille.

LOSHCHILOV I. & HUTTER F. (2019). Decoupled Weight Decay Regularization. *arXiv :1711.05101 [cs, math]*.

MARTIN L., MULLER B., ORTIZ SUÁREZ P. J., DUPONT Y., ROMARY L., DE LA CLERGERIE É., SEDDAH D. & SAGOT B. (2020). CamemBERT : a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 7203–7219, Online : Association for Computational Linguistics.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Classification multilabel de concepts médicaux pour l'identification du profil clinique du patient

Christel Gérardin[1, 3] Pascal Vaillant[2, 4] Perceval Wajsbürt[2, 5] Clément Gilavert[6]
Ali Bellamine[1] Emmanuelle Kempf[1, 7] Xavier Tannier[2, 5]

(1) Assistance Publique – Hôpitaux de Paris, prenom.nom@aphp.fr
(2) Laboratoire d'Informatique Médicale et d'Ingénierie des Connaissances en eSanté (LIMICS)
(3) Institut Pierre Louis d'Epidémiologie et de Santé Publique, Sorbonne Université, Inserm, 27 rue Chaligny, 75012 PARIS, christel.ducroz-gerardin@iplesp.upmc.fr
(4) Université Sorbonne Paris Nord, F-93000, Bobigny, France, vaillant@univ-paris13.fr
(5) Sorbonne Université, prenom.nom@sorbonne-universite.fr
(6) CG Conception, c.gilavert@cg-conception.fr
(7) Département d'oncologie médicale, hôpital Henri Mondor et Albert Chenevier, Créteil, France

RÉSUMÉ

La première tâche du Défi fouille de textes 2021 a consisté à extraire automatiquement, à partir de cas cliniques, les phénotypes pathologiques des patients regroupés par tête de chapitre du MeSH-maladie. La solution présentée est celle d'un classifieur multilabel basé sur un transformer. Deux transformers ont été utilisés : le camembert-large classique (run 1) et le camembert-large *fine-tuné* (run 2) sur des articles biomédicaux français en accès libre. Nous avons également proposé un modèle « bout-en-bout », avec une première phase d'extraction d'entités nommées également basée sur un transformer de type camembert-large et un classifieur de genre sur un modèle Adaboost. Nous obtenons un très bon rappel et une précision correcte, pour une F1-mesure autour de 0,77 pour les trois runs. La performance du modèle « bout-en-bout » est similaire aux autres méthodes.

ABSTRACT

Multilabel classification of medical concepts for patient's clinical profile identification

This year, the first task of the French Text Mining Challenge consisted in automatically extracting the pathological phenotypes of patients from clinical texts, grouped by head's chapter of the MeSH, disease-section. Benefiting from the annotations of previous years. The solution presented is a multilabel classifier based on a transformer. Two transformers were used : the classic Camembert-large (run 1) and a Camembert-large fine tuned on French biomedical articles in free access. We have also proposed an "end-to-end" model, with a first phase of named entity recognition also based on a transformer and a gender information extracted via an Adaboost model. The results obtained are as follows : run 1 : recall = 0.874, precision = 0.696 and F1-measure = 0.775, run 2 : F1-Measure = 0.771 and run 3 : 0.770. The performance of the end-to-end model is comparable to that of other methods.

MOTS-CLÉS : classification multilabel, Transformer, extraction d'entités nommées, concepts médicaux.

KEYWORDS: multi-label classification, Transformer, named entity recognition, medical concepts.

Atelier DÉfi Fouille de Textes (DEFT), pages 21–30.

1 Introduction et données utilisées

L'extraction automatisée des caractéristiques cliniques des patients à partir des comptes rendus médicaux est devenue un enjeu majeur en données médicales depuis l'apparition du dossier patient informatisé. Dans ce contexte, le défi fouille de textes 2021 (Grouin *et al.*, 2021), a organisé une épreuve d'extraction d'entités nommées à partir de cas cliniques.

Les données d'entraînements fournies par le DEFT 2021 regroupent un ensemble de 167 cas-cliniques comprenant les annotations des années précédentes (DEFT 2019 et DEFT 2020), en particulier les entités de type *signe ou symptôme* et *pathologie* avec leurs caractéristiques (négation, hypothèse, lien avec une personne autre que le ou la patiente). L'objectif de la tâche est de réaliser un phénotypage pour chaque cas : c'est-à-dire de déterminer le profil clinique du cas par l'extraction des caractéristiques pathologiques, décrites par tête de chapitre du MeSH, section [C]-Maladies [1].

La liste des intitulés descriptifs est la suivante : Infections bactériennes et mycoses, Maladies virales, Maladies parasitaires, Tumeurs, Maladies ostéomusculaires, Maladies de l'appareil digestif, Maladies du système stomatognathique, Maladies de l'appareil respiratoire, Maladies oto-rhino-laryngologiques, Maladies du système nerveux, Maladies de l'œil, Maladies urogénitales de l'homme, Maladies de l'appareil urogénital féminin et complications de la grossesse, Maladies cardiovasculaires, Hémopathies et maladies lymphatiques, Malformations et maladies congénitales, héréditaires et néonatales, Maladies de la peau et du tissu conjonctif, Maladies métaboliques et nutritionnelles, Maladies endocriniennes, Maladies du système immunitaire, États, signes et symptômes pathologiques, Troubles dus à des produits chimiques, Plaies et blessures. À ces descripteurs complets correspondent respectivement les labels suivants : *infections, virales, parasitaires, tumeur, osteomusculaires, digestif, stomatognathique, respiratoire, ORL, nerveux, oeil, homme, femme, cardiovasculaires, hemopathies, genetique, peau, nutritionnelles, endocriniennes, immunitaire, etatsosy, chimiques, blessures*.

La Figure 1 présente la répartition des labels dans le jeu de données d'entraînement, à titre indicatif. Le label *etatsosy* apparaît dans 141 textes tandis que *stomatognathique* n'est présent que dans 3 textes. La Figure 2 présente le nombre de labels par document, avec une médiane à 3.

Du fait de cette répartition hétérogène et du faible volume du jeu d'entraînement, nous avons ajouté au jeu d'apprentissage l'ensemble des termes du MeSH français, section [C]-Maladies.

2 Description du système

La Figure 3 décrit l'architecture générale du système que nous proposons.

Une analyse du jeu de données d'entraînement nous montre que ce sont les entités de *pathologie* et *signe ou symptôme* (*sosy*) qui donnent lieu à des classifications MeSH-maladies. Les autres entités peuvent donc être ignorées, et parmi les entités de ces deux types, deux cas doivent être ignorés également :

— les entités étiquetées dans le jeu de données comme niées, hypothétiques ou associées à une autre personne que le ou la patiente ;
— les entités n'ayant pas d'attribut particulier dans le jeu de données mais correspondant à des résultats négatifs (examen normal, analyses négatives ...).

1. http://mesh.inserm.fr/FrenchMesh/index.html

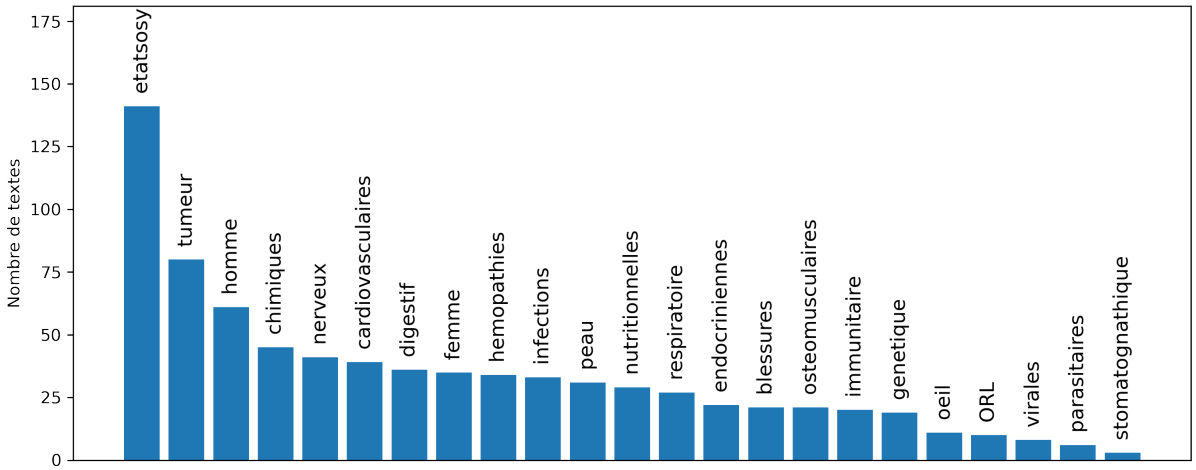

FIGURE 1 – Nombre de textes étiquetés par chaque label, dans le jeu d'entraînement

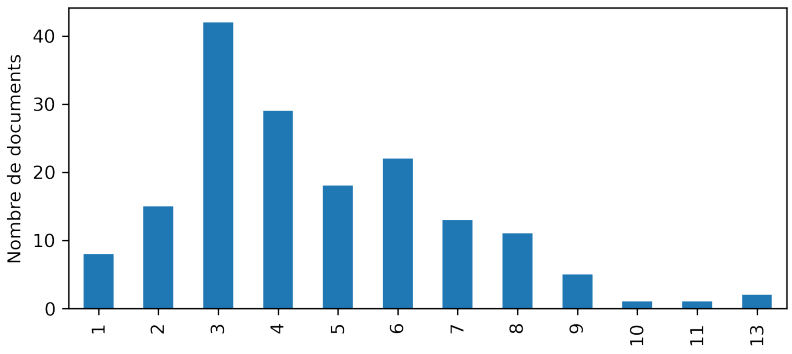

FIGURE 2 – Nombre de labels par document dans le jeu d'entraînement

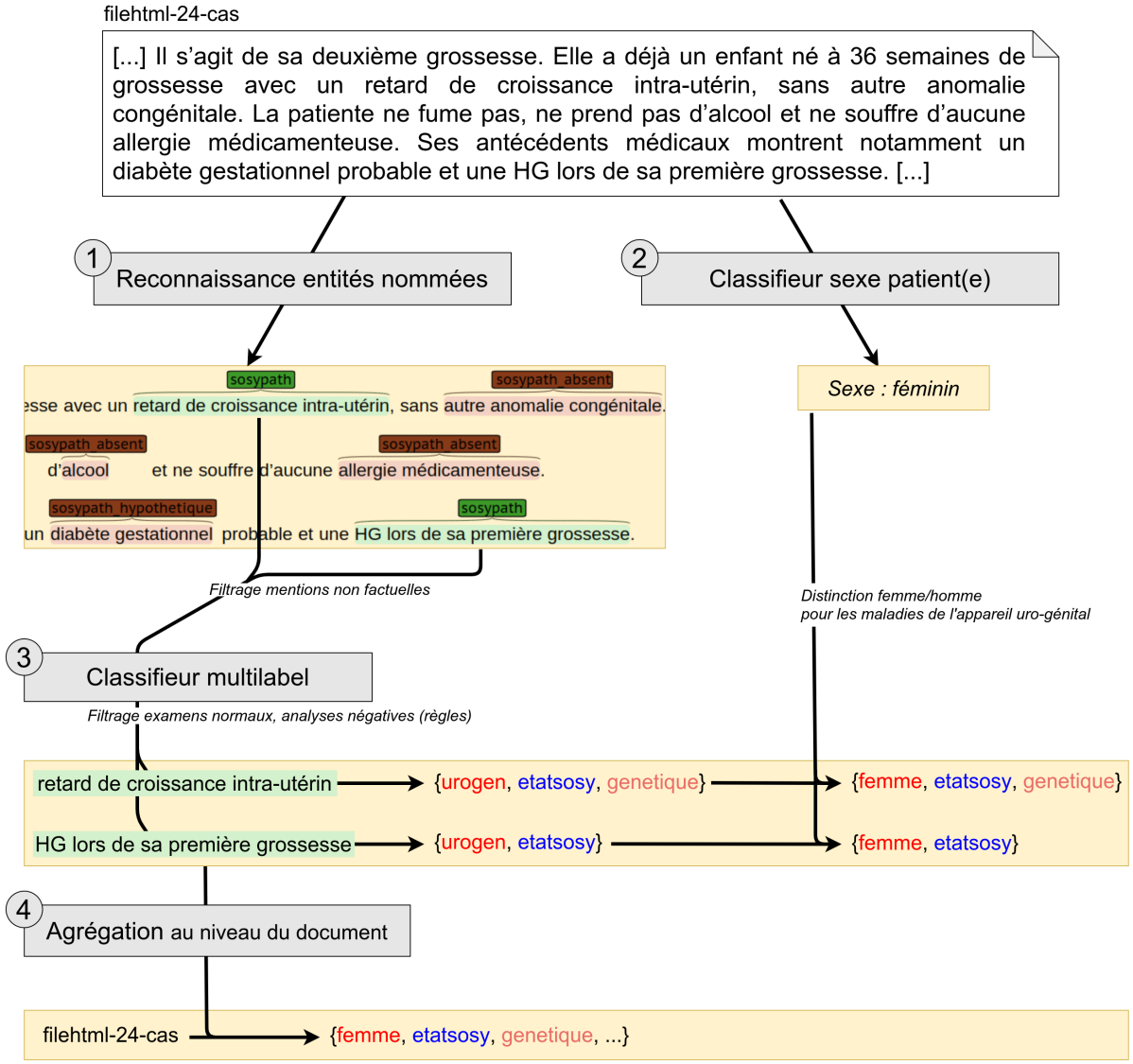

FIGURE 3 – Architecture générale du système

Les entités de type *pathologie* et *sosy*, que nous utilisons pour notre chaîne de traitement, sont fournies par les organisateurs dans le jeu d'entraînement comme dans le jeu de test. Néanmoins, dans l'optique d'évaluer un système « bout-en-bout », nous testons également l'utilisation d'un outil de reconnaissance d'entités nommées (REN) pour réaliser l'extraction de ces entités (run 3). Il s'agit de l'étape 1 présentée à la Figure 3, qui est donc optionnelle. Pour ce système, nous fusionnons les entités *pathologie* et *sosy* en une seule, pour conduire aux entités à extraire : *sosy_path*, *sosy_path_absent* (i.e. nié), *sosy_path_hypothetique*, *sosy_path_non_associe* (i.e. relatif à une autre personne). La fusion se justifie, selon nous, d'une part par la proximité sémantique des deux concepts, d'autre part par l'avantage de regrouper les contextes syntaxiques liés à la la négation, l'hypothèse, la parenté, pour favoriser l'apprentissage de ces notions non triviales.

Par ailleurs, les chapitres MeSH *femme* et *homme* ne se distinguent parfois que par le sexe du ou de la patiente (par exemple, *anurie*). Il est donc nécessaire de construire un classifieur prédisant le sexe à partir du contenu du compte-rendu (étape 2 de la figure).

Une fois les termes d'intérêt extraits par le système, un classifieur a pour tâche de prédire le ou les chapitres MeSH concernés par chaque terme (étape 3). Il s'agit donc d'un classifieur multilabel et multiclasse (les 22 classes représentées dans le jeu de données, en agrégeant *femme* et *homme* en *urogen*). Nous entraînons ce classifieur sur les termes du jeu d'entraînement, mais également sur l'ensemble des termes français de la classification MeSH-maladie contenue dans chaque branche de l'arborescence, classés par tête de chapitre [2].

Enfin, nous agrégeons les informations extraites aux niveaux des termes, pour conclure sur la classification de chaque document.

Les sections qui suivent détaillent chacune de ces étapes.

2.1 Reconnaissance d'entités nommées

Nous présentons ici notre modèle de reconnaissance d'entités nommées, ainsi que le *fine tuning* que nous avons effectué sur le modèle camemBERT (Martin *et al.*, 2020).

2.1.1 Architecture du modèle

Le modèle de reconnaissance d'entités nommées utilisé pour notre système « bout-en-bout » est un évaluateur exhaustif composé d'un Transformer BERT (Devlin *et al.*, 2019) et d'un LSTM bidirectionnel (Hochreiter & Schmidhuber, 1997). La méthode employée est similaire à celle de Yu *et al.* (2020). Les extractions sont des triplets (début, fin, classe). Chaque mot du texte est d'abord divisé en *word pieces* et contextualisé par le Transformer. Les représentations des 4 dernières couches de BERT sont moyennées avec des poids appris, et les *word pieces* d'un mot sont agrégés par *max-pooling* pour construire sa représentation. Un encodage de type char-CNN (Lample *et al.*, 2016) du mot est également concaténé à la représentation précédente. Ces représentations de mots passent par un *Highway LSTM* à 3 couches (Kim *et al.*, 2017) pour obtenir la représentation finale de chaque mot E_i. Enfin, chaque entité possible est évaluée par un produit scalaire entre les représentations de ses bornes de début et de fin :

$$P(span(i,j,k)) = \sigma((W_k^{\text{begin}} \cdot E_i + b_k^{\text{begin}}) \cdot (W_k^{\text{begin}} \cdot E_j + b_k^{\text{end}}) + \text{bias})$$

2. http://mesh.inserm.fr/FrenchMesh/index.html

Les paramètres sont optimisés *via* un objectif de *cross entropy* avec **Adam** (Kingma & Ba, 2015). Nous utilisons un pas d'apprentissage à décroissance linéaire avec un *warmup* de 10 % et deux valeurs initiales : $4 \cdot 10^{-5}$ pour le Transformer, et $6 \cdot 10^{-4}$ pour les autres paramètres.

2.1.2 *Fine-tuning* du camemBERT

Le modèle Transformer utilisé pour la reconnaissance d'entités nommées (ainsi que pour le classifieur) a été au préalable *fine-tuné* sur un jeu de données en accès libres sur Europe-PMC [3] : une première extraction a été réalisée sur l'ensemble des articles. Une fois cette base de d'articles extraite, un traitement de détection de la langue a ensuite été réalisé sur les corps de texte, avec la librairie *langdetect* de Python. Une restriction a ensuite été opérée sur le type d'articles pour ne retenir que les suivants : *case-report* pour environ 1 900 textes, *research-article* (1 060 textes), *brief-report*, *review-article*, *abstract*, *letter*, *chapitre-article*, *discussion*. Le *fine-tuning* a été ensuite réalisé à partir du modèle camemBERT-large (Martin *et al.*, 2020) pour 30 époques. La perplexité calculée sur un jeu de validation de 20 % était de 2,32.

2.2 Classification du sexe du patient ou de la patiente

Le corpus est assez homogène ; il est constitué de documents qui sont des descriptions de cas cliniques. Dans la majorité des cas, ces documents parlent d'une unique personne. L'information sur le sexe de cette personne est dans ce type de textes une donnée déterminante, en particulier pour discriminer entre les deux étiquettes *femme* et *homme*, comme indiqué ci-dessus.

Afin d'entraîner un classifieur à déterminer le sexe, nous avons recueilli un grand nombre de variables candidates afin d'évaluer leur pertinence. Une observation des documents a tout d'abord permis de déterminer que dans un très grand nombre de cas, dans ce type de documents, l'information décrivant le patient se trouvait dans la première phrase. Nous avons donc pondéré les variables par leur distance au début du texte (suivant une pondération fonction du numéro d'ordre de la phrase dans le document, commençant à 1 pour la première phrase et décroissant linéairement jusqu'à 0,5 pour la dernière).

Variable observée	**Information Mutuelle avec la classe « sexe »**
1. genre du mot « patient(e) »	0,5362816
2. genre des adjectifs appliqués à des humains	0,4401735
3. sexe associé aux préfixes caractérisant des morphèmes biomédicaux (organes, maladies, procédures chirurgicales)	0,2874693
4. genre des mots de civilité	0,2178381
5. genre des mots désignant l'individu	0,1874953
6. genre des pronoms personnels 3ps	0,0771561
7. indication explicite du sexe	0,0200703
8. genre des prénoms	0,0185550
autres variables explorées	$< 10^{-2}$

TABLE 1 – Information mutuelle de variables extraites des documents avec la classe *sexe*.

3. http://europepmc.org/

Nous avons ensuite relevé les variables qui nous paraissaient significatives lors d'un premier parcours qualitatif du corpus (voir Table 1). La variable la plus significative est (1) le **genre du mot *patient(e)***. Cette variable, lorsqu'elle est présente (moitié des documents environ), indique sans ambiguïté le sexe. Les autres variables qui contribuent significativement à déterminer le sexe sont, par ordre d'importance : (2) le **genre des adjectifs appliqués à des humains**. Il s'agit des adjectifs qualificatifs qui sont sans ambiguïté utilisés pour des personnes (*« âgé(e) »*, *« né(e) »*, *« enceinte »* ...) ; certains décrivent le contexte de consultation, et nécessitent une petite vérification du contexte (*« adressé(e) (à notre service) »*, *« présenté(e) (aux urgences) »*, *« revu(e) (en consultation) »* ...). (3) Le nombre d'occurrences de **morphèmes faisant référence à des concepts biologiques ou médicaux** spécifiques à un sexe (par exemple *péni-*, *utér-*, *testi-*, *vagin-*. La liste utilisée pour ces morphèmes a été construite en partant de la liste des termes du MeSH français, maintenue par l'INSERM, limitée aux sous-arbres situés dans la hiérarchie sous les nœuds A05 (*appareil urogénital*) pour les termes d'anatomie, C12 (*maladies urogénitales de l'homme*) et C13 (*maladies de l'appareil urogénital féminin et complications de la grossesse*) pour les termes désignant des maladies, enfin E04.950 (*procédures de chirurgie urogénitale*) pour les termes de chirurgie. La liste des termes complets a ensuite été « rabotée » pour factoriser les termes ayant des préfixes communs suffisamment caractéristiques (*stems*). (4) Le **genre des appellatifs de civilité**, sous leurs différentes formes de surface (*« M. »*, *« Mr »*, *« Monsieur »*, *« Mme »*, *« Madame »* ...). (5) Le **genre des noms communs fréquemment utilisés pour désigner un individu humain** (*femme*, *homme*, *enfant* ...).

Plus marginalement, mais encore utilement : (6) Le **genre des pronoms personnels de troisième personne du singulier** utilisés dans le texte (méthode rapide, sans détection de coréférences). (7) L'**indication explicite du sexe** (*« masculin »* ou *« féminin »*). (8) le **genre des prénoms**, déterminé d'après une liste de référence (INSEE) des prénoms les plus fréquemment donnés en France et du genre associé (en effet, dans les études de cas, le patient est souvent désigné par « *M. A.* » ou « *Mme B.* », mais aussi parfois par un prénom).

Pour la collecte de certaines variables, nous avons extrait les catégories morphosyntaxiques (POS, genre, nombre) et les dépendances syntaxiques à l'aide de la librairie *stanza* (Qi *et al.*, 2020).

Nous avons annoté manuellement la classe **sexe du patient** sur le corpus d'entraînement et nous avons entraîné un classifieur supervisé **AdaBoost** (Freund & Schapire, 1997), à partir de ces données, pour déterminer la fonction de prédiction du sexe à partir d'un document texte. Afin de valider cette approche, nous avons entraîné le classifieur sur 80 % des données d'entraînement fournies et l'avons validé sur un jeu de 20 %. Le rappel et la précision étaient tous les deux de 1, y compris sur le jeu de test final. Les cas qui échappent à cette approche de prédiction du sexe du patient sont les rares descriptions de cas cliniques qui concernent plusieurs patients à la fois.

2.3 Classifieur de mentions en chapitres MeSH et agrégation des résultats

Nous effectuons un filtre préalable sur les sorties de la REN ou sur les entités fournies par les organisateurs, de façon à retirer les résultats négatifs (examen normal, analyses négatives). En effet, ces éléments sont souvent annotés comme des *sosy* dans le jeu de données, mais ne doivent pas donner lieu à une annotation MeSH. Ce filtrage est assuré par des expressions régulières simples.

Le classifieur utilisé est un classifieur également composé d'un Transformer BERT (Devlin *et al.*, 2019), plus spécifiquement, un Transformer de type CamembertForSequenceClassification de la librairie Huggingface (Martin *et al.*, 2020; Wolf *et al.*, 2020) comprenant une dernière couche

linéaire en sortie. Pour permettre d'emblée une classification multi-classes, la fonction de perte est la cross-entropie binaire, sommée pour toutes les classes. Deux modèles ont été entrainés : un modèle camembert-large, avec un optimiseur Adam (Kingma & Ba, 2015), pour 50 époques avec un pas d'apprentissage à décroissance linéaire, débutant à $1 \cdot 10^{-5}$. Un jeu de validation de 20 % pour le camembert-large et de 15 % pour le modèle *fine-tuné*.

Pour la prédiction, les scores sont calculés par la fonction sigmoïde en sortie. Le seuil retenu pour la prédiction définitive est celui qui maximise la F1-mesure sur le jeu de validation.

Le deuxième modèle utilisé correspond au CamembertForSequenceClassification *fine-tuné* à partir d'un jeu de données biomédicales françaises en libre-accès sur Europe-PMC, correspondant au même modèle décrit à la section 2.1.2.

Pour finir, la classe *urogen* est séparée en *femme* ou *homme* selon le sexe du ou de la patiente, tel que prédit par l'étape 2.

Enfin, les résultats issus des étapes précédentes sont agrégés de façon triviale pour composer la sortie finale, c'est-à-dire une liste sans doublon des chapitres MeSH-maladie concernés par le compte-rendu (étape 4).

2.4 Configurations d'entraînement et runs soumis

Concernant les configurations, deux runs ont été soumis à partir des annotations des années précédentes et le troisième run correspond à notre système « bout-en-bout ». Pour les deux runs réalisés à partir des annotations des années précédentes, le premier est réalisé avec le CamembertForSequenceClassification entraîné à partir du camembert-large classique et le second à partir du camembert-large *fine-tuné* sur les données Europe-PMC. Le run réalisé avec notre système bout-en-bout est basé sur le camembert-large *fine-tuné* pour l'extraction et pour le classifieur. Ces configurations sont synthétisées à la Table 2.

	Modèle REN	Extraction du genre	Modèle classifieur
Run 1	Aucun	AdaBoost	camembert-large
Run 2	Aucun	AdaBoost	camembert-large FT
Run 3	REN (camembert-large FT)	AdaBoost	camembert-large FT

TABLE 2 – Configurations des différents runs

3 Résultats et discussion

Les résultats des trois runs proposés sont présentés à la Table 3, respectivement aux méthodes proposées table 2. La méthode ayant conduit au meilleur score est celle du camembert-large avec annotations *sosy* et *pathologie* des années précédentes (avec suppression des termes niés, de parenté ou d'hypothèse). De manière particulièrement intéressante, notre modèle « bout-en-bout » *run* 3 parvient à des résultats très proches de ce dernier modèle, sans s'appuyer sur aucune annotation pour la phase de test, et en ayant été entraîné sur la phase REN uniquement sur les 167 compte-rendus d'entraînement. L'étape du *fine-tuning* du camembert-large, *run* 2, n'a pas permis d'améliorer le

	Précision	Rappel	F1-Mesure
Run 1	0,696	0,874	0,775
Run 2	0,677	0,875	0,771
Run 3	0,689	0,872	0,770
Médiane DEFT 2021			0,700
Meilleur DEFT 2021	0,885	0,750	0,812

TABLE 3 – Résultats officiels

	Seuils	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
Run 1	Rappel	0,921	0,908	0,906	0,900	0,893	0,882	0,874	0,860
	Précision	0,619	0,644	0,655	0,659	0,666	0,685	0,696	0,711
	F1-Mesure	0,741	0,753	0,760	0,761	0,763	0,771	**0,775**	**0,778**
Run 2	Rappel	0,922	0,919	0,915	0,904	0,904	0,895	0,889	0,872
	Précision	0,628	0,640	0,657	0,664	0,672	0,677	0,685	0,716
	F1-Mesure	0,747	0,754	0,765	0,766	0,771	**0,771**	0,774	**0,787**
Run 3	Rappel	0,908	0,904	0,896	0,887	0,885	0,872	0,865	0,858
	Précision	0,638	0,650	0,672	0,676	0,686	0,689	0,700	0,728
	F1-Mesure	0,750	0,756	0,768	0,767	0,773	**0,770**	0,774	**0,788**

TABLE 4 – Variation des résultats sur le jeu de test, en fonction du seuil de classification en sortie des 3 modèles. Les valeurs en gras correspondent aux résultats officiels (pour les seuils 0.8 et 0.7, respectivement pour les run 1 et 2-3), et aux meilleurs résultats.

score, ce qui s'explique probablement par le fait que le camembert-large est déjà entraîné sur un très gros volume de données, y compris très hétérogènes, de même que les cas cliniques proposés pour la phase de test. Par ailleurs, le volume de 4 000 articles était probablement insuffisant pour permettre un réel apport au modèle.

A titre d'expérimentation supplémentaire (non soumise aux organisateurs), le modèle bout-en-bout basé sur le camembert-large (Martin *et al.*, 2020), non *fine-tuné*, pour la REN et le classifieur, fourni les résultats suivants avec le script d'évaluation des organisateurs, sur les données de tests : R=0, 871 P=0, 701 F=0, 777. Tous les autres paramètres étant égaux par ailleurs au *run* 1.

Par ailleurs, le seuil en sortie de classifieur nous a paru être un paramètre-clé dans la tache du DEFT 2021. En effet, celui-ci présentait des variations significatives en fonction du volume jeu de données de validation : 20 % pour le classifieur-camemembert large et 15 % pour le modèle *fine-tuné*. Il a été calculé à 0,7 pour le modèle *fine-tuné* et à 0,8 pour le modèle large classique sur ces jeux de validation. Pour illustrer cette importance, nous avons fait varier uniquement ce seuil pour les 3 modèles décrits et calculé les résultats sur le jeu de test avec le script d'évaluation. Les résultats correspondants sont présentés table 4. On observe que les 3 modèles sont très équivalents et que le meilleur score est obtenu pour le modèle « bout-en-bout ».

Enfin, notons que la balance rappel/précision est largement en faveur du rappel dans toutes nos expériences, à l'opposé du système qui se classe premier à la campagne, ce qui pourrait rendre des expériences d'hybridation intéressantes.

Références

DEVLIN J., CHANG M. W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies - Proceedings of the Conference*, **1**, 4171–4186.

FREUND Y. & SCHAPIRE R. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, **55**(1), 119–139. DOI : https ://doi.org/10.1006/jcss.1997.1504.

GROUIN C., GRABAR N. & ILLOUZ G. (2021). Classification de cas cliniques et évaluation automatique de réponses d'étudiants : présentation de la campagne deft 2021. In *Actes de DEFT, Lille*.

HOCHREITER S. & SCHMIDHUBER J. (1997). Long Short-Term Memory. *Neural Computation*, **9**(8), 1735–1780. DOI : 10.1162/neco.1997.9.8.1735.

KIM J., EL-KHAMY M. & LEE J. (2017). Residual LSTM : Design of a Deep Recurrent Architecture for Distant Speech Recognition. In *Interspeech 2017*, volume 2017-Augus, p. 1591–1595, ISCA : ISCA. DOI : 10.21437/Interspeech.2017-477.

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

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, Stroudsburg, PA, USA : Association for Computational Linguistics. DOI : 10.18653/v1/N16-1030.

MARTIN L., MULLER B., ORTIZ SUÁREZ P. J., DUPONT Y., ROMARY L., DE LA CLERGERIE É., SEDDAH D. & SAGOT B. (2020). CamemBERT : a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 7203–7219, Online : Association for Computational Linguistics.

QI P., ZHANG Y., ZHANG Y., BOLTON J. & MANNING C. D. (2020). Stanza : A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics : System Demonstrations*.

WOLF T., DEBUT L., SANH V., CHAUMOND J., DELANGUE C., MOI A., CISTAC P., RAULT T., LOUF R., FUNTOWICZ M., DAVISON J., SHLEIFER S., VON PLATEN P., MA C., JERNITE Y., PLU J., XU C., SCAO T. L., GUGGER S., DRAME M., LHOEST Q. & RUSH A. M. (2020). Transformers : State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing : System Demonstrations*, p. 38–45, Online : Association for Computational Linguistics.

