# *18e Conférence en Recherche d'Information et Applications*
# *16e Rencontres Jeunes Chercheurs en RI*
# *30e Conférence sur le Traitement Automatique des Langues Naturelles*
# *25e Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues*
# (CORIA-TALN) [1]

**Actes de CORIA-TALN 2023.**

**Actes du Défi Fouille de Textes@TALN2023**

Adrien Bazoge, Béatrice Daille, Richard Dufour, Yanis Labrak, Emmanuel Morin, Mickael Rouvier (Éds.)

Paris, France, 5 au 9 juin 2023

1. https://coria-taln-2023.sciencesconf.org/

## Avec le soutien de

















# Préface

Créé en 2005 à l'image des campagnes TREC et MUC, le DÉfi Fouille de Textes est une campagne d'évaluation francophone qui propose chaque année de confronter les méthodes de plusieurs équipes de recherche sur une thématique régulièrement renouvelée.

Dans la continuité de l'édition 2022, cette nouvelle édition du défi portera sur la mise en place d'approches permettant de répondre automatiquement à des questionnaires à choix multiples issus d'annales d'examens de pharmacie. Le corpus utilisé, FrenchMedMCQA, se compose de questions fermées en français provenant d'annales d'examens de pharmacie. Chaque question contient : un identifiant, la question, cinq options et l'ensemble des réponse(s) correcte(s).

Nous proposons d'utiliser pour ce défi le corpus FrenchMedMCQA. Le corpus FrenchMedMCQA est composé de 3 105 questions fermées, extrait d'annales d'examens de pharmacie en français, contenant pour chacune d'entre elles : un identifiant, la question, cinq options et l'ensemble de réponse(s) correcte(s). Pour cette nouvelle édition DEFT 2023, nous proposons deux tâches :

— Tâche principale : identifier automatiquement l'ensemble de réponses correctes parmi les cinq proposées pour une question donnée.
— Tâche annexe : identifier le nombre de réponses (entre 1 et 5) supposément justes pour une question donnée.

# Comités

## Comité scientifique

— Nathalie Camelin (LIUM, Le Mans Université)
— Liana Ermakova (HCTI, Université de Bretagne Occidentale)
— Benoit Favre (LIS, Aix-Marseille Université)
— Corinne Fredouille (LIA, Avignon Université)
— Pierre-Antoine Gourraud (CHU de Nantes)
— Natalia Grabar (STL, CNRS, Université de Lille)
— Cyril Grouin (LISN, CNRS, Université Paris-Saclay)
— Pierre Jourlin (LIA, Avignon Université)
— Fleur Mougin (ISPED, Université de Bordeaux)
— Aurélie Névéol (LISN, CNRS, Université Paris Saclay)
— Didier Schwab (LIG, Grenoble Alpes Université)
— Pierre Zweigenbaum (LISN, CNRS, Université Paris-Saclay)

## Comité d'organisation

— Adrien Bazoge (LS2N, Nantes Université)
— Béatrice Daille (LS2N, Nantes Université)
— Richard Dufour (LS2N, Nantes Université)
— Yanis Labrak (LIA, Avignon Université et Zenidoc)
— Emmanuel Morin (LS2N, Nantes Université)
— Mickael Rouvier (LIA, Avignon Université)

## Table des matières

*Actes de CORIA-TALN 2023*
*Paris, France, 5 au 9 juin 2023*

# Qui de DrBERT, Wikipédia ou Flan-T5 s'y connaît-il le plus en questions médicales ?

Clément Besnard[1] Mohamed Ettaleb[1] Christian Raymond[2] Nathalie Camelin[1]
(1) LIUM, Le Mans Université, France
(2) INSA Rennes, IRISA, France
[prénom].[nom]@univ-lemans.fr, [prénom].[nom]@irisa.fr

RÉSUMÉ

Cet article décrit la participation de l'équipe LIUM-IRISA à la campagne d'évaluation DEFT 2023. Notre équipe a participé à la tâche principale. Cette année, celle-ci porte sur la résolution automatique de questions à choix multiples dans le domaine médical. Nous avons mis en place plusieurs systèmes : un premier qui exploite une base de connaissances, un second interroge un modèle génératif en lui demandant de répondre directement aux questions et le dernier système combine un ensemble de descripteurs.

ABSTRACT

This paper describes the participation of the LIUM-IRISA team in the DEFT 2023 evaluation campaign. Our team participated in the main task, which this year consists of developing approaches for automatically answering medical multiple-choice questions. We have implemented several systems, the first use a knowledge base, a second use generative model-based system, and a final system combining a set of descriptors.

MOTS-CLÉS : base de connaissances, modèles neuronaux pré-entrainés, TF-IDF, corpus spécifique.

KEYWORDS: knowledge base, pre-trained neural models, TF-IDF, specific corpus.

## 1 Introduction

L'édition 2023 du Défi Fouille de Textes (DEFT) porte sur l'exploration d'un corpus de questions fermées en français. Elles proviennent d'annales d'examens de pharmacie et sont réunies dans le corpus *FrenchMedMCQA* (Labrak *et al.*, 2022). Les questions offrent la possibilité de choisir une ou plusieurs réponses parmi celles proposées. Le challenge consiste en la mise en place d'approches afin de répondre automatiquement à ces questions à choix multiples. Il s'agit donc d'associer, de choisir, un ensemble fini de réponses à une question énoncée en langage naturel dans le domaine spécifique du médical.

Une des difficultés de cette tâche provient en partie du fait que le vocabulaire à traiter est très spécifique. Potentiellement, les mots les plus caractéristiques du sens de la question et des réponses n'ont que très peu d'occurrences dans le corpus. Ainsi, appliquer les méthodes classiques de représentation vectorielles des mots par des modèles neuronaux pré-entraînés peut s'avérer plus complexe que pour des mots usuels présentant de nombreuses occurrences et de nombreux contextes différents

d'apparition.

Par ailleurs, cette tâche peut s'apparenter à un cas particulier de la tâche classique question/réponse pour laquelle il existe plusieurs systèmes à l'état de l'art. Ces systèmes combinent généralement un modèle qui recherche un *contexte* permettant de répondre à la question avec un modèle *Lecteur/Générateur* qui à partir du contexte et de la question extrait la réponse.

Comme décrit dans l'article de Weng (Weng, 2020), plusieurs méthodes existent afin de trouver un contexte, mais cela reste limité aux questions ouvertes sur un sujet précis. Il est plus difficile de trouver un contexte dans notre application aux questions fermées. Cependant, des systèmes comme T5 (Raffel *et al.*, 2020) ou GPT-3 (Brown *et al.*, 2020) peuvent s'affranchir de contextes grâce à leur nombre élevé de paramètres. Le nombre de paramètres (plusieurs milliards) que peuvent contenir les deux modèles génératifs précédents permet de stocker de l'information et des connaissances. Cela permet d'atteindre des performances similaires à plusieurs modèles de type BERT (Devlin *et al.*, 2019) associés à un système de recherche de contexte.

Pour cette édition, deux tâches sont proposées. La tâche principale consiste à identifier automatiquement et exactement quelles sont les réponses correctes. La tâche annexe se limite à identifier le nombre de réponses correctes sans indiquer précisément lesquelles sont correctes. Notre équipe a porté son attention sur la tâche principale, pour laquelle nous avons élaboré plusieurs systèmes de classification. Le premier système repose sur de la fouille dans une base de connaissances, le deuxième utilise un modèle génératif tandis que le dernier combine un ensemble de descripteurs.

La structure de l'article est la suivante : après une brève description du corpus et de la tâche dans la section 2, la section 3 présente les différents systèmes proposés. Les résultats des expériences menées sont exposés dans la section 4, suivis d'une synthèse des conclusions dans la section 5.

## 2 Analyse du corpus

Le corpus *FrenchMedMCQA* est une collection de 3 105 questions fermées en français provenant d'annales d'examens de pharmacie. Ce corpus est divisé en trois sous-ensembles, à savoir l'entraînement, le développement et le test. Les questions sont réparties de la manière suivante entre ces trois ensembles : 70% des questions pour l'entraînement, 10% pour le développement et 20% pour le test. Nous avons utilisé l'entrainement pour apprendre nos modèles et le corpus de test n'a été fourni que pendant la phase d'évaluation.

Chaque question est représentée par un identifiant et contient l'énoncé en langage naturel de la question, les cinq énoncés en langage naturel des options de réponse et l'ensemble des réponses correctes à la question. La question contient également le nombre de réponses correctes ainsi que le type de question : *simple* pour une seule réponse et *multiple* pour plusieurs réponses possibles.

On note ainsi une première différence entre les questions : soit on recherche *une seule* réponse, soit *plusieurs* réponses sont correctes.

On note également une deuxième différence, d'un point de vue de la sémantique :

— Soit la réponse recherchée est positivement liée à la question, comme par exemple dans l'Énoncé 1 (Table 1).

— Soit la réponse ne doit pas être vraie vis à vis de la question posée, comme par exemple dans

l'Énoncé 2 (Table 1).

| | |
|---|---|
| Énoncé 1 | *"Parmi les propositions suivantes, une seule est exacte. Laquelle ? La sérotonine est le (la) :"* |
| Énoncé 2 | *"Parmi les affirmations suivantes, une seule est fausse, indiquer laquelle : les particules alpha"* |
| Énoncé 3 | *"Parmi les propositions suivantes, laquelle (lesquelles) est (sont) exacte(s) ?"* |

TABLE 1 – Quelques exemples d'énoncés de questions

Pour finir, nous avons noté une dernière différence entre les questions. Certaines contiennent des informations sémantiquement pertinentes, comme dans l'Énoncé 1 (Table 1) où l'on comprend que la question traite de la *sérotonine*. D'autres en revanche, ne présentent pas de sujet précis dans leur intitulé, comme dans l'Énoncé 3 (Table 1).

Partant de ce constat, nous avons choisi d'appliquer un premier système permettant de déterminer le type de la question avant d'appliquer ensuite un de nos systèmes entraînés pour détecter les réponses à associer aux questions. L'ensemble de ces systèmes est présenté dans la section suivante.

# 3 Systèmes

Nous avons mis en place plusieurs systèmes et un méta-système par apprentissage qui tente de fusionner un ensemble de descripteurs. Ces systèmes ont pour objectif d'estimer la pertinence qu'une réponse candidate soit bonne.

Comme les énoncés demandent de trouver soit la/les bonnes réponse(s) soit le(s) intru(s), nous avons développé un détecteur de type d'énoncé qui va conditionner la manière dont nous allons utiliser le score de pertinence calculé par les systèmes pour associer les réponses recherchées à l'énoncé de la question. Tout ceci est détaillé dans la section suivante.

## 3.1 Détection du type de question et stratégie de réponse

Une première information à connaître afin de répondre à une question est d'identifier si l'association question/réponse est positive ou négative (rechercher la bonne ou la mauvaise réponse). Après avoir analysé les énoncés des questions, nous avons remarqué que cette information pouvait être obtenue à l'aide d'une simple expression régulière qui détecte la présence de certains mots clés. Ainsi, nous avons recherché les mots "sauf", "fausse", "fausses", "ne" "inexacte", "inexactes", "n'", qui sont apparus spécifiques aux questions qui demandent la/les mauvaises réponses.

Une deuxième information utile pour nos systèmes est d'identifier si l'on veut une seule ou bien plusieurs réponses. Cela peut difficilement être réalisé par une expression régulière, car rechercher la présence de certains mots n'est pas suffisant. Un modèle de type CamemBERT (Martin *et al.*, 2020a) avec l'ajout d'une couche de classification permet de réaliser cette tâche. Celui-ci va automatiquement extraire les spécificités de chaque type de question afin de prédire une des deux classes ('simple' ou 'multiple'). Nous avons obtenu un F1-score de 96,90 sur le corpus de développement pour cette tâche.

Un système à base de règles est utilisé afin de prédire la ou les réponses correctes. En entrée du système, on a un score pour chaque réponse, l'information qui nous indique si l'on veut les réponses correctes ou non ainsi que le type de question ('simple' ou 'multiple').

À partir des sorties d'un système, les règles sont les suivantes :

1. Si l'on veut une seule réponse et la réponse correcte, on retourne la réponse avec le plus grand score.
2. Si l'on veut une seule réponse et la réponse fausse, on retourne la réponse avec le plus petit score.
3. Si l'on veut plusieurs réponses et les réponses correctes, on retourne les réponses qui ont un score supérieur au premier quartile des scores.
4. Si l'on veut plusieurs réponses et les réponses fausses, on retourne les réponses qui ont un score inférieur au troisième quartile des scores.

Si l'on cherche une seule réponse, et que l'ensemble des scores est nul, ou bien que plusieurs réponses ont un score égal, le choix de la lettre est réalisé en fonction de la fréquence d'apparition de chaque lettre dans le corpus d'apprentissage. L'ordre défini est différent si l'on cherche la bonne ou bien la mauvaise réponse : **Bonne réponse** : 'd', 'c', 'b', 'a', 'e' ; **Mauvaise réponse** : 'd', 'c', 'e', 'b', 'a'

Lorsque l'on cherche plusieurs réponses et que le score pour chaque réponse est nul, la réponse multiple la plus fréquente dans le corpus d'apprentissage est retournée : **Réponse multiple** : 'bcd'.

## 3.2 Fouille dans une base de connaissances

Notre première approche consiste à vérifier si chacune des réponses candidates peut être associée à sa question en exploitant une base de connaissances.

### 3.2.1 Système FBC-ngram-rule

**Pré-traitements** L'ensemble des pré-traitements suivants ont été appliqués à l'ensemble des textes (énoncés de questions, réponses, titre et contenu des articles) :

— Suppression de la ponctuaction.

— Suppression des stopwords : Nous avons utilisé la liste de mots de la bibliothèque Spacy que nous avons enrichi des mots avec forte occurrence et sans apport sémantique dans l'énoncé des questions : '%exact%', 'proposition%', 'indique%', 'réponse%', 'fausse%', 'affirmation%', 'propos', 'vraie%', 'coche%', 'donner','trouve'.

Les réponses et le contenu des articles ont également subi :

— Modification de la liste de stopword : 'moins', 'plus', 'peu' ainsi que les chiffres ont été retirés de la liste de stopwords. En effet, ces informations apportent des nuances et des indications utiles à la compréhension sémantique de la réponse.

— Remplacement des caractères grecs par leur forme latine

— Remplacement des chiffres romains dans leur notation arabe

— Normalisation des caractères selon la norme NFKD afin d'éviter les variations potentielles

— Suppression des sauts de lignes

— Lemmatisation avec le modèle *fr_core_news_md* de l'outil Spacy [1]

**Construction de la base de connaissances** Deux approches ont été testées afin d'obtenir une liste d'articles pertinents :

1. Modèle Vectoriel : l'énoncé de la question et les titres de chacun des articles sont représentés par leur vecteur de poids tf-idf puis une similarité cosinus est appliquée pour extraire les articles les plus pertinents selon cette mesure.
2. API Wikipedia [2] : l'API de recherche Wikipédia est utilisée avec comme requête l'énoncé de la question. Les articles les plus pertinents sont alors extraits directement par l'API (recherche par présence de mots clés).

Les $n$ articles les plus pertinents vis à vis de l'énoncé de la question constituent alors la base de connaissance pour la question.

Nous avons ensuite choisi de compter le nombre d'occurrences des unigrammes et des bigrammes présents à la fois dans l'énoncé de la réponse et dans cette base de connaissances.

**Association de la base de connaissance aux questions** À partir du nombre d'unigrammes et de bigrammes de chaque réponse, un score est calculé selon la formule :

$$(2 * NbBigram + NbUnigram)/NbTokens \tag{1}$$

Nous avons choisi de donner un poids plus importants aux bigrammes car ceux-ci étaient beaucoup moins fréquents que les unigrammes. Par manque de temps, nous avons appliqué une formule simple avec un poids doublé. Une meilleure proposition consisterait à pondérer le nombre de bigrammes et d'unigrammes avec une formule tf-idf.

Selon cette formule, une réponse plus longue aura mécaniquement un score plus élevé, on normalise donc le score par le nombre de mots dans la réponse pré-traitée.

Les scores calculés sont ensuite utilisés comme décrit dans la section 3.1.

## 3.3 Système Flan-T5

Cette approche utilise *Flan-T5* (Chung *et al.*, 2022), un modèle de langage affiné sur plus de 1 000 tâches (traduction, résumé, classification, question/réponse). Ce type de modèle génère la séquence de mots la plus probable sachant la séquence de mots précédents donnée en entrée du système. L'idée est d'interroger Flan-T5 afin de comparer les réponses candidates à celles qu'il propose.

Il est possible de le spécialiser sur de nouvelles taches grâce à des instructions. Deux instructions ont été définies :

— Une première qui permet d'indiquer au modèle qu'il doit fournir une seule réponse. Le format d'entrée est le suivant : *Choisis la bonne réponse : **{question}** (A) **{réponse A}** (B) **{réponse B}** (C) **{réponse C}** (D) **{réponse D}** (E) **{réponse E}** context : **{contexte}***

1. https://spacy.io/
2. https://fr.wikipedia.org/w/index.php?search=&title=Spécial:Recherche

— Une seconde qui permet de choisir plusieurs réponses : *Choisis les bonnes réponses : **{question}** (A) **{réponse A}** (B) **{réponse B}** (C) **{réponse C}** (D) **{réponse D}** (E) **{réponse E}** context : **{contexte}***

Le contexte ajouté à la fin de l'entrée a été obtenu à l'aide d'un modèle DPR (Dense Passage Retrieval) en français (Karpukhin *et al.*, 2020). Ce type de modèle a été développé dans le cadre de la tâche de questions ouvertes. Il permet de sélectionner pour une question les contextes contenant la réponse. Ce modèle est basé sur une architecture dense avec deux encodeurs. Les données utilisées pour l'entraînement sont pour chaque question les contextes positifs, négatifs et fortement négatifs. Les contextes positifs sont ceux contenant la réponse à la question posée. Le but étant de réduire la distance entre les représentations vectorielles des paires de questions et des contextes positifs.

Les articles Wikipédia [3] en français ont été découpés dans un ensemble de passage de 100 mots chacun avec un recouvrement de 10 mots entre chaque passage du même article. Le passage le plus proche de notre question obtenu par similarité cosinus est utilisé comme contexte. Nous avons utilisé deux modèles déjà entraînés à cette tâche. Un premier modèle (etalab-ia/dpr-question_encoder-fr_qa-camembert) encode la question avec ses réponses dans un vecteur de taille 768. Un second modèle (etalab-ia/dpr-ctx_encoder-fr_qa-camembert) encode les passages. Les deux modèles permettant de réaliser les plongements des questions et des passages ont été réalisés par la branche Intelligence artificielle de l'Etalab [4].

En sortie, le système a été entraîné pour donner les lettres des réponses correctes de A à E séparées par des espaces.

Lors de l'entraînement, la version *Flan-T5-large* (780 millions de paramètres) du modèle a été utilisée sur 10 époques avec un taux d'apprentissage de $5e^{-5}$, un batch de taille 4, un *weight decay* de 0,01. L'aléatoire a été contrôlé en utilisant la graine 0. Après chaque époque, une version du modèle est sauvegardée, celle qui obtient les meilleurs résultats en termes d'*Exact Match ratio* sur l'ensemble de développement a été sélectionnée.

## 3.4 Méta-système

Un dernier système a été implémenté en utilisant un algorithme de boosting appris sur un ensemble de descripteurs.

Dans un premier temps, le corpus de questions est *binarisé* : un ensemble de couples énoncé_question/énoncé_réponseX est créé à partir de chaque question multiple (autant de couples que de réponses candidates). Ensuite, il s'agit de vérifier la validité de l'association question/réponseX grâce au méta-système.

Les descripteurs suivants ont été extraits :

1. **FBC-ngram-rule** : le nombre d'apparitions des unigrammes et bigrammes pour chaque réponse avec les 20 articles, les 5 articles et l'article le plus proche selon l'API de recherche Wikipédia.
2. **Flan-T5** : les scores pour chaque réponse à partir de la distribution de probabilité qui permet au modèle Flan-T5 de générer la lettre de la première réponse.
3. **Descripteur biomédical** : Nous avons utilisé le modèle DrBERT (Labrak *et al.*, 2023) qui est pré-entraîné sur des corpus de données médicales en français. Celui-ci permet de produire

3. https://dumps.wikimedia.org/frwiki/latest/frwiki-latest-pages-articles.xml.bz2
4. Département de la direction interministérielle du numérique

des vecteurs représentatifs d'un énoncé médical (token '[CLS]'). DrBERT est interrogé pour représenter énoncés des questions et énoncés des réponses. Une similarité cosinus est ensuite calculée pour chaque couple question/réponseX.

4. **Enrichissement de l'énoncé de la question** : Un score de similarité (cosinus) est calculé entre chaque réponse et la question enrichie. Pour enrichir la question, nous procédons à deux étapes : 1) Uniquement les 5 mots ayant le plus haut score de tf-idf sont conservés dans l'énoncé de la question ; 2) les 10 noms (NOUN dans Spacy) les plus fréquents dans les 10 pages les plus pertinentes selon l'API Wikipedia sont ajoutés à cet énoncé réduit.
5. **Utilisation d'un système de question/réponse** : L'énoncé de la question est à nouveau réduit à ces 5 mots les plus pertinents et les 10 articles les plus pertinents sont à nouveau considéré. Le modèle francophone de question/réponse *Camembert-base-squadFR-fquad-piaf*[5] est utilisé pour extraire la réponse des articles pertinents en considérant l'énoncé réduit de la question comme requête. Un score de similarité cosinus est ensuite calculé entre la réponse du modèle et la réponse candidate considérée.
6. **Utilisation d'un seul article pertinent** : Le dernier descripteur est obtenu comme le précédent mais en ne considérant qu'un seul article pertinent au lieu de 10.

Tous ces descripteurs ont été utilisés en entrée d'un algorithme de Gradient Boosting afin d'évaluer la validité d'association de chacun de nos couples question/réponseX.

# 4 Expériences et résultats

Les deux métriques utilisées afin d'évaluer la performance des systèmes sont le *Hamming score* (taux de bonnes réponses parmi l'ensemble des hypothèses et références) et l'*Exact Match Ratio* (taux de réponses parfaitement justes).

La table 2 montre les résultats du système *FBC-ngram-rule* selon les 2 méthodes proposées et en considérant plusieurs valeurs $n$.

| | | 1 article | | 5 articles | | 20 articles | |
|---|---|---|---|---|---|---|---|
| | | Hamming | EMR | Hamming | EMR | Hamming | EMR |
| API Wikipédia | Dev | 36,43 | 17,31 | 38,74 | 18,91 | 38,35 | 17,95 |
| | Test | - | - | 36,72 | 17,85 | - | - |
| Modèle Vectoriel | Dev | 35,60 | 17,31 | 34,78 | 16,35 | - | - |

TABLE 2 – Résultats système 1 : FBC-ngram-rule

L'API de Wikipédia est la meilleure méthode pour extraire les documents pertinents. De plus, on note qu'il est intéressant de considérer 5 articles. En revanche, en considérer beaucoup plus comme 20 n'apporte pas de gain significatif. En considérant ce système et le corpus de développement, nous avons observé ceci :

5. Il utilise comme base CamemBERT (Martin *et al.*, 2020b) fine-tuné sur la combinaison de trois jeux de données francophones de questions-réponses : PIAFv1.1 (Keraron *et al.*, 2020), FQuADv1.0 (d'Hoffschmidt *et al.*, 2020), SQuAD-FR (Kabbadj, 2018).

— Recherche d'une réponse *simple* : En considérant les 312 questions du corpus de développement, 6 d'entres elles obtiennent un descripteur nombre d'unigramme et de bigramme nul. 17 questions obtiennent des scores égaux pour plusieurs réponses (dont 11 lorsque l'on recherche une réponse incorrecte).

— Recherche d'une réponse *multiple* : Un ensemble de scores nuls apparait pour 3 questions.

Notons que les résultats sur le test avec ce système restent moins performants que ceux obtenus sur le corpus de développement. Cependant, l'utilisation de notre système assez léger permet d'atteindre des résultats similaires à des plus gros modèles de type BERT ou RoBERTa (Labrak *et al.*, 2022).

Les résultats du système Flan-T5 sont présentés dans la Table 3.

| | Wiki passages DPR | | Sans contexte | |
|---|---|---|---|---|
| | Hamming | EMR | Hamming | EMR |
| Dev | 44,88 | 25,64 | 45,04 | 25,32 |
| Test | 43,24 | 22,19 | - | - |

TABLE 3 – Résultats Flan-T5

Les résultats obtenus sont bien meilleurs qu'avec le premier système. On remarque que l'ajout d'un contexte ne permet pas d'améliorer significativement les résultats. Il est difficile d'associer un contexte utile pour répondre aux questions avec seulement des passages de 100 mots. Par ailleurs, on observe à nouveau une baisse des résultats sur le test par rapport au développement, notamment en termes d'*Exact Match Ratio*. Cela peut s'expliquer par un nombre de questions *multiples* plus important.

Après la phase de test, nous avons réappris le modèle Flan-T5 sur l'ensemble apprentissage+développement sur 3 époques (meilleur paramètre lors de la phase de dev), nous avons améliorer légèrement les résultats sur le Test avec un *Hamming* de **45,84** et un *Exact Match Ratio* de **23,15**.

Les résultats de notre dernier système sont présentés en Table 4.

| Dev | | Test | |
|---|---|---|---|
| Hamming | EMR | Hamming | EMR |
| 44,42 | 24,36 | 35,47 | 18,49 |

TABLE 4 – Méta-système qui combine toutes les features

Les résultats indiquent que le système a obtenu des performances modérées. Il a réussi à identifier correctement la réponse pour environ la moitié des paires de questions et de réponses sur les données de développement (*Hamming* de **44,42**), mais n'a pas été en mesure de trouver toutes les réponses pour la plupart des questions sur les données de test (*Hamming* de **35,47**). Il est essentiel d'analyser les raisons derrière ces résultats et de cibler les points faibles afin d'améliorer les performances. Plusieurs facteurs, tels que la qualité et la taille des données d'entraînement, la complexité des modèles et des algorithmes utilisés, peuvent tous influencer les performances du système. Cependant, pour améliorer les performances et obtenir des correspondances exactes, nous devrons aborder ces facteurs limitants de manière approfondie. L'amélioration de la mesure de similarité entre les questions et les réponses pourrait être une piste à explorer. De plus, une analyse plus rigoureuse de la sélection des caractéristiques pourrait permettre d'identifier des aspects plus pertinents et discriminants pour améliorer la précision des correspondances.

# 5 Conclusion

En conclusion, nous avons proposé trois systèmes. Le premier système est basé sur une approche avec une base de connaissances qui utilise la similarité cosinus entre vecteurs de poids TF-IDF et l'API de recherche Wikipédia pour identifier les articles pertinents. Les réponses sont ensuite classées en fonction du nombre d'unigrammes et de bigrammes qui se trouvent dans ces articles. Le système utilise ensuite des règles pour prédire la ou les réponses correctes. Le deuxième système utilise Flan-T5, un modèle de langage pré-entrainé sur plus de 1 000 tâches, pour générer la séquence de mots la plus probable pour chaque réponse. Le modèle est spécialisé pour les tâches de question à choix multiples en fournissant des instructions pour fournir une seule ou plusieurs réponses. Le troisième système utilise une approche de classification. Le modèle est entraîné sur un ensemble de descripteurs pour prédire la ou les réponses correctes à une question donnée. Après avoir étudié plusieurs architectures, c'est le modèle Flan-T5 qui se distingue le plus. Le nombre important de paramètres lui permet d'obtenir de meilleurs résultats. La base de connaissances qu'est Wikipédia nous permet d'atteindre des résultats équivalents aux systèmes *baseline* mais les règles appliquées rencontrent des limites comme notamment la gestion des questions *multiples* et les négations dans les réponses. Pour finir, les descripteurs calculés à partir de DrBERT ne se sont pas montrés à la hauteur de nos espérances. En effet, le classement des descripteurs par l'algorithme de boosting de notre méta-modèle a montré que le descripteur biomédical était le moins pertinent de tous.

# Références


BROWN T. B., MANN B., RYDER N., SUBBIAH M., KAPLAN J., DHARIWAL P., NEELAKANTAN A., SHYAM P., SASTRY G., ASKELL A., AGARWAL S., HERBERT-VOSS A., KRUEGER G., HENIGHAN T., CHILD R., RAMESH A., ZIEGLER D. M., WU J., WINTER C., HESSE C., CHEN M., SIGLER E., LITWIN M., GRAY S., CHESS B., CLARK J., BERNER C., MCCANDLISH S., RADFORD A., SUTSKEVER I. & AMODEI D. (2020). Language models are few-shot learners.

CHUNG H. W., HOU L., LONGPRE S., ZOPH B., TAY Y., FEDUS W., LI E., WANG X., DEHGHANI M., BRAHMA S., WEBSON A., GU S. S., DAI Z., SUZGUN M., CHEN X., CHOWDHERY A., NARANG S., MISHRA G., YU A., ZHAO V., HUANG Y., DAI A., YU H., PETROV S., CHI E. H., DEAN J., DEVLIN J., ROBERTS A., ZHOU D., LE Q. V. & WEI J. (2022). Scaling instruction-finetuned language models. DOI : 10.48550/ARXIV.2210.11416.

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

D'HOFFSCHMIDT M., VIDAL M., BELBLIDIA W., BRENDL'E T. & HEINRICH Q. (2020). Fquad : French question answering dataset. *ArXiv*, **abs/2002.06071**.

KABBADJ A. (2018). Something new in french text mining and information extraction (universal chatbot) : Largest qa french training dataset (110 000+). [Online ; posted 11-November-2018].

KARPUKHIN V., OGUZ B., MIN S., LEWIS P., WU L., EDUNOV S., CHEN D. & YIH W.-T. (2020). Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 6769–6781, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.emnlp-main.550.

KERARON R., LANCRENON G., BRAS M., ALLARY F., MOYSE G., SCIALOM T., SORIANO-MORALES E. & STAIANO J. (2020). Project PIAF : building a native french question-answering dataset. In *LREC*, p. 5481–5490 : European Language Resources Association.

LABRAK Y., BAZOGE A., DUFOUR R., DAILLE B., GOURRAUD P.-A., MORIN E. & ROUVIER M. (2022). FrenchMedMCQA : A French multiple-choice question answering dataset for medical domain. In *Proceedings of the 13th International Workshop on Health Text Mining and Information Analysis (LOUHI)*, p. 41–46, Abu Dhabi, United Arab Emirates (Hybrid) : Association for Computational Linguistics.

LABRAK Y., BAZOGE A., DUFOUR R., ROUVIER M., MORIN E., DAILLE B. & GOURRAUD P.-A. (2023). Drbert : A robust pre-trained model in french for biomedical and clinical domains.

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

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

RAFFEL C., SHAZEER N., ROBERTS A., LEE K., NARANG S., MATENA M., ZHOU Y., LI W. & LIU P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer.

WENG L. (2020). How to build an open-domain question answering system ? *lilianweng.github.io*.

*Actes de CORIA-TALN 2023*
*Paris, France, 5 au 9 juin 2023*

# SPQR@Deft2023: Résolution automatique de QCM médicaux à partir de corpus de domaine et de mesures de similarité

Julien Bezançon[1,2] Toufik Boubehziz[1] Corina Chutaux[1] Oumaima Zine[1]
Laurie Acensio[1] Caroline Koudoro-Parfait[1,4] Andrea Briglia[1,3] Gaël Lejeune[1,2]
(1) STIH, 28 rue Serpente, 75006 Paris, France
(2) CERES, 28 rue Serpente, 75006 Paris, France
(3) UMR 1253 iBrain, 10 Boulevard Tonnellé, 37000 Tours, France
(4) Obtic, SCAI, 4 Pl. Jussieu, 75005 Paris, France
prenom.nom@sorbonne-universite.fr

RÉSUMÉ

Nous présentons le travail de SPQR (Sorbonne Question-Réponses) au DÉfi Fouille de Textes 2023 sur la réponse automatique à des questionnaires à choix multiples dans le domaine de la pharmacologie. Nous proposons une approche fondée sur la constitution de corpus de spécialité et la recherche de phrases similaires entre ces corpus et les différentes réponses possibles à une question. Nous calculons une similarité cosinus sur des vecteurs en n-grammes de caractères pour déterminer les bonnes réponses. Cette approche a obtenu un score maximal en Hamming de 0,249 sur les données de test (0,305 sur le dev) et de 0,0997 en Exact Match Ratio (0,16 sur le dev).

ABSTRACT

**SPQR@Deft2023 : Answering automatically to MCQ in the medical domain with similarity measures and domain-specific corpora**

We exhibit the approach of the SPQR team in the 2023 French Text Mining Challenge (DEFT). This challenge focused on automatically answering Multiple Choice Questions (MCQ) in the pharmacology domain. We proposed an approach that takes advantage of domain-specific corpora in order to find similarities between possible answers and sentences in the corpora. We compute a cosine similarity on character n-gram vectors to compare them. The best scores we obtained were 0,294 for the Hamming score on the test set (0,305 on the dev set) and 0,997 for the Exact Match ratio (0,16 on the dev set).

MOTS-CLÉS : QCM, FrenchMedMCQA, pharmacologie, similarité, n-grammes de caractères, systèmes de question-réponse.

KEYWORDS: MCQ, FrenchMedMCQA, pharmacology, similarity, character n-grams, question-answering systems.

## 1 Introduction

Cette nouvelle édition du Défi Fouille de Textes (DEFT) porte sur le corpus FrenchMedMCQA (Labrak *et al.*, 2022a) composé de 3 105 questions fermées, issues des annales d'examens de pharmacie en français. Chaque question possède un identifiant, cinq options de réponses et les corrections. Deux tâches sont proposées dans cette édition du DEFT(Labrak *et al.*, 2022b). La tâche principale proposée consiste en l'identification de réponses correctes parmi cinq réponses possibles

proposées. La tâche annexe propose d'identifier le nombre de réponses (entre 1 et 5) potentiellement correctes pour une question. Nous avons participé aux deux tâches. Le corpus FrenchMedMCQA est découpé en 3 sous-parties. Le corpus d'entraînement (70 % du corpus total), le corpus de développement (10 %) et le corpus de test (20 %). Il était attendu que les performances des systèmes soient évaluées avec la métrique *Exact Match Ratio*, ou taux de réponses parfaitement juste, et le *Hamming Score*, taux de bonnes réponses parmi l'ensemble des réponses données par le système.

Nous faisons l'hypothèse que l'identification des bonnes réponses à une question de manière automatique s'apparente à une recherche de similarité entre les réponses possibles et des données textuelles de référence sur le sujet. Plus précisément, qu'il s'agit de chercher comment nous pouvons montrer que les bonnes réponses à une question sont similaires à des phrases trouvées dans un texte de référence. Pour démarrer nos expériences, nous avons exploré différentes techniques pour constituer des corpus de référence, que nous présentons dans l'article. Nous nous sommes appuyés sur la détection de technolectes biomédicaux dans les questions et les réponses du corpus d'entraînement pour les lier soit avec des définitions issues du manuel Merck (Beers *et al.*, 2008) soit pour interroger l'API OpenAI [1] (Brown *et al.*, 2020). Enfin, nous avons constitué un corpus de manière intrinsèque en transformant les questions et les réponses du jeu d'entraînement en énoncés définitoires. Cet article débute par un état de l'art sur l'évaluation des systèmes de questions-réponses à choix multiple (Section 2) puis nous proposons une description du jeu de données FrenchMed MCQA dans la Section 3. La construction des corpus qui servent de base à nos méthodes est présentée dans la Section 4, nous décrivons les différentes méthodes développées dans la Section 5 puis nous présentons nos résultats et des éléments de discussion dans la Section 6.

## 2 Méthodes pour les systèmes de question-réponse

Les systèmes de questions-réponses à choix multiple (ou *Multiple Choice Question Answering*) visent à sélectionner la ou les bonne(s) réponse(s) parmi un ensemble d'options données à une question. Cette tâche est particulièrement utilisée dans le domaine de l'éducation (Touissi *et al.*, 2022) (Soares *et al.*, 2021). Ce défi du traitement du langage naturel (TAL) est un problème complexe du fait qu'il nécessite *a priori* une compréhension fine du langage utilisé pour analyser la question d'une part, et, d'autre part, d'appliquer des techniques de résolution de problèmes différents pour sélectionner la ou les bonne(s) réponse(s) parmi plusieurs options. Ainsi, les systèmes de questions-réponses peuvent fonctionner en analysant la typologie des questions (factuelle, liste, booléenne, définition) selon la catégorisation proposée par (Falco, 2012) et en fonction des réponses qu'ils attendent. Concernant la détection des bonnes réponses, les modèles et techniques sont divers : des méthodes de raisonnement telles que le raisonnement multi-sauts (Clark *et al.*, 2018) ou encore le raisonnement logique (Liu & Lee, 2020; Yu *et al.*, 2020; Baggetto *et al.*, 2022). On trouve aussi dans la littérature des travaux s'appuyant sur des sources sémantiques incluant des connaissances généralistes (ou de sens commun) (Talmor *et al.*, 2018; Mihaylov *et al.*, 2018) et des connaissances spécialisées (ou scientifiques) (Clark *et al.*, 2018; Huang *et al.*, 2019). Il est aussi possible d'adopter des méthodes de déduction par l'erreur, sur le principe de l'élimination des réponses, ces méthodes impliquent la sélection de toutes les mauvaises options afin de faciliter la détermination de la bonne réponse. Cette stratégie utilisée par (Kim & Fung, 2020) vise à entraîner le modèle en imitant la stratégie où le répondant exclut intuitivement les options improbables.

1. https ://openai.com/blog/openai-api

Sans être exhaustif, cet état de l'art montre que les modèles existants se spécifient selon la tâche d'application et ne sont pas directement applicables dans le cadre de notre étude. Le contexte de l'étude induit une complexité linguistique qui se caractérise principalement par la compréhension de la question/réponse qui est relativement courte mais fortement spécialisée. La section suivante de cet article propose une description du jeu de données de questions-réponses du DEFT.

# 3 Description et analyse du jeu de données

Le corpus FrenchMed MCQA est un ensemble de questions à choix multiple extraites d'examens de pharmacologie français. Les questions et les réponses ont été créées manuellement par des experts médicaux et destinés à des étudiants de niveau universitaire.

## 3.1 Considérations générales sur le corpus

Le jeu de données fourni pour le DEFT 2023 se compose de 2 174 questions pour les données d'entraînement et de 312 questions pour les données de développement. Le tableau 1 présente la distribution des données du corpus d'entraînement et de développement. Nous observons par ailleurs que la majorité des questions comportent 1 à 3 bonnes réponses tandis que les questions ayant l'intégralité des bonnes réponses proposées sont minoritaires. Il est à préciser que le nombre de bonnes réponses n'est pas explicitement indiqué dans la formulation dans la question.

| | **Apprentissage** | **Développement** |
|---|---|---|
| Nombre total de mots | 130 290 | 20 428 |
| Nombre total de questions | 2 172 | 313 |
| Nombre total de réponses correctes | 5 159 | 597 |
| Moyenne de réponses correctes par question | 2,37 | 1,9 |

TABLE 1 – Description de données du Deft 2023

Chaque question est associée à un identifiant, cinq réponses possibles étiquetées de A à E, la ou les réponse(s) correcte(s) et le nombre de réponses correctes. Le tableau 2 représente un exemple des questions et de réponses du corpus d'entraînement.

| **ID** | **Question** | **Réponses** | **Bonnes Réponses** |
|---|---|---|---|
| ... | Parmi les substances suivantes, une seule ne traverse pas la barrière placentaire. Laquelle ? | a. Dicoumarine<br>b. Glucose<br>c. Héparine<br>d. Tétracycline<br>e. Amplicilline | c |

TABLE 2 – Exemple de question extraite du corpus d'entraînement

## 3.2 Classification des questions

Les questions du corpus comportent trois parties $P_i$ pour $i = 1, 2, 3$. La séparation entre ces parties est généralement faite avec des virgules. Cela nous donne la forme générale suivante :

$$[P_1][P_2][P_3] \tag{1}$$

— $[P_1]$ est l'entête de la question qui commence souvent par : Parmi les propositions suivantes, cocher la (les) proposition(s) exacte(s), quelle(s) affirmation(s) est(sont) exacte(s), etc.
— $[P_2]$ est donne le type des questions attendues (exacte, inexacte, vraie, fausse) et éventuellement des précisions sur le nombre des réponses possibles.
— $[P_3]$ comporte le contexte (l'information) pharmacologique.

Nous pouvons toutefois avoir d'autres variants à la forme 1 :
— $[P_1 + P_2][P_3]$ ; $[P_1][P_2 + P_3]$ ; $[P_3]$
Parfois, une partie $[P_i]$, pour $i = 1 : 3$, est divisée avec des ponctuations : "." "," ":" "?". Nous pouvons alors classer les questions du corpus d'apprentissage en 5 variantes majeures :

$$\begin{cases} [\mathbf{A} + \cdots][\mathbf{NEG/TF/1N/IMP}][\mathbf{MED}] : \\ [\mathbf{A} + \cdots + \mathbf{MED}][\mathbf{NEG/TF/1N/IMP}][\mathbf{MED}] : \\ [\mathbf{A} + \cdots + \mathbf{NEG/TF/1N/IMP}][\mathbf{MED}] : \\ [\mathbf{A} + \cdots + \mathbf{MED}][\mathbf{NEG/TF/1N/IMP}] : \\ [\mathbf{MED}] : \end{cases} \tag{2}$$

Ces variantes sont composées des éléments récurrents suivants :
— **A** : termes utilisés dans la formulation des questions (parmi, donner, indiquer, cocher, on observe, quelle(s), laquelle(s), sélectionner)
— **TF** : termes utilisés dans l'affirmation ou la négation ((in)exacte(s), juste(s), fausse(s), vraie(s))
— **1N** : indication précise sur le nombre des réponses possibles
— **IMP** : indication implicite sur le nombre des réponses possibles
— **MED** : information médicale
— **NEG** : négation
En termes de structure, on peut trouver :

1. Des questions qui peuvent être de type interrogatif :
   **Exemple :** Parmi les principes actifs suivants, lequel (lesquels) est-il contre-indiqué ou fortement déconseillé d'associer à l'aspirine ?
2. Des questions peuvent être de type assertif :
   **Exemple :** Parmi les propositions suivantes, indiquer celle qui est exacte. Le crack est une forme : . . .")
3. Des cas où le nombre exact de bonnes réponses n'est pas explicite dans la formulation de la question. Néanmoins, certaines formulations indiquent qu'une réponse unique est attendue :
   **Exemple :** Parmi les propositions suivantes, une seule est fausse. Indiquez laquelle ? La rifampicine : (. . .)

On peut aussi classer les questions d'une autre manière en cherchant à répertorier les questions en fonction de la ponctuation utilisée en fin d'énoncé (Tableau 3). Ces deux manières de décrire les énoncés des corpus d'entraînement nous ont permis d'aboutir à la construction du Corpus *feedback* qui, aux côtés de trois autres corpus détaillés et explicités dans la partie 4, servira à la constitution du corpus de référence final.

| | Terminaison de la question | Nombre de questions |
|---|---|---|
| " ?" | (point d'interrogation) | 1147 |
| " :" | (deux points) | 766 |
| " " | (pas de ponctuation finale) | 118 |
| ... | (ellipse) | 78 |
| "." | (point) | 52 |
| "," | (virgule) | 10 |

TABLE 3 – Nombre de questions par type de terminaison

# 4 Construction des corpus de référence

Afin de procéder à la résolution automatique des questions, nous constituons cinq corpus de référence. Ces corpus ont été constitués à partir de diverses ressources (jeu de données d'entraînement du défi, livres numérisés, sites web, ...). L'enjeu est de tester plusieurs ressources médicales afin de déterminer lesquelles sont les plus adaptées à la résolution de la tâche.

**Corpus FEEDBACK** Nous sommes partis du postulat que les questions et les réponses s'entrecroisaient. Nous avons donc procédé à un test de similarité sur le jeu d'entraînement qui a permis une première validation de l'hypothèse. Ainsi, nous avons réfléchi à la construction d'un corpus à partir du jeu d'entraînement fourni et nous avons formulé, à partir des phrases interrogatives et des réponses correctes, des propositions assertives. Nous avons pu répartir les questions en deux catégories principales : questions de forme affirmatives et questions de forme véritablement interrogative. Le jeu d'entraînement contient 1 147 questions interrogatives et 1 024 affirmatives. Nous avons déterminé trois transformations pour la construction du corpus FEEDBACK en nous appuyant sur cette typologie :

1. Concaténation des questions et des réponses correctes (la réponse complète la question) :
   *Parmi les propositions suivantes, indiquer celle qui est exacte. Le crack est une forme :*
   **Réponse** : *De cocaïne*
   **Phrase** : Le crack est une forme *de cocaïne*.
2. Transformation des interrogatives en subordonnées relatives :
   *Dans une des conditions suivantes, l'antigène est tolérogène, laquelle ?*
   **Réponse** : *Administration par voie intra-veineuse*
   **Phrase** : Lors de *l'administration par voie intraveineuse* l'antigène est tolérogène.
3. La paraphrase :
   *Quels sont les toxiques dépresseurs du système nerveux central ?*
   **Réponse** : 1. *Le cannabis*, 2. *Les opiacés*
   **Phrase** : *Le cannabis et les opiacés* sont les toxiques dépresseurs du système nerveux central.

Cette constitution itérative d'un corpus de référence a été rendue possible par les stratégies de classification des questions que nous avons élaborées. Dans ce contexte, il a été possible de concaténer automatiquement, en appliquant des conditions, les phrases se terminant par " :", ".", "dernier mot", "..." et ",", ce qui a représenté 1024 questions sur un total de 2171. Une classification des phrases d'un point de vue syntaxique nous a permis de les rapprocher au niveau de la forme et de traiter la transformation des interrogatives en subordonnées relatives. En ce qui concerne la paraphrase,

la transformation a été plus difficile à réaliser. Nous nous sommes appuyés sur une classification syntaxique et lexicale, effectuée en amont, pour établir un patron et procéder à la modification.

**Corpus FEEDBACK BIS** Bien que l'objectif de ce corpus soit similaire à celui du corpus FEEDBACK, qui consiste à créer un corpus à partir du jeu de données d'entraînement, l'approche adoptée présente quelques nuances. Pour créer ce corpus, on va traiter deux types de questions : celles qui comportent les termes 'exacte(s)' ou 'inexacte(s)'. Elles représentent plus de 46% des questions du jeu d'entraînement. À partir de ces questions et de leurs réponses, nous avons formulé des propositions assertives. Pour construire ce corpus, nous avons identifié deux transformations essentielles :

1. Transformation des questions en assertions avec des expressions régulières pour les deux classes (cf. Section 3.2). Les phrases ainsi obtenues sont purgées de la ponctuation ainsi que les terms spécifiques aux questions ("quelles", "lesquelles", "indiquer", "celles", etc.)
2. Concaténation de chaque assertion obtenue à l'étape 1 avec les réponses correctes (pour la classe "exacte") ou les réponses incorrectes (pour la classe "inexacte"), par exemple :

   *Parmi les affirmations suivantes, indiquer la (les) affirmation(s) inexacte(s). La spectrofluorimétrie est une technique :*
   a : Très sensible,
   b : Applicable à toutes les molécules organiques,
   c : Utilisable pour des dosages quantitatifs,
   d : Ne nécessitant pas de gamme d'étalonnage,
   e : Effectuée sur des solutions congelées à basse température.
   **Réponses correctes** : b, d, e
   **Les assertions ajoutées au corpus** :
   a : La spectrofluorimétrie est une technique très sensible,
   c : La spectrofluorimétrie est une technique utilisable pour des dosages quantitatifs.

Contrairement à FEEDBACK, ici chaque question se voit attribuer une assertion distincte pour chaque réponse correcte. Ainsi, nous évitons de regrouper toutes les réponses sous une seule phrase assertive, comme illustré dans l'exemple ci-dessus. Ce corpus comporte 2 383 phrases.

**Corpus CHATGPT** Afin d'augmenter la taille du corpus FEEDBACK et ainsi élargir la couverture à des termes médicaux absents des données d'entraînement, nous avons choisi de recourir à l'API OpenAI (Brown *et al.*, 2020). Chat GPT a été utilisé pour interroger le modèle GPT-3.5-turbo, une version améliorée du célèbre modèle de langue profond GPT-3, afin de générer des complétions de phrases à partir des assertions du corpus FEEDBACK. Chaque assertion est envoyée au modèle sous forme de requête CURL via l'API qui renvoie un ou plusieurs paragraphes qui viennent compléter l'assertion. Le corpus ainsi généré est présenté sous forme de paires "assertions" + "complétion". Cette approche nous a permis d'élargir la couverture du vocabulaire pharmaceutique lié aux thématiques abordées dans le jeu de données, tout en économisant le temps de recherche de textes pharmaceutiques aléatoires et les coûts liés à la création d'un corpus de phrases médicales. L'exemple suivant présente un extrait du corpus résultant sous forme d'une paire Assertion/complétion :

> "Le crack est une forme de cocaïne" : "le crack est une forme de cocaïne qui est transformée en une substance solide et cristalline, généralement fumée plutôt que sniffée. Le crack est considéré comme une drogue très addictive et dangereuse en raison de son effet rapide et intense sur le système nerveux central."

**Corpus MERCK** Le corpus Merck se base sur le manuel Merck (Beers *et al.*, 2008). Il s'agit d'un manuel de médecine disponible en ligne et dont nous avons récupéré le contenu. Nous l'avons ensuite découpé en phrases afin de constituer un corpus de référence complémentaire de 66 845 phrases.

**Corpus ALL IN ONE** Ce corpus est obtenu par concaténation de tous les corpus décrits précédemment. La création de ce corpus va permettre de nourrir notre système de similarité avec des termes techniques propres à la médecine et à la pharmacologie, des technolectes médicaux ainsi que du contexte.

# 5 Méthodes de résolution automatique

Nous avons procédé à la détection des technolectes médicaux à l'aide d'une similarité *cosinus* entre les phrases de nos corpus de référence et les questions/réponses du défi.

## 5.1 Méthode de détection des termes médicaux

La détection des technolectes médicaux est une étape essentielle dans la résolution automatique des questions du FrenchMedMCQA. Elle permet d'extraire les termes médicaux des questions et de la base de données de référence. Pour réaliser cette tâche, nous avons développé une approche qui consiste à stocker les mots des questions du corpus d'entraînement considérés comme des technolectes médicaux en utilisant une liste préalablement ajustée de mots courants de la langue française. Après avoir dépassé un certain seuil d'apprentissage, la liste ajustée est remplacée par la liste stockée afin de récupérer un maximum de technolectes récurrents dans le corpus d'entraînement. Cela assure le bon fonctionnement de la méthode de résolution des questions présentée dans la section suivante. Afin d'associer chaque question d'un questionnaire donné (jeu de données cible) avec la (les) bonne(s) réponse(s) correspondante(s), nous proposons une approche qui repose sur l'utilisation d'un corpus de référence (voir Section 4) et le jeu de données cible lui-même. La méthode comprend trois phases :

1. Extraction des technolectes médicaux de chaque phrase du corpus de référence et de chaque paire question/réponse
2. Recherche des réponses correctes à l'aide de mesures de similarité
3. Évaluation des résultats obtenus

**Extraction des technolectes** L'extraction des termes médicaux s'opère à la fois sur le corpus de référence et le jeu de données ciblé. Une ressource additionnelle de mots courants en français (décrite dans la Section 5.1) a été créée pour mieux détecter les technolectes. Pour le corpus de référence, nous retirons de chaque phrase tokénisée tous ces mots courants, qui ne sont pas *a priori* des technolectes médicaux. Chaque phrase du corpus de référence devient ainsi une séquence de termes du domaine. Pour le jeu de données ciblé, nous commençons par concaténer les questions avec chacune [Q] de leurs réponses possibles $[R_j]$ pour $j = 1 : 5$.

Nous donnons ici un exemple de traitement d'une question à résoudre :

$$\begin{cases} [Q] \ : \textbf{Parmi les éléments suivants, quel est celui qui n'entre pas dans l'uréogenèse ?} \\ [R_1] \ : \textit{CO2} \\ [R_2] \ : \textit{NH3} \\ [R_3] \ : \textit{Valine} \\ [R_4] \ : \textit{ATP} \\ [R_5] \ : \textit{Ornithine} \end{cases} \tag{3}$$

À partir de cette question et des réponses possibles, on obtient :

— [Q] + [$R_1$], [Q] + [$R_2$], [Q] + [$R_3$] ...

Une fois ce travail effectué, nous tokénisons chaque paire question/réponse concaténée et nous filtrons avec la même méthode les mots courants du français.

**Mesures de similarité** La désignation des réponses correctes pour une question du jeu de données cible est faite à l'aide d'une mesure de similarité de manière à vérifier l'existence d'une phrase similaire du corpus de référence. Si pour une paire question [Q] / réponse ([$R_k$] où $k$ est la $k^{\text{ème}}$ réponse possible pour $k \in \{1, 2, 3, 4, 5\}$ ) une phrase similaire existe dans le corpus de référence, nous ajoutons la réponse de cette paire dans la liste des réponses potentiellement correctes à la question. Le test de similarité est réalisé en faisant varier différents paramètres :

— Type de vectorisation (mots, n-grammes de caractères libres ou à l'intérieur des mots ou mots)
— Taille des n-grammes (bi-grammes, tri-grammes, ...)
— Seuil minimal de similarité pour sélectionner la réponse résultat (0.6, 0.7, 0.8, ...)
— Mesure de similarité (Similarités Cosinus Bray-Curtis, Indices de Dice et de Jaccard)

Nous commençons par vectoriser l'ensemble du corpus de référence à l'aide de la librairie SKLEARN avec une vectorisation en fréquence absolue (COUNTVECTORIZER) et avec une pondération Tf-IDf (TFIDFVECTORIZER). Pour chaque paire question/réponse, nous la vectorisons également selon les mêmes paramètres et l'ajoutons à la matrice de vecteurs du corpus de référence. Nous calculons ensuite la similarité *cosinus*, les autres mesures testées ayant des résultats notablement moindres, au contraire de ce qui a été observé dans (Buscaldi *et al.*, 2020). Afin de vérifier que pour une paire question/réponse, il existe une phrase similaire dans le corpus de référence, nous mettons en place différentes méthodes :

— Recherche par seuil (BYSEUIL) : Toutes les réponses dont la similarité est supérieur à un seuil déterminé sont sélectionnées.
— Recherche par le maximum (BYMAX) : Nous sélectionnons simplement la réponse disposant de la plus grande similarité.
— Fusion des deux méthodes précédentes (BYFUSION) : Nous gardons la réponse avec la similarité auxquelles nous ajoutons toutes les réponses dont la similarité dépasse le seuil.

Pour chaque question du jeu de données cible, nous obtenons une liste de bonnes réponses potentielles selon l'une des trois méthodes décrites. Notons que la méthode BYSEUIL peut ne pas sélectionner de réponse dans le cas où aucune réponse n'a une similarité supérieure au seuil choisi. Cette méthode s'étant avérée nettement moins performante, nous n'en présenterons pas les résultats ici. Nous avons testé toutes les combinaisons des paramètres décrits. En plus d'éliminer la méthode BYSEUIL, ceci

| Hamming | Paramètres | EMR | Corpus |
|---|---|---|---|
| **0.548** | BYFUSION_2-3_char_cosine_0.8 | 0.377 | All In One |
| 0.541 | BYFUSION_3-3_char_cosine_0.8 | **0.388** | All In One |
| 0.532 | BYFUSION_3-3_char_cosine_0.7 | 0.331 | All In One |
| 0.532 | BYFUSION_2-3_char_cosine_0.9 | 0.373 | All In One |
| 0.518 | BYFUSION_2-3_char_cosine_0.7 | 0.252 | All In One |
| 0.505 | BYFUSION_3-3_char_cosine_0.6 | 0.197 | All In One |
| 0.486 | BYFUSION_2-3_char_cosine_0.6 | 0.094 | All In One |
| 0.475 | BYFUSION_3-3_char_cosine_0.8 | 0.352 | Feedback bis |
| 0.468 | BYFUSION_2-3_char_cosine_0.8 | 0.332 | Feedback bis |
| 0.462 | BYFUSION_3-3_char_cosine_0.9 | 0.258 | All In One |

TABLE 4 – 10 meilleurs résultats en Hamming sur le jeu d'entraînement (vectorisation en bi-grammes et tri-grammes de caractères).

nous a amené à écarter la pondération Tf-Idf (moins efficace que la simple valeur absolue) ainsi que les représentations en mots. La section suivante présente en plus en détails les résultats obtenus.

# 6 Résultats et discussion

La chaîne de traitement décrite dans la section 5 nous a permis de chercher les meilleures combinaisons de paramètres pour les deux mesures du défi, le *Hamming Score* et le *Exact Match Ratio*. Tout d'abord, nous avons pu voir que vectoriser en bi-grammes/tri-grammes de caractères était, avec notre méthode, systématiquement plus efficace que de vectoriser en mots (2 à 5 points de pourcentages selon les cas). Le Tableau 4 présente les dix meilleurs résultats obtenus sur le jeu d'entraînement fourni par les organisateurs. Nous avons atteint un score de Hamming maximal de 54 % pour un taux d'exactitude de 37 %. Ces résultats semblaient très prometteurs comparés à ceux obtenus par (Labrak *et al.*, 2022a). En ce qui concerne la tâche annexe, nous obtenons une précision de 49 % pour un F1_macro de 43 %. avec les paramètres BYFUSION_2-3_CHAR_COSINE_0.8 et le corpus All In One.