YU J., BOHNET B. & POESIO M. (2020). Named Entity Recognition as Dependency Parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 6470–6476, Stroudsburg, PA, USA : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.577.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

DEFT 2021: Évaluation automatique de réponses courtes, une approche basée sur la sélection de traits lexicaux et augmentation de données

Timothée Poulain[1] Victor Connes[1]
(1) Université de Nantes, LS2N, 2 Chemin de la Houssinière, 44300 Nantes, France
timothee.poulain@univ-nantes.fr, victor.connes@univ-nantes.fr

RÉSUMÉ

Cet article présente la participation de l'équipe Proofreaders du LS2N au DÉfi Fouille de Textes 2021 (DEFT 2021). La tâche proposée consiste en la poursuite automatique de l'évaluation de réponses courtes d'étudiants (EAQRC) à partir de quelques réponses déjà corrigées par l'enseignant pour chaque énoncé. Une étude comparative de différents traits lexicaux, ainsi qu'une augmentation artificielle de données et de différents modèles de régression pour la notation des réponses courtes est réalisée. Les méthodes sont évaluées en termes de précision, d'erreur quadratique moyenne et de score de corrélation de Spearman. Notre erreur quadratique moyenne varie entre 0.090 et 0.101 et notre précision entre 0.147 et 0.17.
Le code source est disponible à l'adresse suivante : https://github.com/poulain-tim/DEFT_2021

ABSTRACT

DEFT 2021 : Automatic short answer grading, a lexical features selection and data augmentation based approach.

This paper presents the participation of the LS2N Proofreaders team at DÉfi Fouille de Textes 2021 (DEFT 2021). The proposed task consists of Automatic Short Answer Grading (ASAG) continuation from a few answers already corrected by the teacher for each question. A comparative study of different lexical features, as well as artificial data augmentation and different regression models for grading short answers, has been performed. The methods are evaluated in terms of accuracy, mean square error, and Spearman correlation score. Our mean squared error ranges from 0.090 to 0.101 and the accuracy between 0.147 and 0.17.
The source code is available at the following address, https://github.com/poulain-tim/DEFT_2021

MOTS-CLÉS : Questions à réponses courtes (QRC), Evaluation automatique des réponses courtes, e-learning, apprenant, DEFT, sélection de caractéristiques.

KEYWORDS: Short answer questions (SAQ), Automatic short answer grading (ASAG), e-learning, learner, DEFT, feature selection.

1 Introduction

Le développement et l'augmentation du nombre de cours dans des plate-formes et formations digitales dans la sphère éducative traduisent un renouvellement des modalités d'apprentissage (Acquatella, 2018). Le modèle d'apprentissage évoluant, il faut adapter un modèle d'évaluation en conséquence.

Atelier DÉfi Fouille de Textes (DEFT), pages 31–40.

L'assimilation de compétences se matérialise par des tests sur les connaissances de l'apprenant. Dans cette pluralité de stratégies d'évaluation, questionnaire à choix multiples, question à choix uniques, nous nous concentrons sur les questions à réponses courtes. L'épreuve de QRC (questions à réponses courtes) est reconnue comme un dispositif d'évaluation performant permettant à l'élève de formuler ces propres réponses, mais aussi au professeur d'évaluer l'argumentaire, la rédaction et la synthétisation. Contrairement aux questionnaires à choix multiples pour lesquels la restitution automatique des résultats se fait de manière précise et aisée, l'évaluation automatique des questions à réponses courtes est un domaine de recherche à part entière, comme en témoigne l'étude comparative de Galhardi & Brancher (2018) qui prend en compte 44 systèmes différents d'évaluation des QRC. C'est dans ce cadre, que la campagne d'évaluation scientifique francophone, DÉfi et Fouille de textes (DEFT 2021) (Grouin *et al.*, 2021) proposent deux tâches sur la correction automatique de copies électroniques d'étudiants. La première est une évaluation automatique des réponses d'après une référence professorale existante (Tâche n°2) et la deuxième est une poursuite automatique de l'évaluation de réponses d'étudiants à partir de premières évaluations suivant des questions aussi bien ouvertes que fermées (Tâche n°3). Dans cet article, seulement l'évaluation de la tâche n°3 est effectuée.

2 Les méthodes d'évaluation automatique des questions à réponses courtes

Le domaine de l'évaluation automatique des questions à réponses courtes (EAQRC) concerne toutes les approches essayant d'attribuer automatiquement une note à une tentative de réponse courte à une question courte. Les réponses et les questions ne doivent pas dépasser un paragraphe. Habituellement, les corpus sont donc composés de questions posées par des enseignants et de plusieurs tentatives de réponses évaluées par une note. Certaines autres informations peuvent y figurer telles qu'un barème, des réponses de référence ou des éléments de correction. Les ensembles de données habituels pour cette tâche sont [1] [2] [3] [4].

L'approche habituelle pour les EAQRC consiste à extraire des caractéristiques à la fois des réponses et des questions en utilisant des techniques de traitement automatique du langage naturel (TALN) et du web sémantique (WS). La note est ensuite prédite à partir des traits extraits par un algorithme d'apprentissage automatique. Les auteurs de l'étude comparative de ces méthodes (Galhardi & Brancher, 2018) distinguent trois grands types de traits : Lexicaux, Syntaxiques, Sémantiques. Nous faisons un tour d'horizon de ces différents traits à partir des articles suivants Galhardi & Brancher (2018), Galhardi *et al.* (2018), Sultan *et al.* (2016).

Parmi les traits lexicaux, les modèles les plus couramment utilisés sont les n-grammes (y compris les Sac de mots), les méthodes de pondération (TF-IDF) et les méthodes de plongement de mots (principalement Word2Vec, FastText). Nous distinguons dans cet ensemble de traits, des statistiques intrinsèques liées à la réponse (R) et des statistiques exploitant la similarité au niveau lexical entre la réponse traitée (R), sa question associée (Q) et les réponses de référence (RR). Pour les traits des

1. http://web.eecs.umich.edu/~mihalcea/downloads.html
2. www.uni-tuebingen.de/en/research/core-research/collaborative-research-centers/sfb-833/section-a-context/a4-meurers/software-resources-and-corpora.html
3. www.kaggle.com/c/asap-sas
4. www.cs.york.ac.uk/semeval-2013/task7/index.php%3Fid=data.html

statistiques textuelles intrinsèques, on retrouve fréquemment, le nombre de mots, de mots uniques, de caractères, de verbes, de mots mal orthographiés, de longueur moyenne des mots et des phrases, ... À partir de ces statistiques, on infère un certain nombre de rapports tel que la longueur moyenne des mots, le ratio nombre de mots uniques par nombre de mots. On peut aussi s'appuyer sur les statistiques des réponses de références ou de la question pour induire le rapport de longueur (en nombre de mots ou de caractères) entre R/RR ou entre R/Q. Un autre groupe de trait est celui qui utilise une métrique pour mesurer la similarité lexicale entre R/Q et R/RR. Ce groupe se décompose souvent en plusieurs sous-groupes, on y retrouve les mesures de similarité sur les sacs de mots comme les distances Cosine, Euclidienne, Recouvrement, Sorensen-Dice, les similarités établies par les distances d'éditions, Levenshtein, Hamming, Jaro-winkler. On y adjoint les mesures de similarité au niveau de la séquence, tel que LCS (en anglais *longest common subsequence*) ou Ratcliff-Obershelp et les similarités à partir d'algorithme de compression sans perte comme la transformée de Burrows-Wheeler, très utile pour la détection de répétition.

Les traits syntaxiques étudiés sont les étiquetages morphosyntaxiques et leurs similarités dérivées. Les n-grammes, communément utilisés pour modéliser le modèle de langage pour la tâche de EAQRC (Burrows *et al.*, 2015) sont généralement extraits avec un parseur tel que Stanford Parser [5]. On peut aussi extraire un triplet contenant deux mots et leur relation de dépendance.

Enfin, les traits sémantiques sont les caractéristiques inhérentes à un concept, au sens du mot, ayant vocation à déterminer le type de relations lexicales qui existe entre les mots dans une langue et facilitant l'accès sémantique aux mots. Les mesures de similarité à partir de connaissances externes peuvent être considérées comme telles. Ce sont par exemple les méthodes d'étiquetage des rôles sémantiques où l'on attribue des étiquettes aux rôles que les mots représentent dans les phrases en tenant compte de leur aspect sémantique ou encore l'utilisation de la base de données lexicales "Wordnet" pour enrichir et mettre en relation le contenu lexical et sémantique de la langue. D'autres mesures de similarité existent, mais sont cette fois fondées sur le corpus, (Gabrilovich *et al.*, 2007) a démontré l'intérêt d'utiliser une mesure basée sur un corpus (LSA) entraînée sur un corpus spécifique à un domaine. D'autres méthodes acquièrent des informations statistiques pour calculer la relation entre les mots et les documents comme ESA (en anglais *Explicit Semantic Analysis*) et DISCO (en anglais *Extracting DIStributionally similar words using COoccurrences*). Un dernier type de traits sémantiques existe, l'implication textuelle (en anglais *textual entailment*), une méthode qui consiste à juger si un texte peut être déduit d'un autre texte.

D'autres approches plus récentes d'apprentissage profond qui ne sont pas traitées dans cet article ont vu le jour tel que les auto-encoder (Yang *et al.*, 2018), les architectures siamoises (Kumar *et al.*, 2017) ou les modèles à base de transformers pré-entrainés (BERT) par ajustement (*finned tunned*) sur la problématique de EAQRC (Sung *et al.*, 2019).

3 Données

La tâche demandée dans le cadre de cette édition du défi est une tâche de poursuite automatique de l'évaluation de réponses d'étudiants à partir de premières évaluations. Les corpus utilisés se composent d'une centaine d'énoncés en informatique spécialisés en programmation web et bases de données ainsi que des réponses produites par une cinquantaine d'étudiants en moyenne par question, sur deux années d'enseignement. Deux types de question/réponse apparaissent dans ces ensembles de

5. https://nlp.stanford.edu/software/lex-parser.shtml.

données (voir Table 1), les fermées où une réponse syntaxiquement très précise est attendue et les ouvertes où une réponse sémantiquement précise est plus préconisée. Le domaine spécifique de la programmation influence certain de nos choix et nos hypothèses de pré-traitements des données.

Type de question	Exemple de question	Exemple de réponse
Question ouverte	Accessibilité : Indiquez trois précautions à prendre pour garantir [..]	L'enregistrer dans un format qui est couramment utilisé (exemple : .txt) afin de [..]
Question fermée	Donnez le code HTML complet créant le formulaire contenant les champs [...]	<html> <head> <title> Les UEs </title> </head> <body> [...] </body> </html>

TABLE 1 – Exemples de question/réponse pour chaque type de questions

Les données mises à disposition par les organisateurs (Grouin *et al.*, 2021) du défi DEFT 2021 sont composées de deux corpus. Les deux corpus sont séparés au préalable en base d'apprentissage et de test selon les proportions 2/3 et 1/3. L'ensemble d'apprentissage de la tâche n°2 est composé de 50 questions et 50 éléments de réponses du professeur. Parmi ces 50 questions, 30 ont 116 réponses d'étudiants, et les 20 questions restantes ont 17 réponses d'étudiants. L'ensemble de test de la tâche n°2 suit la même logique et est présenté en Table 2.
Le second corpus correspond à la tâche n°3 est composé de 21 questions, sans éléments de réponses du professeur et en moyenne 69 réponses par question (avec un minimum de 41 et un maximum de 116). L'ensemble de test de la tâche n°3 suit la même logique avec en moyenne 9% de réponses d'étudiants déjà corrigées par l'enseignant et est présenté en Table 2.

Corpus	**Entrainement**		**Test**		
	Questions	*Réponses*	*Questions*	*Réponses*	*Réponses Corrigées*
Tâche n°2	50	3820	21	1644	–
Tâche n°3	11	769	6	387	0.09

TABLE 2 – Description des corpus d'entraînement et de test pour les deux tâches

4 Notre Approche

De par la spécificité et la taille de notre corpus, nous choisissons de nous focaliser sur les méthodes d'extraction de traits. En particulier, nous nous intéressons aux traits lexicaux comme présentés dans la section 2. Pour contre-balancer le manque de données, nous employons une méthode de génération artificielle de données, ainsi qu'une méthode de sélection de traits. Enfin en suivant les approches de la littérature scientifique, nous prédisons la note par un algorithme d'apprentissage automatique sur une tâche de régression. Dans les prochaines sections, nous détaillons les différentes modalités de notre approche.

4.1 Pré-traitements des données

Pour tenir compte des particularités linguistiques de notre corpus, nous appliquons un certain nombre de méthodes de pré-traitements aussi bien pour les réponses, que pour les questions. Nous remplaçons les chevrons ouverts et fermants de chaque balises HTML par des symboles "ó" et "ò" en vue de l'utilisation de la méthode de création de plongements lexicaux, FastText. Nous substituons l'ensemble des nombres par le mot NUM, dans le but de réduire notre vocabulaire sachant qu'aucune des questions ne requière de réponses numériques. La suppression de la casse, la suppression des

sauts de lignes "$\backslash n$" ont été aussi réalisées. Après quelques expérimentations, nous avons décidé de ne pas normaliser notre corpus en utilisant la racinisation. En effet, en disséquant les données, et en inspectant le thème informatique et spécialement en programmation web et base de données des énoncés, nous formulons plusieurs hypothèses. La première est la conservation de la ponctuation dans nos modèles. La syntaxe de la programmation web est précise et ponctuée par un ensemble de signes graphiques (deux points, point virgule, accolade, ...) . De surcroît, si nous supprimons celle-ci, cela peut poser problématique pour la notation automatique des questions fermées (par exemple, un type de balise) où chaque signe graphique a son importance. Pour les mêmes raisons, nous décidons de conserver les mots dans leurs flexions originelles et de ne pas faire de racinisation. Par exemple, si le mot "former" apparaît dans le texte, et que nous utilisons la racinisation il est transformé en "form", ce qui est problématique car sans ponctuation, ni racinisation, le mot "form" peut émaner d'une balise html.

4.2 Augmentation des données

Il y un adage en apprentissage automatique qui dit "Plus nous disposons de données, meilleures sont les performances que nous pouvons atteindre". Par conséquent, une augmentation appropriée des données est utile pour améliorer les performances de notre modèle. Néanmoins, acquérir et étiqueter des données supplémentaires manuellement est un travail long, coûteux et fastidieux. Très en vogue dans le domaine de vision par ordinateur, où l'on peut générer facilement d'autres images en la retournant, en la bruitant, la génération de données en TAL est plus compliqué en raison de la complexité du langage. En effet, chaque mot ne possède pas forcément un synonyme, et la substitution peut créer des ambiguïtés. Toutefois, pour notre cas spécifique, nous tentons d'augmenter artificiellement notre corpus de deux manières. La première manière est simplement d'ajouter à notre corpus d'entraînement les données de la tâche n°2, les réponses et les questions.

La deuxième manière a recours à une méthode d'augmentation de données fréquemment utilisée en TALN. Notre objectif ici est de créer pour chaque question et chaque réponse, des nouvelles analogues, les réponses générées héritant de la même note que l'original. Nous générons donc des questions et des réponses similaires à celles déjà présentes dans le jeu de données d'entraînement sauf que nous substituons à chaque entrée un ou plusieurs mots sémantiquement liés. Nous utilisons une méthode d'augmentation originalement proposée par Xie *et al.* (2020). L'idée de base est que les mots qui ont un score TF-IDF faible ne sont pas informatifs et peuvent donc être remplacés sans affecter la phrase. Pour ce faire, nous avons calculé l'ensemble des scores TF-IDF pour chaque mot sur le corpus d'entraînement (agrégation question et réponses) excluant "NUM" et "NO_ANS". Chaque question et chaque réponse constitue ici un document indépendant. Seuls les mots obtenant des scores TF-IDF moyens sur l'ensemble des documents inférieur à 1 (seuil manuellement fixé a partir de la distribution des scores TF-IDF) sont gardés comme candidat à la substitution. A l'aide de la méthode des K plus proche voisin (K-pvv) et d'un modèle de language FastText français, nous générons pour chaque mot un ensemble de mots candidats sur la base de leurs similarités en cosinus avec le mot originel dans l'espace des plongements et nous extrayons les 20 mots plus proches sémantiquement. Et enfin, un score pour chaque mot est calculé comme le produit entre la similarité cosinus entre le plongement du mot originel et le mot traité et un moins le TF-IDF du mot. (voir l'équation 1).

$$score(w) = \text{K-ppv}_{w_{origin},w}(embedding(w_{origin}, w)) \times (1 - \text{TF-IDF}(w)) \quad (1)$$

Cette méthode permet d'aboutir à un lexique de substitution n'affectant pas le sens avec leurs scores

associés, dont voici quelques exemples : ("septembre", "aout", 0.76), ("soucier", "préoccuper" , 0.74), ("continuellement", "constamment", 0.72), ("expliquer", "décriver", 0.70), ("certainement", "incontestablement", 0.63), ("provoque", "occasionne", 0.58), ("manières", "façons", 0.52), ("cependant", "paradoxalement", 0.50). [6]

Les nouvelles questions et réponses sont ensuite générées en remplaçant des mots des phrases originelles par les meilleurs candidats dans le lexique (de 1 à 2 mots).

En pratique nous avons choisi de remplacer au maximum un mot par question ou réponse et de créer 5 nouvelles questions ou réponses avec les 5 meilleurs mots candidats à la substitution.

Avec cette méthode nous obtenons au total 272 questions et 9153 réponses à partir des 67 questions et des 3736 réponses initialement présentes dans notre corpus d'entraînement.

4.3 Traits lexicaux

Ces traits sont aussi bien centrés sur les statistiques extraites de chaque réponse individuellement que les mesures de similarité pour R/Q et R/RR.

Traits liés au comptage : Nombre de mots, Nombre de chevrons ouverts et fermés, nombre de mots uniques, nombre de mots mal orthographiés (utilisant la bibliothèque python "SpellChecker"), le nombre de caractères, longueur moyenne des mots.

Ratios : Le rapport de longueur entre la réponse et les réponses de références exprimés en nombre de mots, le rapport de longueur entre la réponse et la question exprimés en nombre de mots.

Mesure de la similarité sur les sacs de mots : La distance Cosine, Recouvrement, Sorensen-Dice.

Mesure de la similarité sur la fréquence des mots : La distance Cosine.

Mesure de la similarité par distance d'édition : Distance de Levenshtein, Hamming, Jaro-Winckler.

Mesure de similarité au niveau de la séquence Taille de la plus longue séquence de caractère commune, similarité de Ratcliff Obershelp.

Si le nombre de réponses de références est supérieur à un, on extrait aussi bien le maximum, le minimum et la moyenne de la similarité entre R/RR pour chacune des mesures de similarité, distance d'édition etc.

Au total, nous avons extrait 60 traits différents.

4.4 Traits lexicaux issus des plongements de mots

Dans cette section, nous allons exploiter la représentation lexicale des mots dans un espace continu au lieu d'une représentation traditionnelle. Pour créer nos plongements de mots, nous avons une combinaison de notre corpus d'entraînement et d'un moissonnage de Wikilivres spécialisés [7]. Les Wikilivres sont une collection de textes pédagogiques libres rassemblés en livres et écrits en collaboration regroupant une large variétés de sujets.

6. Une liste exhaustive de ce lexique est disponible à https://github.com/poulain-tim/DEFT_2021/blob/main/DATA/corpus/substitutes.json

7. https://www.kaggle.com/dhruvildave/wikibooks-dataset

Pour notre application nous avons choisi ceux dont le titre contient un des mots clés suivants : "HTML", "JavaScript", "PowerPoint", "CSS". Nous avons restreint notre choix à des Wikilivres en français, pour créer notre corpus textuel nous agglomérons les informations du titre, du résumé et du corps de l'article aux questions et réponses de nos données. Ce corpus textuel est pré-traité (voir Section 4.1 est ensuite utilisé pour créer un modèle de language en utilisant la méthode FastText.

De plus, nous avons également testé une méthode de transfert d'apprentissage (transfert learning) en utilisant les vecteurs du modèle pré-entraînés FastText sur Common Crawl et Wikipedia français que nous spécialisons (fine tunnons) à l'aide de notre corpus textuel.

À partir de ces modèles de langage il est possible de calculer des distances entre les phrases. Soit deux phrases, $S_1 = w_1, w_2, ..., w_n$ et $S_2 = w'_1, w'_2, ...w'_m$. Pour mesurer la similarité lexicale entre S_1 et S_2, nous effectuons la moyenne des vecteurs de plongements de mots normalisés par la norme L2, en suivant l'implémentation de la bibliothèque c++ FastText [8]. Nous choisissons deux méthodes de similarité, la similarité euclidienne et la similarité cosinus. A l'instar des traits lexicaux sur la fréquence des mots, nous effectuons une similarité entre le plongement de la réponse traitée et le plongement de la question et entre le plongement de la réponse traitée et les N-plongements des réponses de référence. De la même façon, nous extrayons les scores de similarité minimale, moyenne et maximale par rapport aux N-réponses de référence que nous utilisons en tant que caractéristiques. La dernière caractéristique extraite est la moyenne de ces N-plongements des réponses de références par rapport au plongement de la réponse traitée.

Si nous ajoutons aux 60 traits lexicaux sur la fréquence des mots, les 9 traits lexicaux issus des plongements de mots, nous atteignons un total de 69 traits.

4.5 Sélection des traits

Après avoir créé de nombreux traits, nous choisissons le meilleur sous-ensemble de traits afin de réduire la dimensionnalité, contribuant à prédire de manière optimale notre note. Pour ce faire, nous utilisons l'algorithme Kbest de la bibliothèque python scikit-learn [9]. Cette méthode prend comme paramètre une fonction de score, dans notre cas *f_regression*, le F-score entre la note et le trait pour une tâche de régression. La fonction renvoie un tableau de score, un score par trait. SelectKBest sélectionne les k premières caractéristiques avec les scores les plus élevés. Nous faisons varier la valeur de k de 10 à 30.

Pour l'ensemble de nos expérimentations, nous utilisons la méthode des forêts aléatoires pour la tâche de régression de la librairie scikit-learn [10].

5 Résultats

À l'instar de Mohler *et al.* (2011), nous calculons un score de corrélation de Spearman et de l'erreur moyenne quadratique sur toutes les réponses des élèves de tous les ensembles de données. En effet,

8. https://fasttext.cc/
9. https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html
10. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

comme expliciter dans Burrows *et al.* (2015), nous pouvons voir le problème soit comme un problème de classification, soit comme un problème de régression. Chaque note maximale des questions n'étant pas normalisée en amont, nous pouvons nous retrouver avec une grande variété d'étiquettes de notes finales, dans notre cas, 21 différentes notes entre 0 et 1. En ce sens, nous choisissons d'aborder le problème comme une régression et non une classification, où l'utilisation de la mesure de l'erreur moyenne quadratique semble plus cohérente. Afin d'utiliser aussi la précision comme mesure d'évaluation, nous remplaçons le score prédit par la valeur de la note la plus proche observée dans les données d'entraînements.

La comparaison des résultats (voir tableau 3) montre l'avantage d'ajouter une à une les nouvelles méthodes au regard des trois métriques utilisées. L'amélioration en terme de précision va de 0.147 à 0.17. Toutefois, les différentes étapes impactent de manière variable les différentes métriques ce qui complexifie l'évaluation de l'apport de chacune de ces étapes. À titre d'exemple, SLP améliore grandement les résultats en termes de RMSE mais faiblement en terme de précision et de corrélation. De plus, nous nous sommes aperçus que les mesures de similarité engageant les questions n'étaient jamais retenues dans les traits pertinents sélectionnés.

Méthodes	RMSE	ρ	Précision
SLM	0.101	0.56	0.147
SLM + SLP	0.099	0.57	0.15
SLM + SLP+ AAD	0.090	0.62	0.152
SLM + SLP+ AAD + PP	0.093	0.61	0.17

TABLE 3 – Résultats sur le corpus de test de la tâche n°3 en fonction des différentes méthodes appliquées (SLM : Statistiques Lexicales sur le Mot, SLP : Statistique Lexicale pour les Plongements lexicaux, AAD : Augmentation Artificiel des Données, PP : Plongements lexicaux Pré-entraînés)

5.1 Résultats officiels

Nous présentons dans cette section les résultats officiels de la campagne DEFT 2021. Nous avons soumis trois exécutions pour la tâche n°3. Le tableau 3 résume nos trois exécutions. Nous constatons que la variante V1 obtient les meilleurs résultats. Les résultats sont légèrement modifiés par rapport au Tableau 3, compte tenu du changement des paramètres de racinisation, du nombre minimal de caractéristiques pour l'algorithme de KBest, du nombre maximal de réponses de références par réponse, ...

— Pour V1 : SLM + SLP + AAD + PP
— Pour V2 : SLM + SLP + AAD
— Pour V3 : SLM + SLP

Méthodes	RMSE	ρ	Précision
V1	0.093	0.61	0.17
V2	0.097	0.58	0.159
V3	0.133	0.60	0.133

TABLE 4 – Résultats de nos trois exécutions de modèles pour la tâche n°3

6 Conclusion

Dans ce travail, nous avons décrit la participation de l'équipe Proofreaders du LS2N à DEFT 2021. L'approche suivie par l'équipe modélise le problème EAQRC comme une tâche de régression, la solution proposée s'appuie sur l'extraction de traits lexicaux joint à une méthode de sélection de traits et une méthode d'augmentation de données textuelles. Les résultats obtenus montrent une corrélation entre les notes prédites et les notes de l'enseignant. Néanmoins, les résultats finaux restent faibles pour évaluer l'impact de chacune des étapes du processus. En perspective de ce travail, nous souhaiterions réévaluer l'apport de chacune des étapes de notre méthode de manière plus formelle.

Références

ACQUATELLA F. (2018). *Analyse stratégique du marché de la formation en ligne : les Moocs comme nouvelle variable des écosystèmes de plateformes digitales*. Thèse de doctorat, Paris, ENST.

BURROWS S., GUREVYCH I. & STEIN B. (2015). The eras and trends of automatic short answer grading. *International Journal of Artificial Intelligence in Education*, **25**(1), 60–117.

GABRILOVICH E., MARKOVITCH S. *et al.* (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJcAI*, volume 7, p. 1606–1611.

GALHARDI L. B. & BRANCHER J. D. (2018). Machine learning approach for automatic short answer grading : A systematic review. In *Ibero-american conference on artificial intelligence*, p. 380–391 : Springer.

GALHARDI L. B., DE MATTOS SENEFONTE H. C., DE SOUZA R. C. T. & BRANCHER J. D. (2018). Exploring distinct features for automatic short answer grading. In *Anais do XV Encontro Nacional de Inteligência Artificial e Computacional*, p. 1–12 : SBC.

GROUIN C., GRABAR N. & ILLOUZ G. (2021). Classification de cas cliniques et évaluation automatique de réponses d'étudiants : présentation de la campagne. In *Actes de DEFT. Lille.*

KUMAR S., CHAKRABARTI S. & ROY S. (2017). Earth mover's distance pooling over siamese lstms for automatic short answer grading. In *IJCAI*, p. 2046–2052.

MOHLER M., BUNESCU R. & MIHALCEA R. (2011). Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In *Proceedings of the 49th annual meeting of the association for computational linguistics : Human language technologies*, p. 752–762.

SULTAN M. A., SALAZAR C. & SUMNER T. (2016). Fast and Easy Short Answer Grading with High Accuracy. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 1070–1075, San Diego, California : Association for Computational Linguistics. DOI : 10.18653/v1/N16-1123.

SUNG C., DHAMECHA T., SAHA S., MA T., REDDY V. & ARORA R. (2019). Pre-training bert on domain resources for short answer grading. 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. 6073–6077.

XIE Q., DAI Z., HOVY E., LUONG M.-T. & LE Q. V. (2020). Unsupervised data augmentation for consistency training.

YANG X., HUANG Y., ZHUANG F., ZHANG L. & YU S. (2018). Automatic chinese short answer grading with deep autoencoder. In *International Conference on Artificial Intelligence in Education*, p. 399–404 : Springer.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

DOING@DEFT : utilisation de lexiques pour une classification efficace de cas cliniques

Nicolas Hiot [1] Anne-Lyse Minard[2] Flora Badin[2]
(1) Université d'Orléans, LIFO, Orléans, France
(2) Université d'Orléans, LLL-CNRS, Orléans, France
nicolas.hiot@etu.univ-orleans.fr, anne-lyse.minard@univ-orleans.fr,
flora.badin@univ-orleans.fr

RÉSUMÉ

Nous présentons dans cet article notre participation à la tâche 1 de la campagne d'évaluation francophone DEFT 2021, sur l'identification du profil clinique du patient. Nous proposons une méthode évolutive et efficace en temps et en ressources pour la classification de documents médicaux pouvant être facilement adaptée à d'autres domaines de recherche. Notre système a obtenu les meilleures performances sur cette tâche avec une F-mesure de 0,814.

ABSTRACT

In this paper, we present our participation to the DEFT 2021 task 1. The task focuses on the identification of patient clinical profile. We propose a method that is upgradable and efficient in time and resources for medical document classification. The method can be easily adapted to other domains. Our system has obtained the best results on this task, with an F-measure of 0,814.

MOTS-CLÉS : cas clinique, transducteur fini, lexique, classification.

KEYWORDS: clinical case, final state transducer, lexicon, classification.

1 Introduction

Nous présentons dans cet article le système que nous avons développé pour l'identification du profil clinique du patient, et qui nous a permis de participer à la tâche 1 de DEFT 2021 [1] (Grouin *et al.*, 2021). La tâche consistait à identifier les types de maladies d'un patient décrit dans un cas clinique et correspondant aux entrées génériques du chapitre C du MeSH [2], un thésaurus bilingue anglais-français du domaine médical. Chaque document décrivant le cas clinique d'un patient, l'extraction des classes du MeSH peut être vue comme un problème de classification (pour chaque classe du MeSH on cherche à savoir si la classe est représentée dans le cas ou non).

L'idée était de concevoir un système capable de répondre au problème efficacement en temps et en ressources. Nous avons mis au point une méthode symbolique, basée sur des transducteurs finis et des lexiques. Le système développé peut être utilisé en temps réel, mis à jour facilement et à un coût environnemental faible. En effet, aucun apprentissage n'est effectué et les pré-traitements consistent

1. https://deft.lisn.upsaclay.fr/2021/
2. http://mesh.inserm.fr/FrenchMesh/index.htm

Atelier DÉfi Fouille de Textes (DEFT), pages 41–53.

uniquement à tokeniser le texte et à raciniser les mots. Nous avons également fait le choix de ne pas utiliser les annotations manuelles disponibles dans le corpus de test pour pouvoir utiliser notre système sur n'importe quel texte brut. Nous avons uniquement utilisé les annotations manuelles en genre pour les cas pour lesquels notre méthode n'était pas en mesure de détecter le genre (2 sur 57). Nous montrons dans la section 3.4 que nous aurions pu nous en passer sans que les performances de notre système ne soient impactées. Notre système a obtenu les meilleurs résultats de la campagne DEFT 2021 pour la tâche 1.

Les organisateurs de DEFT nous ont fourni un corpus d'entraînement composé de 167 cas cliniques rédigés en français, pour un total de 69 256 mots. Les cas cliniques sont anonymes et couvrent différentes spécialités médicales (cardiologie, urologie, oncologie, obstétrique, pulmonaire, gastro-entérologie, etc.). Le corpus a été annoté manuellement sous BRAT en pathologies, signes ou symptômes, parties anatomiques, examen, substances, traitement, dose, mode, moment, fréquence, durée, valeur, etc. (e.g. *« échographie» de type «examen»*). Un enrichissement est effectué sur l'absence ou la présence de certains concepts, un changement, un état, une prise, etc. (e.g. *« diminution» de type «changement» et « possible» de type « assertion»*)

Ces cas sont des descriptions de situations cliniques rares utilisées à des fins pédagogiques, scientifiques ou thérapeutiques. Les classer automatiquement permettrait entre autres d'indexer les cas automatiquement et pourrait aider à l'identification de pathologies.

Ce travail a été réalisé dans le cadre du groupe de travail régional DOING [3], qui s'intéresse à la transformation des données en information, puis en connaissance, en favorisant la collaboration de chercheurs en TAL, en bases de données et en IA.

Après avoir fait un rapide état de l'art du domaine (section 2), nous présentons notre système en section 3, puis les résultats obtenus en section 4. Nous terminerons avec une partie discussion autour des choix d'implémentation du système et une analyse des erreurs (section 5).

2 État de l'art

La tâche d'identification de types de maladie s'assimile à une tâche de classification de documents, mais repose en partie sur l'extraction d'information dans le texte, plus particulièrement sur la reconnaissance d'entités. Cette dernière est une tâche très importante en TAL pour le domaine médical, elle est souvent associée à une tâche de linkage d'entités ou normalisation d'entités. Pour l'anglais nous pouvons par exemple citer les travaux sur la détection des maladies et des troubles et leur normalisation via les CUI (Concept Unique Identifier) de l'UMLS (The Unified Medical Language System) dans le cadre de différentes campagnes d'évaluation : ShARe/CLEF eHealth 2013 Evaluation Lab Task 1 (Pradhan *et al.*, 2013), SemEval 2015 tâche 4 (Elhadad *et al.*, 2015), etc. Le système qui a obtenu les meilleures performances à SemEval 2015 tâche 4 (Pathak *et al.*, 2015) utilise une approche supervisée basée sur des CRF (Conditional Random Fields) et des SVM (Support Vector Machine) pour l'identification des entités. Pour la normalisation des entités, ils recherchent d'abord l'entité identifiée dans l'UMLS, puis ils construisent automatiquement des variantes de ces entités et les recherchent, et enfin s'ils n'ont toujours pas trouvé le CUI ils calculent la similarité entre des

3. DOING (https://www.univ-orleans.fr/lifo/evenements/doing/) est un groupe de travail proposé en 2018 dans le cadre du réseau régional DIAMS (https://www.univ-orleans.fr/lifo/evenements/RTR-DIAMS/). En 2020, DOING était également un atelier du GdR MADICS et est devenu une action du GdR en 2021 https://www.madics.fr/actions/doing/.

chaînes proches dans l'UMLS. En français l'annotation d'entités dans des cas cliniques a fait l'objet de la tâche 3 de l'édition de DEFT 2020 (Cardon *et al.*, 2020) qui portait sur l'extraction des examens, des traitements, des signes ou symptômes, etc. (Minard *et al.*, 2020) ont proposé une méthode basée sur une cascade de CRF pour identifier ces informations. Malgré des performances dans l'ensemble plutôt bonnes, le système ne détecte correctement que la moitié des entités de type *signe ou symptôme* et *pathologie* (meilleure F-mesure respectivement de 0,55 et 0,44). Afin d'identifier les types de maladie, nous aurions pu nous baser sur ces deux types d'information, mais les performances ne semblent pas assez bonnes pour y arriver. Nous avons donc décidé de considérer la tâche comme une tâche de classification basée sur un lexique.

Dans la littérature, la classification de documents est souvent traitée comme un problème de classification traditionnelle. Étant donné des individus (documents) représentés sous forme vectorielle, l'objectif est de prédire la classe (un label) de chaque individu à partir d'un modèle qui a été entraîné sur un corpus d'exemples. Lorsque l'on traite des documents textuels, la représentation vectorielle ne parait pas évidente. La méthode souvent retenue est celle du sac de mots, l'idée est de sélectionner un ensemble de mots de taille n qui représenteront les dimensions de notre espace vectoriel. Ainsi, chaque document est représenté par un vecteur de booléen (représentant l'apparition du mot ou non dans le texte) ou un vecteur d'entier représentant le nombre d'occurrences des termes dans le texte. Afin de travailler sur un espace à dimensions réduites, une sélection des mots les plus révélateurs de chaque classe est nécessaire. L'idée est de trier chaque terme selon un certain critère et de sélectionner les m éléments ayant le score le plus élevé. Une mesure simple est le TF-IDF (pour term frequency-inverse document frequency) (Jones, 1972). Intuitivement, cette mesure est un ratio entre la fréquence d'apparition du terme dans une classe donnée et sa fréquence dans l'ensemble du corpus qui repose sur la loi de Zipf (un terme a plus de chance d'être révélateur d'une classe s'il y est souvent présent ; au contraire, si un terme est trop fréquent dans le corpus, il n'est pas assez discriminant). Dans (Weng *et al.*, 2017), les auteurs cherchent à extraire le domaine médical auquel appartiennent un ensemble de documents. Ils comparent une approche utilisant des réseaux de neurones et une approche utilisant des sacs de mots pondérés par TF-IDF avec un classifieur SVM. Ils montrent des résultats similaires pour les deux approches avec un gain en explicabilité pour l'approche sac de mots. D'autres mesures de pondération existent comme l'index de Jaccard. (Mihalcea & Tarau, 2004) proposent une autre méthode de pondération des termes basée sur l'algorithme de PageRank (Brin & Page, 1998) dans un graphe représentant les interactions entre les mots.

Nous avons fait le choix d'utiliser un transducteur fini et un lexique de "mots clés" plutôt que les méthodes décrites précédemment pour avoir une méthode utilisable « en temps réel » et utilisant peu de ressources. Défini dans la section 3.2, les transducteurs finis sont une forme d'automate fini qui reconnaissent un langage mais qui sont aussi capables de produire une sortie. Ils sont formellement définis comme des machines de Turing à deux rubans. A notre connaissance, (Gross, 1987) est le premier à introduire l'utilisation des transducteurs finis dans le traitement automatique de la langue naturelle. Les transducteurs finis peuvent être utilisés pour de l'analyse syntaxique (Briscoe & Carroll, 2002) mais aussi pour l'extraction d'entités (Gaio & Moncla, 2017). (Mihov & Maurel, 2000) ont introduit un algorithme permettant de construire un transducteur minimal à partir d'une liste triée de mots reconnus par le langage avec leur sortie. Cet algorithme est celui implémenté dans pour la construction des FST dans Apache Lucene, un moteur d'indexation de texte notamment utilisé par Apache SolR.

3 Système

Le système utilisé, présenté dans la figure 1, utilise un lexique permettant l'extraction des termes révélateurs de chaque classe (section 3.1). L'extraction est réalisée par un transducteur fini dont le principe général et la construction sont définis dans la section 3.2. Une phase de pré-traitement du texte est nécessaire avant l'extraction des termes. Elle permet de supprimer les mots vides et de normaliser les mots en récupérant leur racine. Un pré-traitement visant à transformer les mots en n-grammes a aussi été envisagé, mais a été jugé trop coûteux en ressources pour un gain de qualité trop faible.

Nous avons également mis en place une phase de post-traitement des annotations retournées par le transducteur afin de nettoyer les résultats obtenus. Elle se charge de gérer les négations (section 3.3) et de traiter l'ambiguïté engendrée par certains termes sur le genre (section 3.4).

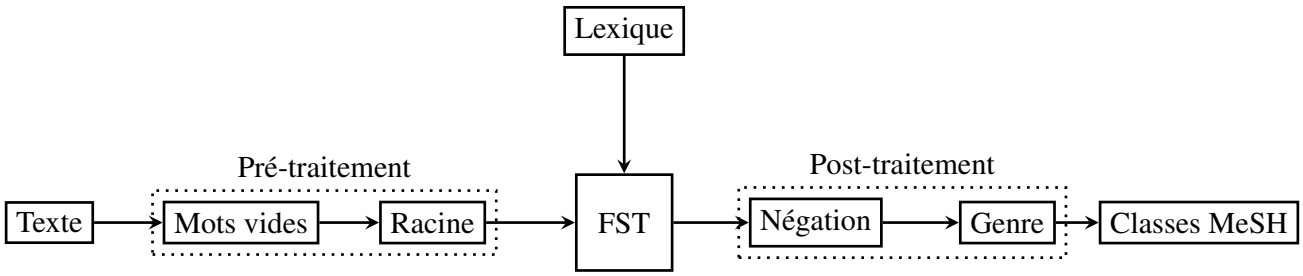

FIGURE 1 – Schéma du système d'identification du profil clinique du patient

3.1 Lexique

Définition Dans cet article nous définissons un lexique comme une collection de valeurs correspondant aux entrées du lexique pour lesquelles sont associées un ensemble de lexèmes la représentant. Dans notre cas les entrées du lexique sont les types de maladie représentés par 23 classes du chapitre C du MeSH (voir section 3.1.1).

Les lexiques sont construits à partir de thésaurus, de terminologies ou de bases de connaissances qui représentent des ressources riches qui s'enrichissent avec le temps (souvent semi-automatiquement à partir de corpus annotés) et qui sont très souvent surveillées par une autorité qui vérifie les informations et se charge du nettoyage des entrées. Le domaine médical ne fait pas exception et fait peut-être partie des plus représentés, notamment avec des institutions comme la NLM (U.S. National Library of Medicine) qui a regroupé, dans le méta-thésaurus UMLS, un grand nombre de ressources [4] pouvant être très utiles pour le traitement du langage naturel. Parmi les ressources de l'UMLS on peut notamment citer MeSH, MedDRA©, SNOMED et RxNORM. Notre méthode s'appuie sur les deux premiers décrits dans la suite de la section.

3.1.1 MeSH

Le MeSH (Medical Subject Headings) est un thésaurus du domaine biomédical, à l'origine en anglais, qui est géré par la NLM. Il permet entre autres d'indexer et d'interroger des bases de données comme

4. https://www.nlm.nih.gov/research/umls/sourcereleasedocs/index.html

MEDLINE/PubMed. Une traduction du MeSH en français a été faite par l'INSERM, elle est mise à jour chaque année. La version bilingue anglais-français peut être interrogée depuis une interface en ligne (http://mesh.inserm.fr/FrenchMesh/), et il est également possible de télécharger le thésaurus au format XML. Le MeSH est organisé en 16 catégories thématiques, mais seule la catégorie C, maladie, a été utilisée pour cette tâche. Elle contient 26 classes, qui correspondent aux types de maladie à identifier dans les cas cliniques. Dans la table 1 nous donnons l'identifiant et le nom de ces classes (trois classes ne sont pas mentionnées dans le tableau car nous n'avions pas à les utiliser). Chaque classe est structurée en une arborescence de descripteurs, eux-mêmes constitués de concepts auxquels sont associés des termes. Nous avons extrait pour chaque classe du chapitre C tous les termes associés à des concepts. Au total 40 052 termes ont été extraits. Il est à noter qu'un même terme peut être associé à plusieurs classes. Par exemple "diabète gestationnel" est associé aux classes C13, C18 et C19.

C01	Infections bactériennes et mycoses	C13	Maladies de l'appareil urogénital féminin et complications de la grossesse
C02	Maladies virales		
C03	Maladies parasitaires	C14	Maladies cardiovasculaires
C04	Tumeurs	C15	Hémopathies et maladies lymphatiques
C05	Maladies ostéomusculaires	C16	Malformations et maladies congénitales, héréditaires et néonatales
C06	Maladies de l'appareil digestif		
C07	Maladies du système stomatognathique	C17	Maladies de la peau et du tissu conjonctif
C08	Maladies de l'appareil respiratoire	C18	Maladies métaboliques et nutritionnelles
C09	Maladies oto-rhino-laryngologiques	C19	Maladies endocriniennes
C10	Maladies du système nerveux	C20	Maladies du système immunitaire
C11	Maladies de l'oeil	C23	États, signes et symptômes pathologiques
C12	Maladies urogénitales de l'homme	C25	Troubles dus à des produits chimiques
		C26	Plaies et blessures

TABLE 1 – Classes du chapitre C du MeSH

3.1.2 MedDRA©

Afin d'augmenter la couverture de notre lexique, nous avons cherché d'autres terminologies pour le français. Nous avons choisi d'utiliser MedDRA© puisque certaines de ses classes semblaient correspondre à celles du MeSH.

MedDRA© [5] (Dictionnaire Médical des Affaires Réglementaires) (Brown *et al.*, 1999) est un dictionnaire terminologique médical utilisé par les autorités réglementaires et l'industrie biopharmaceutique. MedDRA© est disponible en plusieurs langues, dont le français. Il contient aussi bien des termes référant à des symptômes, des examens ou encore des traitements, structurés en 5 niveaux. Le niveau le plus haut étant une classification par discipline médicale. Il existe 26 classes, par exemple *Affections vasculaires*, *Affections du rein et des voies urinaires*, *Affections du système immunitaire*.

Nous avons utilisé les termes de 13 classes de MedDRA©, par exemple *affections congénitales, familiales et génétiques*, *affections gastro-intestinales*, *affections de la peau et du tissu sous-cutané*.

5. 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).

Nous avons relié ces classes aux classes du MeSH, dans les exemples précédents respectivement *congenitales* (C16), *digestif* (C06), *peau* (C17). Le lexique contient ainsi 58 071 termes en plus.

3.1.3 Corpus d'entraînement

Nous avons aussi utilisé le corpus d'entraînement pour supprimer ou ajouter des termes dans le lexique. Les résultats de notre système sur le corpus d'entraînement apportent une indication sur les valeurs repérées et les classes associées à ces valeurs. Il est repéré :

— les faux positifs (101 associations terme/classe) : termes qui ont toujours amené l'identification d'une mauvaise classe
hépatite B / virales ; mastite / femme ; plaie opératoire / blessures ; syphilis / infections ; morsure / chimiques

— les faux négatifs (1110 associations) : termes associés à une classe que notre système ne repère pas
inflexion épidermoïde / peau ; myélome / hemopathies ; tabagisme / chimiques ; cachectique / nutritionnelles

La liste des termes à supprimer a été nettoyée manuellement afin d'ignorer les termes les plus précis comme *accidents cérébrovasculaires*. En effet, nous souhaitons récupérer dans cette liste les termes ambigus ou trop génériques, par exemple *ampoule* pour la classe *peau*. Au total 54 termes ont été supprimés du lexique et 199 termes ont été ajoutés.

3.2 Transducteur fini (FST)

Afin d'extraire les lexèmes d'un lexique en « temps réel », tout en minimisant les ressources utilisées, nous proposons l'utilisation de transducteurs finis.

Définition En théorie des langages, un transducteur fini $T = (\Sigma^{in}, \Sigma^{out}, Q, I, F, \delta)$ est un automate fini qui reconnaît un langage $L = \{w_1, \ldots, w_n\}$ sur un alphabet Σ^{in} où les transitions possèdent deux labels ($l^{in} \in \Sigma^{in}$ et $l^{out} \in \Sigma^{out}$). Le premier caractérise la transition et le second constitue la sortie de l'automate. Q est l'ensemble des états, I les états initiaux, F les états finaux, δ l'ensemble des transitions et ϵ est le mot vide. La sortie de l'automate (quand un mot w est reconnu, c.-à-d. $w \in L$) peut être une somme des labels de sortie, leur concaténation ou, comme ici, une unique valeur (la classe). Un transducteur n'est pas obligatoirement déterministe et peut donc, pour un même mot, retourner plusieurs valeurs de sortie.

Les transducteurs sont des structures plus optimisées en mémoire que d'autres structures comme les tables triées, mais au détriment d'un accès plus coûteux en ressources processeur. Ils sont par conséquent, très utiles pour traiter des langages de grande taille qui ne pourraient pas normalement tenir en mémoire tout en offrant un accès suffisamment rapide. Nous utilisons l'implémentation fournie dans Apache SolR/Lucene [6] par le projet OpenSextant [7]. Elle repose sur l'algorithme de (Mihov & Maurel, 2000) qui permet d'obtenir le transducteur minimal efficacement.

Les transducteurs gardent aussi l'avantage d'être facilement mis à jour. Il est possible d'ajouter ou de supprimer de nouveaux mots dans le langage sans avoir à reconstruire l'automate entièrement.

6. https://solr.apache.org
7. https://github.com/OpenSextant/SolrTextTagger

Construction Comme présenté dans la Section 3.1, nos lexiques sont définis comme une application surjective $\forall v_i \in V \; \exists X_i \subset X, \; Lex : v_i \rightarrow X_i$ où V est l'ensemble des valeurs (ici classes du MeSH) du lexique et X est l'ensemble des lexèmes présents dans le lexique. X_i est défini comme l'ensemble des exemples de la valeur v_i, c.-à-d. pour MeSH, l'ensemble des lexèmes qui représente une classe du MeSH.

Afin de construire notre transducteur pour le lexique $Lex_{MeSH} : V_{MeSH} \rightarrow X_{MeSH}$, nous définissons les alphabets $\Sigma^{in}_{MeSH} = \{t_i \mid t_i \in \textit{token}(x_j) \; \forall x_j \in X_{MeSH}\} \cup \{\epsilon\}$ où *token* est une fonction qui retourne l'ensemble des tokens t_i d'un lexème x_j et $\Sigma^{out}_{MeSH} = V_{MeSH} \cup \{\epsilon\}$. Notre langage L_{MeSH} est alors naturellement défini comme l'ensemble des lexèmes X_{MeSH} du lexique, c.-à-d. chaque lexème est un mot du langage.

La fonction *token* permet l'extraction des tokens utilisés pour la construction du transducteur. Cette fonction est aussi appliquée aux textes en entrée afin de les faire correspondre à l'alphabet Σ^{in}_{MeSH}. Elle a pour rôle :

- Le découpage des lexèmes (mot ou suite de mots) en tokens. Le découpage est réalisé sur les caractères d'espacement, les ponctuations, les traits d'union et les chiffres accolés à du texte (ex : 50mg devient {50, mg}) ;
- Le passage en minuscule de l'ensemble des tokens ;
- Le filtrage des tokens correspondant à des mots vides (basé sur une liste) ;
- La transformation de tous les tokens non ASCII par leur équivalent (suppression des accents) ;
- Le remplacement de chaque token par sa racine en utilisant l'algorithme Snowball (Porter, 2001).

Détection des valeurs Notre système transforme le texte en entrée en une liste de tokens avec l'aide de la fonction *token*. La liste de tokens est ensuite passée dans le transducteur afin d'extraire l'ensemble des valeurs du lexique. Si plusieurs classes sont trouvées pour un même mot de L_{MeSH}, les multiples classes sont gardées. Cependant, si deux mots différents se recoupent (ex : *aggravation transitoire des symptômes* et *symptômes cardiovasculaires*) seulement le plus grand est gardé (ici, *aggravation transitoire des symptômes*). Les tokens restants (*cardiovasculaires*) sont remis en jeux (au cas où ils pourraient former un autre mot de L_{MeSH}). Cette approche permet de sélectionner les plus grands lexèmes qui sont plus discriminant de par leur taille.

Exemple 1. Prenons comme exemple trois lexèmes du MeSH avec des classes arbitraires dans la table 2. A partir des exemples pour chaque classe, nous pouvons construire le transducteur Figure 2.

Classe	Lexème	Tokens
C1	Exacerbation transitoire des symptômes	{exacerb, transitoir, symptom}
C1	Aggravation transitoire des symptômes	{aggrav, transitoir, symptom}
C2	Aggravation passagère des symptômes	{aggrav, passager, symptom}

TABLE 2 – Liste de lexèmes avec leur classe associée et la liste des tokens obtenus avec la fonction *token*

3.3 Détection de la négation et de l'incertitude

Dans les cas cliniques, il est possible que l'auteur notifie l'absence ou l'incertitude de certains symptômes. Bien que utile pour le corps médical, ces informations ne font pas partie du profil du

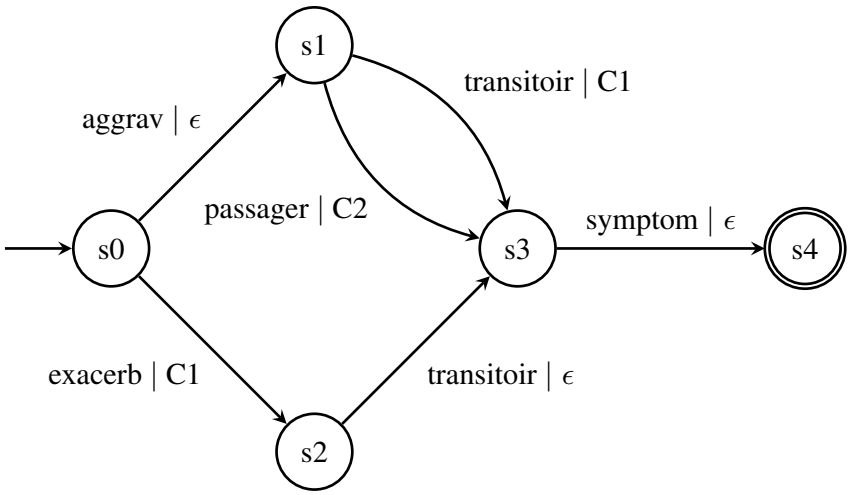

FIGURE 2 – Exemple d'un transducteur fini construit à partir de la table 2

patient. Dans la phrase « Le patient n'avait eu ni **traumatisme**, ni **piqûre d'insecte** » le MeSH nous permet de reconnaître les termes *traumatisme* et *piqûre d'insecte*. Cependant, ils doivent être ignorés pour la classification.

Pour ce faire, nous avons construit un nouveau lexique contenant l'ensemble des marqueurs de négation et d'incertitude extrait automatiquement à partir du corpus CAS (Grabar *et al.*, 2019). Une mesure de distance est ensuite appliquée entre chaque lexème extrait et les marqueurs de négation. Si la distance est inférieure à un certain seuil, l'exemple de la classe est rejeté. Nous utilisons ici la distance en offset de caractères (distance entre la fin du marqueur et le début du lexème). Nous avons testé des seuils de 2, 5, 10 et 20, et nous avons remarqué très peu de différences dans les résultats entre les seuils 2, 5 et 10 (entre 0,736 et 0,737 de F-mesure sur le corpus d'entraînement). Nous avons donc choisi de fixer le seuil à 10, pour améliorer la précision. (Garcelon *et al.*, 2014) propose une méthode plus évoluée pour réduire la porté de la négation, mais dans notre cas, ce n'est pas nécessaire car on recherche uniquement à classer des documents, et si un cas clinique possède une classe C, il y a une grande probabilité que d'autres exemples révélateurs de cette classe se trouvent ailleurs dans le document. Au contraire, un terme hypothétique a plus de chance de n'apparaître que peu de fois et donc d'être supprimé par cette approche. Toutefois, une simple mesure d'occurrences n'est pas suffisante, car certaines classes peuvent n'être reconnues qu'à l'aide d'un seul exemple isolé dans le document.

3.4 Recherche du genre du patient

Dans les cas cliniques du corpus d'entraînement, la classe du genre (homme, femme) est très présente. Cette classe est repérée pour les maladies liées à l'appareil urogénital et aux complications de grossesse. Notre système détecte l'information dans 98 cas sur 167 au total. Pour 62 d'entre eux, un double genre est affecté. Au vu de cette proportion, nous proposons un post-traitement déterminant le genre du patient pour chaque cas.

A l'aide du corpus d'entraînement nous établissons un lexique lié au genre comme « mlle », « madame », « âgée », « hospitalisée », « patiente », « monsieur », « homme », « masculin », etc. Ces mots sont repérés dans le premier paragraphe, le genre étant souvent indiqué en début de cas. Si nous n'avons pas repéré le genre à cette étape alors nous ajoutons au lexique une liste de termes plus

spécifiques à l'anatomie (« testicule », « utérus », « ovaire », etc.) et nous étendons la recherche aux autres paragraphes.

Dans le corpus d'entraînement, pour deux cas nous n'arrivons pas à déterminer le genre, pour l'un il s'agit de plusieurs personnes pour l'autre, aucun indice notable ne permet de le savoir. S'il s'agit de cas cliniques pour lesquels l'information du genre est nécessaire nous gardons l'annotation du transducteur.

Au niveau du corpus test, 57 cas sont liés à ces maladies selon notre système.Pour deux cas notre post-traitement ne nous permet pas de choisir entre la classe homme et la classe femme. L'annotation manuelle effectuée sur les données test sont utilisées pour lever l'ambiguïté. La F-mesure passe de 0.784 à 0.785, il semble donc plus intéressant de se passer de l'annotation manuelle. Dans une prochaine version du système, nous garderons donc les deux genres possibles lorsque nous ne sommes pas en mesure de désambiguïser.

3.5 Suppression de termes non spécifiques au domaine médical

Dans le MeSH nous avons remarqué qu'il y avait certains termes très génériques (e.g. « maladie ») et/ou ambigus (e.g. « pris »). Pour supprimer ces termes qui risquent de nous apporter beaucoup de faux positifs, nous nous sommes basés sur la fréquence des mots dans un corpus non spécifique au domaine médical. Nous avons choisi d'utiliser le corpus Wikipédia FR 2008 [8] pour lequel un fichier avec les fréquences de chaque token est disponible. Grâce à ce corpus, nous avons supprimé les lexèmes du lexique composés d'un seul mot qui sont très fréquents et donc possiblement non spécifiques au domaine médical ou ambigus. Nous avons évalué différents seuils sur le corpus d'entraînement, nous obtenons la meilleure précision avec un seuil à 100, le meilleur rappel avec un seuil à 5000 et la meilleure F-mesure avec un seuil à 1000. Nous avons donc choisi de supprimer les lexèmes présents plus de 1000 fois dans le corpus Wikipedia.

Cette étape constitue un post-traitement, mais elle pourrait également être utilisée lors de la préparation du lexique pour le nettoyer. Cela permettrait à la fois d'avoir un lexique plus petit et d'éviter une étape de post-traitement supplémentaire. Nous intégrerons cette modification dans une nouvelle version du système.

Dans le corpus d'entraînement, nous avons ainsi supprimé 324 termes détectés, qui représentent 33 termes uniques du lexique (e.g. « syndrome », « fièvre », « malade »).

4 Résultats

Dans cette section nous présentons les résultats officiels obtenus à DEFT ainsi que les résultats d'expérimentations supplémentaires sur le corpus de test permettant de mettre en évidence l'apport de chaque post-traitement et l'impact des modifications sur le lexique.

Le corpus de test se compose de 108 cas cliniques rédigés en français, représenté par 41 478 mots. Le corpus comporte des annotations manuelles plus conséquentes que le corpus d'entraînement : genre, âge, poids, taille, température etc.

8. http://redac.univ-tlse2.fr/corpus/wikipedia.html

Nous avons soumis trois runs à DEFT, qui différent selon les ressources utilisées.

— Run1 : MeSH
— Run2 : MeSH + annotation du corpus d'entraînement
— Run3 : MeSH + MedDRA©

Dans la partie gauche de la table 3, nous présentons les résultats officiels fournis par les organisateurs de DEFT. Dans la partie droite nous indiquons le nombre de cas pour lesquels la précision est de 1, c.-à-d. que toutes les classes de ce fichier ont été trouvées, et ceux pour lesquels la précision est de 0, c.-à-d. qu'aucune classe n'a été trouvée pour ce fichier. Nous obtenons les meilleurs résultats avec le run 2, c.-à-d. en nettoyant et complétant le lexique avec les annotations provenant du corpus d'entraînement. Cette configuration nous permet également d'avoir la meilleure précision (0,885). Pour le run 3, nous avons utilisé MedDRA©, ce qui a permis d'augmenter le lexique de façon considérable (+58 071 termes). Cette augmentation permet d'augmenter un peu le rappel par rapport aux run 1 et 2, mais fait chuter la précision à 0,679. Le run1 pour lequel nous n'avons utilisé que le MeSH, nous permet d'obtenir des bons résultats, proche du run 2 et supérieurs à la médiane de la tâche.

Les classes pour lesquelles nous avons obtenues les moins bons résultats sont la classe *blessures* (F-mesure entre 0.49 et 0.55), *chimiques* (F-mesure entre 0.36 et 0.54) et *virales* (1 classe sur 4 a été identifiée dans le corpus de test).

	Évaluation globale			Nombre de cas	
	Précision	Rappel	F1	P=1	P=0
Run1	0,873	0,713	0,785	25	6
Run2	**0,888**	0,750	**0,814**	29	4
Run3	0,686	**0,769**	0,725	42	5

TABLE 3 – Résultats officiels obtenus sur la tâche 1.

Dans la table 4 nous présentons des résultats d'expérimentations supplémentaires sur le corpus de test permettant de mettre en évidence l'impact des post-traitements et de la modification du lexique sur les performances du système.

Configuration	Précision	Rappel	F1
MeSH	0,725	0,739	0,732
MeSH + négation	0,739	0,738	0,738
MeSH + genre	0,739	0,798	0,768
MeSH + négation + genre	0,816	0,738	0,775
MeSH + négation + genre + fréquence	0,873	0,713	0,785
MeSH + négation + genre + fréquence + annotation train	0,888	0,750	0,814

TABLE 4 – Résultats d'expérimentations supplémentaires sur l'impact des post-traitements.

Nous observons que le post-traitement lié à l'identification du genre du patient permet une amélioration importante des performances (+ 0,036 pour la F-mesure). Un gain important est également observé avec l'utilisation des annotations du train pour nettoyer et augmenter le lexique.

5 Conclusion

Dans cet article nous avons proposé une méthode rapide, simple, et efficace pour la classification de documents médicaux. Notre approche montre que les lexiques constituent une ressource riche pour cette tâche, facilement mise à jour, et permettent d'obtenir une bonne qualité de classification.

Une analyse d'erreur rapide nous a permis de mettre en évidence que la plupart des erreurs étaient dues à l'absence de certains termes dans le lexique (e.g. *kystes biliaires hépatiques*), ou au fait que certains termes présents ne sont pas reliés à la classe identifiée manuellement (e.g. *fistule* et *fistule cutanée* sont contenus dans la classe *etatsosy* et *peau* pour le deuxième, mais pas dans la classe *tumeur*, contrairement à ce qui a été annoté manuellement). Nous observons aussi quelques cas qui nécessitent un raisonnement, par exemple *violente chute* qui induit des blessures.

Comme discuté dans les parties précédentes, certaines améliorations du système peuvent être envisagées. Notamment pour le traitement de la négation, où il est envisageable de mettre en place une méthode sémantique cherchant à mieux identifier la porté des marqueurs de négation. Il peut être aussi intéressant de tenter de détecter des négations plus complexes comme dans la phrase « Elle n'a présenté des nausées que durant la nuit et aucun vomissement ».

Pour le traitement du genre, il n'est pas nécessaire d'utiliser les annotations manuelles, si nous avons trop de non détection nous pouvons envisager une augmentation du lexique avec des listes de prénoms ou des participes passés (« s'est présenté »).

La suppression de certains lexèmes dans le lexique basée sur leur fréquence dans le corpus Wikipedia FR 2008 est réalisée en post-traitement. Nous pensons qu'il serait préférable de se servir de ces fréquences pour nettoyer le lexique en amont. L'avantage est d'avoir un lexique plus petit et de ne pas chercher des variants inutilement. Nous avons testé cette nouvelle configuration et nous obtenons pour le run 1 une F-mesure de 0,788 (au lieu de 0,785).

A l'heure des réseaux de neurones énergivores, coûteux en maintenance et en entraînement, notre méthode s'inscrit dans une démarche éco-responsable. Selon (Strubell *et al.*, 2019), entraîner un modèle de réseaux de neurones profond « à l'état de l'art » pour faire de la traduction correspondrait à l'impact de la durée de vie de 5 voitures. Les durées d'entraînement pour ces modèles peuvent aussi aller de quelques jours à plusieurs semaines. Les résultats obtenus montrent que les approches symboliques restent efficaces pour ce genre de tâche. L'utilisation de structures de données adaptées permet de minimiser les ressources nécessaires, ce qui diminue le coût mais aussi le temps d'exécution. Avec notre approche, nous avons pu atteindre un temps $\approx$ 1min pour la classification de 108 documents. Ce système peut aussi être facilement déployé sur des systèmes embarqués.