Les meilleurs résultats viennent de la méthode BYFUSION. Ce résultat n'est pas surprenant puisque les résultats obtenus avec la méthode BYMAX amènent une seule réponse par question (favorisant la précision) et que la méthode BYSEUIL amenait elle un meilleur rappel. La fusion de ces deux méthodes semble donc être l'approche la plus adaptée à la recherche de bons résultats. Le corpus permettant les meilleurs résultats est sans surprise le corpus *All In One*, contenant l'ensemble des corpus assemblés. Le Tableau 5 présente cette fois les meilleurs résultats obtenus avec le jeu de données de développement.

Nous constatons immédiatement des scores inférieurs aussi bien avec Hamming qu'avec l'EMR. Cela est certainement dû à un sur-apprentissage venant de nos corpus issus du jeu de données d'entraînement. De plus, nous remarquons que le corpus le plus productifs n'est plus le corpus *All In One*. Il s'agit des corpus Feedback bis et Feedback. Les ressources extérieures créées sont donc moins productives que les ressources assemblées à partir du jeu de données d'entraînement. Pour la tâche annexe, nous obtenons une précision de 25 % pour un F1_macro de 19 %. avec les paramètres BYFUSION_2-3_CHAR_COSINE_0.6 et le corpus All In One.

Enfin, pour le jeu de données test, nous avons obtenu un Hamming de 24,93 % et un EMR de 8,52 % lors de l'évaluation officielle. Pour la tâche annexe, nous avons obtenu une précision de 23 % et un

| Hamming | Paramètres | EMR | Corpus |
|---|---|---|---|
| **0.303** | BYFUSION_2-3_char_cosine_0.6 | 0.051 | All In One |
| 0.293 | BYFUSION_2-3_char_cosine_0.6 | 0.112 | Feedback bis |
| 0.279 | BYFUSION_2-3_char_cosine_0.6 | 0.041 | Merck |
| 0.277 | BYFUSION_3-3_char_cosine_0.6 | 0.147 | Feedback bis |
| 0.266 | BYFUSION_2-3_char_cosine_0.7 | 0.150 | Feedback bis |
| 0.265 | BYFUSION_3-3_char_cosine_0.6 | 0.099 | All In One |
| 0.261 | BYFUSION_3-3_char_cosine_0.6 | 0.150 | Feedback |
| 0.258 | BYFUSION_3-3_char_cosine_0.7 | 0.157 | Feedback |
| 0.254 | BYFUSION_2-3_char_cosine_0.6 | 0.102 | Feedback |
| 0.253 | BYMAX_3-3_char_cosine | **0.160** | Feedback |

TABLE 5 – 10 meilleurs résultats en Hamming sur le jeu de développement (vectorisation en bi-grammes et tri-grammes de caractères).

| Hamming | Paramètres | EMR | Corpus |
|---|---|---|---|
| **0.329** | BYFUSION_2-3_char_cosine_0.6 | 0.056 | All In One |
| 0.301 | BYFUSION_2-3_char_cosine_0.6 | 0.078 | Merck |
| 0.270 | BYFUSION_2-3_char_cosine_0.6 | 0.074 | Feedback bis |
| 0.269 | BYFUSION_2-3_char_cosine_0.6 | 0.099 | Feedback |
| 0.249 | BYFUSION_3-3_char_cosine_0.6 | 0.085 | All In One |
| 0.243 | BYFUSION_2-3_char_cosine_0.7 | 0.090 | All In One |
| 0.239 | BYFUSION_3-3_char_cosine_0.6 | 0.099 | Feedback bis |
| 0.237 | BYFUSION_2-3_char_cosine_0.7 | 0.117 | Feedback |
| 0.231 | BYMAX_2-3_char_cosine | **0.120** | Feedback |
| 0.231 | BYFUSION_2-3_char_cosine_0.9 | **0.120** | Feedback |

TABLE 6 – 10 meilleurs résultats en Hamming sur le jeu de test (vectorisation en bi-grammes et tri-grammes de caractères), en gris les résultats envoyés pour le défi

F1_macro de 43 %. Les meilleurs résultats obtenus sur le jeu de données de test sont présentés dans le Tableau 6.

Nous remarquons que nous aurions pu obtenir de meilleurs résultats que ceux obtenus lors du défi avec un choix plus judicieux du jeu de paramètre. Tout comme pour le jeu de développement, nous observons que les corpus les plus productifs sont les corpus assemblés à partir du jeu d'entraînement.

En résumé, nous avons présenté un système de question réponse fondé sur la recherche de similarités de phrases. Cette méthode ne se fonde pas sur un apprentissage et ne nécessite qu'un, ou plusieurs, corpus de référence du domaine. Notre méthode cherche la réponse la plus vraisemblable en recherchant dans des corpus spécialisés la ou les phrases les plus similaires aux réponses possibles. Nos corpus de référence ont été constitué de différentes manières : un corpus d'énoncés assertifs reconstruits à partir des données d'entraînement (corpus FEEDBACK), un corpus d'énoncés issus de l'interrogation de ChatGPT (corpus CHATGPT) ainsi que deux corpus issus de données disponibles en lignes (corpus MERCK et corpus CHATGPT). Si cette méthode simple n'obtient pas des résultats aussi élevés que des méthodes supervisées fondées notamment sur du *deep learning*, elle est assez interprétable puisque l'on peut sans rétro-ingénierie complexe retrouver les segments, phrases ou paragraphes par exemple, qui ont présidé au choix de telle ou telle réponse.

# Références


BAGGETTO P., RAMOS S., GARCIA J. & NAVARRO J. R. (2022). Study on text comprehension and mcqa in spanish. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2022), A Coruna, Spain. CEUR Workshop Proceedings. the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, volume 1, p. 4171–4186.

BEERS M. H., PORTER R. S., JONES T. V., KAPLAN J. L. & BERKWITS M. (2008). *Le Manuel Merck de diagnostic et thérapeutique*. Edition d'Après, 4e édition.

BROWN T. B., MANN B., RYDER N., SUBBIAH M., KAPLAN J., DHARIWAL P., NEELAKANTAN A., SHYAM P., SASTRY G., ASKELL A. *et al.* (2020). Language models are few-shot learners. In *Proceedings of the 37th International Conference on Machine Learning*.

BUSCALDI D., FELHI G., GHOUL D., LE ROUX J., LEJEUNE G. & ZHANG X. (2020). Calcul de similarité entre phrases : quelles mesures et quels descripteurs ?(sentence similarity : a study on similarity metrics with words and character strings). In *Traitement Automatique des Langues Naturelles (TALN, 27e édition). Atelier DÉfi Fouille de Textes*, p. 14–25.

CLARK P., COWHEY I., ETZIONI O., KHOT T., SABHARWAL A., SCHOENICK C. & TAFJORD O. (2018). Think you have solved question answering ? try arc, the ai2 reasoning challenge. DOI : 10.48550/ARXIV.1803.05457.

FALCO M.-H. (2012). Typologie des questions à réponses multiples pour un système de question-réponse (typology of multiple answer questions for a question-answering system)[in french]. In *Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 3 : RECITAL*, p. 191–204.

HUANG L., BRAS R. L., BHAGAVATULA C. & CHOI Y. (2019). Cosmos qa : Machine reading comprehension with contextual commonsense reasoning. DOI : 10.48550/ARXIV.1909.00277.

KIM H. & FUNG P. (2020). Learning to classify the wrong answers for multiple choice question answering (student abstract). *Proceedings of the AAAI Conference on Artificial Intelligence*, **34**(10), 13843–13844. DOI : 10.1609/aaai.v34i10.7194.

LABRAK Y., BAZOGE A., DUFOUR R., DAILLE B., GOURRAUD P., MORIN E. & ROUVIER M. (2022a). Frenchmedmcqa : A french multiple-choice question answering dataset for medical domain. p. 41–46.

LABRAK Y., BAZOGE A., DUFOUR R., DAILLE B., GOURRAUD P.-A., MORIN E. & ROUVIER M. (2022b). FrenchMedMCQA : A French multiple-choice question answering dataset for medical domain. In *Proceedings of the 13th International Workshop on Health Text Mining and Information Analysis (LOUHI)*, p. 41–46, Abu Dhabi, United Arab Emirates (Hybrid) : Association for Computational Linguistics.

LIU C.-L. & LEE H.-Y. (2020). Unsupervised multiple choices question answering : Start learning from basic knowledge. *arXiv preprint arXiv :2010.11003*.

MIHAYLOV T., CLARK P., KHOT T. & SABHARWAL A. (2018). Can a suit of armor conduct electricity ? a new dataset for open book question answering. DOI : 10.48550/ARXIV.1809.02789.

SOARES T. G., AZHARI A., ROKHMAN N. & WONARKO E. (2021). Education question answering systems : a survey. In *Proceedings of The International MultiConference of Engineers and Computer Scientists*.

TALMOR A., HERZIG J., LOURIE N. & BERANT J. (2018). Commonsenseqa : A question answering challenge targeting commonsense knowledge. DOI : 10.48550/ARXIV.1811.00937.

TOUISSI Y., HJIEJ G., HAJJIOUI A., IBRAHIMI A. & FOURTASSI M. (2022). Does developing multiple-choice questions improve medical students' learning ? a systematic review. *Medical Education Online*, **27**(1), 2005505.

YU W., JIANG Z., DONG Y. & FENG J. (2020). Reclor : A reading comprehension dataset requiring logical reasoning. *arXiv preprint arXiv :2002.04326*.

*Actes de CORIA-TALN 2023*
*Paris, France, 5 au 9 juin 2023*

# Participation de l'équipe TTGV à DEFT 2023 : Réponse automatique à des QCM issus d'examens en pharmacie

Andréa Blivet[1*] Solène Degrutère[1*] Barbara Gendron[2*] Aurélien Renault[3*]
Cyrille Siouffi[2*] Vanessa Gaudray-Bouju[1*] Christophe Cerisara[2*]
Hélène Flamein[1*] Gaël Guibon[2*] Matthieu Labeau[3*] Tom Rousseau[1*]
(1) DTIPG SNCF, 1-3 avenue François Mitterrand, 93210 Saint-Denis, France
(2) LORIA, Université de Lorraine, CNRS, 54000 Nancy, France
(3) LTCI, Télécom Paris, Institut Polytechnique de Paris, 19 place Marguerite Perey, 91120 Palaiseau, France

andrea.blivet@sncf.fr, solene.degrutere@sncf.fr, barbara.gendron@loria.fr,
aurelien.renault@polytechnique.edu, cyrille.siouffi@loria.fr,
ext.vanessa.gaudray-bouju@sncf.fr, christophe.cerisara@loria.fr,
helene.flamein@sncf.fr, gael.guibon@loria.fr,
matthieu.labeau@telecom-paris.fr, tom.rousseau@sncf.fr

## 1 Introduction

Cette année, l'équipe TTGV (TAL à Très Grande Vitesse) se (re)forme autour de l'équipe TGV qui a participé au défi DEFT en 2022. Comptant désormais 11 membres issus de la SNCF, de Télécom Paris et du LORIA, notre équipe s'est attelée à la résolution des deux tâches proposées : l'identification du nombre de réponses supposément justes à un QCM et la prédiction de l'ensemble de réponses correctes parmi les cinq proposées pour une question donnée. Ces questions proviennent du corpus FrenchMedMCQA, qui regroupe 3 105 questions fermées extraites des annales d'examens de pharmacie en français. Les travaux de (Labrak *et al.*, 2022) décrivent plus en détail les particularités de ce corpus et présentent les premières expérimentations pour aborder ces tâches. Dans notre étude, nous avons décidé de nous appuyer sur différentes approches, tout en tirant des enseignements de la *baseline* établie par (Labrak *et al.*, 2022).

La première partie de notre article se focalisera donc sur les différentes méthodologies mises en oeuvre, explorant ainsi un large éventail d'approches et de techniques pour aborder d'abord la distinction entre les questions appelant une seule ou plusieurs réponses avant de s'interroger sur l'identification des réponses correctes. Nous détaillerons les différentes méthodes utilisées, en mettant en exergue leurs avantages et leurs limites respectives. Ensuite, nous présenterons les résultats obtenus pour chaque approche. Enfin, nous discuterons des limitations intrinsèques aux tâches elles-mêmes ainsi qu'aux approches envisagées dans cette contribution.

*. Contributions égales.

# 2 Méthodologie

La répartition du nombre de réponses attendues pour chaque question dans les jeux de données (*train*, *dev* et *test*) est illustrée dans la figure 1 en annexe. Parmi les 3 105 questions du corpus, on observe que 35% (soit 1 080 questions) nécessitent uniquement une réponse. Alors que le *train* contient davantage de questions demandant trois réponses, les questions ne demandant qu'une seule réponse sont majoritaires dans le *dev* et le *test*. D'une manière générale, la répartition des nombres de réponses suit une tendance similaire entre le *dev* et le *test*. Étant donné la répartition des données, nous allons d'abord nous concentrer sur la distinction des questions n'appelant qu'une seule réponse (dites simples) et celles qui en exigent plusieurs (dites multiples).

## 2.1 Classification des questions simples et multiples

La formulation des questions et des propositions de réponses permet d'estimer si la question attend une ou plusieurs réponses. Un rapide parcours du jeu de données met notamment en lumière les particularités suivantes :

- **Questions** : Certains mots – principalement les dérivés du radical « quel »– donnent un indice sur le nombre de réponses attendues (quel(s), le(s)quel(s), etc.).
- **Réponses** : Dans certains cas, les réponses présentent des affirmations, parfois incompatibles, sur des thèmes apparemment disjoints. En supposant qu'exactement une affirmation par thème est vraie, le nombre de thèmes disjoints dans les réponses pourrait correspondre au nombre de réponses attendues pour la question concernée.

Trois stratégies peuvent être proposées pour estimer le nombre de réponse à une question : 1) **Expressions régulières (ER)** : Cette approche détecte la présence de mots-clefs (pronoms, pronoms démonstratifs...) dans le libellé d'une question pour estimer, selon l'accord en nombre, si elle admet une ou plusieurs réponses. 2) ***Topic modelling* (LSI & LDA)** : Cette approche vise à identifier des thèmes disjoints à partir des mots composant assertion, sans prendre en compte leur sémantique. Parmi les algorithmes reconnus pour effectuer ce type de tâche, nous avons exploré le *Latent Semantic Indexing* (LSI) (Hofmann, 1999) et le *Latent Dirichlet Allocation* (LDA) (Dumais *et al.*, 2004). 3) **Régression Logistique (RL)** : Cette approche statistique est couramment utilisée pour modéliser des événements binaires en fonction de variables indépendantes (Kleinbaum *et al.*, 2002).

S'agissant de tâches de classification – binaire ou multi-classe –, nous évaluons la performance des méthodes proposées avec les indicateurs usuels : précision, rappel, $F_1$-score.

**Analyse des questions par expressions régulières** La présence de pronoms – démonstratifs ou non – dans le libellé des questions donne un indice sur le nombre de réponses attendues. Les expressions régulières permettent de les détecter efficacement. Quelques exemples tirés du jeu d'entraînement sont présentés dans l'annexe B.3. Les exemples n$^{os}$ 68 et 2 161 illustrent que la seule détection des dérivés de « celui » et de « quel » conduit parfois à des ambiguïtés : il faut également assurer qu'ils n'apparaissent pas simultanément au singulier et au pluriel dans le libellé de la question. Nous avons alors créé un système d'expressions régulières permettant de capter les instances de mots dérivés précédemment cités uniquement au singulier (cf. annexe B.1).

Ce système de détection permet de distinguer efficacement les questions à réponse simple ou multiple. En effet, nous obtenons un taux d'exactitude (*accuracy*) de 94% sur le jeu d'entraînement, 89% sur

le jeu de développement. Les scores de classification détaillés en tableau 3 mettent en lumière les principales lacunes de ce classifieur : davantage de faux positifs sur la classe simple ; davantage de faux négatifs sur la classe multiple. A titre d'illustration, quelques exemples de faux positifs dans la classe simple ont été ajouté en annexe B.3. Nous avons alors exploré une approche complémentaire, s'intéressant aux contenu des réponses possibles.

**Analyse des réponses par *topic modelling*** La formulation des réponses peut indiquer combien d'entre elles sont acceptables, *via* le nombre de thèmes différents qu'elles abordent. Le *topic modelling* s'intéresse à la construction automatique de thèmes représentés dans un corpus. Certains algorithmes, à l'instar de LSI et LDA (cf. annexe B.2), résolvent cette tâche par apprentissage statistique. Dans notre contexte, ces deux approches nous ont paru justifiées (i) par la nature monosémique du vocabulaire médical, (ii) par le caractère laconique des réponses, (iii) par le fait que des « thèmes » pourraient se manifester dans les propositions de réponses de plusieurs questions. Sous ces hypothèses, nous avons alors modélisé les thèmes comme suit :

- **Constitution du corpus** : Chacune des cinq réponses proposées pour les 2 171 questions est considérée comme un document. Le corpus regroupe donc $5 \times 2\,171 = 10\,855$ documents.
- **Prétraitement** : Les documents sont prétraités selon les étapes suivantes : découpage des phrases en unités lexicales (*tokenization*) en excluant la ponctuation et mots parasites (*stop-words*), extraction des racines de chaque unité (*stemming*), constitution d'un dictionnaire, transformation des documents en sacs de mots (*Bag of Words*, ou *BoW*), enfin normalisation du corpus transformé *via* TF-IDF, afin d'occulter les mots communs à trop de documents.
- **Hyper-paramétrage** : Estimation du meilleur nombre $N$ de thèmes pour le LSI et le LDA. Pour le LDA, le nombre optimal de thèmes obtenu est de 57.
- **Inférence** : Pour chaque nouvelle question, les cinq réponses proposées sont concaténées en un seul document. Le modèle (LSI ou LDA) en extrait alors des thèmes, que l'on filtre par probabilité d'occurrence *via* un seuillage adaptatif. Le nombre de thèmes retenus correspond alors à notre prédiction de nombre de réponses pour cette question.

Compte tenu de l'écart important de performances entre les meilleurs LSI et LDA trouvés [1], nous ne présentons que les résultats obtenus pour le LDA, donnés dans le tableau 3. En dehors des faibles performances de ce modèle, on constate un fort déséquilibre entre précision et rappel sur chaque classe, que la classification soit binaire ou multiple. Notons aussi que la classe 4 n'a jamais été prédite sur le jeu de développement, alors que ce n'est pas la moins représentée.

Toutefois, l'analyse des réponses apporte des indications utiles, complémentaires à celles tirées du système d'expressions régulières. C'est pourquoi nous avons tenté de croiser les analyses des questions et des réponses en recoupant les prédictions du système d'expressions régulières (appliqué aux questions) et du LDA (appliquée aux réponses). Le croisement a été opéré de la façon suivante : si les expressions régulières prédisent que la réponse sera unique, on se fie à cette prédiction ; sinon, on utilise le LDA pour prédire le nombre de réponse. Les performances de classification sont effectivement meilleures que celles obtenues avec le seul LDA. En effet, le taux d'exactitude sur les deux tâches de classification s'améliore nettement, ce qui tient au fait que la classe 1 est mieux prédite. Cependant, comme l'illustre le tableau 3, cela ne suffit pas à améliorer significativement les performances sur les autres classes, diluant ainsi le gain précédemment relevé.

1. La phase d'hyperparamétrage du LSI n'a pas convergé, contrairement à celle de la LDA.

**Analyse des réponses par *topic modelling*** Notre dernière approche pour essayer de distinguer les questions simples des questions multiples s'appuie sur la régression logistique de *scikit-learn* (Pedregosa *et al.*, 2011). Grâce à cette bibliothèque, nous avons procédé à la tokenisation et à la vectorisation des questions en utilisant un *CountVectorizer*. Cela a permis de représenter les questions sous forme de vecteurs en indiquant quels mots sont présents dans chaque question. Pour réduire la dimensionnalité de la représentation vectorielle, nous avons conservé uniquement les 10 000 mots les plus fréquents dans le corpus pour construire une matrice des vecteurs représentant chaque question. Le modèle de régression logistique a été entraîné sur les données du jeu d'entraînement avec l'algorithme *liblinear*, particulièrement recommandé lorsque les données sont limitées. Étant donné qu'il y avait un déséquilibre entre les étiquettes (plus de questions multiples que de questions simples), nous avons effectué une balance des classes en ajustant les poids des échantillons lors de l'entraînement du modèle. Le modèle a finalement été utilisé pour prédire dans le jeu de données de développement la nature des questions (simple ou multiple) et a obtenu un $F_1$-score de 0,94 (cf. tableau 3 en annexes).

**Analyse croisée des résultats** La régression logistique s'est révélée être le meilleur modèle pour prédire si une question est de type simple ou multiple. Cependant, ses performances ne sont pas satisfaisantes lorsqu'il s'agit d'aller plus loin dans la classification des types de questions. En ce qui concerne la résolution de la tâche annexe proposée dans le cadre du défi, nous avons pris la décision de soumettre les prédictions du modèle combinant ER et LDA ainsi que les prédictions du modèle utilisant la régression logistique qui présente un bon $F_1$-score. Ce dernier ne permettant que de faire la distinction entre les questions simples et multiples, nous avons décidé de soumettre ses prédictions en considérant par défaut que les questions de type multiple appellent deux réponses.

Si la tâche annexe n'est pas résolue à ce stade, le modèle RL pourra trouver son utilité dans les autres méthodes pour aider la résolution de la tâche principale. En outre, les approches les plus encourageantes pour la résolution de la tâche principale – telle que l'approche multi-classe décrite dans la section suivante – pourront également être utilisées pour répondre à la tâche annexe.

## 2.2 Approche multi-classe

Dans cette partie, nous reprenons l'approche multi-classe proposée dans (Labrak *et al.*, 2022) ; de cette manière, le problème revient, pour chaque question, à prédire la bonne classe parmi les 31 possibles, i.e. le nombre total de combinaisons différentes des 5 réponses possibles. Quant au choix de la représentation, nous avons choisi d'uniquement travailler avec le modèle **DrBERT** (Labrak *et al.*, 2023), un modèle de langue français dérivé de **CamemBERT** (Martin *et al.*, 2020) et spécialisé sur des données biomédicales. Le tableau 1 montre le gain de performances de cette architecture comparée à d'autres moins spécialisées.

**Ajout d'un contexte** À l'instar de (Labrak *et al.*, 2022), nous considérons que l'apport de connaissances extérieures constitue un levier intéressant à étudier dans le cadre de cette tâche. Comme eux, nous avons fait le choix de nous appuyer sur le Wikipédia français pour construire des contextes plus denses en informations autour des questions et des réponses de chaque entrée de la base.

Pour mieux cibler la récupération des contextes, une première étape de sélection des termes les plus représentatifs des questions et des propositions de réponses a été intégrée. L'hypothèse ici est que les syntagmes porteurs des informations les plus significatives peuvent être identifiés grâce à la fréquence d'apparition de leurs lemmes dans l'ensemble du jeu de données. Le vocabulaire employé dans le corpus présentant de nombreuses récurrences (« Parmi les propositions suivantes »,

« laquelle », « quelle », « méthode », « indiquer », etc.), nous considérons que plus les lemmes sont fréquents dans le corpus, moins ils seront considérés comme spécifiques. À l'inverse, des lemmes plus rares seront plus susceptibles de représenter les notions centrales des questions et des réponses. En s'appuyant donc sur la fréquence moyenne d'apparition des lemmes dans le corpus et sur leur étiquetage morphosyntaxique, seuls les syntagmes nominaux composés entièrement de termes dont la fréquence d'apparition est supérieure à la moyenne sont retenus. En l'occurrence, cette moyenne s'élève à 12 dans les données du jeu de *test* pour un total de 3 233 lemmes. Cette approche permet de récupérer, pour chaque question, les syntagmes nominaux les plus pertinents en excluant les termes pas suffisamment spécifiques.

Pour extraire des éléments de contexte de Wikipédia, le module Cohere [2] semblait le plus approprié. En effet, il a été utilisé pour vectoriser les paragraphes de millions de pages Wikipédia en de nombreuses langues et permet de récupérer un ou plusieurs paragraphe(s) à partir d'une entrée textuelle grâce à des scores de similarité, obtenus par un produit scalaire. Nous avons testé trois configurations différentes pour récupérer les contextes :

- pour chacune des réponses proposées, concaténation de la question et de la réponse pour retrouver les trois paragraphes les plus similaires pour chacune des cinq entrées ;
- idem mais en ne récupérant qu'un seul paragraphe pour chaque entrée ;
- pour chacune des réponses proposées, concaténation des syntagmes nominaux de la question et de la réponse pour retrouver le paragraphe le plus similaire pour chacune des cinq entrées.

Les paragraphes récupérés sont ensuite concaténés de façon à former un contexte unique par question.

Le tableau 1 montre que dans la configuration actuelle l'ajout d'un contexte ne permet pas d'améliorer les performances ; au contraire, le modèle ne semble pas être en mesure d'extraire les informations pertinentes du contexte fourni. En effet, il semblerait que ces contextes restent encore trop longs pour le modèle. Des tentatives ont été entreprises pour les réduire et les rendre plus concis en incluant des informations pertinentes, mais jusqu'à présent, elles n'ont pas abouti. Parallèlement, nous avons envisagé de limiter nos contextes aux pages Wikipédia provenant des portails Pharmacie et Médecine, mais le manque de temps nous a empêchés d'explorer pleinement cette piste.

**Approche contrastive** Nous proposons également une extension contrastive à cette approche dans le but d'infuser de la connaissance durant la phase de *fine-tuning*, plus spécialement au niveau des entités nommées (Xiong *et al.*, 2020). Pour ce faire, nous obtenons des annotations d'entités en *finetunant* le susmentionné **DrBERT** sur un dataset de NER biomédicale français (QUAERO (Névéol *et al.*, 2014)). Les entités imbriquées ne sont pas gérées, i.e. seule la mention correspondant à l'entité la plus longue est conservée. Durant l'inférence, un mot reçoit une annotation d'entité si au moins le premier *token* de ce dernier est annoté. Ensuite, $k$ exemples négatifs sont créés en remplaçant $50\%$ des entités par d'autres entités du même type. Enfin, nous ajoutons un objectif supplémentaire dont le but est de reconnaître des entités ayant été substituées des entités « réelles ». Si $e$ une entité, $C$ son contexte associé et $E^+$ l'ensemble des vraies mentions d'entités, la tête *contrastive* revient à minimiser la fonction suivante :

$$\mathcal{L}_{\text{contrastive}} = \mathbb{1}_{e \in E^+} \log P(e|C) + (1 - \mathbb{1}_{e \in E^+}) \log(1 - P(e|C))$$
$$\mathcal{L} = \mathcal{L}_{\text{classif}} + \lambda \mathcal{L}_{\text{contrastive}}$$

Dans le cas où une substitution d'entités se serait effectuée sur une entité contenue dans une réponse vraie, la réponse est retirée de l'ensemble des réponses correctes (dans la limite d'avoir toujours

2. https ://txt.cohere.com/embedding-archives-wikipedia/

au moins une bonne réponse). Le tableau 1 montre que l'approche, pour $k = 5$, n'améliore pas significativement les performances sur le jeu de *test*.

| Modèle | Hamming | EMR |
|---|---|---|
| Camembert-base | 33,80 | 14,31 |
| Dr-bert-7GB | **39,08** | **17,68** |
| Dr-bert-7GB$_{\text{contexte}}$ | 35,31 | 15,27 |
| Dr-bert-7GB$_{\text{contrast}}$ | 36,06 | 16,40 |

TABLE 1 – Résultats des approches multi-classe sur le jeu de *test* ; les scores affichés sont, pour chaque modèle, la médiane sur 3 différentes seeds aléatoires

## 2.3 Approches multi-étiquettes

L'approche multi-étiquettes neuronale provient de deux hypothèses principales : 1) l'intégration des questions dans l'encodage peut guider le modèle vers la ou les bonnes classe(s) et 2) chaque classe doit être considérée indépendemment avec une probabilité dédiée.

Ici, on considère qu'une classe ne correspond pas à la réponse finale (par exemple, *a|c*) mais on cherche à prédire indépendamment chaque étiquette (*a* d'une part et *c* d'autre part). Cette approche entend notamment améliorer le score de Hamming, qui valorise chaque bonne réponse trouvée, et intégrer en son sein une hiérarchie et un contrôle de la sortie. Le seuil (*threshold*) à partir duquel un score de probabilité sera considéré comme une prédiction viable est un hyper-paramètre déterminant qu'il est nécessaire d'optimiser. Puisque chaque classe est prédite indépendamment, il faut intégrer pour chacune une fonction de coût de sigmoïde ainsi qu'une cross entropie binaire. Nous avons mis en place cette approche multilabel de manière non neuronale, en considérant des classifieurs plus classiques. Nous avons notamment mis l'accent sur les LGBM (Ke *et al.*, 2017) pour y rechercher les meilleurs paramètres. Malheureusement, toutes ces approches peinent à dépasser les 27% en hamming score, entraînant ainsi la nécessité de considérer des approches neuronales. Lors de la mise en place du classifieur multi-étiquettes neuronal, nous avons fait face à plusieurs obstacles dont la difficulté du choix de la représentation initiale du modèle, le choix du palier pour l'attribution des classes, l'évaluation de ces dernières, et l'équilibre entre les deux métriques d'évaluation. Pour tous nos tests, nous avons considéré l'encodage joint de la question et des réponses associées, en entraînement.