Références

BRIN S. & PAGE L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, **30**(1-7), 107–117.

BRISCOE T. & CARROLL J. A. (2002). Robust accurate statistical annotation of general text. In *Proceedings of the Third International Conference on Language Resources and Evaluation, LREC 2002, May 29-31, 2002, Las Palmas, Canary Islands, Spain* : European Language Resources Association.

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

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 R. CARDON, N. GRABAR, C. GROUIN & T. HAMON, Éds., *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*, p. 1–13, Nancy, France : ATALA. HAL : hal-02784737.

ELHADAD N., PRADHAN S., GORMAN S. L., MANANDHAR S., CHAPMAN W. W. & SAVOVA G. K. (2015). Semeval-2015 task 14 : Analysis of clinical text. In D. M. CER, D. JURGENS, P. NAKOV & T. ZESCH, Éds., *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2015, Denver, Colorado, USA, June 4-5, 2015*, p. 303–310 : The Association for Computer Linguistics. DOI : 10.18653/v1/s15-2051.

GAIO M. & MONCLA L. (2017). Extended Named Entity Recognition Using Finite-State Transducers : An Application To Place Names. In *The Ninth International Conference on Advanced Geographic Information Systems, Applications, and Services (GEOProcessing 2017)*, Nice, France. HAL : hal-01492994.

GARCELON N., SALOMON R. & BURGUN A. (2014). Enrichissement sémantique associé à la détection de la négation et des antécédents familiaux dans un entrepôt de données hospitalier. In *JFIM*, p. 83–93.

GRABAR N., GROUIN C., HAMON T. & CLAVEAU V. (2019). Corpus annoté de cas cliniques en français. In *TALN 2019 - 26e Conference on Traitement Automatique des Langues Naturelles*, p. 1–14, Toulouse, France. HAL : hal-02391878.

GROSS M. (1987). The use of finite automata in the lexical representation of natural language. In *LITP Spring School on Theoretical Computer Science*, p. 34–50 : Springer.

GROUIN C., GRABAR N. & ILLOUZ G. (2021). Classification de cas cliniques et évaluation automatique de réponses d'étudiants : présentation de la campagne deft 2021. In *Actes de DEFT*, Lille.

JONES K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.

MIHALCEA R. & TARAU P. (2004). Textrank : Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, p. 404–411.

MIHOV S. & MAUREL D. (2000). Direct construction of minimal acyclic subsequential transducers. In *Implementation and Application of Automata, 5th International Conference, CIAA 2000, London, Ontario, Canada, July 24-25, 2000, Revised Papers*, volume 2088 de *Lecture Notes in Computer Science*, p. 217–229 : Springer.

MINARD A., ROQUES A., HIOT N., ALVES M. H. F. & SAVARY A. (2020). Doing@deft : cascade de CRF pour l'annotation d'entités cliniques imbriquées (doing@deft : cascade of CRF for the annotation of nested clinical entities). In R. CARDON, N. GRABAR, C. GROUIN & T. HAMON, Éds., *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, Nancy, France, June 8-19, 2020*, p. 66–78 : ATALA et AFCP.

PATHAK P., PATEL P., PANCHAL V., SONI S., DANI K., PATEL A. & CHOUDHARY N. (2015). ezdi : A supervised NLP system for clinical narrative analysis. In D. M. CER, D. JURGENS, P. NAKOV & T. ZESCH, Éds., *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2015, Denver, Colorado, USA, June 4-5, 2015*, p. 412–416 : The Association for Computer Linguistics. DOI : 10.18653/v1/s15-2071.

PORTER M. F. (2001). Snowball : A language for stemming algorithms. Published online. Accessed 11.03.2008, 15.00h.

PRADHAN S., ELHADAD N., SOUTH B. R., MARTÍNEZ D., CHRISTENSEN L. M., VOGEL A., SUOMINEN H., CHAPMAN W. W. & SAVOVA G. K. (2013). Task 1 : Share/clef ehealth evaluation lab 2013. In P. FORNER, R. NAVIGLI, D. TUFIS & N. FERRO, Éds., *Working Notes for CLEF 2013 Conference , Valencia, Spain, September 23-26, 2013*, volume 1179 de *CEUR Workshop Proceedings* : CEUR-WS.org.

STRUBELL E., GANESH A. & MCCALLUM A. (2019). Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv :1906.02243*.

WENG W.-H., WAGHOLIKAR K. B., MCCRAY A. T., SZOLOVITS P. & CHUEH H. C. (2017). Medical subdomain classification of clinical notes using a machine learning-based natural language processing approach. *BMC medical informatics and decision making*, **17**(1), 1–13.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Identification de profil clinique du patient: Une approche de classification de séquences utilisant des modèles de langage français contextualisés

Aidan Mannion[1,2] Thierry Chevalier[3] Didier Schwab[1] Lorraine Goeuriot[1]
(1) Univ. Grenoble Alpes, CNRS, LIG, 38000 Grenoble, France
(2) EPOS, 2-4 Boulevard Des Îles, 92130 Issy Les Moulineaux, France
(3) UFR de Médecine Univ. Grenoble Alpes, Domaine de la Merci, 38700 La Tronche, France

RÉSUMÉ

Cet article présente un résumé de notre soumission pour Tâche 1 de DEFT 2021. Cette tâche consiste à identifier le profil clinique d'un patient à partir d'une description textuelle de son cas clinique en identifiant les types de pathologie mentionnés dans le texte. Ce travail étudie des approches de classification de texte utilisant des plongements de mots contextualisés en français. À partir d'une base de référence d'un modèle constitué pour la compréhension générale de la langue française, nous utilisons des modèles pré-entraînés avec *masked language modelling* et affinés à la tâche d'identification, en utilisant un corpus externe de textes cliniques fourni par SOS Médecins, pour développer des ensembles de classifieurs binaires associant les textes cliniques à des catégories de pathologies.

ABSTRACT

Identification of patient clinical profiles : A sequence classification approach using contextualised French language models

This article summarises our submission to Task 1 of the text mining challenge DEFT 2021. This task involved the identification of the clinical profile of a patient from a textual description of their clinical case by identifying all the types of pathology mentioned in the text. This work investigates the utility of text classification approaches using contextualised French-language vector embeddings. Beginning from a baseline of a model trained for general French-language understanding, we employ both masked-language pre-training and fine-tuning on the DEFT task, using an external corpus of clinical text provided by SOS Médecins, to develop ensembles of binary classifiers to associate pathology types with a given segment of clinical text.

MOTS-CLÉS : TALN biomédicale, Classification des séquences, FlauBERT, plongements de mots contextualisé.

KEYWORDS: Biomedical NLP, Sequence classification, FlauBERT, contextualised word embeddings.

Atelier DÉfi Fouille de Textes (DEFT), pages 54–62.

1 Introduction

La tâche d'identification de profils cliniques à partir de données biomédicales textuelles est d'un grand intérêt pour les institutions, les entreprises et les praticiens du domaine médical. Le problème peut se montrer assez difficile, principalement à cause de la forme non-structurée des données textuelles ainsi que la complexité et la spécificité du domaine. Des développements récents dans le domaine de traitement du langage naturel, et plus particulièrement les plongements de mots contextualisés basés sur l'entraînement de réseaux de neurones avec l'architecture *transformer* (notamment (Devlin *et al.*, 2018)), montrent le potentiel de modélisation des dépendances complexes et à longue portée. Ces nouveaux modèles donnent à la communauté TALN biomédicale des pistes d'expérimentation pour l'amélioration d'extraction d'informations complexes à partir de dossiers de santé textuels.

Certaines études ont montré l'utilité de ces méthodes pour développer des outils en TALN biomédical en anglais (Huang *et al.*, 2019; Alsentzer *et al.*, 2019; Lee *et al.*, 2020) et leur application à diverses tâches du domaine (Yoon *et al.*, 2019; Peng *et al.*, 2019; Blinov *et al.*, 2020). À notre connaissance, il existe peu de travaux sur les applications des modèles de langue neuronaux contextualisés sur des tâches biomédicales sur le français comme celles de DEFT 2021.

Le modèle neuronal FlauBERT (Le *et al.*, 2020), un modèle de type *transformer* (voir la section 2.1) entraîné de manière auto-supervisée sur un corpus de textes de langue générale en français, est utilisé dans ce travail. FlauBERT est un modèle transformer bidirectionnel avec la même architecture que BERT (Devlin *et al.*, 2018) qui a été entraîné sur un corpus français hétérogène extrait de divers sources.

Plus spécifiquement, la tâche 1 du Défi Fouilles de Texte 2021 (Grouin *et al.*, 2021) vise à identifier le profil clinique d'un patient par le type de maladie de toutes les pathologies présente dans le texte associé avec un cas clinique. Nous formulons cette tâche comme un problème de classification des vecteurs représentant des séquences de texte. Parce qu'il peut y avoir plusieurs catégories associées à un document, il n'est pas possible d'utiliser un classificateur multi-label standard (dans plusieurs cas, plusieurs annotations sont mises en association avec le même mot dans le document source). Nous entraînons un classificateur binaire pour chacune des catégories de sortie. De cette manière, les modèles apprennent indépendamment les corrélations entre les plongements vectoriels et les variables cibles, à partir d'une base de réference qui utilise un modèle entraîné sur une tâche non-supervisée, détaillée dans la section 2.3. Les modèles utilisés pour les expériences sont pre-entraînés sur le corpus de SOS Médecins détaillés en section 2.2, et adapté pour la tâche d'identification de profils cliniques de deux manières ; en s'appuyant sur la version étiquetée du corpus de SOS (aussi détaillé en section 2.2), et finalement sur le corpus d'entraînement DEFT fourni pour cette tâche, sur lequel les résultats d'entraînement sont montrés dans la section 3.1, ainsi que les résultats sur le corpus d'évaluation.

2 Entraînement des modèles de langage

Divers types de vecteurs ont été expérimentés pour représenter du texte clinique (Khattak *et al.*, 2019), mais étant donné l'énorme complexité des relations possibles entre les entités biomédicales, et leur importance potentielle pour l'identification des profils cliniques, il est pertinent de penser que les modèles contextualisés peuvent apporter des améliorations aux tâches de classification automatique.

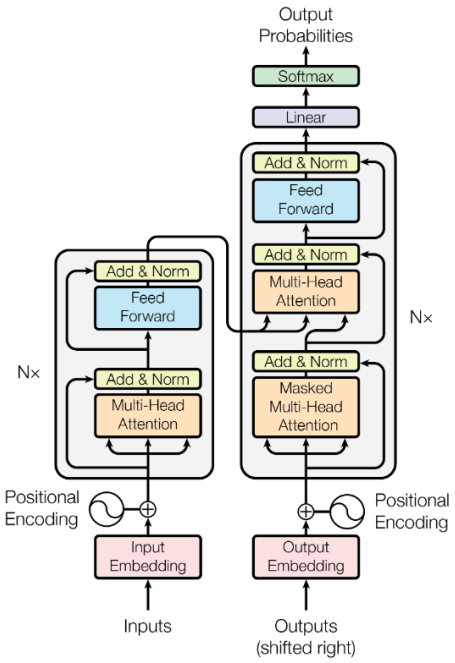

FIGURE 1 – L'architecture d'un transformer : diagramme issu de (Vaswani *et al.*, 2017).

2.1 Les réseaux de neurones *transformer*

Les réseaux transformer (Vaswani *et al.*, 2017) impliquent un modèle encodeur-décodeur neuronal mettant en œuvre une technique appelée auto-attention. Le principal avantage de ce méchanisme est de modèliser les dépendances à long terme dans le texte. Il utilise une opération appelée attention de produit scalaire qui crée une *matrice d'attention* pour chaque séquence de jetons d'entrée dans laquelle chaque composant a une probabilité égale afin d'être associés sémantiquement avec tout autre composant, quel que soit le nombre de composants entre eux. Un diagramme classique de l'architecture est montré en Figure 1.

2.2 Corpus SOS Médecins

Pour adapter les plongements FlauBERT (appris sur du vocabulaire général) au domaine clinique, nous utilisons un corpus de textes cliniques qui est composé de commentaires écrits par des médecins en consultation avec des patients dans le cadre des appels au service SOS Médecins.

L'objectif de l'entraînement supplémentaire avec ce corpus est d'améliorer les performances sur les séquences de texte du corpus DEFT. Étant donné que l'utilisation des réseaux transformer présente plus d'avantage dans le traitement des phrases longues avec plus de complexité en termes de dépendances linguistiques, nous filtrons les documents du corpus pour ne garder que ceux ayant un nombre de mots supérieur à un seuil (28). Cela nous donne un corpus d'apprentissage de 324 753 documents cliniques, avec un longueur (no. mots) moyenne de 41.3 et un longueur maximale de 390.

Nous disposons dans ce corpus de codes de diagnostics ajoutés aux données de la consultation manuellement par le médecin à l'issue de la visite au patient. Le format de ces codes est spécifique à SOS Médecins, mais nous avons bénéficié de l'expertise du deuxième auteur, également médecin, pour les associer aux catégories MeSH correspondantes. Cette conversion impliquait l'effacement de

certaines catégories des codes SOS, parce que ces codes sont plus spécifiquement des classifications de "résultats de la consultation", donc pas forcément toujours un diagnostic de pathologie qui peut être associé avec une chapitre de MeSH pertinente pour la tâche DEFT.

Dans ce travail, nous entraînons et comparons des classificateurs de profil MeSH à partir de trois différentes variantes de l'entraînement supplémentaire avec le corpus SOS Médecins ; un modèle "pre-entraîné" (section 2.3) un modèle adapté directement pour une variante de la tâche DEFT (section 2.4) et un avec les deux ensembles (le pre-entraînement en premier). Les étapes d'optimisation des réseaux de neurones modélisation de mots masqués (section 2.3) ainsi que l'adaptation aux tâches de classification (section 2.4) était fait en utilisant l'algorithme Adam (Kingma & Ba, 2015).

2.3 Pré-entraînement : modélisation des mots masqués

La tâche de pré-entraînement consiste à cacher aléatoirement des jetons dans le corpus d'entrée avec des jetons spéciaux appelés "masques" ; l'objectif de l'entraînement du réseau est alors de prédire le jeton caché.

Nous décrivons ici les expériences faites avec le modèle $\text{FlauBERT}_{\text{BASE}}$, qui consiste en 12 couches, 12 têtes d'attention et qui a une dimension maximale de plongements de 768, pour un total de 138M de paramètres.

Pour suivre la performance du modèle lors de l'entraînement, nous utilisons la perplexité, une métrique largement utilisée pour l'évaluation des modèles prédictifs non-supervisés ; c'est une technique avec ses origines provenant de la théorie d'information qui sert à comparer deux distributions de probabilité en prenant un moyen géométrique pondéré des inverses des probabilités sorties par un modèle pour un certain ensemble de données. Le corpus d'entraînement est divisé en deux parties, "train" et "eval", et à la fin de chaque époque d'entraînement, où le modèle s'entraîne sur le sous-ensemble *train*, la perplexité du modèle est évaluée sur le sous-ensemble *eval*. Le sous-ensemble *train* consiste à 80% du corpus d'entrée, choisi aléatoirement.

Pour notre pré-entraînement avec le corpus SOS Médecins, nous utilisons un seuil de convergence de 0.05 pour la perplexité, c'est-à-dire que l'entraînement s'est arrêté après avoir atteint une époque pour laquelle la perplexité a diminué de moins de 0.05. Le modèle utilisé dans les expériences (nommé "Base + MLM" dans la section 3.1) a complété 8 époques avant d'atteindre ce seuil. Bien qu'il s'agisse d'un nombre d'époques beaucoup plus faible que ce qui est généralement accepté comme raisonnable pour ce type d'entraînement, les contraintes de temps et de ressources ont nécessité l'utilisation ce seuil. La diminution de perplexité à travers des époques d'entraînement est montrée en Figure 2.

Les hyperparamètres utilisés pour l'entraînement étaient les suivants ;

- Une *probabilité de masquage*, c'est-à-dire la proportion des tokens d'entrée qui ont été cachés avec le token spécial `[MASK]`, de 0.15, comme c'est la norme spécifiée par les développeurs de BERT,
- Un taux d'apprentissage de 5×10^{-5},
- Une longueur de séquence maximale de 256 tokens.

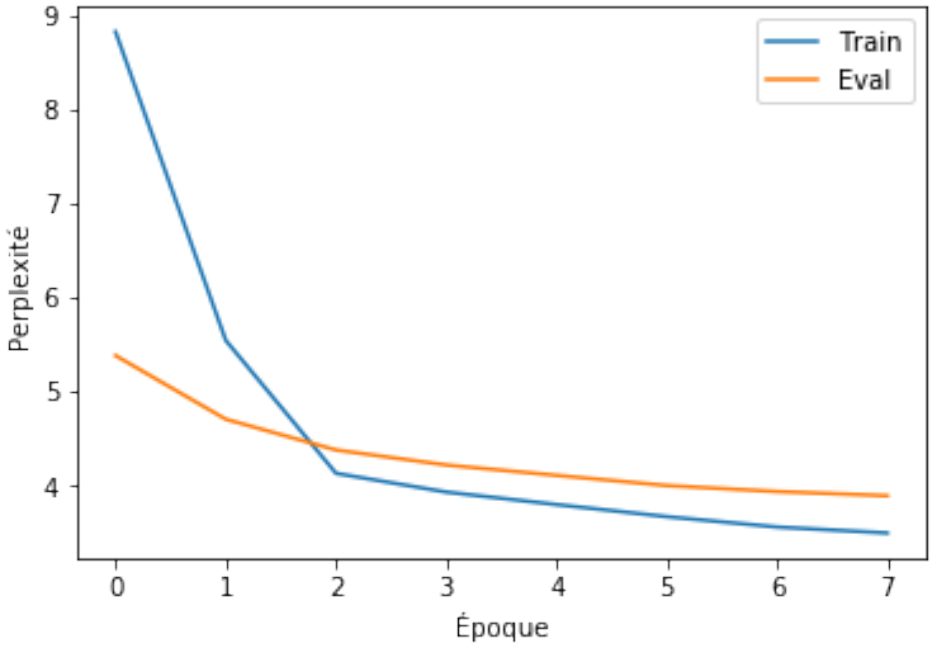

FIGURE 2 – La perplexité de FlauBERT$_{\text{BASE}}$ en s'entraînant sur les sous-ensembles *train* (80% du corpus) et *eval* (20%) du corpus SOS Médecins.

2.4 *Fine-tuning* : adaptation des modèles de langage à la tâche de classification

Pour l'étape de classification, c'est-à-dire l'adaptation des plongements entraînés sur la tâche de pré-entraînement à la tâche d'identification de profils cliniques qui nous intéresse, nous utilisons la méthode de classification intégrée aux plongements de style BERT : un jeton d'agrégation, étiqueté `[CLS]`, le premier jeton d'une séquence, qui est utilisé pour propager la perte à travers le modèle. Comme mentionné précédemment, pour pouvoir associer plusieurs catégories de maladie avec un document, il est nécessaire d'avoir un classifieur par catégorie. Au cas où un document dépasse la limite de longueur de séquences pour le modèle, il est divisé en plusieurs séquences.Dans la sortie finale pour évaluation, les documents sont associés aux chapitres MeSH correspondant à toutes les prédictions positives des classifieurs sur le document (ou au moins un de ses sous-séquences).

Pour construire un corpus d'entraînement pour des classificateurs avec notre corpus SOS Médecins, nous utilisons des codes d'identification diagnostic décrits dans la section 2.2.

En plus de la correspondance approximative entre les catégories de ce corpus et celles de la tâche DEFT, l'apport de ce corpus à l'apprentissage est limité par le déséquilibre entre les deux corpus, pas seulement en terme de taille mais en terme de prévalence des variables cibles. Dans la figure 3 on voit que les catégories de maladies les plus communes dans le corpus DEFT n'apparaissent pas dans le corpus de SOS. Cela veut dire que les distributions de probabilité modélisées par les classifieurs pourraient être assez différentes et un classifieur qui montre des bonnes performances sur la tâche de classification avec le corpus de SOS Médecins pourra bien avoir estimé une frontière de décision qui ne s'appliquera pas très bien à la tâche DEFT.

Après avoir supprimé les documents qui n'ont pas d'étiquette pertinente pour la tâche de classification, comme détaillé dans la section 2.2, il reste un corpus de taille 258 378 avec un longueur moyen de

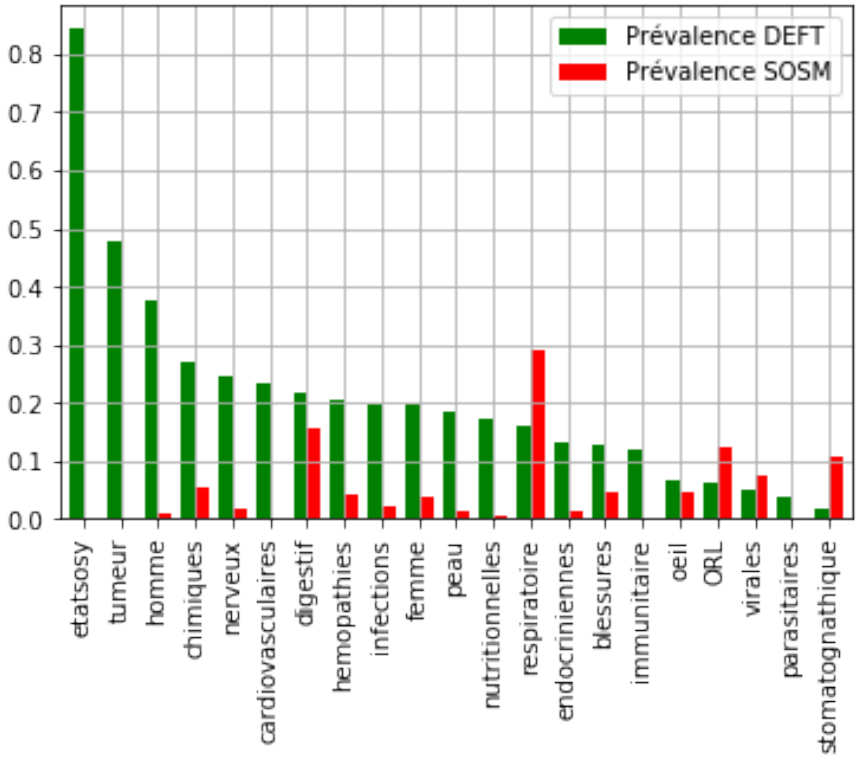

FIGURE 3 – Comparaison de la prévalence des variables cibles dans le corpus d'entraînement DEFT et le corpus de SOS Médecins étiqueté avec les chapitres MeSH pertinents.

41.77 et un longueur maximal de 390.

Les hyperparamètres utilisés pour l'entraînement des classificeurs sur le corpus SOS étaient les suivantes ;

— 4 époques,
— taux d'apprentissage : 2×10^{-5},
— longueur de séquence maximal : 256 tokens

Pour l'entraînement final avec les 167 documents donnés pour la tâche, les hyperparamètres étaient les mêmes sauf que le nombre d'époques était élevé jusqu'à 8.

3 Expériences

3.1 Expériences FlauBERT$_{\text{BASE}}$

Nous présentons l'évaluation de quatre différentes variantes des classifieurs sur la tâche d'identification de profil clinique ;

1. Le modèle FlauBERT$_{\text{BASE}}$ adapté directement à la tâche
2. Un modèle entraîné de manière non-supervisée sur le corpus SOS Médecins non-étiqueté comme décrit en section 2.3 ("Base + MLM" dans la figure 4)
3. FlauBERT$_{\text{BASE}}$ adapté à la classification des documents du corpus SOS Médecins étiqueté ("Base + clf").

	Précision		Rappel		F1	
	Train	**Eval**	**Train**	**Eval**	**Train**	**Eval**
Base	0.422	0.518	0.370	0.402	0.394	0.453
Base + MLM	0.475	0.528	0.457	0.435	0.466	0.477
Base + clf	0.452	0.542	0.477	0.451	0.464	0.492
Base + MLM + clf	0.487	**0.550**	0.520	**0.463**	0.503	**0.503**

FIGURE 4 – Les résultats d'entraînement et d'évaluation des différentes variantes de FlauBERT$_{\text{BASE}}$ sur la tâche. Les scores "Train" correspondent à la performance sur les données d'entraînement de 167 documents, et les scores "Eval" à la performance sur les 108 documents d'évaluation.

	Précision		Rappel		F1	
	Train	**Eval**	**Train**	**Eval**	**Train**	**Eval**
Base	0.366	0.390	0.353	0.444	0.359	0.416
Base + MLM	0.368	0.423	0.360	0.496	0.364	0.457
Base + MLM + clf	0.377	0.398	0.368	0.439	0.372	0.417

FIGURE 5 – Les résultats d'entraînement et d'évaluation des différentes variantes de FlauBERT$_{\text{SMALL}}$ sur la tâche.

4. La combinaison des deux approches précédentes ("Base + MLM + clf").

Les résultats sont présentés dans la figure 4. Comme prévu, il semble que l'addition de l'entraînement supplémentaire sur le corpus SOS Médecins augmente les mesures de performance. Il est intéressant de noter que les performances sur les données de test sont souvent meilleures que celles obtenues sur les données d'entraînement (principalement au niveau de la précision), ce qui suggère que le pouvoir prédictif des classifieurs vient des connaissances apprises pendant l'entraînement sur les corpus externes plutôt que pendant l'adaptation avec le corpus DEFT lui-même.

3.2 Soumission

À cause de certaines contraintes temporelles et de ressources de calculs, nous avons dû soumettre au défi des résultats des modèles entraînés avec FlauBERT$_{\text{SMALL}}$, une version de FlauBERT pas entièrement entraînés sur son corpus de base, donc les résultats pour la compétition n'était pas les meilleurs obtenus avec les modèles abordés dans cet article. Les résultats de ces expériences sont présentés en figure 5.

4 Discussion & Conclusion

Alors que les résultats n'étaient pas satisfaisants, nous suggérons que l'introduction d'un apprentissage supplémentaire sur l'ensemble de données SOS montre des améliorations encourageantes dans les performances des classifieurs qui pourraient être portées à un niveau acceptable avec plus de données d'entraînement de haute qualité, en particulier compte tenu du fait que nous n'avons pas introduit des corrections pour le déséquilibre dans les étiquettes d'entraînement.

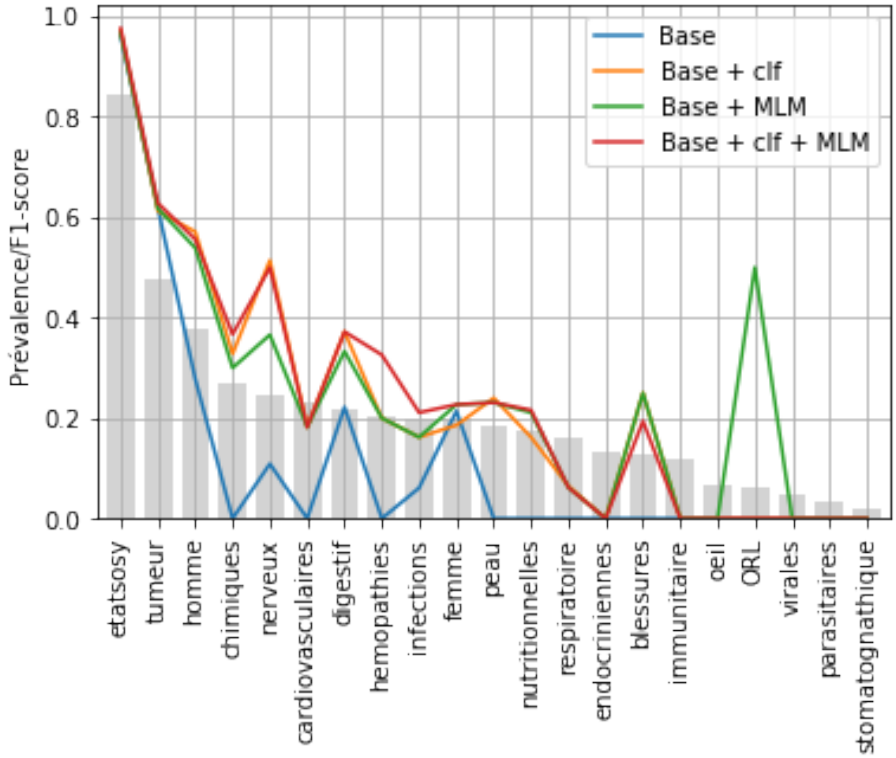

FIGURE 6 – Comparaison entre les prévalences des variables cibles dans le corpus d'entraînement DEFT et la performance des classifieurs entraînés sur le corpus d'évaluation (les barres grises représentent la prévalence de chaque catégorie de maladie).

4.1 Limitations du modélisation

Il y a plusieurs limites à l'efficacité des expériences réalisées dans ce travail qui pourraient être améliorées et constituer la base des futures expériences. Premièrement, en raison de contraintes de calcul et de temps, nous n'avons effectué aucun réglage important des hyperparamètres, en dehors du planificateur du taux d'apprentissage pour l'algorithme d'optimisation Adam dans l'entraînement non-supervisé. Il est généralement considéré comme une meilleure pratique de faire de la validation croisée en entraînant des classificateurs de l'apprentissage automatique, pour réduire l'impact de l'aléatoire dans la séparation des jeux d'apprentissage et de validation, mais ce n'était pas réalisable pour ces expériences, à cause du coût de calcul élevé de l'exécution. Il s'agit d'un amélioration possible pour les futurs expériences de retourner les entraînements avec des différents séparations afin de générer des estimations moins biaisées de la performance générale de ces techniques.

Dans la figure 6, on observe une corrélation entre le nombre d'exemplaires d'une variable cible dans le corpus d'entraînement et le score F1 d'un classificateur sur le corpus d'évaluation, ce qui suggère que la performance de notre système serait améliorée avec plus de données et d'exemples des différents types de maladies. Cette observation n'est surprenante, car il bien connu que les réseaux de neurones comme FlauBERT ont normalement besoin des énormes jeux de données pour sortir des bons résultats.

En plus, nous n'utilisons aucune connaissance externe explicitement encodée, comme les annotations supplémentaires ou des graphes de connaissance. Les graphes de connaissance externe sont souvent appliqués à des cas similaires, lorsque les jeux de données sont relativement petits (Costa *et al.*, 2021;

Chang *et al.*, 2020). Il s'agit d'une piste d'étude assez intéressante pour nous d'essayer de combiner des approches de ce type avec les techniques TALN discutées dans cet article.

Références

ALSENTZER E., MURPHY J. R., BOAG W., WENG W., JIN D., NEUMANN T. & MCDERMOTT M. (2019). Publicly available clinical BERT embeddings. *Proceedings of the 2nd Clinical Natural Language Processing Workshop*.

BLINOV P., AVETISIAN M., KOKH V., UMERENKOV D. & TUZHILIN A. (2020). Predicting clinical diagnosis from patients' electronic health records using BERT-based neural networks. *arXiv :2007.07562*.

CHANG D., BALA˘ZEVI I., ALLEN C., CHAWLA D., BRANDT C. & TAYLOR R. A. (2020). Benchmark and best practices for biomedical knowledge graph embeddings. *arXiv :2006.13774*.

COSTA J. P., STOPAR L., REI L., MASSRI B. & GROBELNIK M. (2021). Exploring biomedical records through text mining-driven complex data visualisation. *medRxiv*. DOI : 10.1101/2021.03.27.21250248.

DEVLIN J., CHANG M., LEE K. & TOUTANOVA K. (2018). BERT : pre-training of deep bidirectional transformers for language understanding. *arXiv :1810.04805v2*.

GROUIN C., GRABAR N. & ILLOUZ G. (2021). Classification de cas cliniques et évaluation automatique de réponses d'étudiants : présentation de la campagne deft 2021. *Actes de DEFT. Lille.*

HUANG K., ALTOSAAR J. & RANGANATH R. (2019). ClinicalBERT : Modeling clinical notes and predicting hospital readmission. *arXiv :1904.05342*.

KHATTAK F., JEBLEE S., POU-PROM C., ABDALLA M., MEANEY C. & RUDZICZ F. (2019). A survey of word embeddings for clinical text. *Journal of Biomedical Informatics X*.

KINGMA D. & BA J. (2015). Adam : A method for stochastic optimisation. *3rd International Conferenct on Learning Representations*.

LE H., VIAL L., FREJ J., SEGONNE V., COUAVOUX M., LECOUTEUX B., ALLAUZEN A., CRABBÉ B., BESACIER L. & SCHWAB D. (2020). FlauBERT : Unsupervised language model pre-training for French. *arXiv :1912.05372v4*.

LEE J., YOON W., KIM S., KIM D., KIM S., SO C. H. & KANG J. (2020). BioBERT : a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics 2020*.

PENG Y., YAN S. & LU Z. (2019). Transfer learning in biomedical natural language processing : and evaluation of BERT and ELMo on ten benchmarking datasets. *arXiv :1906.05474*.

VASWANI A., N.SHAZEER, PARMAR N., USZKOREIT J., JONES L., GOMEZ A., KAISER L. & POLOSUKHIN I. (2017). Attention is all you need. *Advance in Neural Information Processing Systems*.

YOON W., LEE J., KIM D., JEONG M. & KANG J. (2019). Pre-trained language model for biomedical question answering. *arXiv :1909.08229v1*.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Mesure de similarité textuelle pour l'évaluation automatique de copies d'étudiants

Xiaoou Wang [1] Xingyu Liu [1] Yimei Yue [2]
(1) Université Paris Nanterre, France
(2) Inalco, France
xiaoouwangfrance@gmail.com

RESUME

Cet article décrit la participation de l'équipe Nantalco à la tâche 2 du Défi Fouille de Textes 2021 (DEFT) : évaluation automatique de copies d'après une référence existante. Nous avons utilisé principalement des traits basés sur la similarité cosinus des deux vecteurs représentant la similarité textuelle entre des réponses d'étudiant et la référence. Plusieurs types de vecteurs ont été utilisés (vecteur d'occurrences de mots, vecteur tf-idf, embeddings non contextualisés de fastText, embeddings contextualisés de CamemBERT et enfin Sentence Embeddings Multilingues ajustés sur des corpus multilingues). La meilleure performance du concours sur cette tâche a été de 0.682 (précision) et celle de notre équipe 0.639. Cette performance a été obtenue avec les Sentence Embeddings Multilingues alors que celle des embeddings non ajustés ne s'est élevée qu'à 0.55, suggérant que de récents modèles de langues pré-entraînés doivent être fine-tunés afin d'avoir des embeddings adéquats au niveau phrastique.

ABSTRACT

Textual similarity measurement for automatic evaluation of students' answers
This paper describes the Nantalco team's participation in task 2 of the DEFT contest in 2021: Automatic evaluation of students' answers based on an existing reference. We mainly used features based on the cosine similarity of the two vectors representing the textual similarity between student responses and the reference. Several types of vectors were used (count vector, tf-idf vector, non-contextualized embeddings from fastText, contextualized embeddings from Camembert and fine-tuned Multilingual Sentence Embeddings). The best performance of the contest on this task was 0.682 (precision). The maximum performance of our team (0.639) was obtained with the Multilingual Sentence Embeddings, while the best performance of raw embeddings of CamemBERT was only 0.55, suggesting that recent pre-trained language models need to be fine-tuned in order to have adequate embeddings at the sentence level.

MOTS-CLES : évaluation automatique, similarité textuelle, CamemBERT
KEYWORDS: automatic evaluation, textual similarity, CamemBERT

Atelier DÉfi Fouille de Textes (DEFT), pages 63–71.

1. Description de l'équipe et de la distribution des tâches

L'équipe Nantalco est composée de trois Talistes issus de deux établissements : Université Paris Nanterre et Inalco. Parmi les trois tâches proposées (Grouin et al., 2021), nous avons choisi la deuxième tâche visant à évaluer automatiquement des copies d'étudiants d'après une réponse référence fournie par l'enseignant. Le responsable Xiaoou Wang s'est chargé du test des features et de la rédaction de l'article final, Xingyu Liu de l'exploration statistique du corpus et Yimei Yue de la normalisation des données.

2. Quelques caractéristiques du corpus d'entraînement

2.1. Nature de la référence

Les références données par l'enseignant sont de deux types : réponses référence et barèmes. Le premier type de réponses permet directement une mesure de similarité textuelle, le deuxième type requiert un traitement manuel ou un système de traduction qui convertit les barèmes en réponses. Comme le nombre de références du type barèmes est peu élevé, nous avons procédé à une conversion manuelle. A titre d'exemple, trois réponses ont été créées manuellement à partir de la question 1004 où les barèmes sont « head 1, html 0 et meta 0.5 ». Ces réponses créées manuellement ont été intégrées dans le corpus des réponses d'étudiants avec le numéro d'étudiant 0.

« Question 1004 Quelle balise HTML contient les informations destinées au navigateur et aux moteurs de recherche ? head 0 si html 0.5 si meta »

Cette phase manuelle nous a permis d'augmenter le nombre de réponses de 3820 à 3861.

2.2. Répartition des classes dans le jeu de données

Il est important d'examiner la distribution des classes avant de tester des modèles de classification. La FIGURE *1* montre une distribution bimodale avec un nombre important de notes 0 et 1. Nous sommes dès lors confronté à quelle hypothèse il faut faire concernant la distribution des notes dans le corpus test. Dans le cas d'un échantillonnage biaisé, il est possible de faire du downsampling qui consiste à ne tenir compte que d'un sous-ensemble de la classe dominante. Dans le cas des classifieurs comme SVM, il est aussi possible d'ajuster le paramètre C pénalisant davantage une mauvaise classification en faveur de la classe dominante. Cependant, nous avons fait le choix de ne pas tenir compte de ce déséquilibre car il s'agit ici d'une habitude de notation susceptible de se reproduire dans les données de test.

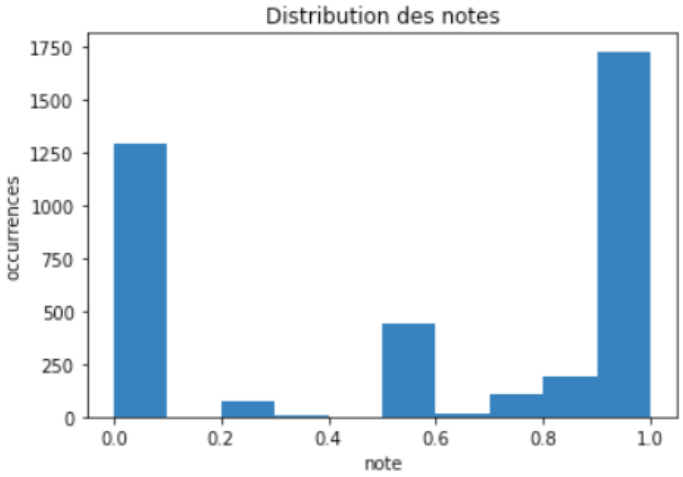

FIGURE 1 : Distribution des notes dans le corpus d'entraînement

3. Normalisation des données

Les processus de normalisations que nous avons effectués sur le corpus sont :

- suppression des tags html entourant les corrections et réponses
- suppression des mots vides avec des listes de mots-vides proposées par 3 bibliothèques : *NLTK* (Bird, 2006), *Spacy* (Honnibal et al., 2020) et *stopwordsiso*[1]
- suppression des mots vides basées sur des POS tags avec la bibliothèque *Stanza* (Qi et al., 2020), seuls les noms, nombres, noms propres, verbes, adverbes et adjectifs ont été gardés
- lemmatisation des tokens restants
- remplacement des espaces multiples avec un seul espace

Nous avons fait en sorte que le mot « pas » soit gardé dans tous les textes car la compréhension de la négation est importante pour la tâche de notation.

Notons que cette normalisation est également révélatrice de notre hypothèse qui suppose que la notation se fait en fonction du nombre de tokens apparaissant à la fois dans la réponse d'étudiant et dans la référence. Cette approche est approximative car dans ce cas les synonymes seront considérés comme des mots non pertinents alors que le fait d'utiliser un mot de sens proche est peu susceptible d'avoir un impact significatif sur la note finale. Des features basés sur des word embeddings seront par la suite ajoutés car ces derniers sont plus robustes dans le cas de l'usage des synonymes.

[1] https://pypi.org/project/stopwordsiso/

4. Élaboration des features

Les features que nous avons élaborés ont tous pour objectif de mesurer la similarité sémantique entre deux textes. Nous distinguons deux grandes classes.

4.1. Features basés sur la statistique textuelle

En utilisant le corpus normalisé, nous avons élaboré quelques features de base en utilisant de simples statistiques textuelles dont les définitions sont les suivantes :

1. ratio d'overlap : le nombre de mots communs apparaissant dans la réponse et la correction divisé par le nombre total de mots dans la correction
2. similarité cosinus calculée sur les vecteurs de compte de la réponse et de la correction
3. similarité cosinus calculée sur les vecteurs tf-idf de la réponse et de la correction
4. différence de nombre de caractères
5. différence de nombre de mots

4.2. Features basés sur des embeddings

Contrairement au comptage de mots qui est basé sur une représentation discrète (une occurrence de voiture n'est pas comparable avec une occurrence de bolide), les word embeddings (Mikolov et al., 2013) constituent une représentation continue de n dimensions qui permettent de mesurer la similarité lexicale. Pour représenter un document, il existe des méthodes telle que doc2vec (Le & Mikolov, 2014) qui, en dehors des vecteurs lexicaux, génère aussi un vecteur pour chaque document. Mais vu la longueur des réponses/références du corpus et la quantité de données d'entraînement, nous avons opté pour la méthode la plus simple consistant à faire la moyenne de tous les vecteurs.

Il existe principalement deux types d'embeddings lexicaux. Des embeddings non contextualisés proposent un seul vecteur pour un mot donné et des embeddings qui varient en fonction du contexte. Nous avons respectivement utilisé fastText (Bojanowski et al., 2017) et CamemBERT (Martin et al., 2019) pour générer ces embeddings en français. Pour ce dernier, il est aussi courant d'utiliser le token spécial [cls] pour représenter un document.

Avant de procéder à l'entraînement proprement dit, nous voudrions, à l'aide d'un simple exemple, montrer la non pertinence de la moyenne des embeddings CamemBERT pour représenter le sens des phrases. Nous utilisons à cet effet les 3 phrases suivantes :

1. J'aime les chats.
2. Je déteste les chats.
3. J'adore les chats.

Deux embeddings ont été utilisées pour calculer la similarité textuelle de ces phrases grâce à la similarité cosinus : la moyenne des embeddings et l'embedding du token [cls]. Notez que les fonctions permettant ces comparaisons ont été publiées dans le package *frenchnlp*[2], développé par notre équipe afin de faciliter l'utilisation des modèles de langue du type CamemBERT. La TABLE *1* montre les résultats :

Paire de phrases	Similarité cosinus basée sur la moyenne des embeddings	Similarité cosinus basée sur le token [cls]
j'aime vs je déteste	0.91	0.99
j'aime vs j'adore	0.99	1

TABLE 1 : Score de similarité entre 3 phrases en utilisant la moyenne des embeddings et [cls]

Il est évident que la différence de polarité entre phrase 1 et phrase 2 est mal représentée par cette méthode. Ce problème a été décrit plus en détail dans (Reimers & Gurevych, 2019), où les embeddings de BERT ont été comparés avec les embeddings Glove (Pennington et al., 2014) sur le benchmark STS (Cer et al., 2017), un corpus de phrases multilingue annoté en similarité textuelle. La mesure de la performance est la corrélation de Spearman et la performance de la moyenne des embeddings de BERT est seulement 46.35 contre 58.02 des embeddings GloVe. Les auteurs ont ensuite affiné les embeddings sur plusieurs benchmarks (dont STS), ce qui aboutit à une performance montée jusqu'à 88.77. Le package *sentence-transformers*[3], créé par les auteurs, permet de générer des sentence embeddings basés sur la méthode décrite dans l'article.

Il n'existe pas de sentence embeddings en français entraînés sur des corpus unilingues, cependant un modèle multilingue entraîné sur un corpus de plus de 50 langues, stsb-xlm-r-multilingual, est disponible. La TABLE *2* montre les résultats de ce modèle.

Paire de phrases	Similarité cosinus basée sur les sentence embeddings multilingues
j'aime vs je déteste	0.46
j'aime vs j'adore	0.96

TABLE 2 : Score de similarité entre 3 phrases en utilisant des sentence embeddings multilingues

[2] https://pypi.org/project/frenchnlp/, state of the art toolkit for Natural Language Processing in French.

[3] https://www.sbert.net/

Grâce à ces nouveaux embeddings, la différence de polarité est mieux représentée.

Par manque de corpus pour tester l'efficacité de ce modèle multilingue à plus grande échelle, nous avons décidé de tester toutes les 4 méthodes de représentation phrastique sur notre tâche.

4 sets de features au total ont été donc établis :

1. Les 5 features de base + similarité cosinus basée sur la moyenne des embeddings fastText
2. Les 5 features + similarité cosinus basée sur la moyenne des embeddings CamemBERT
3. Les 5 features + similarité cosinus basée sur l'embedding du token [cls]
4. Les 5 features de base + similarité cosinus basée sur les sentence embeddings de stsb-xlm-r-multilingual

5. Entraînement

Nous avons considéré la tâche comme une tâche de classification. Notons qu'il est tout à fait possible de modéliser cette tâche avec la régression linéaire, mais la dominance des notes 0 et 1 dans le jeu de données et la nature quasi-discrète des notes (il n'y pas de notes du type 0.11 ou 0.235) nous ont conduits à considérer les notes comme des chaînes de caractères et non des valeurs numériques continues.

Nous avons décidé d'utiliser SVM car ce dernier peut être ajusté avec des fonctions de classification différentes (astuce du noyau) permettant de capturer des relations non linéaires entre features et classes. Outre le noyau linéaire, nous avons aussi testé le noyau polynomial (degré 3) et le noyau RBF.

Pour la séparation des données train et dev, nous avons procédé à 10 folds stratifiés, générant ainsi 10 splits où le pourcentage de chaque classe est respecté. La moyenne de la précision sur les 10 splits a été utilisée comme indice de la performance du modèle.

6. Résultats et discussions

Toutes choses égales par ailleurs, la performance du noyau RBF a été systématiquement meilleure que les deux autres. Nous avons de ce fait présenté, dans la TABLE *3*, uniquement les résultats issus de ce kernel.

	fastText	Moyenne CamemBERT	Embedding [cls] CamemBERT	stsb-xlm-r-multilingual
Précision sur 10 splits	0.53	0.55	0.53	0.67

TABLE 3 : Précision en moyenne calculée d'un SVM au noyau RBF sur 10 splits du corpus.

Les résultats montrent que le score de similarité calculé avec des sentences embeddings ajustés sur un corpus multilingue (stsb-xlm-r-multilingual), conjointement avec les 5 features de base mentionnés plus haut, permettent de mieux prédire les notes des étudiants, alors qu'il n'y a pas d'écart significatif entre l'utilisation de la moyenne des embeddings fastText, des embeddings CamemBERT et du token [cls] de la phrase. Ces résultats vont dans le sens de l'article de (Reimers & Gurevych, 2019) qui montre que Bert, sans fine-tuning, n'est pas plus adéquat que les embeddings non contextualisés pour calculer la similarité sémantique entre deux phrases. En outre, nos résultats suggèrent que les sentence embeddings, bien que entraînés sur des corpus non unilingues, permettent d'obtenir des mesures de similarité plus pertinentes. Le modèle entraîné avec la similarité cosinus des embeddings de stsb-xlm-r-multilingual nous a permis d'atteindre une précision de 0.639 (meilleure performance de notre équipe) sur le jeu de données test.

Par contraintes de temps, nous n'avons pas pu tester d'autres types de sentence embeddings (cf. par exemple InferSent (Conneau et al., 2017) ou Universal Sentence Encoder (Cer et al., 2018)). Cependant, nous espérons avoir au moins montré la non pertinence des embeddings de CamemBERT à l'état brut pour représenter les phrases.

Un reproche pouvant être exprimé à l'égard de notre article est l'abandon des informations contenues dans les questions. En effet, il est possible de construire un système question-réponse où la référence d'enseignant peut être considérée comme un contexte et ensuite de calculer la probabilité d'une réponse spécifique. Des corpus en français comme FQuAD (d'Hoffschmidt et al., 2020) sont disponibles aujourd'hui et peuvent être utilisés afin de concevoir un tel système. Nous explorerons cette piste si l'occasion se présente.

Références

Bird, S. (2006). NLTK: The natural language toolkit. *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, 69–72.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146.

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., & Specia, L. (2017). Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *ArXiv Preprint ArXiv:1708.00055*.

Cer, D., Yang, Y., Kong, S., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Céspedes, M., Yuan, S., & Tar, C. (2018). Universal sentence encoder. *ArXiv Preprint ArXiv:1803.11175*.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., & Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. *ArXiv Preprint ArXiv:1705.02364*.

d'Hoffschmidt, M., Belblidia, W., Brendlé, T., Heinrich, Q., & Vidal, M. (2020). FQuAD: French question answering dataset. *ArXiv Preprint ArXiv:2002.06071*.

Grouin, C., Grabar, N., & Illouz, G. (2021). Classification de cas cliniques et évaluation automatique de réponses d'étudiants: Présentation de la campagne DEFT 2021. *Actes de DEFT. Lille*.

Honnibal, M., Montani, I., Van Landeghem, S., & Boyd, A. (2020). *spaCy: Industrial-strength natural language processing in python*. Zenodo. https://doi.org/10.5281/zenodo.1212303

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. *International Conference on Machine Learning*, 1188–1196.

Martin, L., Muller, B., Suárez, P. J. O., Dupont, Y., Romary, L., de la Clergerie, É. V., Seddah, D., & Sagot, B. (2019). Camembert: A tasty french language model. *ArXiv Preprint ArXiv:1911.03894*.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *ArXiv Preprint ArXiv:1301.3781*.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., & Manning, C. D. (2020). Stanza: A Python natural language processing toolkit for many human languages. *ArXiv Preprint ArXiv:2003.07082*.

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *ArXiv:198.10084 [Cs]*. http://arxiv.org/abs/1908.10084

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Participation d'EDF R&D à DEFT 2021

Philippe Suignard, Alexandra Benamar, Nazim Messous,
Clément Christophe, Marie Jubault, Meryl Bothua
(1) EDF Lab, 7 bd Gaspard Monge, 91120 Palaiseau, France
philippe.suignard@edf.fr, alexandra.benamar@edf.fr, nazim.messous@edf.fr,
clement.christophe@edf.fr, marie.jubault@edf.fr, meryl.bothua@edf.fr

RESUME

Ce papier présente la participation d'EDF R&D à la campagne d'évaluation DEFT 2021. Notre équipe a participé aux deux dernières tâches proposées (T2 et T3), deux tâches sur le calcul de similarité sémantique entre textes courts, et s'est classée 1ère sur ces deux tâches. Cette édition proposait deux nouvelles tâches pour l'évaluation automatique de réponses d'étudiants à des questions d'enseignants. Le corpus se composait d'une centaine d'énoncés en informatique avec la correction de l'enseignant et les réponses d'une cinquantaine d'étudiants en moyenne par question, sur 2 ans. La tâche 2 consistait à évaluer les réponses des étudiants en prenant pour référence la correction produite par l'enseignant et la tâche 3 à évaluer les réponses d'étudiants à partir d'un ensemble composé d'un énoncé et de plusieurs réponses d'étudiants déjà corrigées par l'enseignant.e.

ABSTRACT

EDF R&D Participation to DEFT 2021.
This paper describes the participation of EDF R&D at DEFT 2021. Our team worked on the second and the third tasks (T2 and T3). This edition included two new tasks dealing with automatic evaluation on students' answers to specific questions. The corpus was composed of a hundred questions about computer science, teacher's correction and students' answers. The second task consisted in evaluating students' answers regarding teachers' suggestions and the third task in evaluating students' answers regarding other students' answers, already corrected by the teacher. We finished first for both tasks.

MOTS-CLÉS : détection de similarité sémantique, CamemBERT, Soft cardinalité
KEYWORDS: Semantic Similarity Detection, CamemBERT, Soft cardinality.

1 Introduction

Une partie de l'édition 2021 du défi fouille de textes (Grouin *et al.*, 2021), à savoir les tâches 2 et 3, portait sur des calculs de similarité entre phrases. Participer à DEFT est l'occasion de tester des méthodes de calcul de similarité dont les résultats contribuent directement à EDF Commerce et à d'autres entités du groupe EDF.

2 Tâche 2 : « Evaluation automatique de copies d'après une référence existante »

2.1 Présentation

Cette tâche a pour but d'évaluer les réponses des étudiants à des questionnaires, c'est à dire à fournir une note entre 0 et 1, en prenant pour référence la correction produite par l'enseignant.

La démarche mise en œuvre pour résoudre cette tâche est la suivante :

- Pré-traitement des données textuelles ;
- Calcul de « features » ;
- Entrainement d'un classifieur ;
- Application du classifieur sur les données de test.

2.2 Les prétraitements

2.2.1 Prétraitements des runs 1 de T2 et T3 et remarques sur les indications et consignes de l'enseignant

Dans le fichier « trainT2.tab », la 5ème colonne est constituée de la réponse que l'enseignant aimerait trouver. En plus de la réponse attendue, l'enseignant vient parfois ajouter des commentaires ou des précisions. Ces éléments supplémentaires sont très utiles pour le correcteur, mais peuvent introduire des biais pour notre apprentissage machine :

- 19 questions sur les 50 contiennent ce genre de précisions. 31 autres n'en contiennent pas ;
- A la question 2014, il est précisé : « 0,7 si réponse imprécise ». Ce genre de commentaire est très difficilement interprétable par une machine ;
- A la question 1004 « Quelle balise HTML contient les informations destinées au navigateur et aux moteurs de recherche ? », la consigne de l'enseignant est : « head 0 si html 0.5 si meta », c'est-à-dire qu'il considère la réponse « head » comme bonne, la réponse « html » comme fausse et « meta » comme à moitié bonne. Comme nos *features* sont obtenues à partir de calculs de similarités, les réponses « head », « html » ou « meta » risque d'obtenir les mêmes scores. Dans d'autres cas, il est mentionné « 1 si l'étudiant répond ceci, 0.7 s'il dit cela, 0.5 s'il mentionne tel ou tel mot, etc. ».

De manière générale, ces commentaires ou précisions viennent « perturber » les classifieurs. On décide donc de les enlever pour les runs 1. Les résultats obtenus sur les données d'apprentissage montrent un gain notamment pour prédire la note 0,5.

Les traitements suivants sont appliqués aux données :

- Normalisation des balises "<", ">" en « < » et « > » ;
- Les balises , ,
 en début et fin de texte ont été supprimées ;
- Remplacement des caractères " " par un blanc ;
- Suppression des caractères \n et \t ;
- Insertion d'un caractère blanc avant « < » et après « > » pour faciliter la *tokenisation* en mots pour les calculs de similarité ;

– Utilisation du caractère blanc pour la séparation des phrases en *tokens* ;
– Passage en minuscule.

2.2.2 Prétraitements réalisés pour les runs 2 et 3 de T2 et T3

Pour les runs 2 et 3 des tâches 2 et 3, nous avons réalisé six prétraitements que nous détaillons ci-dessous.

- **Suppression des tags html** : nous avons repéré une liste de tags utilisés uniquement pour la mise en forme : écriture en gras, affichage sous forme de liste et saut de ligne. Nous avons donc supprimé les tags suivants :
  -

  -
  -
  -

  Ces suppressions ont été effectuées sur les fichiers questions réponses des tâches 2 et 3.

- **Suppression ou modification des entités html** : après analyse du fichier trainT2-Q.tab fourni pour la tâche 2, nous avons constaté la présence de plusieurs entités html que nous avons supprimées ou remplacées :
  - Le code entité html de l'espace insécable * *
  - Le code entité html du signe < *&lt.*
  - Le code entité html du signe > *&gt.*

  L'entité html * * a été supprimée de l'ensemble de nos données. Les entités *&lt* et *&gt*, n'ont pas été supprimées car elles sont parfois dans l'objet des questions et les suggestions de l'enseignant. Ces entités sont de plus utilisées pour délimiter le début et la fin de chaque balise html. En d'autres termes, ce ne sont pas des entités superflues. Leur suppression ferait perdre de l'information et influerait sur la similarité entre une réponse suggérée par l'enseignant et la réponse d'un étudiant. Les entités *&lt* et *&gt* ont donc été remplacées par les caractères '<' et ' >'. Voici un exemple ligne 3 du fichier trainT2-Q.tab :

  Quelle est la déclaration de type de document (doctype) d'une page web en HTML 5 (première ligne de la page HTML) ?
 <!DOCTYPE html>

  Devient à la place :

  Quelle est la déclaration de type de document (doctype) d'une page web en HTML 5 (première ligne de la page HTML) ?
 <!DOCTYPE html>

- **Suppression des sauts de ligne et des espaces supplémentaires** : nous avons supprimé les caractères \n et \t qui se trouvaient dans le fichier trainT2-Q.tab et le fichier trainT2-R3.tab.

- **Restructuration de l'information** : nous avons restructuré les deux fichiers trainT2-Q.tab et trainT2-R.tab dans un nouveau fichier .json en vue de l'application de nos modèles et notamment pour nos calculs de similarité.
  Pour la question, "Quelle structure de données la fonction pg_fetch_assoc retourne-t-elle ?", la réponse de l'enseignant est " un tableau associatif 0.7 si tableau ou

array 0.5 pour tuple
 ". Après application des prétraitements, les suggestions sont : 1. tableau ou array, 2. un tableau associatif, 3. Tuple.
Ces différentes suggestions et leurs notations sont très importantes. Nous avons donc divisé la réponse de l'enseignant en plusieurs parties (chaque partie est une suggestion de l'enseignant avec la note associée), puis sauvegardé ces différentes parties séparément en précisant qu'il s'agit de la réponse d'un enseignant. Nous avons ensuite renseigné pour une question donnée les réponses des élèves contenues dans le fichier trainT2-R.tab avec la note associée, en précisant qu'il s'agit d'une réponse élève. Cela nous permettra par la suite pour une question donnée de comparer la similarité entre la réponse donnée par chaque élève et chaque suggestion que l'enseignant a fait. Voici un exemple de résultat obtenus pour la question 1004 :
{**'numero question'**: '1004', **'text'**: 'Quelle balise HTML contient les informations destinées au navigateur et aux moteurs de recherche ?', **'Reponses'**: [{**'AuteurReponse'**: 'Enseignant', **'reponse'**: ' meta', **'note'**: 0.5}, {'AuteurReponse': 'Enseignant', 'reponse': ' html ', 'note': 0}, {'AuteurReponse': 'Enseignant', 'reponse': 'head ', 'note': 1}, {'AuteurReponse': 'eleve', 'id_etudiant': 'student101', 'reponse': 'NO_ANS', 'note': '0'}, {'AuteurReponse': 'eleve', 'id_etudiant': 'student108', 'reponse': 'la balise head <head></head>', 'note': '1'}

- **Suppression de la ponctuation et des lettres capitales.**

2.3 Les « features » utilisées

2.3.1 La « Softcardinalité »

La cardinalité d'un ensemble désigne le nombre d'éléments appartenant à cet ensemble. Si $\boldsymbol{S} = \{"\boldsymbol{a}","\boldsymbol{b}","\boldsymbol{c}"\}$, alors $|\boldsymbol{S}| = \boldsymbol{3}$. Ce nombre est indépendant des relations qu'entretiennent, entre eux, les différents éléments de l'ensemble. Mais si *a* et *b* sont proches l'un de l'autre d'un point de vue sémantique ou d'un point de vue lexical, on pourrait imaginer que cette cardinalité soit plus faible (2.5 ou 2.8 par exemple). C'est ce que permet la « softcardinalité » introduite par (Jimenez, 2015). Si $\boldsymbol{S} = \{\boldsymbol{s_1}, \boldsymbol{s_2}, \dots, \boldsymbol{s_n}\}$, on définit la softcardinalité par $|\boldsymbol{S}|' = \sum_{i=1}^{n} \frac{\boldsymbol{1}}{\sum_{j=1}^{n} \boldsymbol{sim}(\boldsymbol{s_i}, \boldsymbol{s_j})}$. La similarité entre deux termes étant définie ici par : $\boldsymbol{sim}(\boldsymbol{t_1}, \boldsymbol{t_2}) = \frac{\boldsymbol{2} * \left|\boldsymbol{t_1}^{[2:3]} \cap \boldsymbol{t_2}^{[2:3]}\right|}{\left|\boldsymbol{t_1}^{[2:3]}\right| + \left|\boldsymbol{t_2}^{[2:3]}\right|}$ avec $\boldsymbol{t}^{[2:3]}$ l'ensemble des bigrammes et trigrammes du terme t. Si $\boldsymbol{t} = "\boldsymbol{maison}"$,
$\boldsymbol{t}^{[2:3]} = \{"\boldsymbol{ma}","\boldsymbol{ai}","\boldsymbol{is}","\boldsymbol{so}","\boldsymbol{on}","\boldsymbol{mai}","\boldsymbol{ais}","\boldsymbol{iso}","\boldsymbol{son}"\}$

Cette méthode est utilisée car elle avait produit de très bons résultats lors d'une compétition similaire de SemEval 2013 (Dzikovska, 2013).

2.3.2 Similarité de Monge-Elkan

Soit deux suites de mots de longueurs différentes : $\boldsymbol{S} = \{\boldsymbol{s_1}, \boldsymbol{s_2}, \dots, \boldsymbol{s_n}\}$ et $\boldsymbol{T} = \{\boldsymbol{t_1}, \boldsymbol{t_2}, \dots, \boldsymbol{t_p}\}$. La similarité de Monge-Elkan entre ces deux suites de mots est définie de la manière suivante à partir d'une similarité entre deux mots sim_{mot} :

$$sim_{MongeElkan}(S, T, sim_{mot}) = \frac{1}{|S|} * \sum_{i=1}^{|S|} \underset{j=1 \text{ à } |T|}{max} sim_{mot}(s_i, s_j)$$

Pour la similarité entre les mots sim_{mot}, on utilisera une similarité stricte (1 si les mots sont identiques et 0 sinon) ou une similarité de type Damereau-Levenshtein. La similarité de Monge-Elkan n'étant pas symétrique, on construit la similarité symétrique suivante :

$$sim_{ME}(S, T, sim_{mot}) = \sqrt{sim_{MongeElkan}(S, T, sim_{mot}) * sim_{MongeElkan}(T, S, sim_{mot})}$$

2.3.3 Autres similarités

Les autres *features* utilisent les similarités plus connues comme cosinus, Jaro-Wikler, Damereau-Levenshtein et qui n'ont pas besoin d'être présentées.

2.4 Les différents Run

2.4.1 Run1

***Features* utilisées :** A partir de *q* la question posée, *a* la réponse de l'élève (« answer ») et *ra* la réponse proposée par l'enseignant (« request answer »), l'idée consiste à calculer des *features* et similarités croisées (entre *a* et *q*, *a* et *ra*, *q* et *ra*). L'article initial de (Jimenez, 2015) en calculait 42. Mais un premier calcul d'importance des *features* pour la classification a montré que les similarités les plus importantes étaient celles calculées entre *a* et *ra*, c'est-à-dire entre la réponse de l'élève et la réponse proposée par l'enseignant, ce qui paraît assez logique. On a donc réduit le nombre de *features* de l'article initial de 42 à 18 et en avons ajouté d'autres basées sur des similarités croisées entre *a*, *q* et *ra*. Au total, nous avons également 42 *features*.

Quelques précisions : dans le tableau suivant, $X \backslash Y$ correspond à l'ensemble X auquel est enlevé Y. $X \cap Y$ correspond à l'intersection (au niveau des mots) des chaines de caractères X et Y. $X \cup Y$ correspond à l'union (au niveau des mots) des chaines de caractères X et Y.