**Le modèle : un *fine-tuning* à l'aide de *transformers* additionnels.** Notre approche neuronale consiste en l'utilisation d'un modèle de langue comme représentation initiale, en l'occurrence un BERT Mini (Turc *et al.*, 2019; Bhargava *et al.*, 2021) adapté au domaine pharmaceutique mais aux poids non mis à jour, couplé à l'adjonction de cinq couches d'encodeurs transformers ayant chacun 8 têtes d'attention et un dropout entre les couches de 10%. Nous appliquons cet encodeur additionnel en sortie du modèle de langue, c'est-à-dire avant la couche de pooler, qui permet d'unifier les représentations à l'aide d'une fonction d'activation en tangente hyperbolique (tanh). Une particularité de notre approche est également d'utiliser deux fonctions d'activation pour casser la linéarité du modèle, la tangente et le ReLU à la suite avant un dropout de 50%. Ces paramètres extrêmes ont été estimés nécessaires compte tenu de la différence en termes de langue et de domaine. Enfin, un classifieur linéaire suivi d'une fonction sigmoïdale permet de normaliser les sorties entre 0 et 1 et ainsi d'obtenir l'équivalent d'une probabilité pour chaque classe.

**Choix de la représentation initiale.** Notre modèle utilise *in fine* un modèle de langue BERT Mini adapté au domaine. Définir la bonne représentation initiale à adopter est un choix déterminant eu égard aux travaux liés comme la baseline (Labrak *et al.*, 2022) et les autres approches de l'équipe. Pour ce faire, nous avons procédé à une approche empirique en partant du principe que la langue n'est pas un facteur déterminant (Labrak *et al.*, 2022). Ce choix de représentation entraîne toutefois un conflit entre les connaissances généralistes bien encodées dans le modèle de langue et les connaissances spécifiques au domaine (médecine), voire celles spécifiques à la spécialité (pharmacie). Ce conflit entraîne intuitivement une propension forte au *catastrophic forgetting* (Kirkpatrick *et al.*, 2017). C'est dans cette optique que nous avons pallié ce problème par la comparaison empirique d'une représentation vectorielle à partir de zéro (couche d'*embeddings*), à partir d'*embeddings* statiques (FastText (Joulin *et al.*, 2016)), ou encore par l'intégration d'*embeddings* contextuels avec et sans *fine-tuning* (CamemBERT, DistilCamembert, bert, etc.). De manière surprenante, ni le *fine-tuning*, ni la langue cible n'ont apporté un bénéfice certain, les modèles peinant à dépasser la barre des 30% en Hamming. C'est en mettant en place une approche d'adaptation au domaine par un *pre-training* (MLM) continu sur les données d'entraînement (Konlea & Jannidisa, 2020; Wu *et al.*, 2021) que nous avons pu passer ce seuil. Pour cela, il a été nécessaire de trouver une taille adéquate du modèle de langue compte tenu de la taille des données d'entraînement. C'est BERT Mini qui a donné les meilleurs résultats, malgré son entraînement initial réalisé uniquement sur de l'anglais. Notre phase d'adaptation permet donc non seulement au modèle de voir du français mais en même temps de s'adapter à la spécialité et au format des questions et des réponses du jeu d'entraînement. C'est ce MiniBERT adapté qui est fourni en tant que représentation initiale à notre modèle. D'autres données médicales anglaises ont été envisagées mais mises de côté par manque de temps.

**Protocole expérimental.** Le *threshold* choisi dans notre implémentation est l'hyper-paramètre permettant de favoriser la métrique d'EMR ou celle de Hamming. En effet, un *threshold* habituel est $0.5$, mais ce dernier s'avère insuffisant, voire trop élevé. Par tests empiriques, nous sommes arrivés à la conclusion que le meilleur *threshold* était de $0.4$. Ce dernier permettant au modèle d'essayer davantage d'étiquettes, il nous a permis d'augmenter significativement le score de Hamming (+ 5-7%) au détriment du score d'EMR, quasi-inexistant.

Face à la cupidité du modèle et au *threshold* bas, nous avons considéré un modèle de contrôle du comportement prédictif de celui-ci. Pour ce faire, un modèle de classification a été mis en place afin de déterminer si connaître le nombre de bonnes réponses (tâche annexe) permettrait d'augmenter l'EMR. Seul un modèle de classification binaire utilisant BERT a été mis au point par contrainte de temps. Ce modèle atteint 86% d'*accuracy*, ce qui demeure moins que l'approche par régression logistique susmentionnée, qui atteint 94% d'*accuracy* et qui a donc été choisie.

Un contrôle de la prédiction utilisant la sortie de ce classifieur binaire a été intégré. Si le type prédit est simple, alors l'étiquette avec la plus forte probabilité est prédite. Si le type prédit est multiple, alors une prédiction classique de *multi-label* est réalisée. Cette logique assez simple permet de pallier légèrement le problème du score d'EMR inexistant.

## 2.4 Méthode baseline avec GPT-2 & BioGPT avec traduction avec Opus

Avec l'objectif de voir s'il serait possible d'agrandir le corpus d'entraînement de certains modèles, de se servir des ressources plus vastes qui sont proposées en anglais et afin de profiter des modèles spécialisés en biologie, en partant de l'hypothèse que le vocabulaire technique comporte énormément de similarités en anglais et en français, nous avons essayé d'utiliser des modèles basés sur GPT-2

(Radford *et al.*, 2019) comme BioGPT (Luo *et al.*, 2022), après avoir effectué une traduction sur le *dataset* (originellement français), et de les comparer à sa référence.

Nous avons utilisé un modèle de traduction spécialisé dans la conversion du français vers l'anglais, Opus-MT (Tiedemann & Thottingal, 2020), et avons f*ine-tuné* les modèles GPT sur le jeu d'entraînement fourni après traduction en classification mono-label, multi-output, multi-classe, comme dans la *baseline* proposée. Avec cette approche sur GPT-2 simple, nous avons constaté un *overfitting* très tôt dans l'entraînement, peu importe les hyperparamètres utilisés, et évoqué l'hypothèse que le modèle pourrait soit manquer de connaissances sur les sujets évoqués dans les questions, soit que la tâche de classification telle que définie n'était pas adaptée à l'objectif. En effectuant la même expérience sur BioGPT, nous observons exactement le même schéma et avons décidé de supposer qu'il ne s'agissait pas d'un problème de connaissances du modèle.

## 2.5 Approches génératives

**Galactica (1.3b)** Dans cette partie, nous avons essayé de voir, avec une approche *zero-shot* sur un modèle génératif supposé spécialisé dans le domaine des sciences, si nous pouvions simplement lui faire répondre aux questions. Nous avons utilisé les méthodes avec lesquelles a été entraîné Galactica (Taylor *et al.*, 2022), qui sont spécifiées plus en détails dans le dépôt GitHub du modèle, pour lui poser les questions du sujet. Nous l'avons d'abord tenté en brut, en essayant d'extraire les réponses de la sortie du modèle génératif, qui n'était pas toujours dans le même format.

Nous avons aussi cherché s'il existait une forme de prompt plus adapté pour les questions, en tentant diverses approches, basées sur *Language Mostly Know What They Know* (Kadavath *et al.*, 2022), qui cherche à déterminer si l'on peut savoir si un modèle de langage connaît la réponse à la question, ce qui aurait pu permettre d'effacer ou d'ajuster les réponses incertaines du résultat.

**BloomZ** Dans cette partie, nous donnons au modèle BloomZ-176b (Muennighoff *et al.*, 2022) un prompt de la forme : « Question : ... Choix : (A) ... (B) ... (C) ... (D) ... (E) ... Réponses : »en copiant tels quels les éléments de chaque question à la place des « ... », puis nous demandons au modèle de générer les 20 tokens suivants les plus probables, sans sampling. Nous détectons ensuite dans ces 20 *tokens* les lettres majuscules A B C D E : dès qu'une lettre apparaît, la réponse correspondante est considérée comme donnée. Dans les détails, nous avons testé plusieurs variantes, presque toutes en mode *zero-shot*, donc sans utiliser le corpus d'apprentissage de DEFT :

- Nous avons testé d'autres modèles pré-entraînés qui n'ont pas subi d'*instruction fine-tuning* : Bloom-176b, Bloom-7b1 et Vicuna-13b (Chiang *et al.*, 2023). Ces 3 modèles donnent de mauvais résultats, car ils n'arrivent pas à répondre correctement à la question posée sous cette forme. La phase d'*instruction-tuning* réalisée sur BloomZ est donc primordiale pour que le modèle puisse générer une réponse sous la forme attendue.
- Nous avons testé une poignée de variantes légèrement différentes du prompt, par exemple avec ou sans espaces, retour-chariot, etc. De plus, les instructions du prompt peuvent être en anglais (« Question : », « Choices : », « Correct answers : »), tout en gardant le contenu de la question en français ; les résultats sont toujours à peu près les mêmes.
- Nous avons testé en ajoutant de 1 à 5 phrases de contexte, avant la question elle-même et en préfixant par « Contexte : ». Ces phrases en anglais sont issues de Wikipedia par une recherche neuronale sémantique via l'API de Cohere-AI, en lui donnant le texte de la question et des réponses. Les résultats sont à peu près les mêmes, légèrement meilleurs mais pas

significativement. Ceci suggère que l'ajout d'information contextuelle n'apporte rien, et donc que BloomZ-176b possède déjà les informations demandées. Nous pensons que le goulot d'étranglement en termes de performance est lié à l'interprétation de la question par le modèle.

- Ajouter 5 questions-réponses aléatoires du corpus d'apprentissage, formatées de la même manière, en mode *in-context few-shot learning* n'améliore pas les performances.
- Nous avons testé en ajoutant un unique vecteur de paramètres au début du contexte (*soft-prompt-tuning*), que nous avons appris sur le corpus d'apprentissage de DEFT. Les résultats étaient un peu meilleurs, mais très peu, et au vu du coût bien plus élevé de cette approche, nous ne l'avons pas choisie pour le modèle final.
- Nous avons fait des expériences en modifiant l'ordre des réponses, ce qui donne des résultats différents, parfois un peu meilleurs, parfois moins bons. BloomZ-176b n'est donc pas sensible aux variations mineures du prompt (y compris la langue des instructions), mais est sensible aux modifications majeures du *prompt*. Nous n'avons pas optimisé le prompt pour avoir les meilleurs résultats possibles sur le corpus de développement, mais nous avons exploité cette variabilité dans notre système final comme une forme de « mesure de confiance ».

L'approche finalement choisie est du *zero-shot* basé sur BloomZ-176b, exécutée 3 fois pour chaque question : une exécution de base, décrite précédemment, que nous appelons ABCDE ; une exécution en inversant l'ordre des réponses, EDCBA ; et une exécution en translatant l'ordre des réponses, BCDEA. Nous extrayons ensuite les réponses communes des 2 variantes EDCBA et BCDEA : Si ces réponses n'apparaissent pas dans ABCDE, alors nous les y ajoutons. La réponse finale est la réponse de ABCDE ainsi complétée. Nous avons choisi cette approche car nous avons remarqué que BloomZ favorise les réponses uniques, donc intuitivement, lorsque les 2 variantes sont d'accord entre elles, nous pouvons considérer cette réponse commune comme ayant une bonne confiance.

# 3 Résultats

Afin de distinguer toutes ces approches et de choisir trois d'entre elles à envoyer, nous avons utilisé le score hamming comme métrique principale. La tâche annexe ayant servi a essayer d'améliorer les résultats de la tâche principale (voir sections 2.1 et 2.3), nous avons mis nos efforts en priorité sur cette dernière. Le tableau 2 montre les résultats finaux envoyés pour le défi. Ces résultats sont conformes à ce que nous avions prédit, à l'exception de l'approche *multi-class contrastive* qui obtenait de meilleurs résultats que l'approche *multi-label* sur le corpus de validation. Il serait intéressant d'en examiner la raison, sachant que la répartition des étiquettes est similaire (voir figure 1).

# 4 Limites et perspectives

Les approches que nous avons menées se sont heurtées à certaines limites provenant de la spécificité de la tâche. En effet, il ne s'agit pas d'un problème de *multi-label* ou de *multi-classes* classique : les différentes classes étudiées n'ont pas de réelle continuité (les différentes réponses $a$, par exemple, ne partagent pas davantage de caractéristiques communes qu'avec les autres classes). L'apprentissage ne doit donc pas se faire au niveau de l'étiquette. Pour avoir un apprentissage qui se concentre davantage sur le contenu en tant que tel, on pourrait préférer une approche basée sur une *loss* de *ranking* qui renseigne sur la qualité des prédictions. Une autre méthode serait de prendre une loss

Tâche principale

| Modèle | Hamming | EMR |
|---|---|---|
| DrBERT Multiclass contrastive | 37,22 | 15,43 |
| DAPT Multilabel avec type prédit (LR) | 39,15 | 11,58 |
| BloomZ zero-shot | **41,54** | **23,95** |

Tâche annexe

| Modèle | macro F1 | Accuracy |
|---|---|---|
| LDA | 13,26 | 19,13 |
| DrBERT Multiclass contrastive | **31,51** | 60,45 |
| Régression logistique | 27,98 | **62,54** |

TABLE 2 – Résultats principaux

basée sur une métrique de classification exigeante, typiquement le MCC (*Matthews correlation coefficient*) (Matthews, 1975). Cette dernière a été mise en place à l'aide d'un simili seuil par fonction sigmoïdale (Abhishek & Hamarneh, 2021) mais les résultats n'ont pas été concluants sur la tâche de *multi-label*. Malgré tout, il semble légitime de se demander s'il existe une *loss* permettant d'apprendre une telle tâche, puisque que l'expression du hamming ne permet pas de le différencier d'une manière exploitable. Pour ces raisons, les solutions présentées ici semblent n'être que palliatives.

Par ailleurs, cette tâche nécessitant des connaissances scientifiques très spécifiques, il nous semble important de fournir au modèle le plus de savoir possible sur le domaine étudié. Nous avons notamment utilisé Cohere à ces fins mais avons vu que cette piste pourrait encore être améliorée, par exemple en filtrant davantage les résultats, afin de ne pas récupérer d'informations provenant de pages Wikipédia hors domaine, ou encore en affinant la recherche au niveau des termes, pour ne récupérer que des passages mentionnant ceux présents dans les propositions de réponse. Il serait en outre intéressant de réaliser des tests plus approfondis sur la longueur des contextes fournis.

L'autre approche envisagée, l'adaptation au domaine par spécialisation d'un mini-BERT, présente plusieurs limites, à commencer par un vocabulaire non-exhaustif et un pré-entraînement sur des données en anglais. De plus, l'entraînement du modèle et donc son évaluation dépend du *threshold* qu'on fixe pour la classification *multi-label*. Une méthode évoquée mais non implémentée pourrait consister en l'utilisation d'un *threshold* approximatif en appliquant une sigmoïde (Abhishek & Hamarneh, 2021). Finalement, on observe une absence de corrélation entre la fonction de coût et la tâche finale, ce qui pose un problème pour l'évaluation du modèle. Cela rejoint le point précédent sur la difficulté de construire un entraînement pertinent, d'autant plus que dans ce cas on utilise une BCELoss qui considère des classes réelles, ce qui est fondamentalement différent de la démarche du choix des réponses possibles. Pour améliorer ce point, il serait possible de considérer l'EMR en tant que fonction de coût, ce qui a plus de sens vis à vis de la tâche.

Au final, notre équipe s'est attelée à explorer différentes familles d'approches afin d'aborder au mieux cette tâche. Ces approches n'ont toutefois pas réussi à dépasser le score obtenu en *zero-shot* par Bloomz, que ce soit en Hamming ou en EMR. Nous avons toutefois bon espoir quant au fait qu'une fonction de coût dédiée à cette tâche permettrait l'apprentissage de modèles dédiés plus efficaces.

# Remerciements

Ces travaux ont bénéficié d'un accès aux moyens de calcul de l'IDRIS au travers de l'allocation de ressources 2023-AD011011668R3 attribuée par GENCI. Certaines expériences présentées dans cet article ont été réalisées sur le banc d'essai Grid'5000, soutenu par un groupement d'intérêt scientifique hébergé par l'Inria et comprenant le CNRS, RENATER et plusieurs universités ainsi que d'autres organisations.

# Références

ABHISHEK K. & HAMARNEH G. (2021). Matthews correlation coefficient loss for deep convolutional networks : Application to skin lesion segmentation. In *The IEEE International Symposium on Biomedical Imaging (ISBI)*.

BHARGAVA P., DROZD A. & ROGERS A. (2021). Generalization in nli : Ways (not) to go beyond simple heuristics.

CHIANG W.-L., LI Z., LIN Z., SHENG Y., WU Z., ZHANG H., ZHENG L., ZHUANG S., ZHUANG Y., GONZALEZ J. E., STOICA I. & XING E. P. (2023). Vicuna : An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

DUMAIS S. T. *et al.* (2004). Latent semantic analysis. *Annu. Rev. Inf. Sci. Technol.*, **38**(1), 188–230.

HOFMANN T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, p. 50–57.

JOULIN A., GRAVE E., BOJANOWSKI P., DOUZE M., JÉGOU H. & MIKOLOV T. (2016). Fasttext.zip : Compressing text classification models. *CoRR*, **abs/1612.03651**.

KADAVATH S., CONERLY T., ASKELL A., HENIGHAN T., DRAIN D., PEREZ E., SCHIEFER N., HATFIELD-DODDS Z., DASSARMA N., TRAN-JOHNSON E., JOHNSTON S., EL-SHOWK S., JONES A., ELHAGE N., HUME T., CHEN A., BAI Y., BOWMAN S., FORT S., GANGULI D., HERNANDEZ D., JACOBSON J., KERNION J., KRAVEC S., LOVITT L., NDOUSSE K., OLSSON C., RINGER S., AMODEI D., BROWN T., CLARK J., JOSEPH N., MANN B., MCCANDLISH S., OLAH C. & KAPLAN J. (2022). Language models (mostly) know what they know.

KE G., MENG Q., FINLEY T., WANG T., CHEN W., MA W., YE Q. & LIU T.-Y. (2017). Lightgbm : A highly efficient gradient boosting decision tree. In I. GUYON, U. V. LUXBURG, S. BENGIO, H. WALLACH, R. FERGUS, S. VISHWANATHAN & R. GARNETT, Éds., *Advances in Neural Information Processing Systems*, volume 30 : Curran Associates, Inc.

KIRKPATRICK J., PASCANU R., RABINOWITZ N., VENESS J., DESJARDINS G., RUSU A. A., MILAN K., QUAN J., RAMALHO T., GRABSKA-BARWINSKA A. *et al.* (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, **114**(13), 3521–3526.

KLEINBAUM D. G., DIETZ K., GAIL M., KLEIN M. & KLEIN M. (2002). *Logistic regression*. Springer.

KONLEA L. & JANNIDISA F. (2020). Domain and task adaptive pretraining for language models. *Proceedings http ://ceur-ws. org ISSN*, **1613**, 0073.

LABRAK Y., BAZOGE A., DUFOUR R., DAILLE B., GOURRAUD P.-A., MORIN E. & ROUVIER M. (2022). FrenchMedMCQA : A French multiple-choice question answering dataset for medical domain. In *Proceedings of the 13th International Workshop on Health Text Mining and Information Analysis (LOUHI)*, p. 41–46, Abu Dhabi, United Arab Emirates (Hybrid) : Association for Computational Linguistics.

LABRAK Y., BAZOGE A., DUFOUR R., ROUVIER M., MORIN E., DAILLE B. & GOURRAUD P.-A. (2023). Drbert : A robust pre-trained model in french for biomedical and clinical domains.

LUO R., SUN L., XIA Y., QIN T., ZHANG S., POON H. & LIU T.-Y. (2022). BioGPT : generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, **23**(6). bbac409, DOI : 10.1093/bib/bbac409.

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. DOI : 10.18653/v1/2020.acl-main.645.

MATTHEWS B. W. (1975). Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, **405**(2), 442–451.

MUENNIGHOFF N., WANG T., SUTAWIKA L., ROBERTS A., BIDERMAN S., SCAO T. L., BARI M. S., SHEN S., YONG Z.-X., SCHOELKOPF H., TANG X., RADEV D., AJI A. F., ALMUBARAK K., ALBANIE S., ALYAFEAI Z., WEBSON A., RAFF E. & RAFFEL C. (2022). Crosslingual generalization through multitask finetuning.

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

PEDREGOSA F., VAROQUAUX G., GRAMFORT A., MICHEL V., THIRION B., GRISEL O., BLONDEL M., PRETTENHOFER P., WEISS R., DUBOURG V., VANDERPLAS J., PASSOS A., COURNAPEAU D., BRUCHER M., PERROT M. & DUCHESNAY E. (2011). Scikit-learn : Machine learning in Python. *Journal of Machine Learning Research*, **12**, 2825–2830.

RADFORD A., WU J., CHILD R., LUAN D., AMODEI D. & SUTSKEVER I. (2019). Language models are unsupervised multitask learners.

TAYLOR R., KARDAS M., CUCURULL G., SCIALOM T., HARTSHORN A., SARAVIA E., POULTON A., KERKEZ V. & STOJNIC R. (2022). Galactica : A large language model for science.

TIEDEMANN J. & THOTTINGAL S. (2020). OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, p. 479–480, Lisboa, Portugal : European Association for Machine Translation.

TURC I., CHANG M., LEE K. & TOUTANOVA K. (2019). Well-read students learn better : The impact of student initialization on knowledge distillation. *CoRR*, **abs/1908.08962**.

WU H., XU K., SONG L., JIN L., ZHANG H. & SONG L. (2021). Domain-adaptive pretraining methods for dialogue understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2 : Short Papers)*, p. 665–669, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.acl-short.84.

XIONG W., DU J., WANG W. Y. & STOYANOV V. (2020). Pretrained encyclopedia : Weakly supervised knowledge-pretrained language model. In *International Conference on Learning Representations*.

# A Données

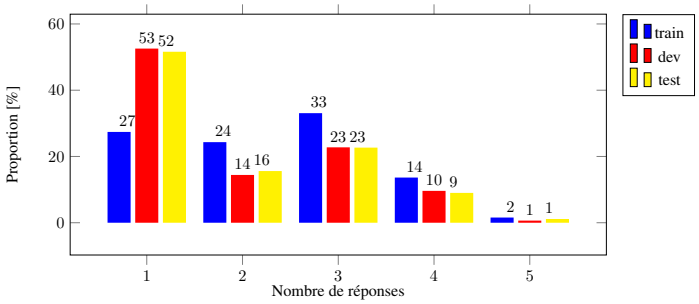


FIGURE 1 – Répartition du nombre de réponses dans les jeux de données fournis.

# B Détails de modélisation

## B.1 Expressions régulières (ER)

Le classifieur binaire à expressions régulières utilisé fonctionne comme suit :

- `regexp_1 = "quel|quelle|lequel|laquelle|celui|celle"`
- `regexp_2 = "quels|quelles|lesquels|lesquelles|ceux|celles"`
- `regexp_3 = "\(s\)|\(nt\)"`
- `Détecteur = match(regexp_1) ET NON match(regexp_2) ET NON match(regexp_3)`
- `Classifieur = "simple" SI Détecteur == VRAI SINON "multiple"`

## B.2 *Topic modelling*

***Latent Semantic Indexing* (LSI)** Cet algorithme s'appuie sur la décomposition en valeurs singulières (SVD) de la matrice terme-document $M$ construite à partir d'un corpus : $M = U\Sigma V^\top$, avec $U$ et $V$ unitaires et $\Sigma$ diagonale. Chaque composante singulière figurant dans $U$ permet de relier un terme à un concept, un **thème**. Au-delà de permettre d'identifier des thèmes dans un corpus à partir des seuls mots qui le composent, LSI permet aussi de ne retenir que les thèmes les plus saillants et d'imposer le nombre maximum de termes associé à un thème donné.

***Latent Dirichlet Allocation* (LDA)** Cet algorithme génératif suppose qu'un corpus est le fruit du tirage de mots associés à un nombre de thèmes fixé : c'est un modèle de mélange probabiliste. Les tirages sont supposés suivre une loi de Dirichlet. Ses paramètres sont calibrés par apprentissage statistique.

## B.3 Exemples d'erreurs de classification

- **Exemple n° 1** : « Parmi les affirmations suivantes, une seule est fausse, indiquer **laquelle** : » ;

- **Exemple n° 2** : « Parmi les propositions suivantes, indiquer **celle** qui est exacte. Le crack est une forme : » ;
- **Exemple n° 68** « Parmi les marqueurs suivants, indiquer celui (ceux) qui reflète(nt) l'activité ostéoblastique. » ;
- **Exemple n° 2161** : « Parmi les propositions suivantes concernant Escherichia coli, quelle(s) est(sont) celle(s) qui est(sont) exacte(s) ? ».
- **Exemple n° 1188** : « Concernant la validation des méthodes d'analyse, laquelle de ces propositions est exacte ? ». Cette question appelle explicitement une unique réponse, mais la correction en donne deux.
- **Exemple n° 1585** : « Quelle est la morphologie de Fasciola hepatica ? » admet trois bonnes réponses. Si la question ne le laisse pas présupposer, les réponses possibles donnent un indice : trois thèmes semblent se dégager parmi les assertions.
- **Exemple n° 1650** : « Dans le cadre d'une enquête épidémiologique [...] » appelle une seule réponse, alors que deux sont en réalité acceptées.

## B.4 Résultats détaillés

Le tableau 3 montre l'ensemble des résultats obtenus pour la tâche de classification à l'aide des approches mises en place pour la classification des questions simples et multiples.

Pour décider du modèle pré-entraîné à prendre en compte dans nos expérimentations pour la tâche de *multilabel*, nous avons comparé plusieurs modèles sans pré-supposer de la prévalence de l'adéquation de la langue de pré-entraînement. Le tableau 4 nous montre une autre preuve, s'il en fallait une, que la langue n'est pas le critère principal pour ces données de spécialité pharmaceutique. Il s'en dégage alors deux candidats : le BERT-mini et le BERT-small. Le choix du BERT-mini s'est ensuite décidé à partir de leurs performances respectives après l'application de la seconde phase de pré-entraînement par *Masked Language Modelling*. Lors de celle-ci, BERT-mini a donné les meilleurs résultats, tandis que CamemBERT adapté a semblé faire preuve de *catastrophic forgetting*. Nous supposons que cela est dû au prior très éloigné du domaine de spécialité, qu'un modèle plus compact et à l'espace vectoriel moindre dans son état caché (256 au lieu de 512) favorise une adaptation plus efficace. Bien qu'il serait intéressant de mettre en place des tests supplémentaires à ce propos, nous nous sommes limités à ces derniers dans le cadre de ce défi.

| Tâche | Classe | Précision | Rappel | $F_1$-score | Support |
|---|---|---|---|---|---|
| Binaire (ER) | Simple | 1,00 | 0,79 | 0,88 | 164 |
| | Multiple | 0,81 | 1,00 | 0,89 | 148 |
| | Macro moyenne | 0,90 | 0,89 | 0,89 | 312 |
| Binaire (LDA) | Simple | 0,47 | 0,26 | 0,34 | 164 |
| | Multiple | 0,45 | 0,68 | 0,54 | 148 |
| | Macro moyenne | 0,46 | 0,47 | 0,44 | 312 |
| Binaire (RL) | Simple | 0,99 | 0,90 | 0,94 | 164 |
| | Multiple | 0,90 | 0,99 | 0,94 | 148 |
| | Macro moyenne | 0,94 | 0,94 | 0,94 | 312 |
| Multiple (LDA) | 1 | 0,47 | 0,26 | 0,34 | 164 |
| | 2 | 0,12 | 0,13 | 0,12 | 45 |
| | 3 | 0,11 | 0,03 | 0,04 | 71 |
| | 4 | 0,00 | 0,00 | 0,00 | 30 |
| | 5 | 0,01 | 1,00 | 0,03 | 2 |
| | Macro moyenne | 0,14 | 0,28 | 0,11 | 312 |
| Multiple (ER + LDA) | 1 | 0,63 | 0,50 | 0,56 | 164 |
| | 2 | 0,15 | 0,18 | 0,16 | 45 |
| | 3 | 0,14 | 0,03 | 0,05 | 71 |
| | 4 | 0,00 | 0,00 | 0,00 | 30 |
| | 5 | 0,00 | 0,00 | 0,00 | 2 |
| | Macro moyenne | 0,18 | 0,14 | 0,15 | 312 |

TABLE 3 – Performances des différentes combinaisons d'approches (ER, LDA, RL) pour les tâches de classifications binaire (réponse simple ou multiple) et multiple (nombre de réponses attendues, de 1 à 5) sur le jeu de développement.

| Modèle pré-entraîné | Couches | États cachés | Hamming↑ | Loss↓ |
|---|---|---|---|---|
| bert-tiny | 2 | 128 | 0.251 | 0.635 |
| bert-mini | 4 | 256 | 0.269 | 0.624 |
| bert-small | 4 | 512 | **0.283** | **0.619** |
| bert-medium | 8 | 512 | 0.277 | **0.619** |
| camembert-base | 12 | 512 | 0.210 | 0.682 |
| camembert-base-wikipedia-4gb | 12 | 512 | 0.195 | 0.682 |

TABLE 4 – Évaluation sur les données de validation pour différentes tailles de modèles BERT pré-entraînés puis évalués sur la tâche principale.