$\|a\|'$	$\frac{\|a \cap ra\|'}{\min(\|a\|', \|ra\|')}$	$cosinus_{bi}(a, ra)$
$\|q\|'$	$\frac{\|a \cap ra\|'}{\max(\|a\|', \|ra\|')}$	$cosinus_{bi}(q, ra)$
$\|ra\|'$	$\frac{\|a \cap ra\|' * (\|a\|' + \|ra\|')}{2 * \|a\|' * \|ra\|'}$	$cosinus_{tri}(a, q)$[1]
$\|a \cup q\|'$	$\|a \cup ra\|' - \|a \cap ra\|'$	$cosinus_{tri}(a, ra)$
$\|a \cup ra\|'$	$cosinus_{mot}(a, q)$[2]	$cosinus_{tri}(q, ra)$
$\|q \cup ra\|'$	$cosinus_{mot}(a, ra)$	$cosinus(a, q \cup ra)$
$\|a \cap ra\|'$	$cosinus_{mot}(q, ra)$	$cosinus_{bi}(a, q \cup ra)$
$\|a \backslash ra\|'$	$similarité_{Damereau\_Levenshtein}(a, q)$[3]	$cosinus_{tri}(a, q \cup ra)$
$\|ra \backslash a\|'$	$similarité_{Damereau\_Levenshtein}(a, ra)$	$sim_{ME}(a, q, sim_{DamereauLevenshtein})$[4]
$\frac{\|a \cap ra\|'}{\|a\|'}$	$similarité_{Damereau\_Levenshtein}(q, ra)$	$sim_{ME}(a, ra, sim_{DamereauLevenshtein})$
$\frac{\|a \cap ra\|'}{\|ra\|'}$	$similarité_{JaroWinkler}(a, q)$[3]	$sim_{ME}(q, ra, sim_{DamereauLevenshtein})$
$\frac{\|a \cap ra\|'}{\|a \cup ra\|'}$	$similarité_{JaroWinkler}(a, ra)$	$sim_{ME}(a, q, sim_{stricte})$[5]
$\frac{2 * \|a \cap ra\|'}{\|a\|' + \|ra\|'}$	$similarité_{JaroWinkler}(q, ra)$	$sim_{ME}(a, ra, sim_{stricte})$
$\frac{\|a \cap ra\|'}{\sqrt{\|a\|' * \|ra\|'}}$	$cosinus_{bi}(a, q)$[6]	$sim_{ME}(q, ra, sim_{stricte})$

Tableau 1 : Description des 42 features

Une fois les *features* calculées, un classifieur est entraîné à l'aide du logiciel Weka (Hall, 2009). Plusieurs classifieurs ont été testés et c'est Random Forest qui a obtenu le meilleur score sur les données d'entraînements. Pour ce *run1*, le nombre de classes à prédire a été ramené à 3 :

- 0 si la note était inférieure à 0,25 ;
- 0,5 si la note était comprise entre 0,25 et 0,75 ;
- 1 si la note était supérieure à 0,75.

2.4.2 Run 2

2.4.2.1 Calcul des embeddings de suggestions enseignant.e.s et de réponses étudiant.e.s

Afin de calculer les similarités entre les suggestions d'un.e enseignant.e pour une question donnée et la réponse d'un.e étudiant.e, nous avons utilisé le modèle de langue CamemBERT (Martin *et al.*, 2019), en français, basé sur l'architecture RoBERTa (Liu *et al.,* 2019). Nous avons ainsi encodé

[1] - cosinus calculé sur la chaine de caractères complète découpée en trigrammes
[2] - cosinus calculé sur les occurrences des mots
[3] - similarité calculée sur la chaine de caractères complète
[4] - calculé sur la chaine de caractères complète découpée en mots
[5] - calculé sur la chaine de caractères complète découpée en trigrammes
[6] - cosinus calculé sur la chaine de caractères complète découpée en bigrammes

l'ensemble de *n* mots sous la forme d'un vecteur, obtenant *n* vecteurs de taille 768. Nous obtenons une matrice de taille *n**768. Nous avons ensuite utilisé la librairie Flair afin de calculer nos embeddings de phrases à partir de la matrice. Cette librairie propose trois méthodes différentes pour générer des embeddings de phrases :

- *max* : une fois notre matrice *n**768 obtenue, nous prenons pour chaque colonne, la valeur maximum afin de former l'embedding final de notre document.
- *mean* : une fois notre matrice *n**768 obtenue, nous prenons pour chaque colonne la moyenne de la colonne afin de former l'embedding final de notre document.
- *min* : une fois notre matrice *n**768 obtenue, nous prenons pour chaque colonne la valeur minimum afin de former l'embedding final de notre document.

2.4.2.2 Calcul de distance et similarité avec seuils

Une fois les embeddings obtenus, nous avons mesuré la similarité entre la réponse de chaque étudiant.e avec chacune des suggestions de l'enseignant.e. L'enseignant.e ne donne pas systématiquement plusieurs suggestions : il ou elle peut en donner plusieurs ou se contenter de donner la réponse avec la note 1, qui est toujours donnée. C'est la raison pour laquelle nous avons défini des seuils de similarité pour affecter les notes 0, 0.5 ou 1. Nous n'avons pas cherché à rentrer dans la complexité de prédiction de notes intermédiaires comme 0.2, 0.7 ou 0.8. De manière empirique, nous avons fixé le premier seuil à 0.92. Quand le score de similarité obtenu était inférieur à 0.92 alors la note de la suggestion de l'étudiant.e n'était pas affectée à l'élève. Si aucune suggestion valant 0.5 point n'a été faite par l'enseignant.e, mais qu'il a suggéré une réponse valant 0, alors on affecte à l'élève la note 0. Dans le cas contraire, on affecte la note 0.5. Si l'enseignant.e n'a fait aucune suggestion valant 0 ou 0.5 point, alors on définit un second seuil de 0.905. Le plus grand score de similarité obtenu est supérieur à ce seuil. Nous affections par conséquent la note de 0.5 sinon la note 0.

2.4.2.3 Résultats obtenus en phrase d'entraînement

Une fois les embeddings calculés et les distances obtenues, nous affectons à l'élève la note de la suggestion pour laquelle nous avons obtenu le plus grand score de similarité. Les résultats suivants ont été obtenus sur la phase d'entraînement, par calcul de la précision en fonction des distances et similarité calculés et en fonction des paramètres d'embeddings choisis :

Distances et similarité	**Précision**		
	max	*mean*	*min*
Euclidean distance	*0.570*	*0.549*	*0.580*
Wasserstein distance	*0.537*	*0.538*	*0.548*
Cosinus Similarity	***0.588***	*0.579*	*0.579*
Manhattan distance	*0.417*	*0.412*	*0.417*
Jaccard distance	*0.396*	*0.396*	*0.333*

Le meilleurs score obtenu est celui avec le calcul de similarité cosinus avec le paramètre max pour le calcul de l'embedding. C'est la méthode que nous avons sélectionné pour l'évaluation du run 2 sur le jeu de test.

2.4.3 Run 3

Dans cette partie, nous utilisons Sentence-BERT (Reimers *et al.*, 2019), un réseau de neurones siamois qui a pour objectif de prédire une similarité cosinus entre deux phrases. Dans notre étude, nous avons utilisé le modèle CamemBERT (Martin *et al.*, 2019) en français, basé sur l'architecture RoBERTa (Liu *et al.,* 2019). Pour utiliser cette architecture, nous avons besoin de paires de phrases parallèles et de scores de similarité entre ces phrases.

Dans cette étude, nous avons testé l'utilisation de trois sources de données d'apprentissage :

- La question posée par le ou la professeur.e et la réponse de l'étudiant.e.
- La réponse attendue par le ou la professeur.e et la réponse de l'étudiant.e.
- La concaténation entre la question posée par le ou la professeur.e et la réponse attendue par le ou la professeur.e (ConcatProf.) et la réponse de l'étudiant.e.

Nous avons divisé le jeu de données en trois sous-ensembles : l'ensemble d'apprentissage (60% des paires), l'ensemble de validation (20% des paires) et l'ensemble de test (20% de paires). Les résultats obtenus sont présentés dans la Table 2. Ayant obtenu de meilleurs scores en utilisant uniquement la réponse des professeur.e.s, nous utilisons cette source de données pour notre soumission finale.

Données	**Précision**
Question prof. + Réponse étud.	*0.687*
Réponse prof. + Réponse étud.	***0.742***
ConcatProf. + Réponse étud.	*0.725*

Tableau 2 : Comparaison des résultats obtenus sur l'ensemble d'apprentissage avec Sentence-CamemBERT.

2.5 Résultats obtenus en phase de test sur la tâche 2

Run	**Evaluation**
Run 1 :	**P=0,682**
Run 2 :	P=0,594
Run 3 :	P=0,638
Maximum	**P=0,682**
Médiane	P=0,627
Moyenne	P=0,607
Minimum	P=0,448

Tableau 3 : résultats de la tâche 2

3 Tâche 3 : « Poursuite automatique de l'évaluation de réponses d'étudiants à partir de premières évaluations »

3.1 Présentation

Pour un ensemble composé d'un énoncé et de plusieurs réponses d'étudiant.e.s déjà corrigées par l'enseignant.e, il s'agit d'évaluer les autres réponses d'étudiant.e.s pour cet énoncé (fournir des notes comprises entre 0 et 1).

3.2 Les différents runs

3.2.1 Run 1

Les prétraitements appliqués sont les mêmes que pour le run1 de la tâche 2. Dans cette tâche, un petit nombre de réponses d'étudiant.e.s notées sont connues. La démarche suivie ici va s'appuyer sur ces notes connues pour prédire les notes inconnues. Pour cela, pour une question donnée, on va calculer la similarité ($\boldsymbol{sim_{ME}}$décrite en section 2.3.2 et calculée sur les trigrammes de caractères**)** entre une réponse d'étudiant.e dont on ne connait pas la note et les réponses des étudiant.e.s qui ont été notées. On retiendra la réponse avec laquelle la similarité est la plus élevée et on attribuera la note correspondante.

3.2.2 Runs 2 et 3

Pour cette tâche, nous utilisons Sentence-BERT, présenté en Section 2.4.3 de l'article. Ici, l'objectif est de prédire la note d'un.e étudiant.e en connaissant celles des autres pour cette question. Pour cela, nous construisons notre schéma en utilisant les élèves ayant obtenu une note maximale de 1/1. Nous souhaitons ensuite calculer la similarité des réponses des autres élèves aux réponses complètes. Pour cela, nous avons besoin de beaucoup de données. Pour répondre à ce problème, nous construisons tout d'abord deux tables de données. La première contient les réponses de tous les étudiant.e.s, les notes associées à ces réponses et l'identifiant de la question posée. La deuxième contient les mêmes métadonnées, mais concerne uniquement les réponses ayant obtenu des notes maximales. Nous procédons ensuite à une projection de ces deux tables sur la colonne des identifiants de questions. En d'autres termes, pour chaque question, nous associons toutes les paires possibles *réponse de l'étudiant.e - réponse correcte d'un.e autre étudiant.e.* Nous obtenons ainsi un jeu de données contenant 34 156 paires de réponses. Avec Sentence-BERT, nous essayons ensuite de prédire la similarité cosinus de ces paires de réponses, en utilisant la note obtenue à la question.

En sortie de Sentence-BERT, nous obtenons une similarité cosinus associée à chaque paire de réponses. Ces valeurs doivent ensuite être transformées en note pour cette tâche. Pour cela, nous divisons le corpus en trois notes possibles : 0, 0.5 et 1. Empiriquement, les seuils permettant d'obtenir les meilleurs résultats sur le jeu de données d'entraînement sont : les valeurs de similarités inférieures à 0.33 correspondent à des notes de 0, celles inférieures à 0.66 correspondent à des notes de 0.5 et les autres à des notes de 1 (run 2). Nous proposons une deuxième solution, qui consiste à arrondir les notes à une décimale (run 3).

3.3 Résultats

Run	Evaluation
Run 1 :	**P=0,510 (corrélation = 0,65),**
Run 2 :	P=0,382 (corrélation = 0,56)
Run 3 :	P=0,292 (corrélation = 0,59)
Maximum	**P=0,510 (corrélation = 0,65)**
Médiane	P=0,241
Moyenne	P=0,264
Minimum	P=0,133 (corrélation = 0,60)

Tableau 4 : résultats de la tâche 3

4 Conclusion

L'équipe texte de la R&D EDF a participé pour la 4e année consécutive au Défi Fouille de Texte dans le cadre de la conférence TALN (Traitement Automatique du Langage Naturel). Cette campagne 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. Nous sommes arrivés 1ers sur la tâche 2 et 1ers sur la tâche 3. Participer à ce défi est pour nous l'occasion d'échanger sur des méthodes de traitement automatique du langage avec des universitaires et des industriels.

Références

DZIKOVSKA, M. O., NIELSEN, R. D., BREW, C., LEACOCK, C., GIAMPICCOLO, D., BENTIVOGLI, L., ... & DANG, H. T. (2013). *Semeval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge*. NORTH TEXAS STATE UNIV DENTON.

GROUIN, C., GRABAR, N., ILLOUZ, G. (2021). Classification de cas cliniques et évaluation automatique de réponses d'étudiants : présentation de la campagne DEFT 2021. In : Actes de DEFT. Lille.

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.

JIMENEZ, S., GONZALEZ, F. A., & GELBUKH, A. (2015). Soft cardinality in semantic text processing: experience of the SemEval international competitions. *Polibits*, (51), 63-72.

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

MARTIN, L., MULLER, B., SUAREZ, P. J. O., DUPONT, Y., ROMARY, L., DE LA CLERGERIE, É. V., ... & SAGOT, B. (2019). Camembert: a tasty french language model. *arXiv preprint arXiv:1911.03894*.

REIMERS, N., & GUREVYCH, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Participation de Berger-Levrault (BL.Research) à DEFT 2021 : de l'apprentissage des seuils de validation à la classification multi-labels de documents

Mokhtar Boumedyen Billami, Lina Nicolaieff, Camille Gosset, Christophe Bortolaso
Berger-Levrault, 64 Rue Jean Rostand, 31670 Labège, France
{mb.billami, lina.nicolaieff, camille.gosset,
christophe.bortolaso}@berger-levrault.com

RESUME

Cet article présente notre participation à l'édition 2021 du DÉfi Fouille de Textes (DEFT) et plus précisément à la première tâche liée à l'identification du profil clinique du patient. Cette tâche consiste à sélectionner, pour un document décrivant l'état d'un patient, les différents types de maladies rencontrées correspondant aux entrées génériques des chapitres du MeSH (*Medical Subject Headings*). Dans notre travail, nous nous sommes intéressés aux questions suivantes : (1) Comment améliorer les représentations vectorielles de documents, voire de classes ? (2) Comment apprendre des seuils de validation de classes ? Et (3) Une approche combinant apprentissage supervisé et similarité sémantique peut-elle apporter une meilleure performance à un système de classification multi-labels ?

ABSTRACT

Berger-Levrault (BL.Research) submission to DEFT 2021: from learning validation thresholds to multi-label document classification.

This article presents the participation of Berger-Levrault team to the DEFT's 2021 challenge, and more precisely to the first task related to the identification of the patient's clinical profile. This task consists in selecting, for a document describing a patient's health status, the different types of diseases encountered corresponding to the generic entries of the MeSH (Medical Subject Headings) chapters. In our work, we were interested in the following questions: (1) How to improve vector representations of documents, or even classes? (2) How to learn class validation thresholds? And (3) Can an approach combining supervised learning and semantic similarity bring a better performance to a multi-label classification system?

MOTS-CLES : Apprentissage supervisé, Représentation sémantique de classes, Similarité sémantique, Réentraînement de plongements lexicaux, MeSH.

KEYWORDS: Supervised learning, Semantic representation of classes, Semantic similarity, Fine-Tuning, MeSH.

1 Introduction

L'édition 2021 du DÉfi Fouille de Textes (DEFT) (Grouin et al., 2021) est consacrée à trois tâches différentes, à savoir : (1) l'identification du profil clinique du patient ; (2) l'évaluation automatique

de copies d'étudiants d'après une référence existante ; et (3) la poursuite automatique de la correction d'après de premières corrections. Berger-Levrault s'est fortement intéressée à participer à la première tâche dont l'enjeu scientifique est la classification de cas cliniques. La direction BL.Research a tenu à proposer des systèmes de classification multi-labels traitant des données provenant du domaine de la santé. Nous avons appelé notre équipe BL.Santé pour faire référence aux données cliniques.

La première tâche de DEFT 2021 s'inscrit dans la continuité des deux éditions précédentes (Cardon et al., 2020 ; Grabar et al., 2019), à savoir : le traitement des cas cliniques rédigés en français (descriptions de situations cliniques rares utilisées à des fins pédagogiques, scientifiques ou thérapeutiques). L'édition DEFT 2019 s'est concentrée sur la recherche et l'extraction d'informations (*âge*, *genre*, *origine* et *issue*) à partir de documents. L'édition 2020, quant à elle, s'est poursuivie en partie sur la tâche d'extraction d'information avec de nouveaux types autour des patients (*anatomies*), de la pratique clinique (*examen*, *pathologie*, *signe* ou *symptôme*), des traitements médicamenteux et chirurgicaux (*substance*, *dose*, *durée*, *fréquence*, *mode d'administration*, *traitement* (*chirurgical* ou *médical*) et *valeur*) et autour du temps (*date* et *moment*). Ces différentes informations ont été proposées comme annotations dans les corpus de données de DEFT 2021.

Les données DEFT 2021 proviennent d'un ensemble plus vaste composé de cas cliniques, porteur d'annotations. Les cas cliniques sont anonymes et couvrent différentes spécialités médicales (*cardiologie, urologie, oncologie, obstétrique, pulmonaire, gastro-entérologie, etc.*). Ils décrivent des cas qui se sont produits dans différents pays francophones (France, Belgique, Suisse, Canada, pays africains, pays tropicaux, etc.). L'objectif principal que nous nous fixons pour la première tâche de DEFT 2021 est le suivant : pour un cas clinique donné, nous nous intéresserons à identifier le profil clinique du patient concerné par le type de maladie de toutes les pathologies présentes dans le cas.

Après avoir présenté en section 2 les corpus d'apprentissage et de test de DEFT 2021, nous décrivons notre méthodologie de classification multi-labels en section 3. À ce stade, nous présentons différents systèmes que nous avons proposés lors de la campagne. Par la suite, dans la section 4, nous décrivons les résultats d'évaluation avant de conclure en section 5.

2 Corpus de données

Le corpus se compose de cas cliniques décrits dans un format textuel. Il regroupe des documents pour l'apprentissage (cf. sous-section 2.1) et d'autres documents pour le test (cf. sous-section 2.2). Chaque corpus est accompagné d'annotations provenant des deux éditions précédentes de DEFT, avec 25 types au total pour l'apprentissage (*poids*, *taille*, *changement*, *état*, *prise*, *AnnotatorNotes*, *date*, *âge*, *origine*, *durée*, *fréquence*, *issue*, *norme*, *mode*, *genre*, *dose*, *pathologie*, *moment*, *assertion*, *traitement*, *valeur*, *substance*, *examen*, *anatomie* et *sosy*) et 28 types pour le test dont 3 nouveaux (*température*, *organisme* et *fonction*). Par la suite, nous présentons dans la sous-section 2.3 une analyse comparative entre les deux corpus. Pour le contenu textuel des corpus, le lecteur peut consulter le travail mené par Grabar et al. (2018) pour plus de détails.

2.1 Corpus d'apprentissage

Ce corpus regroupe 167 documents. Pour l'ensemble des 25 types d'annotation, nous avons 15 802 instances. Par exemple, *rachianesthésie* est une instance pour le type *traitement*, *épileptique* est une instance pour *pathologie*, voire *légère somnolence* pour *sosy*. Le corpus d'apprentissage a la particularité d'avoir de nouvelles annotations par rapport aux éditions précédentes. En effet, nous

disposons pour DEFT 2021 de nouvelles instances associées aux chapitres du MeSH. Nous appellerons ces instances dans ce qui suit par "termes-clés". Par exemple, *pyurie itératives* est un terme-clé du chapitre *infections*, *plombémie élevée* pour *chimiques*, *adénocarcinome à cellules claires* pour *tumeur*, voire *lombalgie gauche* pour *etatsosy*. Il est à noter que ce corpus d'apprentissage propose des instances d'annotation pour 23 chapitres du MeSH. La FIGURE 1 présente le nombre d'instances/occurrences et la taille du vocabulaire (termes-clés uniques) associés à chaque chapitre du MeSH. Le corpus d'apprentissage dispose de 2 115 instances de chapitres au total pour avoir 17 917 annotations tout type confondu.

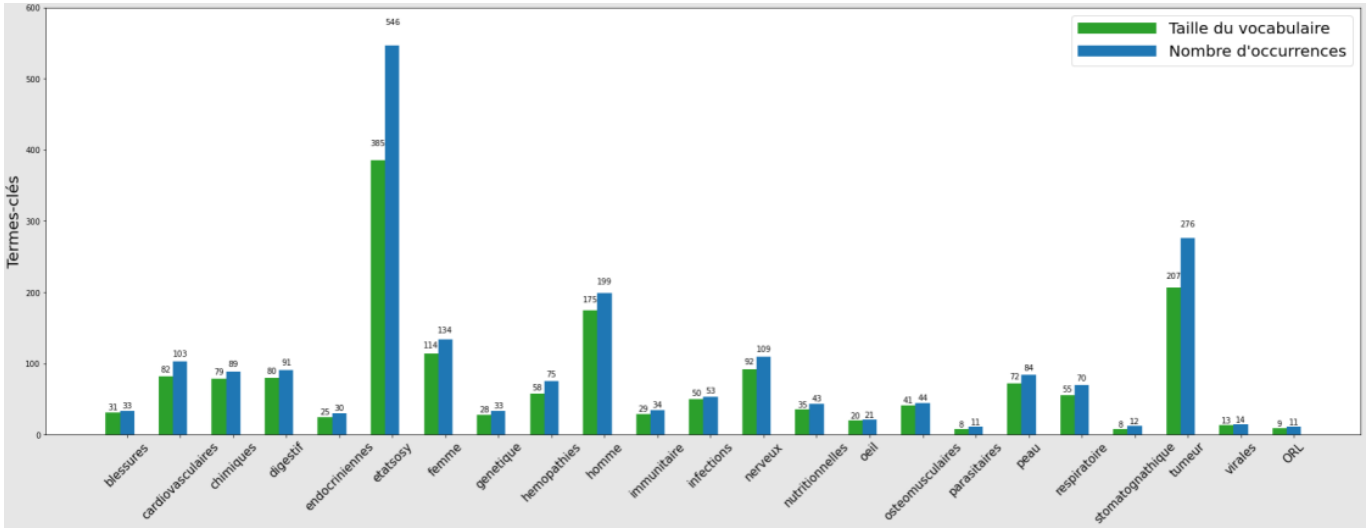

FIGURE 1: Distribution du nombre de termes-clés par chapitre du MeSH

Par ailleurs, le nombre d'annotations en chapitres dans ce corpus est 773. Sur l'ensemble des 167 documents, nous avons en moyenne 4,63 (≈ 5) classes (chapitres) par cas clinique. La classe la plus fréquente dans les documents est *etatsosy* avec 141 cas sur 167 contre la classe la moins fréquente *stomatognathique* avec 3 cas sur 167.

2.2 Corpus de test

Ce corpus regroupe 108 documents. Contrairement au corpus d'apprentissage, nous ne disposons pas d'instances de chapitres. Pour les annotations fournies dans ce corpus, nous avons 9 856 instances pour 28 types. Le type le plus fréquent est *sosy* avec 2 127 annotations contre le type le moins fréquent *état* avec 2 instances. En prenant en considération les chapitres de référence, nous constatons que sur l'ensemble des 108 documents, nous avons en moyenne 5,01 (≈ 5) chapitres par cas clinique. Cela revient à la même moyenne en comparaison avec le corpus d'apprentissage.

2.3 Analyse comparative des deux corpus

Dans cette sous-section, nous présentons une analyse comparative de la distribution des documents par chapitre entre apprentissage et test. En prenant en considération les annotations de référence des deux corpus, nous illustrons dans la figure FIGURE 2 le nombre d'exemples fournis pour chaque chapitre. Nous constatons qu'à l'exception des chapitres *ostéomusculaires* et *stomatognathique* avec un nombre d'exemples équitables entre apprentissage et test, la plupart du temps, nous avons plus d'exemples en apprentissage.

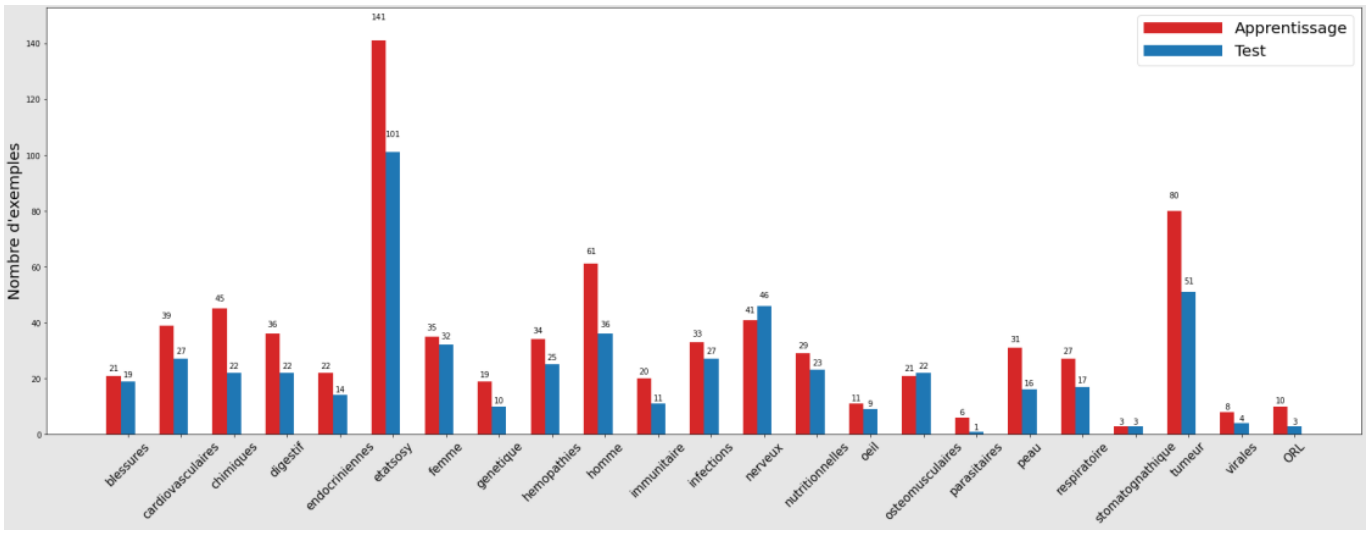

FIGURE 2: Distribution du nombre d'exemples (documents) par chapitre entre apprentissage et test

Le chapitre le mieux représenté en apprentissage est *etatsosy* mais cela revient aussi du fait qu'il est couvert par 84,4 % du corpus. Le chapitre *tumeur*, quant à lui, est couvert par 47,9 % du corpus d'apprentissage contre 47,22 % pour le corpus du test. Les trois chapitres *stomatognathique*, *parasitaires* et *virales* sont les moins représentés dans les deux corpus, avec moins de 10 exemples pour chacun.

3 Méthodologie

Dans cette section, nous présentons deux approches différentes pour répondre à la tâche de classification automatique multi-labels. La première approche (cf. sous-section 3.1) consiste à utiliser des plongements lexicaux provenant du domaine général et les réentraîner sur un corpus de spécialité, c'est-à-dire, le corpus d'apprentissage de DEFT 2021. Cette même approche consiste à créer des représentations sémantiques de documents, et de classes, et permet d'apprendre des seuils de validation pour valider les rapprochements sémantiques entre les documents et les classes. La deuxième approche (cf. sous-section 3.2), quant à elle, consiste à utiliser une succession de classificateurs binaires basés sur des représentations vectorielles de sacs de mots (*Bag-of-Words*). Nous proposons ensuite une combinaison des deux approches pour augmenter la couverture des classes non prédites par l'une des deux approches (cf. sous-section 3.3).

Par la suite, nous présentons dans cette même section une méthode d'extraction de terme-clés pour des documents provenant du corpus de test et faisant référence à des instances pour les chapitres du MeSH (cf. sous-section 3.4). Cette méthode repose principalement sur l'utilisation d'expressions régulières apprises à partir du corpus d'apprentissage.

3.1 Réentraînement de plongements lexicaux standards et apprentissage des seuils de validation

Cette première approche consiste à utiliser principalement des plongements lexicaux (*Word Embeddings*) et permet d'apprendre des seuils de validation des chapitres pour des cas cliniques. La figureFIGURE 3 présente l'architecture globale de notre approche. Dans l'ensemble du processus d'apprentissage, 8 étapes sont essentielles.

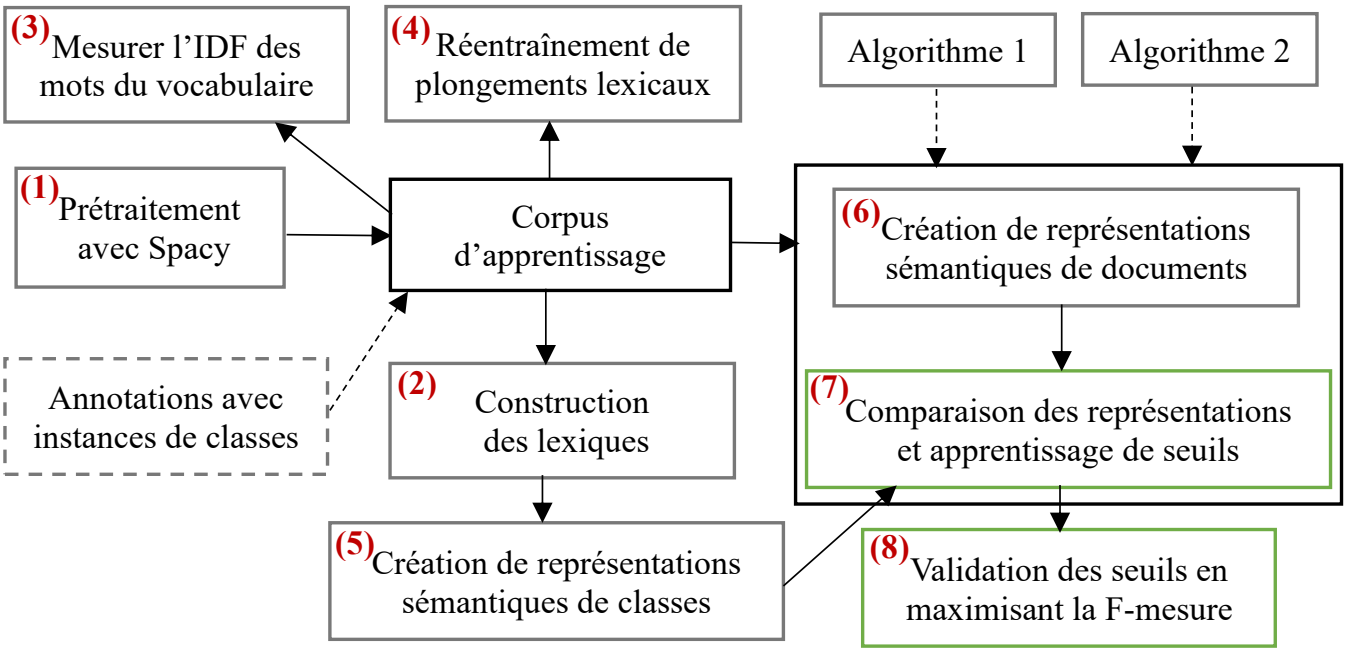

FIGURE 3: Architecture globale du premier système – apprentissage des seuils

Le principe de cette approche consiste à projeter dans un même espace vectoriel les documents et les classes. Pour cela, des vecteurs de documents et des vecteurs de classes sont créés. Nous détaillons ci-après chaque étape du processus.