*Actes de CORIA-TALN 2023*
*Paris, France, 5 au 9 juin 2023*

# Participation d'EDF R&D au défi DEFT 2023 : réponses automatiques à des questionnaires à choix multiples à l'aide de « Larges Modèles de Langue »

Meryl Bothua, Leila Hassani, Marie Jubault, Philippe Suignard *
EDF R&D
7, boulevard Gaspard Monge 91120 Palaiseau
prenom.nom@edf.fr

RÉSUMÉ

Ce papier présente la participation d'EDF R&D à la campagne d'évaluation DEFT 2023. Notre équipe a participé à la tâche de réponse automatique à des questions à choix multiples issus d'annales d'examens en pharmacie en français. Le corpus utilisé est FrenchMedMCQA. Nous avons testé des Large Language Models pour générer des réponses.

ABSTRACT

**EDF RD Participation to DEFT 2023**

This paper describes the participation of EDF R&D at DEFT 2023. Our team worked on the Multiple Choice Questions Answering task proposed. The corpus was FrenchMedMCQA and is composed of questions from pharmacy exam annals. We used Large Language Models to predict the answers.

MOTS-CLÉS : Gros modèles de langue, Bloom, BloomZ, ChatGPT, Questions à choix multiples, pharmacie.

KEYWORDS: Large Language Models, Bloom, BloomZ, ChatGPT, MCQA, pharmacy.

## 1 Introduction

L'objectif du défi DEFT 2023 est de répondre automatiquement à des questionnaires à choix multiples issus d'annales d'examens de pharmacie. Le corpus utilisé, FrenchMedMCQA (Labrak *et al.*, 2022), se compose de questions fermées en français provenant d'annales d'examens de pharmacie. Chaque question contient : un identifiant, la question posée, cinq réponses possibles et l'ensemble des réponse(s) correcte(s).

À la différence des défis des années passées, il est impossible de répondre aux questions posées sans avoir accès à des connaissances extérieures. Plusieurs possibilités existent comme utiliser des bases de données spécialisées dans le domaine concerné (comme PubMed, par exemple), utiliser le Web comme source de données ou bien utiliser les Larges Modèles de Langues (LLM en anglais) construits sur de grosses quantités de données et qui ont ainsi « acquis » un certain nombre de connaissances (Wei *et al.*, 2022).

∗. Cités par ordre alphabétique

En participant à ce défi, nous avons choisi de tester plusieurs "Larges Modèles de Langue" comme GPT3 et Bloom, ainsi que leur sur-couche ChatGPT et BloomZ.

# 2 ChatGPT

L'arrivée de ChatGPT (OpenAI, 2021) a provoqué une onde choc assez impressionnante dans le monde du TAL. Cette technologie est très prometteuse et offre de belles perspectives pour la résolution des tâches classiques du TAL mais également pour de nouvelles. Nous en voulons pour preuve les nombreuses solutions concurrentes qui commencent à émerger comme « Open Assistant [1] ». Plusieurs articles utilisent déjà cette technologie pour répondre à des questions à choix multiples tels que (Kung *et al.*, 2023) et (Liévin *et al.*, 2022). Il nous a semblé intéressant d'utiliser le cadre de ce défi pour tester cette technologie en mode *zero-shot*, c'est-à-dire sans entraînement particulier et en demandant directement à ChatGPT quelles étaient les bonnes réponses parmi les cinq réponses possibles.

## 2.1 Aspect technique

ChatGPT se présente comme un modèle que l'on peut interroger avec différents paramètres. Deux distinctions principales existent : le mode "*chat*" et le mode "*completion*". Dans le mode "*completion*", on lui indique un début de phrase qu'il va venir compléter ainsi que le nombre de mots que doit fournir ChatGPT, ce qui est un peu la "fonctionnalité de base" d'un LLM. Dans le mode "*chat*", la réponse sera plus complète et prendra plus de temps. C'est ce mode que nous avons utilisé.

Le modèle utilisé est `gpt-3.5-turbo`, avec 0 comme valeur de température. Cette valeur est comprise entre 0 et 2. Plus la valeur est proche de 0 et plus la réponse sera précise, plus elle est proche de 2 et plus la réponse sera "créative".

Enfin, plusieurs façons d'interroger le modèle lui-même sont possibles dont cURL, pour *client URL request library*, une interface en ligne de commande pour requêter des ressources informatiques accessibles sur un réseau, le format d'échange de données étant JSON.

## 2.2 Formatage des données

Une des clés de l'utilisation de cette technologie est la manière de poser la question (*prompt engineering*). La manière de poser la question est ici inspirée par (Liévin *et al.*, 2022) : « Q : » suivi du texte de la question, puis par les 5 réponses possibles précédées de « A) », « B) », etc. avec « \n » venant séparer les différents champs, comme dans l'exemple suivant :

```
Q : Texte de la question \n A) Réponse 1 \n B) Réponse 2 \n C)
Réponse 3 \n D) Réponse 4 \n E) Réponse 5.
```

ChatGPT fournit une réponse qu'il faut ensuite analyser via des expressions régulières. Sur les 2171 questions du corpus d'entraînement, les réponses ont pour type :

— « `B) Réponse 2 \n C) Réponse 3` » (s'il a jugé que B et C étaient les bonnes réponses) (2063 cas sur 2171). Il faut donc récupérer les lettres majuscules A,B,C, D ou E suivi d'une parenthèse pour obtenir la réponse : « BC » ;

1. disponible à l'adresse : https ://open-assistant.io/

— « `toutes les (réponses|propositions|affirmations) sont (vraies|possibles|correctes|exactes)` ». (37 cas sur 2171). Dans ces cas-là, on génère la suite « ABCDE », puisque toutes les réponses sont justes ;
— « `Réponse : A et D sont exactes.\n \n Explication : ...` ». Dans ce cas, il faudrait extraire A et D, mais qui ne sont pas suivis d'une parenthèse. Le mot « Réponse » pouvant d'ailleurs être facultatif ;
— D'autres formes de réponses sont également possibles, mais sans qu'un lien puisse facilement être établi entre les réponses possibles et la réponse fournie (pas de lettre, ni de parenthèse). Dans ces deux derniers cas de figure (71 cas sur 2171), les expressions régulières retournent une chaîne de caractère vide, la réponse « ABCDE » est donc attribuée arbitrairement.

## 2.3 Résultats obtenus sur le corpus d'entraînement

Les scores obtenus par ChatGPT sur le corpus d'entraînement en *zero-shot*, sont de 29,13% en EMR (Exact Match Ratio) et 58,26% pour Hamming.

Quand on analyse le corpus d'entraînement, les réponses sont équitablement réparties. Les nombres de réponses A, B, C et D sont pratiquement égales. Par contre, le nombre de réponses à E est légèrement inférieur. Si on analyse maintenant les résultats obtenus par ChatGPT sur ce même corpus, on constate, à peu près la même répartition, avec un nombre de réponses à E encore inférieur. Le nombre moyen de réponses apportées par ChatGPT est très légèrement inférieur au nombre réel de réponses dans le corpus d'entraînement (2,22 réponses contre 2,37 réponses par question).

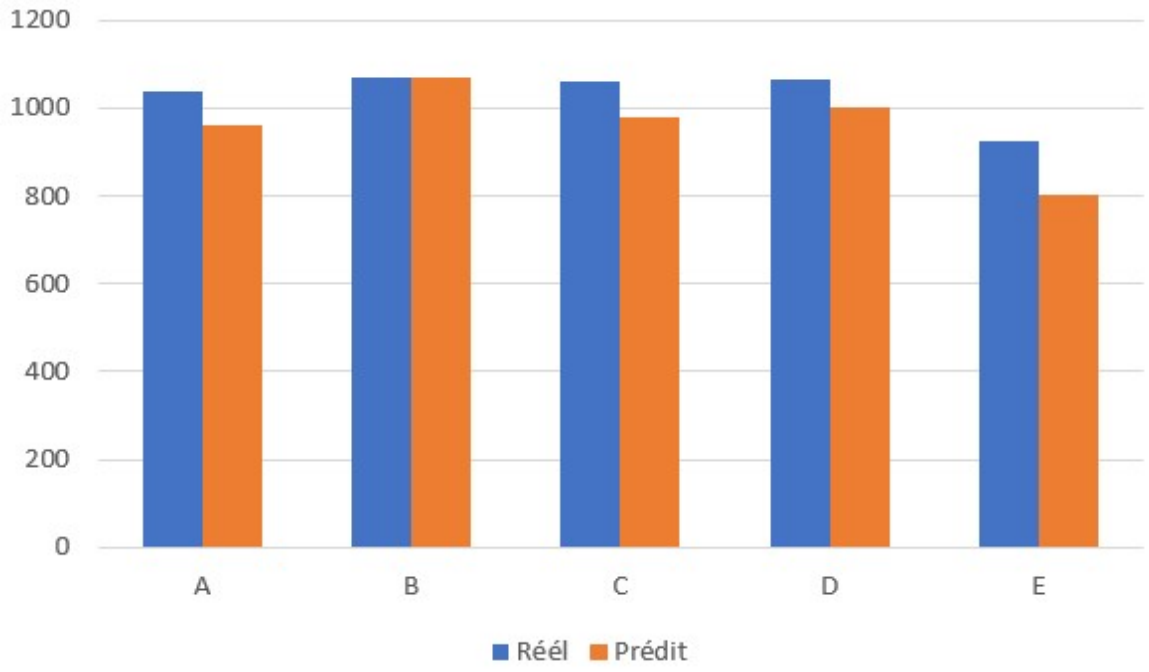


FIGURE 1 – Comparaison entre les réponses prédites par ChatGPT et les bonnes réponses

# 3 BloomZ

BloomZ (Workshop *et al.*, 2023) est un grand modèle de langue Open-Source et gratuit entraîné dans le cadre d'un projet international et collaboratif appelé BigScience, partiellement financé par le gouvernement français. Plus particulièrement, BloomZ est une itération de Bloom entraîné spécifiquement sur les tâches de compréhension d'instructions en *zero-shot*, c'est-à-dire sans exemple de résolution de la tâche attendue dans le prompt en entrée.

## 3.1 BloomZ 1.1B et 7.1B en inférence

Pour des questions de ressources matérielles, nous avons travaillé sur l'évaluation de BloomZ de 1.1 et 7.1 milliards de paramètres, et non pas sur le modèle le plus conséquent, qui atteint 176 milliards de paramètres et requiert des GPUs très performants. Le chargement du modèle 1.1B a été effectué à l'aide d'un GPU de 20Go, tandis que le modèle 7.1 milliards a dû être chargé à l'aide d'un GPU de 40Go.

Nous les avons testés directement en inférence avec du *zero-shot* et en *few-shot*. Les résultats sont, sans grande surprise, légèrement meilleurs avec le plus gros modèle.

Le tableau ci-dessous récapitule les résultats obtenus pour chaque *run*, à partir du Hamming Score (HS) et de l'Exact Match Ratio (EMR).

| | 1.1B | 7.1B |
|---|---|---|
| *Zero-Shot* | HS : 0,53<br>EMR : 0,08 | **HS : 0,57**<br>**EMR : 0,1** |
| *One-Shot* | HS : 0,53<br>EMR : 0,08 | HS : 0,56<br>EMR : 0,09 |
| *Two-Shot* | HS : 0,5<br>EMR : 0,05 | HS : 0,55<br>EMR : 0,09 |
| *Three-Shot* | HS : 0,52<br>EMR : 0,07 | HS : 0,55<br>EMR : 0,09 |

TABLE 1 – Résultats obtenus pour du ZSL et du FSL en inférence avec BloomZ 1.1 & 7.1B

Les meilleurs résultats pour le Hamming Score et l'EMR sont donc avec des prompts en *zero-shot*, bien que les différences de résultats sont parfois marginales. De plus, nous avons remarqué que pour le modèle 7.1B particulièrement, le choix des exemples envoyés au modèle a un impact plus ou moins fort. Par exemple, dans le cas du *one-shot learning* et avec le même prompt, les scores sont légèrement meilleurs si l'exemple envoyé au modèle est une question avec plusieurs bonnes réponses. Les résultats en *one-shot* avec une question qui n'accepte qu'une bonne réponse sont les suivantes :

* **BloomZ 1.1B** : 0,53 pour le Hamming Score, et 0,8 pour l'Exact Match Ratio.

* **BloomZ 7.1B** : 0,55 pour le Hamming Score, et 0,8 pour l'Exact Match Ratio.

La différence n'est qu'infime dans le cas de l'EMR, mais plus significative dans le cas du HS.

### 3.2 BloomZ 1.1B avec apprentissage

Afin d'adapter le modèle BloomZ à la tâche de MCQA sur le domaine pharmaceutique, nous avons fait le choix d'outrepasser les méthodes classiques de *Fine-Tuning* afin d'expérimenter avec le *Prompt Tuning* (Lester *et al.*, 2021), un nouvelle méthode d'adaptation des grands modèles de langue qui ne requiert pas de modifier les paramètres du modèle de base.

Pour ce faire, nous avons utilisé la librairie PEFT (*Parameter-Efficient Fine-Tuning*) (Sourab Mangrulkar, 2022). Nous avons tenté le Prompt Tuning de BloomZ-1b1 seulement. Les résultats sont décevants : évalué sur des prompts en One-Shot, il n'améliore que très légèrement les résultats du Hamming Score (0.54 contre 0.53 pour le modèle 1b1 non entraîné), et n'améliore pas du tout l'Exact Match Ratio, qui reste de 0.08.

## 4 Résultats

Les résultats obtenus sur la tâche principale sont très satisfaisants : BloomZ obtient des scores équivalents aux méthodes présentées dans (Labrak *et al.*, 2022) et ChatGPT présente des scores environ 2 fois supérieurs, ce qui prouve sa qualité et justifie l'engouement qu'il suscite.

| Système | Hamming | EMR |
|---|---|---|
| ChatGPT | **64,40** | **46,46** |
| BloomZ - run1 | 26,34 | 14,63 |
| BloomZ - run2 | 35,90 | 15,27 |
| BloomZ - run3 | 37,93 | 12,70 |

TABLE 2 – Résultats obtenus sur la tâche principale

Pour BloomZ, le *run1* correspond à un test avec du *zero-shot*, alors que les *run2* et *run3* correspondent à des tests en *one-shot* (avec des exemples différents).

Pour la tâche annexe (identifier le nombre de réponses supposément justes pour une question donnée), nous nous sommes contentés de calculer le nombre de réponses considérées comme étant justes par ChatGPT.

| Système | Accuracy | F1-score macro |
|---|---|---|
| ChatGPT | 65,92 | 44,36 |

TABLE 3 – Résultats obtenus sur la tâche annexe

## 5 Conclusion

La participation à la campagne DEFT 2023 nous a permis de tester les nouveaux *Large Language Models*, à la fois fermés via API, tel GPT3, et ouverts tels Bloom et BloomZ. Ces modèles, alliés aux mécanismes de *Prompt Engineering* sont très prometteurs pour le traitement des données textuelles au sein de EDF Commerce et d'autres entités du groupe EDF.

# Références


KUNG T. H., CHEATHAM M., MEDENILLA A., SILLOS C., DE LEON L., ELEPAÑO C., MADRIAGA M., AGGABAO R., DIAZ-CANDIDO G., MANINGO J. *et al.* (2023). Performance of chatgpt on usmle : Potential for ai-assisted medical education using large language models. *PLoS digital health*, **2**(2), e0000198.

LABRAK Y., BAZOGE A., DUFOUR R., DAILLE B., GOURRAUD P.-A., MORIN E. & ROUVIER M. (2022). FrenchMedMCQA : A French multiple-choice question answering dataset for medical domain. In *Proceedings of the 13th International Workshop on Health Text Mining and Information Analysis (LOUHI)*, p. 41–46, Abu Dhabi, United Arab Emirates (Hybrid) : Association for Computational Linguistics.

LESTER B., AL-RFOU R. & CONSTANT N. (2021). The power of scale for parameter-efficient prompt tuning.

LIÉVIN V., HOTHER C. E. & WINTHER O. (2022). Can large language models reason about medical questions ? *arXiv preprint arXiv :2207.08143*.

OPENAI (2021). Gpt-3.5 language model. [online]. disponible sur https ://openai.com/gpt-3-5/.

SOURAB MANGRULKAR, SYLVAIN GUGGER L. D. Y. B. S. P. (2022). Peft : State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft.

WEI J., TAY Y., BOMMASANI R., RAFFEL C., ZOPH B., BORGEAUD S., YOGATAMA D., BOSMA M., ZHOU D., METZLER D. *et al.* (2022). Emergent abilities of large language models. *arXiv preprint arXiv :2206.07682*.

WORKSHOP B., :, SCAO T. L., FAN A., AKIKI C., PAVLICK E., ILIĆ S., HESSLOW D., CASTAGNÉ R., LUCCIONI A. S., YVON F., GALLÉ M., TOW J., RUSH A. M., BIDERMAN S., WEBSON A., AMMANAMANCHI P. S., WANG T., SAGOT B., MUENNIGHOFF N., DEL MORAL A. V., RUWASE O., BAWDEN R., BEKMAN S., MCMILLAN-MAJOR A., BELTAGY I., NGUYEN H., SAULNIER L., TAN S., SUAREZ P. O., SANH V., LAURENÇON H., JERNITE Y., LAUNAY J., MITCHELL M., RAFFEL C., GOKASLAN A., SIMHI A., SOROA A., AJI A. F., ALFASSY A., ROGERS A., NITZAV A. K., XU C., MOU C., EMEZUE C., KLAMM C., LEONG C., VAN STRIEN D., ADELANI D. I., RADEV D., PONFERRADA E. G., LEVKOVIZH E., KIM E., NATAN E. B., TONI F. D., DUPONT G., KRUSZEWSKI G., PISTILLI G., ELSAHAR H., BENYAMINA H., TRAN H., YU I., ABDULMUMIN I., JOHNSON I., GONZALEZ-DIOS I., DE LA ROSA J., CHIM J., DODGE J., ZHU J., CHANG J., FROHBERG J., TOBING J., BHATTACHARJEE J., ALMUBARAK K., CHEN K., LO K., WERRA L. V., WEBER L., PHAN L., ALLAL L. B., TANGUY L., DEY M., MUÑOZ M. R., MASOUD M., GRANDURY M., ŠAŠKO M., HUANG M., COAVOUX M., SINGH M., JIANG M. T.-J., VU M. C., JAUHAR M. A., GHALEB M., SUBRAMANI N., KASSNER N., KHAMIS N., NGUYEN O., ESPEJEL O., DE GIBERT O., VILLEGAS P., HENDERSON P., COLOMBO P., AMUOK P., LHOEST Q., HARLIMAN R., BOMMASANI R., LÓPEZ R. L., RIBEIRO R., OSEI S., PYYSALO S., NAGEL S., BOSE S., MUHAMMAD S. H., SHARMA S., LONGPRE S., NIKPOOR S., SILBERBERG S., PAI S., ZINK S., TORRENT T. T., SCHICK T., THRUSH T., DANCHEV V., NIKOULINA V., LAIPPALA V., LEPERCQ V., PRABHU V., ALYAFEAI Z., TALAT Z., RAJA A., HEINZERLING B., SI C., TAŞAR D. E., SALESKY E., MIELKE S. J., LEE W. Y., SHARMA A., SANTILLI A., CHAFFIN A., STIEGLER A., DATTA D., SZCZECHLA E., CHHABLANI G., WANG H., PANDEY H., STROBELT H., FRIES J. A., ROZEN J., GAO L., SUTAWIKA L., BARI M. S., AL-SHAIBANI M. S., MANICA M., NAYAK N., TEEHAN R., ALBANIE S., SHEN S., BEN-DAVID S., BACH S. H., KIM T., BERS T., FEVRY T., NEERAJ T., THAKKER U., RAUNAK

V., TANG X., YONG Z.-X., SUN Z., BRODY S., URI Y., TOJARIEH H., ROBERTS A., CHUNG H. W., TAE J., PHANG J., PRESS O., LI C., NARAYANAN D., BOURFOUNE H., CASPER J., RASLEY J., RYABININ M., MISHRA M., ZHANG M., SHOEYBI M., PEYROUNETTE M., PATRY N., TAZI N., SANSEVIERO O., VON PLATEN P., CORNETTE P., LAVALLÉE P. F., LACROIX R., RAJBHANDARI S., GANDHI S., SMITH S., REQUENA S., PATIL S., DETTMERS T., BARUWA A., SINGH A., CHEVELEVA A., LIGOZAT A.-L., SUBRAMONIAN A., NÉVÉOL A., LOVERING C., GARRETTE D., TUNUGUNTLA D., REITER E., TAKTASHEVA E., VOLOSHINA E., BOGDANOV E., WINATA G. I., SCHOELKOPF H., KALO J.-C., NOVIKOVA J., FORDE J. Z., CLIVE J., KASAI J., KAWAMURA K., HAZAN L., CARPUAT M., CLINCIU M., KIM N., CHENG N., SERIKOV O., ANTVERG O., VAN DER WAL O., ZHANG R., ZHANG R., GEHRMANN S., MIRKIN S., PAIS S., SHAVRINA T., SCIALOM T., YUN T., LIMISIEWICZ T., RIESER V., PROTASOV V., MIKHAILOV V., PRUKSACHATKUN Y., BELINKOV Y., BAMBERGER Z., KASNER Z., RUEDA A., PESTANA A., FEIZPOUR A., KHAN A., FARANAK A., SANTOS A., HEVIA A., UNLDREAJ A., AGHAGOL A., ABDOLLAHI A., TAMMOUR A., HAJIHOSSEINI A., BEHROOZI B., AJIBADE B., SAXENA B., FERRANDIS C. M., CONTRACTOR D., LANSKY D., DAVID D., KIELA D., NGUYEN D. A., TAN E., BAYLOR E., OZOANI E., MIRZA F., ONONIWU F., REZANEJAD H., JONES H., BHATTACHARYA I., SOLAIMAN I., SEDENKO I., NEJADGHOLI I., PASSMORE J., SELTZER J., SANZ J. B., DUTRA L., SAMAGAIO M., ELBADRI M., MIESKES M., GERCHICK M., AKINLOLU M., MCKENNA M., QIU M., GHAURI M., BURYNOK M., ABRAR N., RAJANI N., ELKOTT N., FAHMY N., SAMUEL O., AN R., KROMANN R., HAO R., ALIZADEH S., SHUBBER S., WANG S., ROY S., VIGUIER S., LE T., OYEBADE T., LE T., YANG Y., NGUYEN Z., KASHYAP A. R., PALASCIANO A., CALLAHAN A., SHUKLA A., MIRANDA-ESCALADA A., SINGH A., BEILHARZ B., WANG B., BRITO C., ZHOU C., JAIN C., XU C., FOURRIER C., PERIÑÁN D. L., MOLANO D., YU D., MANJAVACAS E., BARTH F., FUHRIMANN F., ALTAY G., BAYRAK G., BURNS G., VRABEC H. U., BELLO I., DASH I., KANG J., GIORGI J., GOLDE J., POSADA J. D., SIVARAMAN K. R., BULCHANDANI L., LIU L., SHINZATO L., DE BYKHOVETZ M. H., TAKEUCHI M., PÀMIES M., CASTILLO M. A., NEZHURINA M., SÄNGER M., SAMWALD M., CULLAN M., WEINBERG M., WOLF M. D., MIHALJCIC M., LIU M., FREIDANK M., KANG M., SEELAM N., DAHLBERG N., BROAD N. M., MUELLNER N., FUNG P., HALLER P., CHANDRASEKHAR R., EISENBERG R., MARTIN R., CANALLI R., SU R., SU R., CAHYAWIJAYA S., GARDA S., DESHMUKH S. S., MISHRA S., KIBLAWI S., OTT S., SANG-AROONSIRI S., KUMAR S., SCHWETER S., BHARATI S., LAUD T., GIGANT T., KAINUMA T., KUSA W., LABRAK Y., BAJAJ Y. S., VENKATRAMAN Y., XU Y., XU Y., XU Y., TAN Z., XIE Z., YE Z., BRAS M., BELKADA Y. & WOLF T. (2023). Bloom : A 176b-parameter open-access multilingual language model.

*Actes de CORIA-TALN 2023*
*Paris, France, 5 au 9 juin 2023*

# LIS@DEFT'23 : les LLMs peuvent-ils répondre à des QCM ? (a) oui; (b) non; (c) je ne sais pas.

Benoit Favre[1]
(1) Aix Marseille Université, CNRS, LIS, Marseille, France. benoit.favre@lis-lab.fr

RÉSUMÉ

Cet article présente un ensemble d'expériences sur la tâche de réponse à des questions à choix multiples de DEFT 2023. Des grands modèles de langage sont amorcés avec les questions afin de collecter les réponses générées. Les résultats montrent que les modèles ouverts sans affinage obtiennent des performances similaires à celles d'un système supervisé fondé sur BERT, et que l'affinage sur les données de la tâche apporte des améliorations.

ABSTRACT

**LIS@DEFT'23 : can LLMs answer MCQs ? (a) yes ; (b) no ; (c) I don't know.**

This paper presents a set of experiments on the multiple-choice question answering task of DEFT 2023. Large language models are prompted with the questions and responses are collected from generated completions. Results show that open models without refinement achieve similar performance to that of a supervised system based on BERT, and that refinement on the task data brings improvements.

MOTS-CLÉS : Questions à choix multiples, DEFT, grands modèles de langage, Amorce, GPT, Affinage LoRA.

KEYWORDS: Multiple choice questions, DEFT, large language models, Prompt, GPT, LoRA finetuning.

## 1 Introduction

Les récents progrès en traitement automatique des langues ont montré que les grands modèles de langage (LLM, pour large language models) ont des propriétés de généralisation qui leur permettent d'adresser un grand nombre de tâches, y compris sans y avoir été soumis en entraînement, et ont la capacité de puiser dans les connaissances accumulées dans leurs données d'entraînement pour construire des raisonnements simples. Dans ce travail nous nous intéressons aux capacités de ces modèles pour répondre à des questions à choix multiples dans le domaine médical. Cet article présente en particulier le système proposé par le LIS [1] pour la tâche partagée DEFT 2023.

Nous tentons de répondre aux questions suivantes :

1. Quelles sont les différences de performances entre modèles ouverts sur la tâche de QCM ?
2. Quel est le lien entre taille des LLM et performances attendues ?
3. L'affinage des modèles à bas coût est-il bénéfique pour traiter la tâche ?

1. Code source sur https://gitlab.lis-lab.fr/benoit.favre/deft2023-llm

```
{
  "id": "e3e35ba581919533a9d7e75fa6437c201837f4cc6698c5bb2e7c8fd2580366f8",
  "question": "Parmi les propositions suivantes concernant le métabolisme
    du calcium, indiquer celles qui sont exactes.",
  "answers":{
    "a": "La majorité du calcium de l'organisme se trouve dans le plasma",
    "b": "L'hormone parathyroïdienne favorise la réabsorption tubulaire
      du calcium",
    "c": "La sécrétion de calcitonine est régulée par le calcium ionisé",
    "d": "La vitamine D favorise l'absorption intestinale du calcium",
    "e": "L'hormone parathyroïdienne inhibe l'action de la 1-alpha
      hydroxylase rénale"
  },
  "correct_answers": ["b","c","d"],
  "subject_name": "pharmacie",
  "type": "multiple",
  "nbr_correct_answers": 3
}
```


FIGURE 1 – Exemple d'instance du corpus DEFT 2023, contenant la question, les réponses possibles et les réponses correctes au format JSON.

L'article présente d'abord la tâche (section 2), il détaille ensuite l'approche et la méthode employée (section 3), puis présente les expériences et résultats associés (section 4), avant de conclure.

# 2 Réponse à des QCM

## 2.1 Travaux reliés

La tâche de question-réponse est une tâche emblématique du TAL et a été traitée principalement sous trois formes : la génération de réponses à des questions ouvertes (*question answering*), souvent des questions de connaissances générales, la localisation de réponses dans un texte (*machine reading comprehension*), et la sélection de réponses à des questions à choix multiples. De nombreuses revues de littérature présentent les avancées dans le domaine (Soares & Parreiras, 2020; Allam & Haggag, 2012; Hao *et al.*, 2022; Bouziane *et al.*, 2015).

Les QCM se présentent sous la forme d'une question suivie d'un certain nombre de réponses possibles, dont un sous-ensemble peut être correct. Les méthodes développées par la communauté pour répondre automatiquement à des QCM ont suivi les avancées en TAL. Par exemple, la famille de réseaux de neurones pré-entraînés BERT a été exploitée en mettant en compétition plusieurs encodeurs qui prennent en entrée la question et un choix possible et génèrent une réponse binaire, des entrées parfois complétées par un contexte trouvé par recherche d'information dans un corpus de connaissances du domaine cible (Pal *et al.*, 2022; Roy *et al.*, 2021; Labrak *et al.*, 2022; Le Berre & Langlais, 2020). La génération précédente d'approches exploitait les réseaux de neurones récurrents comme les LSTM dans des conditions similaires (Guo *et al.*, 2017).

Les grands modèles de langage, après affinage sur des instructions correspondant à des tâches variées, sont devenus une méthode versatile pour traiter bon nombre de tâches de TAL dont la réponse à des

QCM. Ils offrent des performances prometteuses dans le domaine médical (Nori *et al.*, 2023), mais les performances ne sont pas toujours exceptionnelles, comme par exemple dans le cas de questions sur du code source, un type de contenu peu présent dans les données de pré-entraînement (Savelka *et al.*, 2023).

### 2.2 La tâche DEFT 2023

La campagne d'évaluation DEFT 2023 consiste en la génération automatique de réponses à des QCM provenant d'annales d'examens de pharmacie. Elle tire parti du corpus FrenchMedMCQA (Labrak *et al.*, 2022), calqué sur le corpus MedMCQA en anglais (Pal *et al.*, 2022). Ce corpus contient 3105 questions accompagnées de 5 choix possibles, dont au moins un est vrai. Un exemple est donné dans la figure 1. Ce corpus est divisé en entraînement (2171 instances), validation (312) et test (622). Deux tâches sont proposées : répondre aux questions (principale) , et déterminer le nombre de réponses correctes (annexe). Nous avons participé à la première tâche et dérivé des sorties pour la seconde à partir de celles de la première. Pour la tâche principale, les performances des systèmes sont évaluées avec deux métriques : la correspondance exacte entre les réponses produites (*exact match ratio*, EMR), et la distance de hamming entre les réponses produites et les réponses de référence (*Hamming score*). Nous rapportons dans cet article principalement la métrique EMR qui nous apparaît canonique pour la tâche. La tâche annexe est évaluée selon le taux de bonne classification sur le nombre de réponses.

## 3 Approche : amorçage et affinage de grands modèles de langage

### 3.1 Modèles et affinage

Les modèles de langage autoregressifs sont fondés sur l'idée de calculer la probabilité d'un texte étant donnée la langue, ou à défaut étant donné un corpus d'apprentissage. Cette probabilité peut être marginalisée sur les mots selon l'ordre classique de factorisation du début à la fin du texte, puis la probabilité d'un mot étant donné son historique est en général approximée par un contexte de taille fixe. Depuis l'avènement des n-grammes jusqu'aux transformers, les modèles de langage sont donc entraînés à prédire le mot suivant étant donné un contexte. Afin de limiter le nombre d'unités lexicales, les modèles les plus récents travaillent sur des tokens (mots ou facteurs de mots) plutôt que les mots eux-mêmes.

Depuis GPT (Brown *et al.*, 2020), l'architecture la plus courante pour les modèles de langage est le transformer. Il s'agit d'un empilement de couches comprenant un mécanisme d'attention multi-tête permettant de capturer les interactions entre les paires de mots (paires d'interactions, paires de paires, etc. quand on monte dans les couches). Ces modèles peuvent être implémentés de manière relativement efficace sur GPU et sont entraînés par rétro-propagation. Ils peuvent être exploités par amorçage (donner le début d'un texte), puis génération token par token d'une suite, les amorces étant choisies pour que la génération corresponde à une tâche de TAL.

Lorsque l'on augmente le nombre de paramètres des modèles de langage, leur entraînement fait émerger des propriétés remarquables, comme la spécialisation de têtes d'attention à des phénomènes linguistiques connus. À partir de plusieurs milliards de paramètres, les grands modèles de langage (LLM, *large language models*) peuvent faire des tâches sans avoir été spécialisés dessus comme

répondre à des questions, traduire ou résumer un texte, et leur comportement peut être calibré à cet effet en les affinant sur de nombreux exemples d'instructions et réponse attendue. Les plus grands modèles (BLOOM, GPT-3, OPT, PaLM...), avec plusieurs centaines de milliards de paramètres (175 pour GPT-3, 540 pour PaLM), nécessitent une infrastructure spécifique en entraînement et en inférence, avec de nombreux noeuds de calcul équipés chacun de plusieurs GPU de dernière génération.

La série de modèles LLaMa (Touvron *et al.*, 2023) a été rendue publique sous une licence permettant la recherche. Ces modèles, fondés sur l'architecture transformers avec quelques innovations proposées dans GPT-3 et PaLM comme les rotary embeddings (Su *et al.*, 2021), sont disponibles en plusieurs tailles : 7 milliards (7B), 13 milliards (13B), 30 milliards (30B) et 65 milliards (65B) de paramètres. Ils sont entraînés sur un jeu de données constitué de Common Crawl, C4 (variante de pré-traitement de Common Crawl), Wikipédia, Github, ArXiv, Books et StackExchange. Les corpus anglophones sont filtrés pour conserver en majorité cette langue, alors que Wikipédia contient explicitement 20 langues dont le français. Les plus petits modèles sont entraînés sur $10^{12}$ tokens, alors que les plus gros le sont sur $1.4 \times 10^{12}$ tokens, suivant les observations de passage à l'échelle des LLM (Kaplan *et al.*, 2020). Tous ces modèles prennent en compte un historique maximum de 2048 tokens. Leurs paramètres exacts peuvent être trouvés dans l'article original.

L'affinage des LLM demandant beaucoup de mémoire GPU pour conserver l'état de l'optimiseur ainsi que les activations intermédiaires des modèles, plusieurs techniques d'adaptation à moindre coût ont été proposées. L'adaptation à faible rang (LoRA, *Low Rank Adaptation*) repose sur l'idée intuitive que l'ensemble des paramètres n'a pas besoin d'être modifié lors d'un affinage avec relativement peu d'exemples (Hu *et al.*, 2021). Pour cela, on considère le modèle original comme gelé, puis on ajoute à chaque matrice de paramètres une matrice de faible rang calculée comme le produit de deux petites matrices. Le nombre de paramètres ajoutés au modèle est faible et la mémoire requise est raisonnable. Conjuguée à un stockage des paramètres du modèle d'origine sur 8 bits ou moins, cette technique permet d'affiner un modèle 65B sur un seul GPU A100 à 80G de mémoire.

## 3.2 Amorce et extraction de réponse

Dans les expériences présentées dans la suite, nous exploitons une amorce simple dans laquelle sont introduites la question et les réponses possibles, telle que présentée dans la figure 2. Cette amorce est constituée d'une description de la tâche donnant le contexte et une contrainte sur le format de sortie, puis de la question et de la liste des cinq réponses possibles, préfixées par des lettres entre parenthèses, et enfin suivies d'une sollicitation de la réponse, contenant selon les modèles une parenthèse ouvrante pour les forcer à présenter les réponses sous formes de lettres. Cette amorce a été sélectionnée parmi plusieurs candidates prometteuses mentionnant par exemple un "corrigé des épreuves de pharmacie" comme contexte, ou prenant la forme d'une mise en scène de dialogue entre deux interlocuteurs, l'un spécialiste du domaine et l'autre posant des questions, ou encore avec une description en anglais de la tâche comme c'est le cas dans les instructions de BLOOMz (Muennighoff *et al.*, 2022). Ces variantes n'ont pas donné de meilleurs résultats que l'amorce présentée ici lors de tests préliminaires, elle est donc exploitée dans l'ensemble des expériences.

Malgré les contraintes, la réponse d'un modèle de langage à ce type d'amorce peut être variée en termes de formattage et de contenu (copier ou non les réponses choisies, utiliser ou non les parenthèses autour des lettres, utiliser ou non la numérotation des réponses, développer ou non des explications ou un raisonnement avant de donner la réponse). Afin d'extraire la ou les réponses à une question,

Ceci est une question de QCM de l'examen de pharmacie. Réponds avec la ou les lettres correspondant à la bonne réponse.

La diminution d'une unité pH correspond à une concentration en H+ :
(a) 2 fois plus forte.
(b) 10 fois plus faible.
(c) 10 fois plus forte.
(d) 100 fois plus forte.
(e) 100 fois plus faible.
Réponse(s) : (

FIGURE 2 – Exemple d'amorce pour une question de QCM du corpus de développement de DEFT 2023. L'amorce est constituée d'une description de la tâche (bleu), de la question, d'une liste de réponses possibles (magenta), et d'un incitateur (marron) suivi éventuellement une parenthèse ouvrante forçant le début de la réponse (rouge).

nous avons développé une stratégie simple : (1) supprimer tout le texte correspondant à une répétition du début de l'amorce car nombre de modèles continuent de générer une liste de questions/réponses après avoir donné leur réponse, (2) détecter tous les caractères uniques (a-e) entre parenthèses, et (3) si (2) ne renvoie aucune réponse, détecter tous les caractères uniques (a-e) sans parenthèses. Pour les modèles sans affinage, nous exploitons les 32 premiers tokens générés après l'amorce.

Les modèles avec affinage reposent sur le même motif d'amorce et la même stratégie d'extraction de réponses, mais ils sont supervisés avec des réponses qui mentionnent explicitement la lettre et le texte des choix corrects, afin que les modèles associent explicitement les réponses aux choix possibles, et ne se contentent pas de capturer les régularités dans les lettres associés aux choix. La figure 3 montre le format d'une instance pour laquelle le modèle a généré plusieurs choix de réponses. Les modèles affinés sont limités à la génération de 128 tokens.

# 4 Expériences et résultats

Une première série d'expériences teste les compétences sur la tâche d'une série de modèles disponibles pour la recherche sur le hub huggingface [2]. Tous les modèles testés sont des modèles "instruits", qui ont été préalablement affinés sur des jeux de triplets (instruction, entrée et réponse attendue) à réaliser les tâches décrites en amorce plutôt que générer du texte ressemblant à leur corpus d'entraînement. Malgré le fait que le format d'instruction recommandé pour chaque modèle puisse être différent, nous avons conservé l'amorce décrite ci-avant. La plupart des modèles sont entraînés sur des corpus majoritairement en anglais, mais contenant du français, et affinés avec des instructions en anglais. Le tableau 1 présente les résultats en termes d'EMR sur l'ensemble de développement de DEFT 2023 pour les modèles : BLOOMz (Muennighoff *et al.*, 2022), issu du projet BigScience et affiné sur environ 80 millions d'instructions des corpus xP3/xP3-mt synthétisées à partir de corpus de TAL existants, Flan-T5 (Chung *et al.*, 2022) et Flan-UL2 (Tay, 2023), deux modèles affinés sur environ

2. huggingface.co/models/

Ceci est une question de QCM de l'examen de pharmacie. Réponds avec la ou les lettres correspondant à la bonne réponse.

Les complications d'une hépatite virale aiguë peuvent être à plus ou moins long terme :
(a) Une lithiase vésiculaire.
(b) Une hépatite chronique.
(c) Un cancer du foie.
(d) Une cirrhose.
(e) Une pancréatite aiguë.
Réponse(s) : (b) Une hépatite chronique ; (c) Un cancer du foie ; (d) Une cirrhose.

FIGURE 3 – Exemple de sortie sur le test de DEFT 2023 pour le système LLaMa-65B affiné sur les données d'entraînement. La partie générée par le système est explicitée en bleu : les réponses sont prefixées de leur lettre entre parenthèses, séparées par un point virgule, et terminées par une fin de ligne.

15 millions d'instructions multilingues et variées, Tk-Instruct (Wang *et al.*, 2022) un modèle T5 affiné avec des instructions naturelles collectées auprès d'humains, Pythia affiné sur les instructions naturelles d'OpenAssistant (Biderman *et al.*, 2023) ciblant le cas d'utilisation d'un assistant, OPT-IML (Iyer *et al.*, 2022) un modèle affiné sur 2000 tâches de TAL, Galactica (Taylor *et al.*, 2022) un modèle entraîné majoritairement sur des publications scientifiques, MPT (Team, 2023) un modèle ouvert reproduisant LLaMa et instruit sur de petits jeux d'instructions naturelles, PMC-LLaMa (Wu *et al.*, 2023) une version de LLaMa affinée sur des articles médicaux. Ces modèles ont été sélectionnés pour leur pertinence potentielle à la tâche, selon plusieurs tailles représentatives, et avec la contrainte de fonctionner sur un seul GPU. On observe les tendances suivantes dans ces résultats : dans une famille de modèles, la taille est corrélée avec les performances ; certains modèles ont des performances bien supérieures à taille égale, possiblement à cause d'une meilleure représentation du français ; les modèles entraînés spécifiquement sur des données médicales ou scientifiques n'offrent pas d'avantage par rapport aux modèles génériques. On observe aussi que les meilleures performances obtenues par tk-instruct-11b dans un contexte zero-shot sont proches de celles rapportées par Labrak et al. sur le corpus de test de DEFT'23 avec des systèmes supervisés fondés sur BERT/BART dont les instances sont augmentées par un contexte provenant d'articles scientifiques (Labrak *et al.*, 2022).

Une seconde série d'expériences présente des résultats à partir de la famille de modèles LLaMa (Touvron *et al.*, 2023) avec ou sans affinage. Dans ces expériences, les paramètres des modèles LLaMa sont convertis en 8 bits afin d'occuper moins de mémoire GPU, puis ils sont affinés pendant une époque avec la méthode LoRA décrite plus haut, et avec comme paramètres $R = 4$ (rang des matrices de paramètres), $\alpha = 16$, un dropout de $0.05$, une taille de batch de 24, un taux d'apprentissage de $3 \times 10^{-4}$, une longueur maximale de 256 tokens pour le mécanisme d'attention, et une optimisation par la méthode Adam avec un warmup de $5\%$ des batches. Seules les matrices de paramètres de projection de la clé et la valeur du mécanisme d'attention sont modifiées par LoRA (`q_proj` et `v_proj` dans le modèle). La taille de micro-batches est ajustée pour maximiser la vitesse d'entraînement ; le modèle 65B peut être entraîné sur un GPU A100 avec 80GB de RAM avec une taille de micro-batch de 1. Les modèles sont affinés selon quatre jeux d'entraînement : les 2171 instances d'entraînement de DEFT 2023 qui correspondent exactement à la tâche ciblée, le jeu de données

| Modèle | Taille | EMR |
|---|---|---|
| bloomz-560m | 0.5B | 0.0737 |
| bloomz-3b | 3B | 0.1442 |
| bloomz-7b1 | 7.1B | 0.1602 |
| bloomz-7b1-mt | 7.1B | 0.1762 |
| flan-t5-xxl-11b | 11B | 0.1794 |
| flan-ul2-20b | 20B | 0.1570 |
| tk-instruct-3b-def | 3B | 0.1346 |
| tk-instruct-11b-def | 11B | 0.1826 |
| oasst-sft-1-pythia-12b | 12B | 0.0705 |
| opt-iml-1.3b | 1.3B | 0.0673 |
| opt-iml-30b | 30B | 0.1442 |
| mpt-instruct-7b | 7B | 0.0641 |
| galactica-6.7b | 6.7B | 0.0352 |
| pmc-llama-7b | 7B | 0.0224 |

TABLE 1 – Performances des modèles huggingface affinés sur des instructions non spécifiques à la tâche. Les performances sont données en termes de réponses exactes sur le jeu de développement.

Alpaca (Taori *et al.*, 2023) contenant 52k instructions génériques générées à partir de ChatGPT, le jeu de données Alpaca-fr, traduit par nos soins en français de manière automatique à l'aide du modèle nllb-200-distilled-1.3B [3], et le jeu de données Vicuna (Chiang *et al.*, 2023) contenant 70k dialogues sélectionnés par des utilisateurs de ChatGPT pour leur pertinence. La génération est effectuée sur un maximum de 128 tokens, avec une température de 0.1 et en utilisant 4 faisceaux, exploitant les 40 meilleurs candidats par token avec une masse de probabilité totale minimale de 0.75.

On observe dans les résultats du tableau 2 que : la représentation en (int8) diminue les performances par rapport à la représentation d'origine (fp16), les performances augmentent avec la taille des modèles, les modèles de langage génériques de grande taille offrent des performances similaires aux modèles affinés de petite taille, l'affinage sur des instructions est bénéfique par rapport à un modèle générique de même type et taille, et l'affinage sur les données de la tâche cible permettent des gains substantiels par rapport à un affinage générique. On remarque aussi que les performances après affinage sur Alpaca-fr ne sont pas différentes de celles d'un modèle affiné sur de l'anglais, suggérant que les LLM entraînés sur une petite partie d'une autre langue sont déjà capables de traitements multilingues. Nous n'avons pas beaucoup expérimenté avec cette capacité, et il serait souhaitable de mieux la cerner dans des travaux futurs.

Une troisième série d'expériences explore cette fois les performances de modèles fermés mis à disposition par les industriels à travers des API. Nous testons, toujours avec le même schéma d'amorce, les modèles des entreprises Cohere [4], AI21 [5] et OpenAI [6]. Les détails techniques des modèles associés ne sont pas toujours disponibles, les poids des modèles ne sont pas accessibles, et certains modèles ne sont plus accessibles au public depuis que nous les avons utilisés. Nous avons testé

3. Traduction automatique de la structure json complète de chaque instance pour conserver le contexte, aboutissant à 48k instructions après filtrage des instances mal formées à l'issue de la traduction.
4. https://cohere.com/
5. https://www.ai21.com
6. https://openai.com/

| Modèle | Taille | Affinage | EMR |
|---|---|---|---|
| llama (int8) | 7B | - | 0.0576 |
| llama (int8) | 7B | alpaca | 0.1217 |
| llama (int8) | 7B | alpaca-fr | 0.1185 |
| llama (int8) | 7B | deft | 0.1378 |
| llama (int8) | 13B | - | 0.0769 |
| llama (int8) | 13B | alpaca | 0.1474 |
| llama (int8) | 13B | vicuna | 0.1538 |
| llama (int8) | 13B | deft | 0.1730 |
| llama (int8) | 30B | - | 0.1442 |
| llama (fp16) | 30B | - | 0.1891 |
| llama (int8) | 30B | alpaca | 0.1923 |
| llama (int8) | 30B | deft | 0.2467 |
| llama (int8) | 65B | - | 0.1730 |
| llama (fp16) | 65B | - | 0.2179 |
| llama (int8) | 65B | deft | 0.3044 |

TABLE 2 – Performances des variantes du modèles LLaMa en termes de réponses exactes, sur le jeu de développement.

le modèle command-xlarge-beta [7] de Cohere, le modèle j1-jumbo de A2AI, et code-cushman-001[6], code-davinci-002[6], text-curie-001, text-davinci-003, gpt-3.5-turbo-0301 (ChatGPT), et gpt-4-0314 (GPT-4) de OpenAI. Dans ces expériences, aucun effort n'a été fait pour adapter les amorces ou l'analyse de la réponse aux modèles. Le coût associé à l'utilisation de ces API est non nul, mais reste raisonnable, avec par exemple un coût de 2 USD pour faire tourner l'inférence de GPT-4 sur le test de DEFT2023, étant donné un tarif entre 0.03 et 0.06 USD pour 1000 tokens selon qu'ils appartiennent à l'amorce ou à la partie générée.

Les résultats sur les API fermées donnés dans le tableau 3 montrent que les performances en termes d'EMR varient beaucoup, avec des modèles aux performances similaires à celles des modèles publics, et d'autres modèles aux performances bien supérieures. Il est intéressant de constater par exemple

7. Modèle qui n'est plus disponible à l'heure actuelle

| Fournisseur | Modèle | EMR |
|---|---|---|
| cohere | command-xlarge-beta | 0.1057 |
| ai21 | j1-jumbo | 0.0833 |
| openai | code-cushman-001 | 0.1121 |
| openai | code-davinci-002 | 0.3108 |
| openai | text-curie-001 | 0.1217 |
| openai | text-davinci-003 | 0.2884 |
| openai | gpt-3.5-turbo-0301 | 0.4551 |
| openai | gpt-4-0314 | 0.7788 |

TABLE 3 – Performances des modèles fermés disponibles à travers une API, sur l'ensemble de développement et en termes de réponses exactes. Certains modèles ne sont plus ouverts au public.

que code-davinci-002 obtient des performances similaires à LLaMa 65B affiné alors qu'il a été entraîné principalement sur du code source. Les résultats obtenus par GPT-4 sont similaires à ceux obtenus sur le jeu de données MedMCQA en anglais (Nori *et al.*, 2023) par le même modèle. Il n'est malheureusement pas possible de tirer de conclusions scientifiques de ces expériences car les détails des modèles, leurs données d'entraînement, etc. ne sont pas publiques.

| | | Tâche principale | | Tâche annexe | |
|---|---|---|---|---|---|
| **Nom du système** | **Repro.** | **Hamming** | **EMR** | **F1-Score** | **Accuracy** |
| LIS/llama-65b-lora | ✓ | 52.94 | 33.76 | 42.42 | 68.65 |
| LIS/llama-30b-lora | ✓ | 47.43 | 27.81 | 35.26 | 65.92 |
| LIS/llama-13b-lora | ✓ | 35.93 | 17.85 | 34.52 | 65.11 |
| LIS/gpt-3.5-turbo-0301_prompt0 | | 64.75 | 46.95 | 47.51 | 68.17 |
| LIS/gpt-4-0314_prompt0 | | 85.17 | 72.83 | 71.57 | 79.58 |

TABLE 4 – Résultats officiels sur le test de DEFT 2023 pour les systèmes soumis dans les deux tâches. Les systèmes reproductibles sont dénotés par ✓.

Enfin, le tableau 4 liste les résultats officiels selon toutes les métriques sur le test pour les systèmes soumis dans la catégorie *reproductible* et dans la catégorie *sans restriction*. Les systèmes reproductibles sont basés sur le modèle LLaMa affiné sur les données d'entraînement DEFT comme indiqué plus haut. Seuls deux systèmes sans limites ont été lancés sur le test : ChatGPT et GPT-4. Les résultats en termes d'EMR sont proches de ceux obtenus sur l'ensemble de développement pour tous les systèmes. On peut remarquer que sur la tâche annexe, de grandes différences en EMR n'impliquent pas de grandes différences en précision ou F-score, sauf pour GPT-4 qui semble être plus précis dans sa compréhension du nombre de réponses attendues.

# 5 Conclusion et perspectives

Les expériences montrent que certaines familles de modèles ont des performances très différentes à taille similaire, ce qui pourrait être expliqué par les compétences en français des modèles, et donc par la quantité de données en français sur lesquelles ils ont été entraînés. Nous observons aussi que la taille des modèles, à travers une famille, est un bon prédicteur des performances. Ceci peut s'expliquer par une meilleure compréhension du texte et une meilleure fidélité des connaissances représentées dans les modèles. Toutefois, une autre explication pourrait provenir de la présence des questions et réponses du corpus dans les données d'apprentissage des modèles. Il serait intéressant de tester cette hypothèse avec la méthode MELD (Nori *et al.*, 2023) qui détermine si une question a été mémorisée par un modèle en générant la fin de la question à partir de son début avec une température de 0 et en la comparant à l'original. Une autre façon de faire serait de retrouver les questions dans les données d'apprentissage des modèles.

D'autres perspectives pour améliorer les systèmes seraient d'explorer l'espace des amorces afin de comprendre les caractéristiques d'une amorce associées à la tâche, de donner des exemples de réponses attendues (few-shot learning), de générer un contexte à partir de connaissances externes comme des ouvrages du domaine, de faire des ensembles à partir de modèles initialisés différemment. D'un point de vue des ressources nécessaires pour affiner les grands modèles, des méthodes alternatives à LoRA devraient être étudiées.

# Références


ALLAM A. M. N. & HAGGAG M. H. (2012). The question answering systems : A survey. *International Journal of Research and Reviews in Information Sciences (IJRRIS)*, **2**(3).

BIDERMAN S., SCHOELKOPF H., ANTHONY Q., BRADLEY H., O'BRIEN K., HALLAHAN E., KHAN M. A., PUROHIT S., PRASHANTH U. S., RAFF E., SKOWRON A., SUTAWIKA L. & VAN DER WAL O. (2023). Pythia : A suite for analyzing large language models across training and scaling.

BOUZIANE A., BOUCHIHA D., DOUMI N. & MALKI M. (2015). Question answering systems : survey and trends. *Procedia Computer Science*, **73**, 366–375.

BROWN T. B., MANN B., RYDER N., SUBBIAH M., KAPLAN J., DHARIWAL P., NEELAKANTAN A., SHYAM P., SASTRY G., ASKELL A., AGARWAL S., HERBERT-VOSS A., KRUEGER G., HENIGHAN T., CHILD R., RAMESH A., ZIEGLER D. M., WU J., WINTER C., HESSE C., CHEN M., SIGLER E., LITWIN M., GRAY S., CHESS B., CLARK J., BERNER C., MCCANDLISH S., RADFORD A., SUTSKEVER I. & AMODEI D. (2020). Language models are few-shot learners.

CHIANG W.-L., LI Z., LIN Z., SHENG Y., WU Z., ZHANG H., ZHENG L., ZHUANG S., ZHUANG Y., GONZALEZ J. E., STOICA I. & XING E. P. (2023). Vicuna : An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

CHUNG H. W., HOU L., LONGPRE S., ZOPH B., TAY Y., FEDUS W., LI E., WANG X., DEHGHANI M., BRAHMA S., WEBSON A., GU S. S., DAI Z., SUZGUN M., CHEN X., CHOWDHERY A., NARANG S., MISHRA G., YU A., ZHAO V., HUANG Y., DAI A., YU H., PETROV S., CHI E. H., DEAN J., DEVLIN J., ROBERTS A., ZHOU D., LE Q. V. & WEI J. (2022). Scaling instruction-finetuned language models. DOI : 10.48550/ARXIV.2210.11416.

GUO S., LIU K., HE S., LIU C., ZHAO J. & WEI Z. (2017). IJCNLP-2017 task 5 : Multi-choice question answering in examinations. In *Proceedings of the IJCNLP 2017, Shared Tasks*, p. 34–40, Taipei, Taiwan : Asian Federation of Natural Language Processing.

HAO T., LI X., HE Y., WANG F. L. & QU Y. (2022). Recent progress in leveraging deep learning methods for question answering. *Neural Computing and Applications*, p. 1–19.

HU E., SHEN Y., WALLIS P., ALLEN-ZHU Z., LI Y., WANG L. & CHEN W. (2021). Lora : Low-rank adaptation of large language models.

IYER S., LIN X. V., PASUNURU R., MIHAYLOV T., SIMIG D., YU P., SHUSTER K., WANG T., LIU Q., KOURA P. S. *et al.* (2022). Opt-iml : Scaling language model instruction meta learning through the lens of generalization.

KAPLAN J., MCCANDLISH S., HENIGHAN T., BROWN T. B., CHESS B., CHILD R., GRAY S., RADFORD A., WU J. & AMODEI D. (2020). Scaling laws for neural language models. *arXiv preprint arXiv :2001.08361*.

LABRAK Y., BAZOGE A., DUFOUR R., DAILLE B., GOURRAUD P.-A., MORIN E. & ROUVIER M. (2022). FrenchMedMCQA : A French multiple-choice question answering dataset for medical domain. In *Proceedings of the 13th International Workshop on Health Text Mining and Information Analysis (LOUHI)*, p. 41–46, Abu Dhabi, United Arab Emirates (Hybrid) : Association for Computational Linguistics.

LE BERRE G. & LANGLAIS P. (2020). Attending knowledge facts with bert-like models in question-answering : Disappointing results and some explanations. In *Advances in Artificial Intelligence : 33rd Canadian Conference on Artificial Intelligence, Canadian AI 2020, Ottawa, ON, Canada, May 13–15, 2020, Proceedings 33*, p. 356–367 : Springer.

MUENNIGHOFF N., WANG T., SUTAWIKA L., ROBERTS A., BIDERMAN S., SCAO T. L., BARI M. S., SHEN S., YONG Z.-X., SCHOELKOPF H., TANG X., RADEV D., AJI A. F., ALMUBARAK K., ALBANIE S., ALYAFEAI Z., WEBSON A., RAFF E. & RAFFEL C. (2022). Crosslingual generalization through multitask finetuning.

NORI H., KING N., MCKINNEY S. M., CARIGNAN D. & HORVITZ E. (2023). Capabilities of gpt-4 on medical challenge problems. *arXiv preprint arXiv :2303.13375*.