- **Prétraitement des données :** Avec l'utilisation d'une chaîne de traitement du langage naturel comme Spacy[1] (Honnibal et al., 2020), nous prenons en considération seulement les mots portant du sens (noms, adjectifs, adverbes et verbes) dans leur format lemmatisé. Ce prétraitement est effectué sur le contenu textuel des documents et est pris en considération lors de l'utilisation des annotations.
- **Construction des lexiques :** nous prenons en considération uniquement les annotations associées aux instances de chapitres. Ces annotations, considérées comme termes-clés, alimentent des lexiques. Concrètement, nous construisons un lexique pour chaque classe (chapitre).
- **Mesurer la fréquence inverse du document pour chaque mot :** la fréquence IDF, *Inverse Document Frequency,* (Jones, 1972) est calculée sur le corpus d'apprentissage. Elle sert principalement à apporter des pondérations dans la création des représentations sémantiques de documents et classes (cf. étapes 5 et 6).
- **Réentraînement de plongements lexicaux :** à cette étape, nous prenons en considération un modèle Word2Vec (Mikolov et al., 2013) déjà entraîné (Fauconnier, 2015)[2] sur un corpus de grande taille, à savoir : FRWAC (Baroni et al., 2009), *The WaCky wide web* pour le français. Ce modèle est de type CBOW (*Continuous Bag of Words*) avec 500 dimensions pour la taille des vecteurs *embeddings*. Nous effectuons ensuite un réentraînement de type *fine-tuning* sur le corpus d'apprentissage à l'aide de la bibliothèque Gensim[3] tout en gardant un vocabulaire plus riche. À cette étape, le corpus est fourni dans un format prétraité avec Spacy. Par exemple, « ... *Le matin du jour trois, étant donné la persistance des nausées, des*

1 https://spacy.io/
2 http://fauconnier.github.io/#data
3 https://radimrehurek.com/gensim/models/word2vec.html

vomissements et de l'hypersalivation, l'équipe traitante augmente l'ondansétron à … » est considéré de la façon suivante « … *matin jour donner persistance nausée vomissement hypersalivation équipe traitant augmenter ondansétron …* ». Le nombre d'*epochs* a été fixé à 500. Cela n'a pas posé de problèmes pour les temps de calcul puisque la taille du corpus d'apprentissage reste relativement petite.

- **Création de représentations sémantiques de classes (chapitres) :** à partir du lexique construit pour chaque classe, nous créons des vecteurs centroïdes (moyens) pondérés avec l'IDF. Les vecteurs de mots proviennent du modèle fine-tuné.
- **Création de représentations sémantiques de documents :** dans le même principe que l'étape précédente, des vecteurs moyens pondérés sont créés en prenant en considération seulement les mots pleins des documents. Il est à noter qu'en phase de test, et dans le cas où de nouveaux mots apparaissent, l'IDF est considéré à une valeur égale à 1. Cela est dans le but de prendre en considération les vecteurs de mots du corpus de test non reconnus dans le corpus d'apprentissage. Dans cette étape, nous créons deux types de représentation sémantique pour les documents : (1) soit plusieurs vecteurs, chacun est associé à un paragraphe donné du document à traiter (cf. algorithme 1), (2) soit un vecteur représentant le texte intégral (cf. algorithme 2). Nous avons utilisé deux algorithmes dans la même approche afin d'associer le meilleur des deux pour chaque classe (chapitre).
- **Comparaison des représentations sémantiques, apprentissage et validation des seuils :** selon l'algorithme utilisé, nous comparons les vecteurs de classes avec les vecteurs de documents. Formellement, l'équation 1 ci-après fait référence au premier algorithme et l'équation 2 fait référence au deuxième algorithme.

$$Sim_1(C_i, Doc_j) = \underset{p \in paras\,(Doc_j)}{argmax} \; (1 - DistanceCosinus\,(Vec_{C_i}, Vec_p)) \quad (1)$$

$$Sim_2(C_i, Doc_j) = 1 - DistanceCosinus\,(Vec_{C_i}, Vec_{Doc_j}) \quad (2)$$

Avec Sim_1 et Sim_2 comme fonctions de similarité sémantique entre un chapitre C_i et un document Doc_j ; $paras$ représente l'ensemble des paragraphes d'un document Doc_j, Vec_{C_i} le vecteur du chapitre C_i, Vec_p le vecteur du paragraphe p et Vec_{Doc_j} le vecteur du document Doc_j.

Étant donné les annotations de référence du corpus d'apprentissage, nous avons appris les seuils de validation pour chaque classe permettant d'obtenir les meilleurs scores de F-mesure (mesure détaillée en section 4.1). Pour cela, nous avons varié les seuils de $0{,}1$ à 1 par pas de $0{,}01$.

Le Table 1 présente les seuils appris pour chaque algorithme et les meilleurs scores de F-mesure obtenus. Pour le premier algorithme, nous constatons que le seuil de validation minimal revient à 0,58 (pour *etatsosy*), à l'exception de la classe *tumeur* où le seuil est 0,46. Pour le deuxième algorithme, les seuils sont plus élevés puisque le minimum revient à 0,61 pour la classe *etatsosy*. De plus, dans la plupart du temps et pour chaque chapitre, les seuils pour le deuxième algorithme sont plus élevés à l'exception des classes *ostéomusculaires*, *stomatognathique* et *ORL*. Par ailleurs, l'utilisation du texte intégral permet d'avoir de meilleures scores pour la F-mesure à l'exception des classes *génétique, respiratoire, virales* et *ORL*. Ainsi, nous avons pris le choix d'utiliser l'algorithme 1 pour ces 4 classes et l'algorithme 2 pour toutes les autres classes.

Chapitre	Paragraphe le plus proche		Texte intégral	
	Best Seuil	**F-mesure**	**Best Seuil**	**F-mesure**
blessures	0,74	0,345	0,75	0,478
cardiovasculaires	0,65	0,505	0,70	0,553
chimiques	0,64	0,727	0,68	0,737
digestif	0,69	0,416	0,77	0,506
endocriniennes	0,74	0,444	0,76	0,489
etatsosy	0,58	**0,916**	0,61	**0,936**
femme	0,64	0,418	**0,80**	0,467
génétique	**0,79**	0,364	**0,80**	0,323
hémopathies	0,67	0,488	0,72	0,558
homme	0,65	0,618	**0,80**	0,662
immunitaire	0,65	0,545	0,67	0,558
infections	0,66	0,449	0,70	0,467
nerveux	0,59	0,466	0,68	0,545
nutritionnelles	0,68	0,514	0,73	0,549
œil	0,73	0,333	0,78	0,429
ostéomusculaires	0,77	0,32	0,72	0,333
parasitaires	0,70	0,222	0,74	0,364
peau	0,73	0,481	0,76	0,515
respiratoire	0,65	0,458	0,69	0,412
stomatognathique	0,69	0,667	0,67	0,667
tumeur	0,46	0,721	0,71	0,836
virales	0,69	0,5	0,70	0,471
ORL	0,70	0,333	0,64	0,235

TABLE 1 : Apprentissage du seuil de validation pour chaque chapitre du MeSH et présentation de la meilleure F-mesure obtenue

Toutefois, deux classes peuvent porter une certaine ambigüité, à savoir : (1) *Maladies urogénitales de l'homme (homme)* et *Maladies de l'appareil urogénital féminin et complications de la grossesse (femme)*. Afin de différencier ces deux classes et dans le cas où les deux classes sont prédites, nous utilisons une stratégie d'identification du genre. Concrètement, nous avons pris les annotations associées au type *genre* du corpus d'apprentissage. Le chapitre *homme* est validé pour un cas clinique donné seulement et seulement si aucun genre féminin n'est identifié. Cela est le cas aussi pour le chapitre *femme* à sa validation, c'est-à-dire, aucun genre masculin n'est identifié.

3.2 Apprentissage supervisé par utilisation de représentations vectorielles à base de sacs de mots (*Bag-of-Words*)

Cette deuxième approche consiste à utiliser des modèles d'apprentissage supervisé pour satisfaire le besoin de la classification multi-labels. Nous proposons d'utiliser plusieurs classificateurs binaires, chacun pour prédire un chapitre donné du MeSH. Ainsi, nous avons 23 classificateurs. Plusieurs

modèles de classification ont été testés, à savoir : la régression logistique, la classification naïve bayésienne, un classificateur de forêts aléatoires (*Random Forest Classifier*, RFC), voire un classificateur linéaire par vecteurs de support faisant partie de la famille des machines à vecteurs de support (SVM). Pour les représentations vectorielles de documents, nous nous sommes intéressés aux vecteurs TF-IDF (*Term Frequency-Inverse Document Frequency*) (Jones, 1972) et sacs de mots (*Bag-of-Words*). L'avantage de ces types de représentation, en comparaison aux représentations par plongements lexicaux, est qu'ils nous permettent de nous affranchir du réentraînement d'un modèle de langage, sur le vocabulaire spécifique médical.

En utilisant seulement le corpus d'apprentissage, et afin de sélectionner le meilleur modèle, nous avons mis en place une stratégie de validation basée sur 5 jeux de tests construits en gardant une même répartition des classes présentées dans ce corpus. L'équilibrage des classes dans les 5 jeux de données a été réalisé à l'aide la librairie *Scikit-multilearn*[4] spécialisée dans la classification multi-labels et basée sur Scikit-learn (Buitinck et al., 2013). Le nombre de classes prédites correspond au nombre de classes différentes en sortie du système. Le meilleur modèle aura ainsi un juste équilibre entre la F-mesure et une bonne répartition des classes dans les prédictions. Le tableau 2 présente les résultats obtenus pour la validation du modèle le plus pertinent à sélectionner.

Modèle	**F-mesure**		**Nombre de classes prédites**	
	TF-IDF	**Sac de mots**	**TF-IDF**	**Sac de mots**
Régression Logistique	**0,78**	0,61	4	**20**
Naïve Bayes	0,74	0,64	5	16
RFC	0,72	**0,66**	7	9
SVM	0,57	0,57	**15**	19

TABLE 2 : Résultats des performances de chacun des modèles testés en fonction de la représentation vectorielle choisie

L'approche ayant retenu notre attention par l'obtention des meilleurs validations combine des représentations de sac de mots avec un modèle de régression logistique. Cette méthode offre une meilleure diversité dans les classes prédites ainsi qu'une F-mesure considérable. La figure 4 illustre ainsi l'architecture globale de ce deuxième système.

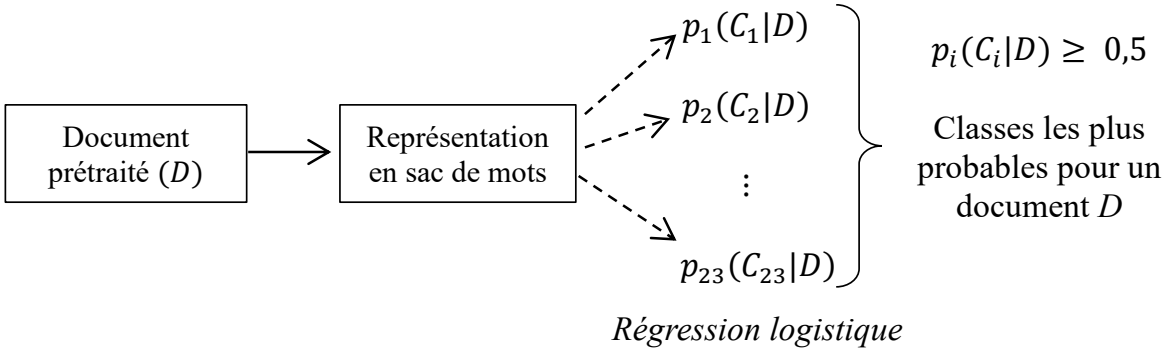

FIGURE 4: Architecture globale du deuxième système – apprentissage supervisé

Tout d'abord, nous tenons à mentionner qu'un prétraitement a été effectué sur tout le contenu textuel des documents de façon à ne garder que les mots portant du sens. Cela dit, les caractères spéciaux et

[4] http://scikit.ml/

les mots-vides (*stopwords*) sont supprimés. La fonction de perte de la régression logistique est optimisée à l'aide du solveur *liblinear*[5] et la diversité des classes prédites est améliorée grâce à la régularisation $L1$. Formellement, l'équation 3 permet de mesurer la probabilité d'associer la classe C_i au document D :

$$p_i(C_i|D) = \frac{e^{\beta_K X}}{1 + e^{\beta_K X}} \quad (3)$$

Avec k le nombre de mots retenus pour construire le dictionnaire servant à l'élaboration des vecteurs de sacs de mots (*Bag-of-Words*), X la matrice contenant l'ensemble des vecteurs représentant les documents et β les coefficients de régression estimés par le modèle. Le modèle permet d'estimer une probabilité de la classe C_i pour le document D. Par exemple, si le système offre une probabilité de 0,6 alors le document à 60 % de chance d'appartenir à cette classe.

3.3 Approche hybride : et si nous tenions compte des deux premiers systèmes

Pour un troisième système, nous nous sommes intéressés à prendre le cumul des deux premiers systèmes. Concrètement, nous nous intéressons ici à mieux couvrir les classes non prédites par l'un des deux systèmes. En effet, cela permettra d'augmenter la mesure du rappel (cf. sous-section 4.1). Toutefois, le risque d'une diminution de la mesure de précision (cf. sous-section 4.1) reste présent. Les résultats obtenus sur le corpus de test sont discutés dans la sous-section 4.2.

3.4 Extraction de termes-clés et association « Terme-Clé – Chapitre du MeSH »

Afin de satisfaire le besoin d'associer un chapitre à un terme-clé se trouvant dans un cas clinique, nous proposons une méthode d'extraction de termes-clés et d'association d'un terme-clé à un chapitre donné. Cette méthode s'inspire de l'utilisation des lexiques que nous avons créés et détaillés dans les sections précédentes.

Le principe de notre méthode est d'identifier automatiquement des termes-clés similaires à ceux des lexiques. Pour cela, nous utilisons les expressions régulières. L'utilisation de ces expressions offre plusieurs avantages. Tout d'abord, cela permet d'englober toutes les formes fléchies d'un terme-clé, c'est-à-dire, avoir la possibilité de récupérer un terme-clé sous différentes formes. En effet, la suite de termes-clés peut-être sous forme singulière ou plurielle (par exemple, *kyste pyélogénique* ou *kystes pyélogéniques*). Aussi, on peut récupérer plusieurs temps de conjugaison pour un verbe donné. Cela permet de retrouver dans un texte une suite de termes-clés au présent comme au passé (par exemple, *perdu conscience* ou *perdre conscience*).

Par ailleurs, nous tenons à récupérer des motifs de phrases contenant des termes-clés. Par exemple, dans l'expression *10 fois la dose*, l'intérêt est de pouvoir retrouver des expressions du genre *N fois la dose* où N est un entier > 0. En effet, dans le corpus de test, si la dose est prescrite, elle ne sera pas forcément d'une parfaite égalité à 10 (comme dans le corpus d'apprentissage). Pour le traitement de ces cas, nous récupérons la forme lemmatisée à l'aide de Spacy (Honnibal et al., 2020) pour chacun des termes-clés. Le traitement des chiffres est effectué en utilisant l'expression régulière ([0-9](, ?))+. Cette expression signale qu'il est possible de récupérer une suite de chiffres séparables à tout moment par une virgule. Le TABLE 3 présente un exemple pour l'application de ces expressions régulières.

[5] https://www.csie.ntu.edu.tw/~cjlin/liblinear/

Exemple tiré du corpus d'apprentissage	*10*	*fois*	*la*	*dose*
Expression régulière associée (concept)	([0-9](, ?))+	foi*	l*	dos*
Exemple concordant	5	fois	les	doses

TABLE 3 : Exemple de traitement de termes-clés avec des expressions régulières

Enfin, pour le traitement des majuscules, deux cas sont possibles : (1) nous pouvons avoir un terme complet en lettres capitales. Pour ce cas, l'expression régulière prend en compte la forme à la fois en capitale et en minuscule ; (2) nous pouvons avoir une majuscule seulement sur la première lettre des mots composant le terme-clé. Pour ce cas, l'expression régulière prend en compte la forme normalisée.

4 Résultats et discussion

Dans cette section, nous présentons tout d'abord les mesures d'évaluation recommandées par la campagne (cf. sous-section 4.1) avant de présenter les résultats obtenus (cf. sous-section 4.2).

4.1 Mesures d'évaluation

Souvent, le rappel (R), la précision (P) et la F-mesure (F) sont utilisés pour évaluer les performances des systèmes de classification automatique de textes. Le rappel permet de répondre à la question : Quelle proportion de résultats positifs réels est identifiée correctement ? La précision, quant à elle, répond à la question : Quelle proportion d'identifications positives est effectivement correcte ? La F-mesure est une moyenne harmonique du rappel et de la précision. Elle permet de mesurer la capacité d'un système à donner toutes les solutions pertinentes et à refuser les autres. Formellement, la description mathématique de ces mesures pour une étiquette de classe donnée C_i (cf. un chapitre C du MeSH), avec $C_i \in [1, 23]$, est présentée dans les équations suivantes :

$$P_{C_i} = \frac{TP_{C_i}}{TP_{C_i} + FP_{C_i}} \quad (4)$$

$$R_{C_i} = \frac{TP_{C_i}}{TP_{C_i} + FN_{C_i}} \quad (5)$$

$$F_{C_i} = \frac{2 \times P_{C_i} \times R_{C_i}}{P_{C_i} + R_{C_i}} \quad (6)$$

Dans les équations, 3 variables sont à déterminer, à savoir TP_{C_i}, FP_{C_i} et FN_{C_i} :

- TP_{C_i} représente le nombre de documents correctement classés pour la C_i-ième classe (vrais positifs)
- FP_{C_i} représente le nombre de documents qui sont incorrectement classés en C_i-ième classe (faux positifs)
- FN_{C_i} est le nombre de documents qui appartiennent à la C_i-ième classe, mais qui sont incorrectement classés (faux négatifs)

Pour le calcul des mesures sur toutes les classes (cf. mesures globales), la moyenne sur la somme est obtenue.

4.2 Résultats d'évaluation

Nous avons testé nos trois systèmes sur le corpus de test ayant 108 cas cliniques. Les résultats obtenus sont présentés dans le tableau 4. Nous nous comparons dans ce tableau avec le meilleur système ayant participé à la campagne DEFT 2021 (Grouin et al., 2021). Dans le tableau, le Run1 fait référence à notre premier système utilisant les plongements lexicaux et les seuils validés par apprentissage. Le Run2 fait référence au deuxième système à base d'apprentissage supervisé et utilisant la régression logistique. Le Run3 fait référence au troisième système utilisant la combinaison des deux premiers Runs.

Système	Rappel	Précision	F-mesure
Run1	0,677	0,570	0,619
Run2	0,471	**0,786**	0,589
Run3	**0,730**	0,558	**0,633**
Meilleur système	**0,750**	**0,885**	**0,812**

TABLE 4 : Résultats obtenus pour la classification multi-labels en chapitres du MeSH

Les résultats montrent que nous obtenons un bon rappel avec le Run3. Toutefois, le Run1 est le système ayant permis d'accroître cette bonne couverture. Nous constatons aussi que le Run2 possède un faible rappel. Cependant, il propose une meilleure précision que le Run1. Comme pressenti, la combinaison des deux (cf. Run3) permet seulement d'augmenter le rappel sans faire autant pour la précision. Néanmoins, ce Run3 permet d'avoir notre meilleur score pour la F-mesure. Par ailleurs, en comparaison avec le meilleur système de la campagne, le rappel du Run3 est proche avec un écart de 2 %. Pour la précision, l'écart avec le Run2 est près de 10 %. La figure 5 présente les résultats de F-mesure pour chaque chapitre du MeSH.

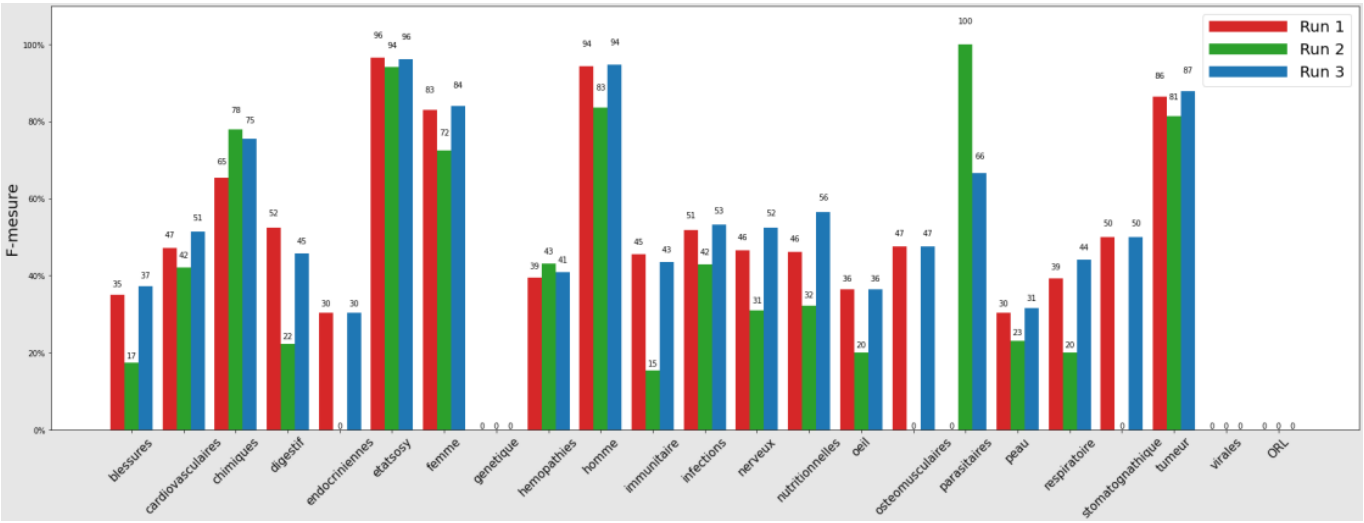

FIGURE 5: Résultats en F-mesure des trois systèmes (Runs) proposés pour chaque chapitre du MeSH

Sur ces résultats plus détaillés, nous constatons une certaine cohérence entre les validations effectuées sur le corpus d'apprentissage et les résultats obtenus en test. En effet, nous avons pour la prédiction du chapitre *etatsosy* une F-mesure de 96 % pour le Run1 et 94 % pour le Run2 (101 documents de référence sur 108). Les chapitres *femme* et *homme* sont aussi bien traités avec 83 % (*femme*, Run1) et

94 % (*homme*, Run1). Toutefois, le Run2 offre une bonne F-mesure pour certains chapitres comme *chimiques*, *hémopathies* voire *parasitaires*. Par ailleurs, aucun de nos systèmes n'a pu faire une identification des chapitres *génétique* (10 cas), *virales* (4 cas) ou *ORL* (3 cas). L'enrichissement des lexiques pour ces chapitres est une bonne piste afin d'améliorer les performances de nos trois Runs.

5 Conclusion

Notre participation à la campagne DEFT 2021 nous a permis de proposer deux approches totalement différentes pour satisfaire le besoin de la classification automatique multi-labels pour une application dans le cadre du domaine médical. De l'apprentissage des seuils et la comparaison de représentations sémantiques à la proposition d'un modèle d'apprentissage supervisé, nous avons testé ces techniques sur des données médicales. Nous constatons que ces méthodes peuvent être améliorées si nous enrichissons les lexiques construits, avec une utilisation de bases de connaissances (par exemple, des ontologies du domaine médical). En effet, si l'écart entre les représentations sémantiques de classes est grand, cela ne peut que diminuer le nombre de faux positifs et faux négatifs pour ainsi augmenter les valeurs de la précision et du rappel.

Remerciements

Nous tenons à remercier toutes les personnes ayant contribué à la réalisation de ce travail dans sa globalité. Nos remerciements vont tout particulièrement aux organisateurs de DEFT 2021 pour la disponibilité, la qualité et tout l'effort de l'annotation du corpus de travail.

Références

BARONI M., BERNARDINI S., FERRARESI A. & ZANCHETTA E. (2009). The WaCky wide web : a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, p. 209–226.

BUITINCK L., LOUPPE G., BLONDEL M., PEDREGOSA F., MUELLER A., GRISEL O., NICULAE V., PRETTENHOFER P., GRAMFORT A., GROBLER J., LAYTON R., VANDERPLAS J., JOLY A., HOLT B. & VAROQUAUX G. (2013). API design for machine learning software: experiences from the scikit-learn project. *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, p. 108–122.

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. *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*, Nancy, France. p. 1–13. HAL : hal-02784737v3.

FAUCONNIER J.-P. (2015). French Word Embeddings. URL : http://fauconnier.github.io.

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. *DEFT 2019 - Défi fouille de texte*, Toulouse, France. p. 1–10. HAL : hal-02280852.

GRABAR N., CLAVEAU V. & DALLOUX C. (2018). CAS: French Corpus with Clinical Cases. *LOUHI 2018 - The Ninth International Workshop on Health Text Mining and Information Analysis*, Bruxelles, France. p. 1–7. HAL : hal-01937096.

GROUIN C., GRABAR N. & ILLOUZ G. (2021). Classification de cas cliniques et évaluation automatique de réponses d'étudiants : présentation de la campagne DEFT 2021. *Actes de DEFT*. Lille.

HONNIBAL M., MONTANI I., VAN LANDEGHEM S. & BOYD A. (2020). spaCy: Industrial-strength Natural Language Processing in Python, *Zenodo*, DOI :10.5281/zenodo.1212303.

JONES K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, vol. 28, p. 11–21.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013). Efficient Estimation of Word Representations in Vector Space. *Proceedings of the International Conference on Learning Representations*, p. 1–12.

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

QUEER*@DEFT2021 : Identification du Profil Clinique de patients et Notations Automatique de Copies d'Étudiants

Yoann Dupont[1] Carlos-Emiliano González-Gallardo[1] Gaël Lejeune[1]
Alice Millour[1] Jean-Baptiste Tanguy[1, 2]
(1) Sens Texte Informatique Histoire (STIH), Sorbonne Université
(2) Observatoire des Textes et des Connaissances (OBTIC), Sorbonne Université
{yoann.dupont, carlos-emiliano.gonzalez-gallardo, gael.lejeune,
alice.millour, jean-baptiste.tanguy} @sorbonne-universite.fr

RÉSUMÉ

Nous présentons dans cet article notre contribution aux 3 tâches de la campagne d'évaluation du défi Fouille de Texte 2021. Dans la tâche d'identification de de profil clinique (tâche 1) nous présentons une méthode de recherche d'information basé sur un index dérivé du MeSH. Pour la tâche de notation automatique à partir d'une correction (tâche 2), nous avons expérimenté une méthode de similarité de vecteurs de chaînes de caractères. Pour la tâche de notation à partir de copies déjà notées (tâche 3) nous avons entraîné un réseau de neurones LSTM.

ABSTRACT

QUEER@DEFT2021 : Patients Clinical Profile Identification and Automatic Student Grading

We present in this article our contribution to the 3 tasks of the DEFT 2021 evaluation campaign. For the task of clinical profile identification (task 1) we present an information retrieval method using an index derived from the MeSH. For the automatic grading task from a correction (task 2), we computed similarity on character n-gram vectors. For the scoring task from already graded copies (task 3) we trained an LSTM neural network.

MOTS-CLÉS : MESH, Modèles en caractères, modèle LSTM.

KEYWORDS: MESH, Character-level Models, LSTM model.

1 Introduction

Le Défi Fouille de Textes (DEFT), créé en 2005, porte pour l'édition 2021(Grouin *et al.*, 2021) sur la classification de cas cliniques et la correction automatique de copies d'étudiants à travers trois tâches. Pour un cas clinique donné, la tâche 1 vise à identifier le profil clinique du patient concerné par le type de maladie de toutes les pathologies présentes dans le cas. Elle s'appuie sur les annotations antérieures en informations démographiques et cliniques (pathologies, signes ou symptômes, etc.). Les tâches 2 et 3 concernent la prédiction de notation de réponses d'étudiants à des questions courtes liées au domaine de l'informatique. Les réponses attendues peuvent être de différents types.

— réponse courte correspondant à une question ouverte ;
— réponse exacte correspondant à l'exécution d'un programme court ;
— code XML, HTML.

∗. Question Understanding for Event Extraction and Retrieval

Atelier DÉfi Fouille de Textes (DEFT), pages 95–107.

Remarquons d'entrée de jeu que les tâches 2 et 3 cherchent à développer des modèles capables, pour une réponse donnée, de proposer une note uniquement en observant le contenu intrinsèque de cette réponse. L'intersection des jeux de questions entre l'ensemble d'apprentissage et celui d'évaluation est vide, et l'identité de l'étudiant auteur de la réponse n'est naturellement pas pris en considération. Ces tâches visent ainsi à modéliser la langue (*rationnelle* dans le cas des réponses demandant du code, *naturelle* sinon) en fonction d'un système de notation.

Nous avons décidé de participer au trois tâches où nous explorons des différentes techniques de Recherche d'information, analyses statistiques et réseaux de neurones artificielles. Les jeux de données et métriques d'évaluation étant propres à chaque tâche, nous les présentons successivement.

2 T1 - Identification du profil clinique du patient

La tâche 1 s'articule autour de deux ressources : un corpus de documents annotés au format BRAT [1] (Stenetorp *et al.*, 2012) et un fichier de réponses attendues au format tabulaire. Les annotations disponibles sont celles de DEFT 2020 (Cardon *et al.*, 2020). Le corpus d'entraînement est constitué de 167 documents annotés au format BRAT, celui d'évaluation de 108 documents au même format.

Nous pouvons remarquer plusieurs éléments. Le premier est que la distribution entre les classes est très déséquilibrée, la classe "etatsosy" étant représentée dans environ 90% des documents, là où les classes minoritaires entre 1% et 5% des documents. Nous remarquons également que les distributions des classes dans l'ensemble d'entraînement et d'évaluation sont similaires. La seule différence notable est pour la classe "nerveux", qui passe d'un support relatif de 24,6% dans l'ensemble d'entraînement à 42,6% pour l'ensemble d'évaluation.

2.1 Méthode

D'abord nous avons considéré traiter la tâche comme un problème d'apprentissage automatique, plus précisément, comme un problème de classification automatique multiclasses étant donné l'existence d'un corpus d'entraînement avec 167 cas cliniques et 24 classes différentes. Après avoir effectué quelques expériences exploratoires, nous nous sommes aperçu que la distribution déséquilibrée entre les classes, ajouté au nombre réduit de cas cliniques et la très grande variabilité des justifications pour chaque élément du profil clinique d'un patient produisent un système de classification très silencieux. Finalement nous avons décidé d'aborder cette tâche comme un problème de Recherche d'information (RI) car nous bénéficions de la base de connaissance de référence dans le domaine biomédical. Dans le reste de cette section, nous présenterons les différents modules de notre système et une évaluation interne utilisée pour décider quelles configurations de modules seront sélectionnées pour l'évaluation officielle de DEFT2021.

2.1.1 Index inversé du MeSH

Le Médical Subject Headings (MeSH) est le thésaurus de référence dans le domaine biomédical. Nous avons obtenu le MeSH bilingue anglais-français (fMeSH) produit par l'Institut national de la

1. http://brat.nlplab.org

santé et de la recherche médicale (Inserm) et disponible en format XML. Il comprend 16 catégories thématiques mais pour nos besoins nous avons effectué une étape de filtrage pour garder uniquement les entrées correspondant au chapitre [C] - Maladies. Ce chapitre contient une arborescence de 26 grands groupes de maladies.