PAL A., UMAPATHI L. K. & SANKARASUBBU M. (2022). Medmcqa : A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on Health, Inference, and Learning*, p. 248–260 : PMLR.

ROY S., EHTESHAM N., ISLAM M. S. *et al.* (2021). Augmenting bert with cnn for multiple choice question answering. In *2021 24th International Conference on Computer and Information Technology (ICCIT)*, p. 1–5 : IEEE.

SAVELKA J., AGARWAL A., BOGART C. & SAKR M. (2023). Large language models (gpt) struggle to answer multiple-choice questions about code. *arXiv preprint arXiv :2303.08033*.

SOARES M. A. C. & PARREIRAS F. S. (2020). A literature review on question answering techniques, paradigms and systems. *Journal of King Saud University-Computer and Information Sciences*, **32**(6), 635–646.

SU J., LU Y., PAN S., MURTADHA A., WEN B. & LIU Y. (2021). Roformer : Enhanced transformer with rotary position embedding. *arXiv preprint arXiv :2104.09864*.

TAORI R., GULRAJANI I., ZHANG T., DUBOIS Y., LI X., GUESTRIN C., LIANG P. & HASHIMOTO T. B. (2023). Stanford alpaca : An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

TAY Y. (2023). A new open source flan 20b with ul2.

TAYLOR R., KARDAS M., CUCURULL G., SCIALOM T., HARTSHORN A., SARAVIA E., POULTON A., KERKEZ V. & STOJNIC R. (2022). Galactica : A large language model for science.

TEAM M. N. (2023). Introducing mpt-7b : A new standard for open-source, commercially usable llms.

TOUVRON H., LAVRIL T., IZACARD G., MARTINET X., LACHAUX M.-A., LACROIX T., ROZIÈRE B., GOYAL N., HAMBRO E., AZHAR F., RODRIGUEZ A., JOULIN A., GRAVE E. & LAMPLE G. (2023). Llama : Open and efficient foundation language models.

WANG Y., MISHRA S., ALIPOORMOLABASHI P., KORDI Y., MIRZAEI A., ARUNKUMAR A., ASHOK A., DHANASEKARAN A. S., NAIK A., STAP D., PATHAK E., KARAMANOLAKIS G., LAI H. G., PUROHIT I., MONDAL I., ANDERSON J., KUZNIA K., DOSHI K., PATEL M., PAL K. K., MORADSHAHI M., PARMAR M., PUROHIT M., VARSHNEY N., KAZA P. R., VERMA P., PURI R. S., KARIA R., SAMPAT S. K., DOSHI S., MISHRA S. D., REDDY S. C., PATRO S., DIXIT T., DONG SHEN X., BARAL C., CHOI Y., HAJISHIRZI H., SMITH N. A. & KHASHABI D. (2022). Benchmarking generalization via in-context instructions on 1,600+ language tasks.

WU C., ZHANG X., ZHANG Y., WANG Y. & XIE W. (2023). Pmc-llama : Further finetuning llama on medical papers.

*Actes de CORIA-TALN 2023*
*Paris, France, 5 au 9 juin 2023*

# Tâches et systèmes de détection automatique des réponses correctes dans des QCMs liés au domaine médical : Présentation de la campagne DEFT 2023

Yanis Labrak[1, 3] Adrien Bazoge[2] Béatrice Daille[2] Richard Dufour[2]
Emmanuel Morin[2] Mickael Rouvier[1]
(1) Laboratoire Informatique d'Avignon (LIA), Avignon Université, France
(2) Laboratoire des Sciences du Numérique de Nantes (LS2N), Nantes Université, France
(3) Zenidoc, France
{prenom.nom}@univ-avignon.fr, {prenom.nom}@univ-nantes.fr

## RÉSUMÉ

L'édition 2023 du DÉfi Fouille de Textes (DEFT) s'est concentrée sur le développement de méthodes permettant de choisir automatiquement des réponses dans des questions à choix multiples (QCMs) en français. Les approches ont été évaluées sur le corpus FrenchMedMCQA, intégrant un ensemble de QCMs avec, pour chaque question, cinq réponses potentielles, dans le cadre d'annales d'examens de pharmacie. Deux tâches ont été proposées. La première consistait à identifier automatiquement l'ensemble des réponses correctes à une question. Les résultats obtenus, évalués selon la métrique de l'Exact Match Ratio (EMR), variaient de 9,97 % à 33,76 %, alors que les performances en termes de distance de Hamming s'échelonnaient de 24,93 à 52,94. La seconde tâche visait à identifier automatiquement le nombre exact de réponses correctes. Les résultats, quant à eux, étaient évalués d'une part avec la métrique de F1-Macro, variant de 13,26 % à 42,42 %, et la métrique *Accuracy*, allant de 47,43 % à 68,65 %. Parmi les approches variées proposées par les six équipes participantes à ce défi, le meilleur système s'est appuyé sur un modèle de langage large de type LLaMa affiné en utilisant la méthode d'adaptation LoRA.

## ABSTRACT

**Tasks and systems for automatic question-answering in the medical field : presentation of the DEFT 2023 campaign.**

The 2023 edition of the text mining challenge *DÉfi Fouille de Textes* (DEFT) focused on the development of methods for automatically selecting answers in multiple-choice question (MCQ) in French. The approaches were evaluated on the FrenchMedMCQA corpus, which includes a set of MCQ with five potential answers for each question, based on pharmacy exam archives. Two tasks have been proposed. The first one aimed to automatically identify all the correct answers to a question. The obtained results, evaluated using the Exact Match Ratio (EMR) metric, ranged from 9.97% to 33.76%, while the performances in terms of Hamming score ranged from 24.93 to 52.94. The second task aimed to automatically identify the exact number of correct answers. The results, on the other hand, were evaluated using the F1-Macro metric, ranging from 13.26% to 42.42%, and the Accuracy metric, ranging from 47.43% to 68.65%. Among the various approaches proposed by the six participating teams in this challenge, the best system relied on a large language model of the LLaMa type, fine-tuned using the LoRA adaptation method.

MOTS-CLÉS : Question à choix multiples ; Domaine médical ; Modèle de langue large ; TALN.

KEYWORDS: Multiple Choice Question ; Medical field ; Large Language Model ; NLP.

# 1 Introduction

Le DÉfi Fouille de Textes (DEFT) est une campagne d'évaluation annuelle francophone qui permet à plusieurs équipes, souvent issues du monde académique et/ou industriel, de confronter des méthodes originales en traitement automatique du langage naturel (TALN) sur une ou plusieurs tâches régulièrement renouvelées.

Pour cette édition 2023 du défi [1], nous avons proposé de travailler sur le corpus FrenchMedMCQA (Labrak *et al.*, 2022), intégrant un ensemble de QCMs en français issus d'annales d'examens de pharmacie. Une des difficultés, et originalités, du corpus, est que chaque question contient une inconnue sur le nombre de réponses associées, là où d'autres corpus attendent une seule réponse par question. Cette difficulté a permis aux équipes participantes d'explorer et de proposer des approches pouvant s'écarter de celles actuellement proposées pour des tâches plus classiques en TALN. Deux tâches ont été proposées aux participants pour cette édition :

1. Sélectionner automatiquement le sous-ensemble de réponses correctes parmi l'ensemble proposées pour une question donnée en s'aidant, ou non, de connaissances externes au corpus fourni.
2. Identifier, pour chaque question, le nombre exact de réponses correctes.

La campagne a été lancée le 27 février 2023. L'accès aux données d'entraînement était possible après signature d'un accord par tous les membres de l'équipe participante. La phase d'entraînement s'est déroulée sur pratiquement deux mois (27 février 2023 au 23 avril 2023). La phase de test s'est déroulée du 24 avril au 7 mai 2023. Six équipes se sont inscrites, et sont allées jusqu'au terme de la campagne :

— *ALMANACH-ARKHN* (Meoni *et al.*, 2023) : Équipe jointe entre l'entreprise Arkhn et l'INRIA.
— *LIS* (Favre, 2023) : LIS (Aix-Marseille Université).
— *LIUM-IRISA* (Besnard *et al.*, 2023) : Équipe jointe entre le LIUM (Le Mans Université) et l'IRISA (Université de Rennes).
— *SEQUOIA* (Bothua *et al.*, 2023) : Entreprise EDF R&D.
— *SPQR* (Bezançon *et al.*, 2023) : Équipe jointe entre le STIH (Sorbonne Université), le L3I (La Rochelle Université) et l'OBTIC (Sorbonne Université).
— *TTGV* (Blivet *et al.*, 2023) : Équipe jointe entre l'entreprise SNCF R&D, le LORIA (Université de Lorraine) et le LTCI (Telecom Paris).

# 2 Corpus

Le corpus FrenchMedMCQA (Labrak *et al.*, 2022) contient un ensemble de 3 105 QCMs en français portant sur le domaine médical, proche de ce que l'on retrouve dans d'autres langues telles que l'anglais avec le corpus MedMCQA (Pal *et al.*, 2022) ou SciQ (Welbl *et al.*, 2017). Ce corpus a été constitué en collectant des questions et leurs réponses associées à partir d'annales d'examens réels de pharmacie obtenus du site *Remede.org* [2]. Chaque QCM contient cinq réponses potentielles, parmi

1. https://deft2023.univ-avignon.fr
2. http://www.remede.org/internat/pharmacie/qcm-internat.html

lesquelles se trouve une ou plusieurs réponses correctes. Ces QCMs ont été réalisés manuellement par des experts médicaux et utilisés lors d'examens de pharmacie.

Le Tableau 1 fournit la distribution du jeu de données FrenchMedMCQA selon son découpage pour l'apprentissage, le développement et l'évaluation. Nous constatons que le corpus est composé de 1 080 questions ayant une réponse unique parmi les cinq potentielles ($\#Réponses = 1$), et de 2 025 questions avec de multiples réponses ($\#Réponses > 1$). Au total, le corpus contient 3 105 questions. Afin de garder un corpus équilibré, 70% des questions sont utilisées pour le corpus d'apprentissage, alors que 10 % ont été conservées pour le corpus de développement et 20% pour l'évaluation.

| # Réponses | Apprentissage | Développement | Évaluation | Total |
|---|---|---|---|---|
| **1** | 595 | 164 | 321 | **1 080** |
| **2** | 528 | 45 | 97 | **670** |
| **3** | 718 | 71 | 141 | **930** |
| **4** | 296 | 30 | 56 | **382** |
| **5** | 34 | 2 | 7 | **43** |
| **Total** | **2 171** | **312** | **622** | **3 105** |

TABLE 1 – Distribution du corpus FrenchMedMCQA selon son découpage en apprentissage, développement et test.

Chaque instance du corpus comprend un identifiant, une question, cinq réponses potentielles (étiquetées dans le corpus de $A$ à $E$), et la (ou les) réponse(s) correcte(s). La longueur moyenne des questions est de 14,17 mots et la longueur moyenne des réponses est de 6,44 mots. Le vocabulaire compte 13 000 mots, sachant que 3 800 d'entre-eux (soit environ 29 %) sont spécifiques au domaine médical. Dans le détail, en moyenne, chaque question contient 2,5 mots spécifiques au domaine médical (représentant 17 % des mots en moyenne dans une question) et chaque réponse en contient 2 (représentant 36 % des mots en moyenne dans une réponse). Enfin, toujours en moyenne, un mot spécifique au domaine médical ciblé apparaît dans 2 questions et dans 8 réponses. La Figure 1 donne un exemple d'une instance pour une question contenant plusieurs réponses correctes.


```
{
  "id": "6979d46501a3270436d37b98cf351439fbcbec8d5890d293dabfb8f85f723904",
  "question": "Cocher la (les) proposition(s) exacte(s) : Le métronidazole :",
  "answers": {
    "A": "Est un dérivé du pyrazole",
    "B": "Peut induire un effet antabuse",
    "C": "Peut être administré par voie parentérale intraveineuse",
    "D": "Peut être utilisé dans certaines parasitoses à protozoaires",
    "E": "Est inefficace dans les infections à germes anaérobies"
  },
  "correct_answers": ["B", "C", "D"],
  "nbr_correct_answers": 3,
}
```


Listing 1: Exemple d'une instance du corpus FrenchMedMCQA, comprenant un identifiant, une question, cinq réponses potentielles (étiquetées de $A$ à $E$) et les réponses correctes.

Lors de ce défi, deux tâches liées aux QCMs médicaux ont été proposées. La tâche principale (voir Section 2.1.1) a consisté à choisir automatiquement, selon une question posée, l'ensemble des réponses correctes parmi cinq réponses possibles fournies. La tâche annexe (voir Section 2.1.2) a consisté à identifier automatiquement le nombre de réponses correctes. À ces deux tâches, nous avons proposé,

pour chacune d'entre-elles, deux pistes (voir Section 2.2) que les équipes pouvaient développer : 1) *recherche reproductible*, avec des approches et modèles dont les données d'entraînement ou de référence étaient connues et contrôlées, et 2) *aucune restriction*, laissant libre chaque participant de fournir des propositions sans contrainte de reproductibilité. À noter que la seconde piste de recherche n'apparaît qu'à des fins de recherche et ne compte pas dans le classement final des équipes à la campagne DEFT 2023.

Pour l'ensemble des tâches et pistes, les participants ont eu à leur disposition les données d'entraînement et de développement comme décrites dans la Section 2. Les annotations du corpus de test, sur lequel toutes les équipes ont été évaluées, n'a jamais été fourni durant la campagne d'évaluation, mais a été rendu disponible librement, tout comme les corpus d'entraînement et de développement, à la fin de la campagne.

Enfin, nous avons fourni à l'ensemble des équipes un système état-de-l'art (*baseline*) pour chacune des deux tâches. Ces systèmes ont été transmis sous la forme de recettes disponibles librement en ligne [3] que chaque participant pouvait entraîner. Ces premiers résultats permettaient aux équipes d'avoir un repère quant aux performances de leurs approches.

## 2.1 Tâches proposées

### 2.1.1 Tâche principale : Choix automatique des réponses correctes dans un QCM

**Présentation** La tâche principale consiste à identifier automatiquement la ou les bonne(s) réponse(s) parmi l'ensemble de réponses proposées. Les participants ont alors à leur disposition la question posée ainsi que les cinq réponses potentielles, leur système devant choisir celles qui sont correctes.

**Évaluation** Contrairement à une tâche de classification classique où il est demandé d'associer une étiquette à un problème donné, notre tâche principale peu impliqué le fait d'avoir une réponse partiellement correcte. Par exemple, si nous devons retrouver deux réponses correctes parmi les cinq options disponibles pour une question ciblée, mais que notre système automatique n'est capable que d'en retrouve une seule, alors la réponse n'est pas juste, car incomplète. Il faut donc, dans ce cas, mettre en place une métrique permettant de prendre en compte la proportion de réponses justes, tout en pénalisant la/les réponse(s) incorrecte(s) dans le but d'éviter que les systèmes proposés répondent aux questions par l'ensemble du champ des possibilités. Dans cette optique, deux métriques différentes ont été utilisées, à savoir la correspondance exacte entre les réponses produites et la référence (*Exact Match Ratio*, EMR) ainsi que la Distance de Hamming (*Hamming Score*).

$$\text{Exact Match Ratio (EMR)} = \frac{1}{N}\sum_{i=1}^{N}[\hat{y}_i = y_i]$$

où $N$ est le nombre de questions, $\hat{y}_i$ est l'ensemble de réponses prédites pour la $i$-ième question, $y_i$ est l'ensemble des bonnes réponses pour la $i$-ième question, et $[x]$ est une fonction indicatrice qui vaut 1 si $x$ est vrai et 0 dans le cas contraire.

3. https://github.com/qanastek/DEFT-2023/tree/main/training_scripts

$$\text{Hamming Score} = \frac{1}{N}\sum_{i=1}^{N}\frac{|y_i \cap \hat{y}_i|}{|y_i \cup \hat{y}_i|}$$

où, $N$ est le nombre de questions, $y_i$ est l'ensemble des bonnes réponses pour la $i$ème question, $\hat{y}_i$ est l'ensemble des réponses prédites pour la $i$ème question, $|y_i \cap \hat{y}_i|$ est la taille de l'intersection des réponses vraies et prédites, et $|y_i \cup \hat{y}_i|$ est la taille de l'union des bonnes réponses et des réponses prédites.

**Système *Baseline*** Le problème de réponse automatique à une question peut être considéré dans notre cas comme étant un problème de classification multi-étiquettes. Nous avons donc proposé aux participants un système *baseline* s'appuyant sur l'affinage (*fine-tuning*) du modèle CamemBERT 138 GB OSCAR (Martin *et al.*, 2020), un modèle de langue générique pré-entraîné pour le français et fondé sur l'architecture RoBERTa (Liu *et al.*, 2019). Notons que la séquence d'entrée du modèle est composée de la question suivie des cinq réponses possibles, toutes séparées avec un token [SEP], selon le format suivant : [CLS] <question> [SEP] (A) <answer.a> [SEP] (B) <answer.b> [SEP] (C) <answer.c> [SEP] (D) <answer.d> [SEP] (E) <answer.e> [SEP] [EOS]. Pour ce qui est de la sortie, nous avons une couche de classification de dimension 5 suivie d'une SIGMOID et représentant, à partir d'un certain palier, soit l'absence ou la présence d'une classe, ici les lettres des réponses de $A$ à $E$.

### 2.1.2 Tâche annexe : Nombre de réponses correctes dans un QCM

**Présentation** La tâche annexe à la campagne d'évaluation consiste à identifier automatiquement le nombre de réponses correctes, pour chaque question, parmi l'ensemble des réponses potentielles dans un QCM.

**Évaluation** Contrairement à la tâche principale, qui se retrouve très proche d'un problème multi-étiquettes, il s'agit ici de retrouver la valeur exacte correspondant au nombre de réponses correctes, cette valeur étant comprise entre 1 et 5 inclus. La tâche peut alors être vue comme un problème multi-classes (ici, 5 classes) : nous avons donc choisi d'évaluer les systèmes en termes de taux correct de classification (Accuracy) et de macro F-mesure (F1-Macro).

**Système *Baseline*** Le système *baseline* permet aux participants de résoudre cette tâche comme un problème de classification multi-classes, fondé, à l'instar de la tâche principale, sur l'affinage du modèle CamemBERT 138 GB OSCAR. La séquence d'entrée du modèle est exactement la même que pour la tâche principale (voir Section 2.1.1), mais ici, le modèle doit ne fournir, en sortie, qu'une seule et unique classe parmi les cinq qui sont possibles et qui représente le nombre de réponses correctes. Ici aussi, une fonction SIGMOID est appliquée sur le vecteur de sortie, mais nous choisissons par défaut la classe donnant le score le plus élevé.

## 2.2 Pistes développées

Pour cette campagne d'évaluation, nous avons proposé d'ouvrir deux pistes aux équipes : *Recherche reproductible* et *Aucune restriction*. Chacune des deux tâches, présentées précédemment dans la Section 2.1, peuvent s'inscrire dans ces deux pistes. Dans la piste *Recherche reproductible*, seuls les systèmes qui respectent les deux conditions suivantes sont acceptés : 1) ne pas rechercher sur Internet les originaux des données fournies, et 2) utiliser des modèles pré-entraînés dont les données d'entraînement sont connues. Pour la piste *Aucune restriction*, tous les systèmes sont acceptés sans limite de recherche, que ce soit au niveau des données collectées ou des modèles pré-entraînés utilisés.

Le classement des équipes se fait uniquement sur les sorties des systèmes déposés dans la piste *Recherche reproductible*, où un système doit obligatoirement être déposé. Dans cette piste, les participants sont autorisés à soumettre jusqu'à trois sorties de système (*run*) par tâche. En revanche, la participation à la piste *Aucune restriction* n'étant pas obligatoire, et difficile à contrôler, le classement des équipes dans la campagne d'évaluation n'intégrera pas les systèmes proposés dans cette piste. Notons également que le nombre de soumissions autorisées n'y a pas été limité.

# 3 Résultats

Dans la piste *Recherche reproductible*, six équipes ont participé à la tâche principale et trois équipes à la tâche annexe. Dans le cadre de la piste *Aucune restriction*, deux équipes ont participé aux deux tâches proposées.

Dans les sections suivantes, nous présentons, pour chacune des deux tâches, tout d'abord les résultats officiels de la campagne DEFT 2023, intégrant une description succincte des systèmes proposés par chaque équipe, correspondant à la piste *Recherche reproductible* (voir Section 3.1), alors que les systèmes correspondant à la piste *Aucune restriction* sont décrits dans la Section 3.2.

## 3.1 Piste : Recherche reproductible

Les résultats décrits dans cette partie, pour la tâche principale (Section 3.1.1) et la tâche annexe (Section 3.1.2), constituent les performances officielles des systèmes proposés par les participants à la campagne d'évaluation DEFT 2023.

### 3.1.1 Tâche principale

Les six participants ont chacun soumis trois fichiers de prédictions. Le Tableau 2 présente les résultats en termes de distance de Hamming et EMR obtenus par chaque équipe pour chaque fichier de prédiction (*Run*). Le classement de chaque équipe selon la métrique ciblée est fourni. Notons que nous avons également intégré dans ce tableau les résultats de notre méthode *baseline* (Section 2.1.1) et ceux de la *classe majoritaire*.

**Méthodes des participants** Les participants ont utilisé des méthodes variées pour cette tâche principale. Nous observons cependant un intérêt commun pour les grands modèles de langue (*Large*

| Équipe | Run | Hamming | Classement | EMR | Classement |
|---|---|---|---|---|---|
| LIS | 1 | 52,94 | 1 | 33,76 | 1 |
| | 2 | 47,43 | - | 27,81 | - |
| | 3 | 35,93 | - | 17,85 | - |
| LIUM-IRISA | 1 | 43,24 | 2 | 22,19 | 3 |
| | 2 | 37,24 | - | 18,65 | - |
| | 3 | 35,47 | - | 18,49 | - |
| TTGV | 1 | 41,54 | 3 | 23,95 | 2 |
| | 2 | 39,15 | - | 11,58 | - |
| | 3 | 37,22 | - | 15,43 | - |
| SEQUOIA | 1 | 26,34 | - | 14,63 | - |
| | 2 | 35,90 | - | 15,27 | 4 |
| | 3 | 37,93 | 4 | 12,70 | - |
| ALMANACH-ARKHN | 1 | 33,27 | - | 12,22 | - |
| | 2 | 33,67 | - | 14,15 | 5 |
| | 3 | 35,96 | 5 | 13,67 | - |
| SPQR | 1 | 24,93 | 6 | 8,52 | - |
| | 2 | 22,38 | - | 9,32 | - |
| | 3 | 23,94 | - | 9,97 | 6 |
| Baseline | - | 36,24 | - | 16,55 | - |
| Classe majoritaire | - | 23,93 | - | 13,67 | - |

TABLE 2 – Résultats et classement des équipes participantes pour la tâche principale dans la piste *Recherche reproductible*.

*Language Models* - LLMs) de la part de la majorité des équipes. L'équipe en tête du classement (LIS) a utilisé pour ses soumissions le grand modèle de langue LLaMa (Touvron *et al.*, 2023) affiné sur les données d'entraînement de DEFT 2023 en utilisant la méthode d'adaptation LoRA (Hu *et al.*, 2021) (méthode d'adaptation à faible rang, *Low Rank Adaptation*) permettant de réduire le coût machine d'un tel affinage et de le rendre réalisable sur le matériel utilisé par l'équipe (NVIDIA A100 80 GB). Les différentes soumissions de l'équipe correspondent aux variantes de différentes tailles de ce modèle : 13 milliards (13B), 30 milliards (30B) et 65 milliards (65B) de paramètres. D'autres LLMs ont été explorés par les équipes participantes, tels que BloomZ (Muennighoff *et al.*, 2022) (SEQUOIA, TTGV, LIS), Flan-T5 (Chung *et al.*, 2022) (LIUM-IRISA, ALMANACH-ARKHN, LIS) et Vicuna 13B (Chiang *et al.*, 2023) (TTGV).

Des modèles de langue pré-entraînés pour le français et fondés sur RoBERTa ont aussi été utilisés, en particulier des modèles spécialisés dans le domaine médical, tels que DrBERT (Labrak *et al.*, 2023) (LIUM-IRISA, TTGV) et Camembert-BIO (Touchent *et al.*, 2023) (ALMANACH-ARKHN). À noter que ces derniers restent compétitifs, puisque les équipes LIUM-IRISA et TTGV obtiennent alternativement la deuxième et troisième place du classement selon la métrique étudiée.

Certaines équipes ont aussi exploré des approches différentes de celles maintenant classiques liées à l'affinage de modèles de langue. L'équipe SPQR s'est par exemple appuyée sur l'utilisation d'un corpus externe pour *vérifier* les réponses données dans les QCMs. Après plusieurs traitements des données, dont l'extraction de mots-clés médicaux, leurs systèmes intégraient différentes mesures de similarité afin de rapprocher les données des QCMs et des ressources externes biomédicales. L'équipe LIUM-IRISA a, quant à elle, proposé un système de fouille dans une base de connaissances, avec des approches TF.IDF et données Wikipédia, voire un méta-système intégrant plusieurs sources de connaissances (Flan-T5, DrBERT, etc.). Enfin, l'équipe TTGV a exploré un très grand nombre d'approches différentes, parmi lesquelles nous pouvons citer des méthodes par expressions régulières, modélisation par topics, régression logistique, approches multi-classes et multi-étiquettes, etc.

Enfin, nous observons que le modèle *baseline* fourni aux équipes participantes, se placerait à la cinquième position considérant la distance de Hamming, et à la quatrième position selon la métrique EMR.

### 3.1.2 Tâche annexe

Sur la tâche annexe, trois équipes ont proposé des systèmes originaux. Le Tableau 3 présente les résultats en termes de taux correct de classification (*Accuracy*) et F1-Macro, ainsi que le classement associé pour ces métriques. À l'instar de la tâche principale, nous intégrons dans ce tableau les résultats de notre *baseline* et ceux de la classe majoritaire.

| Équipe | Run | F1-Macro | Classement | Accuracy | Classement |
|---|---|---|---|---|---|
| LIS | 1 | 42,42 | 1 | 68,65 | 1 |
| | 2 | 35,26 | - | 65,92 | - |
| | 3 | 34,52 | - | 65,11 | - |
| TTGV | 1 | 27,98 | - | 62,54 | 2 |
| | 2 | 13,26 | - | 19,13 | - |
| | 3 | 31,51 | 2 | 60,45 | - |
| SPQR | 1 | 22,99 | 3 | 43,57 | - |
| | 2 | 15,29 | - | 47,43 | 3 |
| | 3 | 21,05 | - | 46,78 | - |
| Baseline | - | 28,79 | - | 67,04 | - |
| Classe majoritaire | - | 13,62 | - | 51,61 | - |

TABLE 3 – Résultats et classement des équipes participantes pour la tâche annexe dans la piste *Recherche reproductible*.

**Méthode des participants** Ainsi, l'équipe tête du classement (LIS) a dérivé ses sorties du système LLaMA de la tâche principale pour la tâche annexe.

L'équipe TTGV a testé plusieurs méthodes : celle ayant obtenu les meilleurs résultats a consisté à utiliser un modèle de langue pré-entraîné dans le domaine médical pour le français (DrBERT) affiné sur les données d'apprentissage de DEFT 2023 afin de résoudre le problème sous la forme de classification multi-classes.

Enfin, l'équipe SPQR s'est appuyée sur les résultats obtenus dans la tâche principale avec leur approche par similarité pour fournir le nombre de réponses correctes par question.

Nous observons qu'en termes de F1-macro, seule l'approche intégrant des LLMs surpasse largement les résultats de la *baseline* fournie aux participants. Notons cependant que cette observation n'est pas aussi franche en termes d'accuracy, où finalement le modèle CamemBERT affiné reste compétitif face à ces grands modèles.

## 3.2 Piste : Aucune restriction

La piste *Aucune restriction* n'était pas une piste obligatoire, celle-ci laissant la possibilité aux équipes participantes d'évaluer n'importe quelle approche. Contrairement à la piste *Recherche reproductible*, aucun classement n'a été fait dans cette piste, les résultats n'étant reportés qu'à titre informatif. Deux

équipes ont choisi de participer à la piste *Aucune restriction* dans les deux tâches (principale et annexe). Le Tableau 4 présente les résultats obtenus par chacune des équipes.

| Équipe | Run | Tâche principal | | Tâche annexe | |
|---|---|---|---|---|---|
| | | Hamming | EMR | F1-Macro | Accuracy |
| LIS | 1 - GPT-3.5-turbo | 64,75 | 46,95 | 47,51 | 68,17 |
| | 2 - GPT-4 | 85,17 | 72,83 | 71,57 | 79,58 |
| SEQUOIA | 1 - GPT-3.5-turbo | 64,40 | 46,46 | 44,36 | 65,92 |

TABLE 4 – Résultats et classement des équipes participantes pour la tâche principale dans la piste *Aucune restriction*.