Une fois ces entrées identifiées, nous avons obtenu les termes (descripteurs) français associés a chaque entrée et son code correspondant dans l'arborescence. Ainsi dans l'index inversé un descripteur ou terme sera associé à une ou plusieurs maladies. Pour le processus d'indexation nos avons créé trois index inversés (Manning *et al.*, 2008) en suivant différents types de pré-traitements pour chaque descripteur et terme. Pour des raisons pratiques, dans la suite de l'article nous ferons référence aux index inversés simplement comme index.

— *simple* : seuls les caractères alphanumériques, les apostrophes et les traits d'union sont retenus, les espaces sont considérés comme séparateurs pour la tokenisation.
— *spacy_mots* : un token est considéré comme un élément du terme s'il n'est pas un mot vide ou un signe de ponctuation. La tokenisation, filtrage des mots vides et étiquetage morpho-syntaxique pour la détection de signes de ponctuation ont été effectués avec la bibliothèque Python Spacy [2] et le modèle pré-entraîné "fr_core_news_md" [3].
— *spacy_lemmes* : équivalent à *spacy_mots*, mais les lemmes des mots sont stockés dans l'index plutôt que leurs formes.

Le tableau 1 montre la quantité de descripteurs et termes différents dans chacun des index. Nous pouvons observer que le nombre de termes entre *simple* et *spacy_mots* ne varie pas beaucoup, néanmois *spacy_lemmes* présente une réduction de 4,36% par rapport à l'index *simple*.

index	**nombre de termes (descripteurs)**
simple	21 658
spacy_mots	21 384
spacy_lemmes	20 713

TABLE 1 – T1 : Nombre de termes et descripteurs pour chaque index

2.1.2 Détection des phrases négatives

Une des difficultés principales concerne les phrases dites négatives ou hypothétiques. Seules les maladies attestées dans le cas clinique, y compris celles dans le passé, doivent être conservées pour établir le profil du patient. Les maladies mentionnées mais absentes ou hypothétiques ne doivent pas être annotées. Nous avons attesté ce phénomène dans une proportion importante de cas cliniques ce qui a une grande répercussion au niveau des faux positifs. Pour cela nous avons donc implémenté une phase de détection des phrases négatives basée sur les couples mot $\leftrightarrow$ classe grammaticale décrits dans le tableau 2. Si une phrase contient au moins un de ces couples, la phrase est considérée comme négative et elle n'est pas prise en considération.

2. https://spacy.io/
3. https://github.com/explosion/spacy-models/releases//tag/fr_core_news_md-3.0.0

mot	classe grammaticale
ni	conjonction
pas	adverbe
aucun	déterminant
absence	nom
négative	verbe

TABLE 2 – T1 : Couples mot ↔ classe grammaticale

2.1.3 Filtrage des annotations hommes/femmes

Une difficulté supplémentaire de la tâche vient de l'existence de deux classes mutuellement exclusives, à savoir *homme* et *femme*, pour les maladies et complications liées à l'appareil uro-génital (ainsi que les complication dues à la grossesse pour les *femmes*). La particularité de ces annotations vient du fait que la classe demande connaissance extérieure à la justification donnée. Par exemple, une *anurie* sera la justification d'une maladie de l'appareil uro-génital, mais il faudra par ailleurs trouver la justification concernant le sexe du patient. Pour prendre en compte cette exclusion mutuelle, nous avons d'abord annoté *femme* et/ou *homme* en fonction des entrées de l'index trouvées dans le texte, puis nous avons appliqué des filtres en cas d'incohérence. Ces filtres fonctionnent sur le même principe que l'index, mais vont ici servir à supprimer une classe plutôt que d'en rajouter une.

2.1.4 Augmentation de l'index

L'un des problèmes principaux de notre méthode est le relativement faible rappel en comparaison de la précision. Afin de pallier ce problème, nous avons utilisé deux méthodes pour rendre l'index plus couvrant : 1. augmentation avec l'ensemble d'entraînement et 2. ajout manuel d'entrées.

Pour la première méthode, nous avons intégré des entrées correspondant aux justifications présentes dans l'ensemble d'entraînement. Cette méthode est représentée par l'indice *au* dans le tableau 4.

Nous avons également ajouté manuellement de nouvelles entrées à l'index constitué depuis le MeSH afin d'y intégrer des connaissances, aussi bien du domaine que linguistiques. Cette méthode est représentée par l'indice *up* dans la section 2.1.5. Certaines entrées du MeSH sont générales, comme par exemple "abus de drogues" ou "addiction à une substance". Ces éléments, bien que présents dans le MeSH, n'apparaissent pas dans le texte étant donné leur nature générique. Pour le cas de la classe *chimiques*, nous avons ajouté divers noms de drogues, comme par exemple "cannabis", "ecstasy", ou "alcool". Nous avons également ajouté divers termes en rapport avec les drogues concernées, comme par exemple "tabagisme", "alcoolique", "addiction" ou encore "empoisonnement". Pour la classe *endocriniennes*, nous avons rajouté "hormone" ainsi que différents dérivés flexionnels. De même pour les classes *femme* et *homme* où nous avons rajouté des termes spécifiques comme "ovaire", "testicule". Le tableau 3 donne un aperçu de la quantité d'ajouts manuels.

classe	nombre de termes ajoutés
virales	1
homme	13
femme	21
endocriniennes	6
chimiques	33

TABLE 3 – T1 : Nombre de termes ajoutés par classe

2.1.5 Expériences et évaluation interne

Nous avons conduit neuf expériences à partir des différentes configurations des modules de notre système. Pour toutes les configurations, le pré-traitement et la tokenisation des cas cliniques correspondent à celui de l'index utilisé.

- $T1_{bs}$: les requêtes se font sur l'index *simple*
- $T1_{bs\_scy}$: les requêtes se font sur l'index *spacy_mots*
- $T1_{up}$: $T1_{bs}+$ augmentation de l'index en suivant la méthode d'ajout manuel d'entrées
- $T1_{scy\_up}$: $T1_{bs\_scy}+$ augmentation en suivant la méthode d'ajout manuel d'entrées
- $T1_{scy\_up\_neg}$: $T1_{scy\_up}+$ détection des phrases négatives
- $T1_{scy\_up\_neg\_au}$: $T1_{scy\_up\_neg}+$ augmentation en suivant la méthode d'augmentation avec l'ensemble d'entraînement
- $T1_{scy\_up\_neg\_au\_n15}$: $T1_{scy\_up\_neg\_au}+$ n-grams de taille jusqu'à 15
- $T1_{scy\_neg\_au\_n15\_lem}$: les requêtes se font sur l'index *spacy_lemmes* avec détection des phrases négatives, augmentation en suivant la méthode d'augmentation avec l'ensemble d'entraînement et n-grams de taille jusqu'à 15
- $T1_{scy\_up\_neg\_au\_n15\_lem}$: $T1_{scy\_neg\_au\_n15\_lem}+$ augmentation en suivant la méthode d'ajout manuel d'entrées

Le tableau 4 donne un aperçu de la performance de chaque expérience effectuée. Nous pouvons observer que chaque configuration affecte différemment les scores de précision, rappel et F-mesure. L'ajout du filtrage des phrases négatives impacte très positivement la précision, tandis que l'utilisation de lemmes produit les meilleurs résultats en termes de rappel ; néanmoins la précision est fortement affectée négativement. Du point de vue de la F-mesure, utiliser les mots à la place des lemmes mais implémenter le reste des modules, semble être la meilleure option suite à la valeur maximale obtenue.

2.2 Résultats

Dans cette section, nous présenterons les résultats de nos systèmes sur la tâche 1 et analyserons diverses erreurs du systèmes. Les résultats officiels de QUEER sur la tâche 1 sont donnés dans le tableau 5 et le détail par classe est donné dans la figure 1.

Comme nous pouvons le voir dans le tableau 5, la principale source d'erreur de notre système est le silence. Les classes les plus concernées sont "nerveux" (21), "blessures" (14), "digestif" (13), "infections" et "osteomusculaires" (10). Nous nous concentrerons sur ces classes dans cette section.

Pour la classe "nerveux", une partie des silences ont rapport au sommeil, comme "endormissement" ou "somnolente". Ces termes apparaissent bien dans le MeSH, mais dans un contexte différent ou

Système	Précision	Rappel	F-mesure
$T1_{bs}$	0,783	0,673	0,724
$T1_{bs\_scy}$	0,760	0,702	0,729
$T1_{up}$	0,788	0,728	0,757
$T1_{scy\_up}$	0,766	0,753	0,759
$T1_{scy\_up\_neg}$	**0,810**	0,730	0,767
$T1_{scy\_up\_neg\_au}$	0,808	0,763	0,784
$T1_{scy\_up\_neg\_au\_n15}$	0,809	0,765	**0,786**
$T1_{scy\_neg\_au\_n15\_lem}$	0,759	0,739	0,748
$T1_{scy\_up\_neg\_au\_n15\_lem}$	0,762	**0,779**	0,770

TABLE 4 – T1 : Résultats sur la tâche 1 sur une validation croisée à 5 plis

Configuration	Précision	Rappel	F-mesure
$T1_{bs}$	0,819	0,677	0,741
$T1_{scy\_up\_neg}$	**0,843**	0,684	0,755
$T1_{scy\_up\_neg\_au\_n15}$	0,838	**0,734**	**0,782**

TABLE 5 – T1 : Résultats officiels QUEER

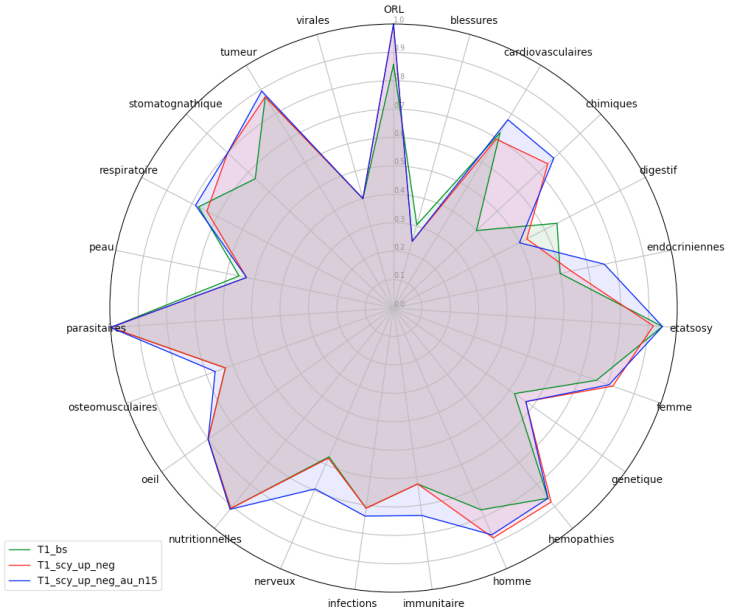

FIGURE 1 – T1 : Résultats officiels QUEER par classe (F-mesure)

une sous forme différente. Par exemple "endormissement" dans l'ensemble d'évaluation apparaît comme "trouble de l'endormissement". "Somnolente" est présent dans le MeSH sous la forme du nom commun "somnolence". Une piste pour améliorer le rappel de notre système pourrait être de nominaliser les formes trouvées dans le texte ("somnolente" deviendrait donc "somnolence"). En ce qui concerne la classe "blessures", le même constat semble se faire : il est détaillé différents types de plaies et autres blessures, cependant absents du MeSH. Pour les termes de la catégorie

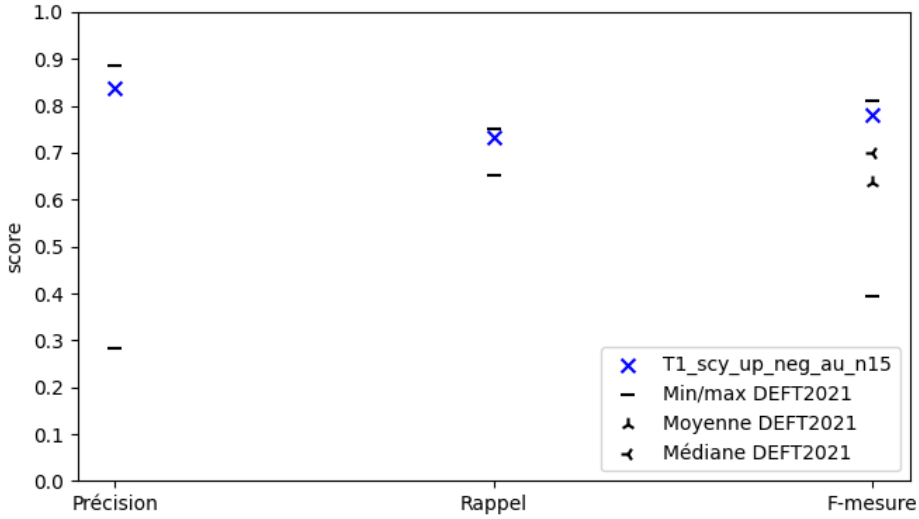

FIGURE 2 – T1 : Résultats QUEER VS DEFT2021

"digestif", le problème de rappel est surtout dû à la généricité des entrées du MeSH comme évoqué dans la section 2.1.4. En effet, de nombreux faux négatifs sont des instances particulières de "tumeur maligne gastrique" ou "tumeur hépatique". Pour la classe "infections", nous avons principalement relevé des noms de bactéries présentes dans le MeSH comme "escherichia coli" ou "klebsiella" qui apparaissaient seules dans le texte mais sous la forme de "infections à <bactérie>" dans le MeSH.

Il semble y avoir principalement deux décalages entre le MeSH et les cas cl²niques qui nuisent au rappel de notre système. Certains termes du MeSH sont plus précis que leur usage, là où d'autres termes du MeSH sont plus généraux que dans l'usage. Pour le premier cas, nous pourrions recourir à la fouille de motifs pour détecter les entrées ayant des constructions similaires pour en extraire des éléments spécifiques (eg : les bactéries des infections recensées dans le MeSH). Pour le second cas, nous pourrions rechercher des lexiques afin d'instancier les termes génériques utilisés dans le MeSH.

La performance de la meilleure configuration de notre système ($T1_{scy\_up\_neg\_au\_n15}$) par rapport au reste de participants à cette tâche est présentée dans la figure 2. Il est intéressant d'analyser le rang de valeurs concernant la précision et rappel. En ce qui concerne la précision, nous pouvons observer un intervalle de valeurs de 0,602 qui est très large si nous le comparons à l'intervalle du rappel qui est de 0,099. Cette différence de rang nous amène à conclure qu'il est beaucoup plus compliqué de diminuer les faux négatifs indépendamment de la méthode utilisée étant donnée un point de départ relativement élevé par rapport à la précision. Il est aussi possible que ce comportement est un résultat de la distribution déséquilibrée des classes. Notre système se positionne très proche des scores maximaux de précision, rappel et F-mesure avec une différence de -0,047 , -0,016 et -0,032 respectivement. En ce qui concerne la F-mesure, notre système est positionné bien au dessus de la moyenne et dans le 3^e quartile.

3 T2 - Évaluation automatique de copies d'après une référence

La tâche 2 s'articule autour de deux ressources : un fichier de questions et un fichier de réponses produites par des étudiants. Outre la question elle-même, le fichier de questions contient, pour certaines d'entre elles, un ou plusieurs éléments de réponse attendus. Ceux-ci s'accompagnent dans certains cas d'indications de notation à l'usage du correcteur. De ce fait, le fichier de questions contient également des éléments de réponse incorrects.

Les données d'entraînement et d'évaluation sont composées de questions distinctes (entraînement : 50, évaluation : 21 questions). En revanche, les réponses (entraînement : 3 820, évaluation : 1 644) ont été fournies par le même ensemble de 118 étudiants.

3.1 Méthode de correction fondée sur des similarités de chaînes de caractères

Nous avons vu la tâche 2 comme un problème consistant à vérifier la proximité entre la correction et la réponse donnée. Nous nous sommes basés sur un calcul de similarité entre la réponse de l'étudiant et une réponse modèle. Nous montrons que cette méthode peut être utilisée avec simplement le corrigé mais qu'elle fonctionne mieux avec des exemples de réponses déjà corrigées.

Nous avons testé différentes manières de vectoriser : en mots, en chaînes de caractères à l'intérieur des mots ou non. Nous avons comparé différentes mesures de similarité afin de comprendre leur influence sur la qualité des résultats, dans l'esprit de ce qui a été fait dans Buscaldi *et al.* (2020). Dans cet article, il était aussi montré que vectoriser des n-grammes de caractères mots ou non-mots permettait d'encoder des relations de l'ordre de la sémantique de manière plus souple qu'avec des mots. Nous y voyons deux raisons principales. D'une part, la vectorisation en n-grammes de caractères rend possible une racinisation non-supervisée : les racines de mots pertinentes sont en effet calculées en fonction des données traitées. D'autre part, avec des n-grammes suffisamment longs, il est possible d'encoder des relations séquentielles entre les mots. À nouveau, cela est fait de manière plus souple qu'avec des mots puisque certains descripteurs extraits sont en réalité des combinaisons "forme + racine". Enfin, les n-grammes de caractères sont plus robustes aux problèmes de variation locale (casse, orthographe…) que les approches en mots. Concernant la tâche elle-même, la méthode appliquée fonctionne comme suit :

1. Une similarité de 0 est attribuée aux non-réponses ;
2. Une similarité de 1 est attribuée aux réponses qui sont strictement contenues dans la correction ;
3. On vectorise le reste des réponses et la correction ;
4. On utilise le score de similarité multiplié par un correctif C avec $C >= 1$;
5. On arrondit au plus proche sachant les notes possibles.

Le coefficient est le produit de deux éléments : le nombre de points maximum que l'on peut obtenir sur la question (qui peut être différent de 1) et un correctif destiné à compenser la tendance du score de similarité à sous-estimer la proximité entre les bonnes réponses et la correction (en particulier sur les questions ouvertes où les vecteurs sont plus creux). Pour les réponses ne rentrant pas dans les cas 1 et 2 décrits ci-dessus on calcule donc la note comme suit (avec $corr$ la correction, rep la réponse de l'étudiant, $coeff$ le nombre de points maximum et C le correctif) : $note = sim(corr, rep) * coeff * C$. Enfin, cette note est arrondie à la valeur la plus proche parmi les notes possibles.

	Unigrammes	**Bigrammes**	**`char_wb(1,9)`** [4]	**`char(1,9)`**
Données brutes	0,607(p=0,044)	0,598(p=0,047)	0,566(p=0,048)	0,62 (p=0,035)
Minuscules	0,604(p=0,05)	0,599(p=0,045)	0,584(p=0,063)	0,632(p=0,034)
Sans balises p	0,604(p=0,05)	0,598(p=0,047)	0,566(p=0,048)	0,623(p=0,034)
Minuscules sans balises p	0,604(p=0,05)	0,598(p=0,188)	0,583(p=0,061)	0,636(p=0,033)

TABLE 6 – T2 : Corrélation de Spearman et *p-value* sur le jeu de données d'entraînement selon les pré-traitements effectués et le type de vectorisation (mesure de similarité : cosinus)

3.2 Paramétrage de la méthode et résultats

Nous présentons dans le tableau 6 l'importance des différents types de caractéristiques utilisées pour la vectorisation (en colonnes) et de pré-traitements (en lignes). Concernant les caractéristiques utilisées, nous comparons les mots (uni-grammes et bi-grammes) et les chaînes de caractères (restreintes à l'intérieur des mots ou libres). La tokenisation en mots est réalisée avec le tokeniseur par défaut de *scikit-learn* (Pedregosa *et al.*, 2011). Nous avons restreint les deux représentations en n-grammes de caractères à l'intervalle $1 \leq N \leq 9$.

La première observation que nous pouvons faire est qu'utiliser les caractères à l'intérieur des mots (`char_wb` dans le tableau) fonctionne moins bien qu'utiliser les mots eux-mêmes. De façon assez inattendue pour nous, les bi-grammes de mots donne de moins bons résultats que les uni-grammes. Enfin, utiliser des chaînes de caractères sans restriction sur les frontières de mots (colonne `char`) donne les meilleurs résultats. C'est en particulier vrai pour les réponses impliquant du code pour lesquelles la représentation en mots est inadaptée (on peut voir un résultat concordant sur de la détection de plagiat dans Brixtel *et al.* (2009)). Ces observations se reflètent à la fois sur la corrélation de Spearman et sur la *p-value*.

Concernant les pré-traitements, nous en avons testé deux très simples : passage en minuscules et suppression des balises "p" entourant la correction et les réponses. On n'observe pas d'influence significative sur les résultats des représentations en mots. L'apport est par contre plus significatif sur les approches en caractères. Le passage en minuscules est nettement plus influent et la combinaison avec le débalisage offre un certain gain pour l'approche en n-grammes de caractères.

Nous avons testé différentes mesures de similarité. Il apparaît que l'indice de Jaccard et la distance de Bray-Curtis donnaient les moins bons résultats. Le coefficient de Dice donnait les meilleurs résultats, le cosinus ayant tendance à sous-estimer les notes. En corrigeant la similarité par le coefficient C nous obtenions alors les meilleurs résultats avec la distance cosinus, $C = 3$ étant la valeur entière avec laquelle nous avions les meilleurs résultats.

4 T3 - Poursuite automatique de correction à partir de premières corrections

La tâche 3 s'articule autour de deux ressources : un fichier de questions et un fichier de réponses. Aucun élément de réponse n'est fourni. Néanmoins, au moins une des réponses produites par les

4. Nous utilisons ici la terminologie du COUNTVECTORIZER de *scikit-learn* : *characters between word boundaries*

étudiants a obtenu la note maximale.

Les données d'entraînement et d'évaluation sont composées de questions distinctes (entraînement : 11, évaluation : 6 questions). Les réponses du jeu de données d'évaluation (387 réponses) ont été produites par un sous-ensemble de 197 étudiants parmi les 213 ayant produit les réponses du jeu d'entraînement (769 réponses).

4.1 Méthodes

Trois méthodes ont été explorées pour la poursuite automatique de correction à partir de premières corrections : (M1) une méthode par pondération TF-IDF (Aizawa, 2003; Ramos *et al.*, 2003), (M2) une méthode par régression linéaire (Aggarwal & Zhai, 2012) et (M3) une méthode utilisant un réseau de neurones artificiels LSTM (Zhou *et al.*, 2015; Xiao *et al.*, 2018).

Les méthodes de pondération TF-IDF et de régression linéaire ayant été entraînées à prédire une note pour une question donnée – et l'ensemble de test ne comprenant pas les mêmes question que l'ensemble d'apprentissage –, elles n'ont pas été présentées au challenge. Néanmoins, nous les décrivons *infra* ainsi que leurs résultats sur l'ensemble d'apprentissage.

4.1.1 (M1) Méthode de pondération TF-IDF

Cette méthode fait intervenir la méthode de pondération TF-IDF [5]. Pour chaque question de l'ensemble d'apprentissage, les valeurs des TF-IDF de toutes les réponses entre elles sont calculées et stockées dans une matrice carrée. Le calcul de cette matrice a été réalisé en utilisant la classe *TfidfVectorizer* de la librairie *scikit-learn* (version 0.23.2) (Pedregosa *et al.*, 2011). Ensuite, pour chaque réponse de l'ensemble d'apprentissage, une note peut lui être prédite : celle de la réponse qui maximise le TF-IDF – il s'agit de la note de la réponse qui a un TF-IDF avec la réponse courante supérieur à tous les autres TF-IDF de la question courante.

4.1.2 (M2) Régression linéaire

Cette méthode apprend, pour chaque question, un modèle de régression linéaire. Afin d'apprendre à prédire les notes, les réponses sont vectorisées comme suit :

- stylométrie : longueur de la réponse ; nombre de caractères numériques ; nombre de caractères non numériques ; nombre de caractères d'espacement ; nombre de caractères n'étant pas d'espacement ; nombre de caractères de mots ; nombre de caractères n'étant pas de mots ; nombre de caractère en capital ;
- stemmes (*PorterStemmer* de la librairie *nltk* (Bird, 2006)) : chaque dimension du vecteur correspond à un stemme de l'ensemble des réponses de l'ensemble d'apprentissage.

Une fois les réponses vectorisées, l'apprentissage est réalisé en utilisant la classe *LinearRegression* de la librairie *scikit-learn*.

5. *term frequency-inverse document frequency*

4.1.3 (M3) Réseau de neurones LSTM

La troisième méthode, la seule présentée, est un réseau de neurones constitué avec la libraire Python *keras* (version 2.4.3) (Brownlee, 2016). À la différence des deux premières, cette méthode n'est pas spécialisée par question. Un unique réseau est constitué pour toutes les questions et réponses. Il y a donc une prétention à modéliser la valeur (notative) d'une réponse, quelle que soit la question. Théoriquement, cela pose problème : une réponse peut être correcte pour une question et incorrecte pour une autre. Une voie d'amélioration serait l'intégration de l'énoncé de la question à la modélisation.

Pré-traitements des données Les réponses sont mises en bas de casse et les caractères non alphanumériques remplacés par des espaces. La segmentation est opérée par le *Tokenizer* de *keras*.

Structure du réseau Le réseau est constitué d'une première couche *Embedding* qui vectorise les réponses (nombre de dimension des vecteurs : 100). À cette première couche succèdent les couches profondes du réseau, spécifiques à chaque *run* présenté :

run 1 : Une couche *LSTM (120, dropout = 0,2)*, suivie d'une couche *Dense(120, activation = "relu")*, suivie d'une couche *Dense(21, activation = "linear")*, suivie d'une couche *Dense(21, activation = "softmax")*. Apprentissage réalisé sur 10 époques.

run 2 : Une couche *LSTM (100, dropout = 0,2)*, suivie d'une couche *Dense(100, activation = "relu")*, suivie d'une couche *Dense(100, activation = "linear")*, suivie d'une couche *Dense(21, activation = "softmax")*. Apprentissage réalisé sur 10 époques.

run 3 : Une couche *LSTM (120, dropout = 0,2)*, suivie d'une couche *Dense(120, activation = "relu")*, suivie d'une couche *Dense(21, activation = "linear")*, suivie d'une couche *Dense(21, activation = "softmax")*. Apprentissage réalisé sur 100 époques.

4.2 Résultats

4.2.1 M1 et M2 : méthodes non présentées

Questions	M1 Précision	M1 Rappel	M1 F-mesure	M2 r2
5002	0,200	0,104	0,078	0,022
5003	0,151	0,389	0,218	0,156
5004	0,111	0,244	0,152	0,090
5006	0,581	0,344	0,357	0,479
5007	0,395	0,589	0,453	-0,719
5008	0,439	0,267	0,191	0,715
5010	0,603	0,189	0,214	-0,137
5013	0,370	0,444	0,354	-0,943
5014	0,102	0,193	0,111	-0,186
2011	0,800	0,845	0,781	###[6]
2013	0,584	0,319	0,205	-0,276

TABLE 7 – T3 : Résultats, sur l'ensemble d'apprentissage, pour les méthodes M1 (Précision, Rappel et F-mesure) et M2 (coefficient de détermination r2), non présentées

Le tableau 7 montre pour la méthode M1 des résultats peu satisfaisants, dans la mesure où ils peinent à dépasser les résultats obtenus sur l'ensemble de test par la meilleure équipe ($F - mesure = 0,510$). On observe aussi des disparités entre les questions, la question numéro 5002 obtenant des résultats très proches de 0.

Les résultats de cette première méthode *baseline* ont conforté l'idée de procéder à l'apprentissage de modèle de notation spécifique à chaque question, lesquelles sont substantiellement différentes les unes des autres (comme proposer du code HTML *vs.* proposer une réponse en langage naturel). Le tableau 7 montre aussi des résultats mitigés pour la méthode M2. La question 5006 est corrigée relativement correctement quand les questions à $r2 < 0$ ne le sont pas du tout.

4.2.2 M3 : méthode présentée

Questions	**run 1**	**run 2**	**run 3**
2012	0,413	0,375	0,356
5001	0,113	0,038	0,094
5005	0,266	0,172	0,188
5009	0,532	0,532	0,362
5011	0,149	0,170	0,085
5012	0,000	0,000	0,000
Moyenne	0,278	0,241	0,212

TABLE 8 – T3 : Précision des 3 runs sur l'ensemble de test

Le tableau 8 montre que les trois *runs* offrent des performances similaires, avec une supériorité pour le *run 1*. Effectivement, mis à part l'agencement des couches profondes et le nombre d'époques, ils sont égaux. On comprend donc bien que l'agencement des couches profondes et le nombre d'époques n'ont pas un impact majeur sur les performances du système. En outre, on observe aussi que la question 5012 est toujours mal notée et que la question 5009 est toujours la mieux notée.

5 Conclusion

Dans cet article, nous avons présenté différentes approches pour les trois tâches du Défi Fouille de Textes 2021. En ce qui concerne la tâche 1 nous avons décidé de créer un système inspiré de la recherche d'information pour identifier le profil d'un patient à partir de son cas clinique. Notre système s'est montré très performant avec une F-mesure de 0,782 ; bien au-dessus de la moyenne des équipes. Néanmoins le taux de faux négatifs est un élément à travailler pour diminuer le silence du système avec des analyses plus précises du contenu des cas cliniques. Dans la tâche 2 (notation automatique à partir d'une correction) nous avons proposé une méthode fondée sur une similarité de distribution de chaînes de caractères (mots ou non-mots). Cette méthode a obtenu une F-mesure de 0,63 ce qui la situe juste au-dessus de la médiane des participants au défi. Nous envisageons d'appliquer cette méthode à la tâche 3. Cette dernière a été abordée en utilisant un réseau de neurones LSTM et a obtenu une F-mesure de $0,278$ qui la situe au-dessus de la moyenne et de la médiane.

6. Les trois dièses ### signifient que la valeur est très grande ($< 10^{24}$) mais négative.

Références

AGGARWAL C. C. & ZHAI C. (2012). A survey of text classification algorithms. In *Mining text data*, p. 163–222. Springer.

AIZAWA A. (2003). An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, **39**(1), 45–65.

BIRD S. (2006). NLTK : the natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, p. 69–72.

BRIXTEL R., LESNER B., BAGAN G. & BAZIN C. (2009). De la mesure de similarité de codes sources vers la détection de plagiat : le "Pomp-O-Mètre". In *Actes des 7èmes Journées MajecSTIC'09*, p. 8 p., Avignon, France. Actes électroniques, HAL : hal-01066127.

BROWNLEE J. (2016). *Deep learning with Python : develop deep learning models on Theano and TensorFlow using Keras*. Machine Learning Mastery.

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 R. CARDON, N. GRABAR, C. GROUIN & T. HAMON, Éds., *DEFT@JEP/TALN/RECITAL 2020*, p. 14–25, Nancy, France : ATALA. HAL : hal-02784738.

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 (presentation of the deft 2020 challenge : open domain textual similarity and precise information extraction from clinical cases). In *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*, p. 1–13.

GROUIN C., GRABAR N. & ILLOUZ G. (2021). Classification de cas cliniques et évaluation automatique de réponses d'étudiants : présentation de la campagne deft 2021. In *Conférence sur le Traitement Automatique des Langues Naturelles (TALN, 28e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 23e édition). Atelier DÉfi Fouille de Textes*.

MANNING C. D., SCHÜTZE H. & RAGHAVAN P. (2008). *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge.

PEDREGOSA F., VAROQUAUX G., GRAMFORT A., MICHEL V., THIRION B., GRISEL O., BLONDEL M., PRETTENHOFER P., WEISS R., DUBOURG V. *et al.* (2011). Scikit-learn : Machine learning in python. *the Journal of machine Learning research*, **12**, 2825–2830.

RAMOS J. *et al.* (2003). Using TF-IDF to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, p. 29–48 : Citeseer.

STENETORP P., PYYSALO S., TOPIĆ G., OHTA T., ANANIADOU S. & TSUJII J. (2012). BRAT : a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, p. 102–107.

XIAO L., WANG G. & ZUO Y. (2018). Research on patent text classification based on word2vec and LSTM. In *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, volume 1, p. 71–74 : IEEE.

ZHOU C., SUN C., LIU Z. & LAU F. (2015). A C-LSTM neural network for text classification. *arXiv preprint arXiv :1511.08630*.