**Méthode des participants** L'agent conversationnel privé ChatGPT (GPT-3.5-Turbo) [4] de l'entreprise OpenAI, a été utilisé par les deux équipes LIS et SEQUOIA. Les *Run 1* des deux équipes utilisent le modèle GPT-3.5 accessible par l'API de la firme, ce qui explique les résultats très proches (des petites différences de résultats pourraient être imputées à la manière d'interagir avec le modèle). Le *Run 2* de l'équipe LIS utilise, quant à lui, le modèle GPT-4 [5]. Il est intéressant de voir que les prédictions fondées sur le modèle GPT-4 obtiennent de meilleurs résultats que le système le plus performant de la piste *Recherche reproductible*. Toutefois, étant donné que nous ne disposons pas d'informations concernant la présence des données d'entraînement dans le corpus d'apprentissage, nous pouvons difficilement tirer des conclusions quant à cette performance.

# 4 Conclusion

L'édition 2023 du DÉfi Fouille de Textes (DEFT) s'est concentrée sur le développement de méthodes permettant de choisir les réponses dans des questions à choix multiples (QCMs) en français.

La première tâche a rassemblé six équipes et consistait à sélectionner automatiquement le sous-ensemble de réponses correctes parmi celles proposées pour une question donnée. Les équipes participantes avaient à leur disposition des données d'entraînement et de développement fournies avec le corpus FrenchMedMCQA, et pouvaient également s'aider de connaissances externes. Les résultats obtenus sur les données de test de ce corpus ont été fournis selon la métrique de l'Exact Match Ratio, variant de 9,97 % à 33,76 %, alors que les performances en termes de distance de Hamming s'échelonnaient de 24,93 à 52,94. Notre système *baseline* a obtenu 16,55 % et 36,24 respectivement avec l'EMR et la distance de Hamming. L'utilisation de grands modèles de langue comme LLaMa affiné sur les données mises à disposition en utilisant la méthode d'adaptation LoRA s'est révélée la plus efficace.

Quant à la deuxième tâche, elle a rassemblé trois équipes et consistait à identifier, pour chaque question, le nombre exact de réponses correctes. Les résultats ont été fournis en utilisant la métrique de F1-Macro, variant de 13,26 % à 42,42 %, ainsi que la métrique *Accuracy*, allant de 47,43 % à 68,65 %. Notre système *baseline* a obtenu une F1-Macro de 28,79 % et une *Accuracy* de 67,04 %.

Cette nouvelle édition de DEFT se termine avec une grande variété de méthodes testées sur chacune des tâches proposées, et montre que l'utilisation de grands modèles de langue s'avère très efficace,

4. https://openai.com/product/chatgpt
5. https://openai.com/waitlist/gpt-4-api

alors même que certains de ces modèles ne sont pas adaptés au domaine traité (ici, le domaine médical).

## Remerciements

Le comité d'organisation de DEFT 2023 tient à remercier chaleureusement l'ensemble des équipes (ALMANACH-ARKHN, LIS, LIUM-IRISA, SEQUOIA, SPQR, TTGV) pour l'engagement et la qualité des systèmes proposés durant cette campagne d'évaluation. Le comité d'organisation tient également à remercier Cyril Grouin, pour son aide précieuse à la mise en place de l'atelier, ainsi que le comité scientifique de DEFT 2023 (Nathalie Camelin, Liana Ermakova, Benoit Favre, Corinne Fredouille, Pierre-Antoine Gourraud, Natalia Grabar, Cyril Grouin, Pierre Jourlin, Fleur Mougin, Aurélie Névéol, Didier Schwab et Pierre Zweigenbaum).

## Références

BESNARD C., ETTALEB M., RAYMOND C. & CAMELIN N. (2023). Qui de DrBERT, Wikipédia ou Flan-T5 s'y connaît le plus en questions médicales ? In *Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles. Atelier DÉfi Fouille de Textes (DEFT)*.

BEZANÇON J., BOUBEHZIZ, TOUFIK ANX CHUTAUX C., ZINE O., ACENSIO L., BRIGLIA A., KOUDORO-PARFAIT C. & LEJEUNE G. (2023). SPQR@Deft2013 : Similarité Sorbonne Pour les Systèmes de Question Réponse. In *Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles. Atelier DÉfi Fouille de Textes (DEFT)*.

BLIVET A., DEGRUTÈRE S., GENDRON B., RENAULT A., SIOUFFI C., GAUDRAY-BOUJU V., CERISARA C., FLAMEIN H., GUIBON G., LABEAU M. & ROUSSEAU T. (2023). Participation de l'équipe TTGV à DEFT 2023 : Réponse automatique à des QCM issus d'examens en pharmacie. In *Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles. Atelier DÉfi Fouille de Textes (DEFT)*.

BOTHUA M., HASSANI L., JUBAULT M. & SUIGNARD P. (2023). Participation d'EDF R&D au défi DEFT 2023 : réponses automatiques à des questionnaires à choix multiples à l'aide de « Larges Modèles de Langue ». In *Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles. Atelier DÉfi Fouille de Textes (DEFT)*.

CHIANG W.-L., LI Z., LIN Z., SHENG Y., WU Z., ZHANG H., ZHENG L., ZHUANG S., ZHUANG Y., GONZALEZ J. E., STOICA I. & XING E. P. (2023). Vicuna : An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

CHUNG H. W., HOU L., LONGPRE S., ZOPH B., TAY Y., FEDUS W., LI E., WANG X., DEHGHANI M., BRAHMA S. *et al.* (2022). Scaling instruction-finetuned language models. *arXiv preprint arXiv :2210.11416*.

FAVRE B. (2023). LIS@DEFT'23 : les LLMs peuvent-ils répondre à des QCM ? (a) oui ; (b) non ; (c) je ne sais pas. In *Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles. Atelier DÉfi Fouille de Textes (DEFT)*.

HU E. J., SHEN Y., WALLIS P., ALLEN-ZHU Z., LI Y., WANG S., WANG L. & CHEN W. (2021). LoRa : Low-rank adaptation of large language models. *arXiv preprint arXiv :2106.09685*.

LABRAK Y., BAZOGE A., DUFOUR R., DAILLE B., GOURRAUD P.-A., MORIN E. & ROUVIER M. (2022). FrenchMedMCQA : A French multiple-choice question answering dataset for medical domain. In *Proceedings of the 13th International Workshop on Health Text Mining and Information Analysis (LOUHI)*, p. 41–46, Abu Dhabi, United Arab Emirates (Hybrid) : Association for Computational Linguistics.

LABRAK Y., BAZOGE A., DUFOUR R., ROUVIER M., MORIN E., DAILLE B. & GOURRAUD P.-A. (2023). DrBERT : A Robust Pre-trained Model in French for Biomedical and Clinical domains. In *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics (ACL'23), Long Paper*, Toronto, Canada : Association for Computational Linguistics.

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

MARTIN L., MULLER B., ORTIZ SUÁREZ P. J., DUPONT Y., ROMARY L., DE LA CLERGERIE É. V., SEDDAH D. & SAGOT B. (2020). CamemBERT : a Tasty French Language Model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

MEONI S., TOUCHENT R. & DE LA CLERGERIE (2023). Passe ta pharma d'abord ! In *Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles. Atelier DÉfi Fouille de Textes (DEFT)*.

MUENNIGHOFF N., WANG T., SUTAWIKA L., ROBERTS A., BIDERMAN S., SCAO T. L., BARI M. S., SHEN S., YONG Z.-X., SCHOELKOPF H. *et al.* (2022). Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv :2211.01786*.

PAL A., UMAPATHI L. K. & SANKARASUBBU M. (2022). MedMCQA : A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on Health, Inference, and Learning*, p. 248–260 : PMLR.

TOUCHENT R., ROMARY L. & DE LA CLERGERIE E. V. (2023). CamemBERT-bio : Un modèle de langue français savoureux et meilleur pour la santé. In *Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles*.

TOUVRON H., LAVRIL T., IZACARD G., MARTINET X., LACHAUX M.-A., LACROIX T., ROZIÈRE B., GOYAL N., HAMBRO E., AZHAR F. *et al.* (2023). Llama : Open and efficient foundation language models. *arXiv preprint arXiv :2302.13971*.

WELBL J., LIU N. F. & GARDNER M. (2017). Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, p. 94–106, Copenhagen, Denmark : Association for Computational Linguistics. DOI : 10.18653/v1/W17-4413.

*Actes de CORIA-TALN 2023*
*Paris, France, 5 au 9 juin 2023*

# Passe ta pharma d'abord !

Simon Meoni [1, 2, *] Rian Touchent [1, *] Éric de la Clergerie [1]
(1) Inria, Paris, France
(2) Arkhn, Paris, France
simon.meoni@arkhn.com, rian.touchent@inria.fr,
eric.de_la_clergerie@inria.fr

RÉSUMÉ

Nous présentons les 3 expériences menées par l'équipe ALMAnaCH - Arkhn et leurs résultats pour le DÉfi Fouille de Textes (DEFT) 2023. Les scores sont encourageants mais suggèrent surtout de nouveaux éléments à prendre en compte pour réussir ce défi. Nous avons exploré différentes approches avec des modèles de tailles variables et modélisé la tâche de différentes manières (classification multi-labels, implication textuelle, séquence à séquence). Nous n'avons pas observé des gains de performance significatifs. Nos expériences semblent montrer la nécessité de l'utilisation de bases de connaissances externes pour obtenir de bons résultats sur ce type de tâche.

ABSTRACT

**Graduate Pharma First !**

We present 3 experiments and results obtained by the ALMAnaCH - Arkhn team for the Text Mining Challenge (DEFT) 2023. We have explored various approaches with models of varying sizes and by modeling the task differently (multi-label classification, natural language inference, sequence-to-sequence). The results are encouraging but suggest new elements to consider to succeed in this challenge. We have not observed significant performance gains. Our experiments indicate the necessity of using external knowledge bases to achieve good results on this type of task.

MOTS-CLÉS : biomédical, pharmacologie, QCM, implication textuelle, affinage avec instruction, TAL.

KEYWORDS: biomedical, pharmacology, MCQA, textual entailment, prompt-tuning, NLP.

## 1 Introduction

Dans cet article, nous présentons notre participation au DÉfi Fouille de Textes (DEFT) 2023, une campagne d'évaluation francophone. L'objectif de ce défi est de développer des approches permettant de répondre automatiquement à des questionnaires à choix multiples issus d'annales d'examens de pharmacie.

*. Ces auteurs ont contribué de manière égale à ce travail

## 2 Corpus

FrenchMedMCQA (Labrak *et al.*, 2022) est composé de 3105 questions fermées, extraites d'annales françaises d'examens de pharmacie. Chaque question est associée à un identifiant et cinq réponses possibles. Le nombre de bonnes réponses par question varie entre 1 et 5. Le corpus est divisé en trois sous-ensembles : entraînement, développement et test. Le corpus d'entraînement représente 70% des questions, celui de développement 10%, et celui de test 20%.

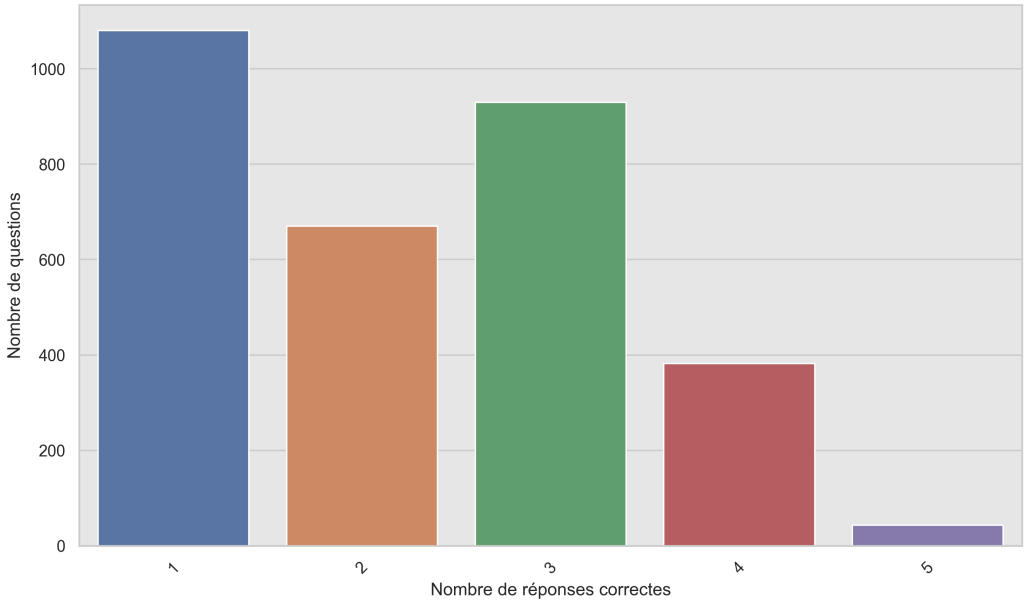


FIGURE 1 – Répartition du nombre de réponses correctes par question

Une exploration des données Fig. 1 montre un faible nombre de questions à 4 ou 5 réponses correctes. La majorité des questions n'ont qu'une seule bonne réponse, ce qui est souvent explicité dans la question elle-même. Le nombre de bonnes réponses n'est pas précisé dans la question quand il est supérieur à 1.

## 3 Description de la tâche

Le défi comporte deux tâches principales. La première tâche consiste à identifier automatiquement l'ensemble des réponses correctes parmi les cinq options proposées pour chaque question. L'évaluation de cette tâche est basée sur le taux de réponses parfaitement justes (*Exact Match Ratio*, EMR) et le taux de réponses justes parmi l'ensemble des réponses et références (*Hamming Score*). Le classement final des équipes se fait en fonction de l'Exact Match Ratio.

La deuxième tâche annexe consiste à estimer le nombre de réponses supposément justes pour chaque question, qui peut varier de 1 à 5. L'évaluation de cette tâche se fait à l'aide des métriques de précision et de score F1.

Seule la première tâche a été étudiée dans notre participation.

# 4 Méthode

## 4.1 Découpage du corpus et protocole expérimental

| Run | Modèle | Taux d'apprentissage | Taille de lots | Accumulation de gradients | Epochs |
|---|---|---|---|---|---|
| run 1 | `camembert-base` | $1e^{-5}$ | 8 | 2 | 25 |
| run 2 | `camembert-bio-base` | $4e^{-5}$ | 4 | 4 | 25 |
| run 3 | `flan-t5-xl` | $4e^{-5}$ | 1 | 16 | 10 |

TABLE 1 – Hyperparamètres retenus pour chaque run

Pour la run 1 et la run 2, nous avons exploré les modèles `camembert-base` (Martin *et al.*, 2020), `camembert-bio-base` (Touchent *et al.*, 2023) et `DrBERT-7GB` (Labrak *et al.*, 2023) et le taux d'apprentissage entre $1e^{-5}$ et $4e^{-5}$. Nous avons ensuite sélectionné les hyperparamètres (Table 1) donnant les meilleurs scores sur le jeu de développement en utilisant optuna (Akiba *et al.*, 2019)

Pour la run 3, nous avons exploré les modèles `t5-base`, `t5-large`, `flan-t5-xl`, `SciFive-base-Pubmed_PMC`, `mt5-base`.

Nous avons adapté le découpage initial du corpus à nos besoins pour l'ensemble de nos expériences comme suit :

— notre corpus d'entraînement contient 80% du jeu de données d'entraînement initial ;
— notre corpus de validation est le même que le jeu de données de développement initial ;
— notre corpus de test contient 20% du jeu de données d'entraînement initial ;
— notre corpus de soumission correspond au jeu de test initial ;

Lors de l'entraînement et pour chaque *epoch*, nous mesurons l'Exact Match Ratio sur le jeu de développement afin de sélectionner les meilleurs paramètres du modèle lors de la phase d'entraînement. Le corpus de test nous a permis d'évaluer nos approches avant la soumission. Nous sélectionnons le modèle ayant obtenu le meilleur score sur le corpus de test afin de l'utiliser pour la prédiction sur le corpus de soumission.

## 4.2 run 1 : Classification multi-labels

Notre première approche est basée sur de la classification multi-labels. La question et les 5 réponses correspondantes sont concaténées (Fig.2), séparées par un token spécifique, avant d'être encodées. La prédiction est ensuite réalisée avec une simple couche linéaire pour obtenir 5 logits, correspondants à la probabilité estimée par le réseau de neurones pour chaque question. La prédiction finale est obtenue en gardant les probabilités supérieures à 0.5, notre seuil, comme positifs.

## 4.3 run 2 : NLI

Dans la seconde approche, nous modélisons le défi comme une tâche de prédiction d'implication textuelle. Pour chaque question, nous construisons 5 paires de question-réponse. Où chacune de ses paires de question-réponse correspondent à la question suivi d'une des 5 propositions de réponses

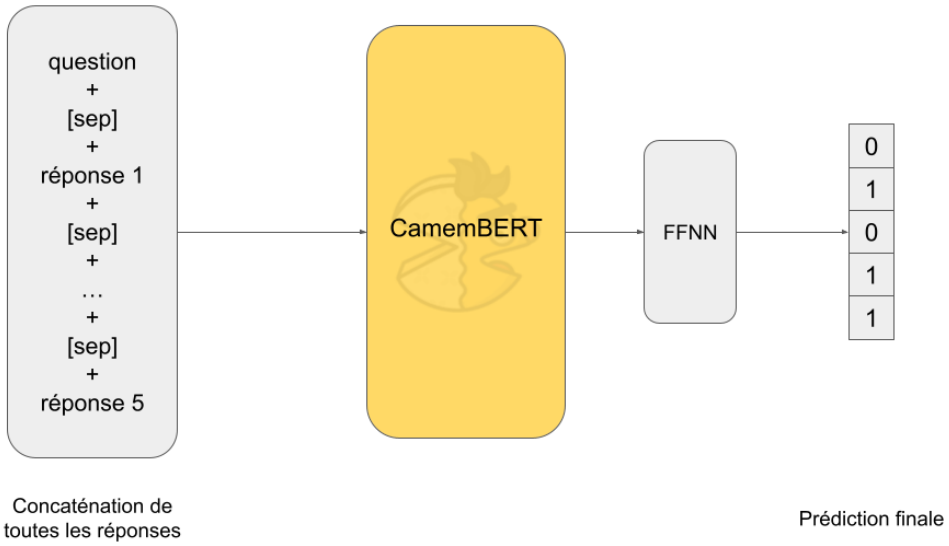


FIGURE 2 – Schéma de notre approche par classification multi-labels

(Fig.3). Chacune de ces paires est encodée par `camembert-base`, puis une couche linéaire va prédire si la question implique la réponse ou non, dans une tâche de classification binaire.

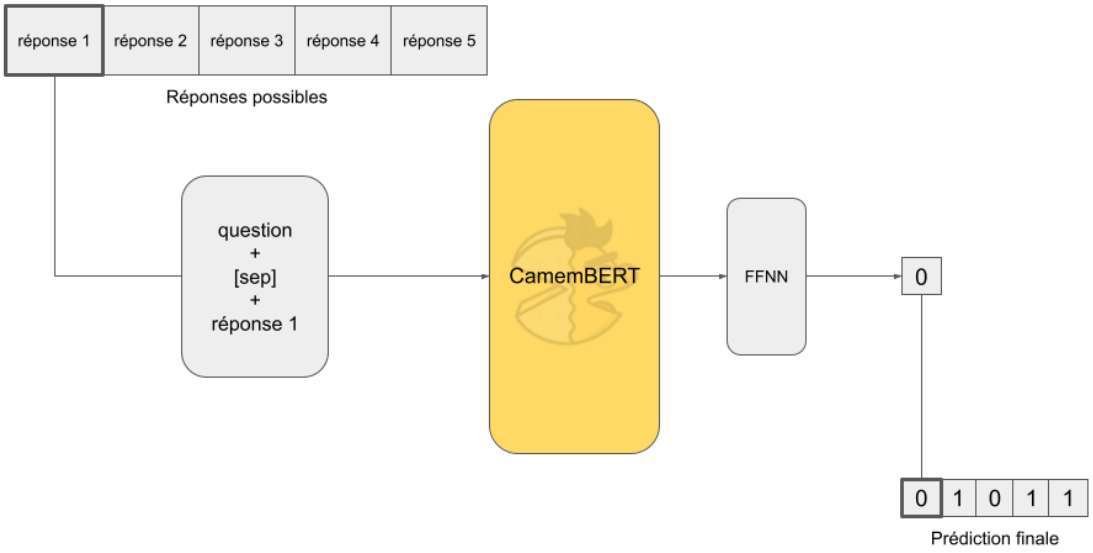


FIGURE 3 – Schéma de notre approche par NLI

La prédiction est alors réalisée 5 fois par question. Les 5 probabilités sont ensuite concaténées, et deviennent positives ou négatives en fonction du seuil, ce qui constitue la prédiction finale.

## 4.4 run 3 : Instruction-seq2seq

Pour cette run, nous nous sommes inspirés des travaux de Wang *et al.* (2022). Dans ce papier, les auteurs convertissent des tâches d'extraction d'entités nommées en une tache de séquence à séquence afin de les adapter à des modèles de type encodeur-décodeur tel que `t5` (Raffel *et al.*, 2020). Le but est de fournir au modèle une instruction en entrée afin d'avoir en sortie les résultats désirés comme illustrés sur la figure 4. La phase d'entraînement est un affinage classique sur la tâche. Quant à la phase d'évaluation, elle consiste à structurer la sortie textuelle de `t5` en une sortie interprétable pour un script d'évaluation donné ou un cas d'usage précis. Pour les besoins de la tâche, nous avons adapté cette technique en utilisant seulement une seule tâche principale. D'autre part, Nous avons essayé différents modèles tels que `t5`, `t5-large` et `flan-t5-xl` (Chung *et al.*, 2022). Pour la soumission nous avons sélectionné le meilleur, à savoir `flan-t5-xl`.

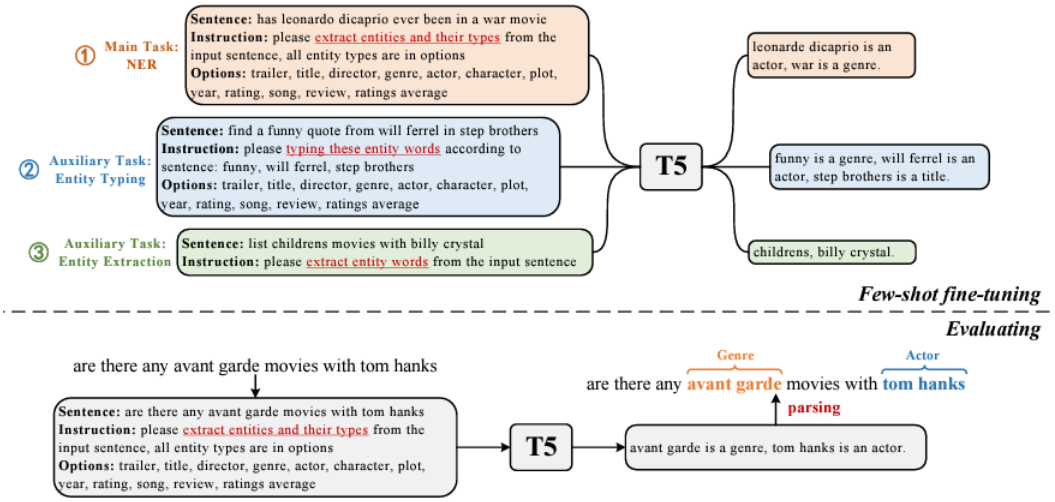


FIGURE 4 – Représentation du *framework* InstructionNER de Wang *et al.* (2022)

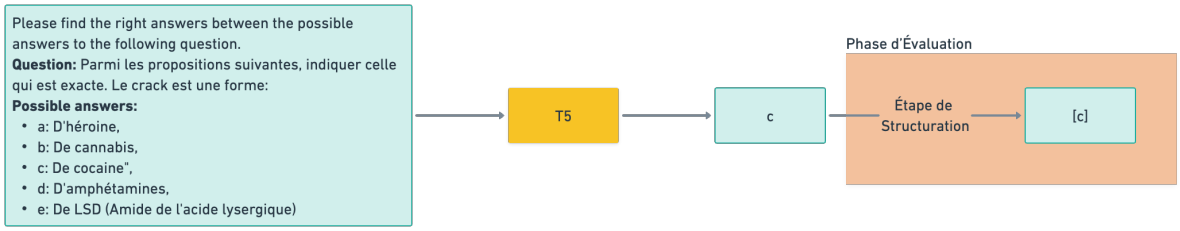


FIGURE 5 – Schéma de notre méthode par instruction sur un modèle de séquence à séquence

# 5 Résultats

En termes de taux de réponses parfaitement justes (Exact Match Ratio - EMR), le système NLI s'est légèrement démarqué des autres, bien que les différences ne soient pas significatives. En effet, les scores EMR des trois systèmes se situent dans le même ordre de grandeur et sont relativement faibles (Table 2). Nous n'avons pas réussi à obtenir des systèmes assez fiables pour cette tâche. Nous

| Nom du système | Hamming | EMR |
|---|---|---|
| multilabel-classification | 33.27 | 12.22 |
| nli | 33.67 | 14.15 |
| instruction-seq2seq | 35.96 | 13.67 |

TABLE 2 – Comparaison des performances entre les différents systèmes

pensons que c'est du à un manque d'accès à des connaissances précises dans certains cas de figures (par exemple sur des valeurs numériques). L'utilisation de connaissances externes aurait pu être une option.

En revanche, en considérant le score de Hamming, qui mesure la proportion de réponses correctes identifiées parmi les options proposées, le système instruction-seq2seq s'est avéré légèrement plus performant par rapport aux autres systèmes. Il a démontré une meilleure capacité à identifier les bonnes réponses parmi les options proposées. Afin d'améliorer ce système, nous aurions pu filtrer les réponses proposées par ce modèle à l'aide d'un modèle entraîné à cette ou à l'aide de techniques basées sur des graphes de connaissances.

La run 3 met en évidence un constat intéressant. Malgré l'utilisation de modèles plus larges tels que `flan-t5-xl`, nous n'avons pas observé d'augmentation significative des performances par rapport à l'utilisation de modèles de taille plus réduite. Ce qui signifie que la part de données pharmacologiques durant le pré-entraînement de `flant-t5-xl` n'est pas assez importante ou que les tâches de pré-entraînement ne sont pas assez adaptées à notre cas d'usage.

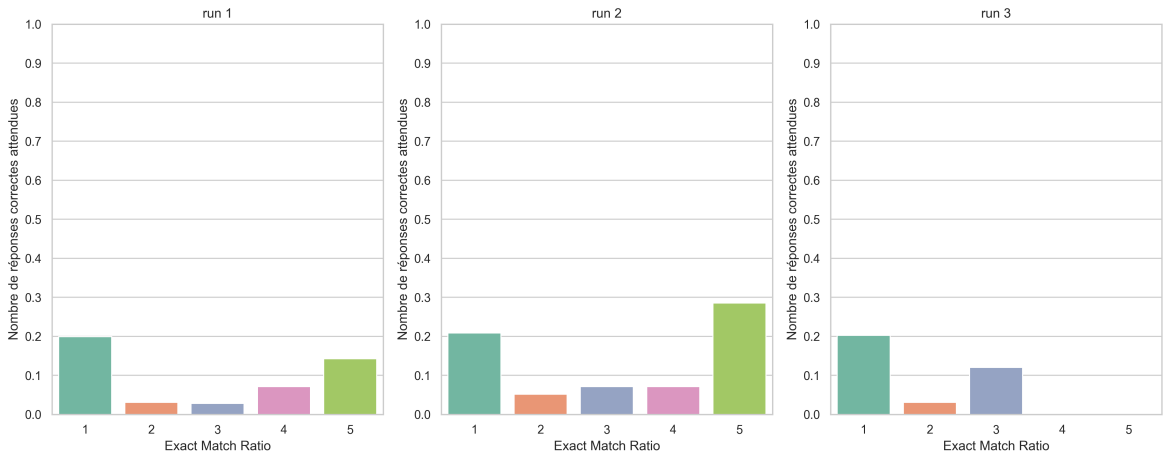


FIGURE 6 – EMR pour chaque nombre de bonnes réponses attendues

Dans la Fig. 6, on observe que la run 3 obtient un EMR de 0 si on ne prend en compte que les questions où entre 4 et 5 réponses sont attendues. Cela s'explique par une quasi absence de prédiction à 5 réponses de la part de `flan-t5-xl` (Fig.7). Il semblerait que T5 a amplifié le biais de la distribution du jeu d'entraînement, en se concentrant principalement sur des prédictions à 1 ou 3 réponses. C'est cependant aussi T5 qui obtient assez légèrement les meilleurs résultats sur ces deux catégories.

La run 2 semble moins sensible aux biais de la distribution du jeu d'entraînement. Le modèle de la run 2 semble moins frileux à prédire 5 réponses pour une question, ce qui lui permet d'obtenir un très bon score EMR pour les questions à 5 réponses (Fig.6). C'est également la run qui a le meilleur

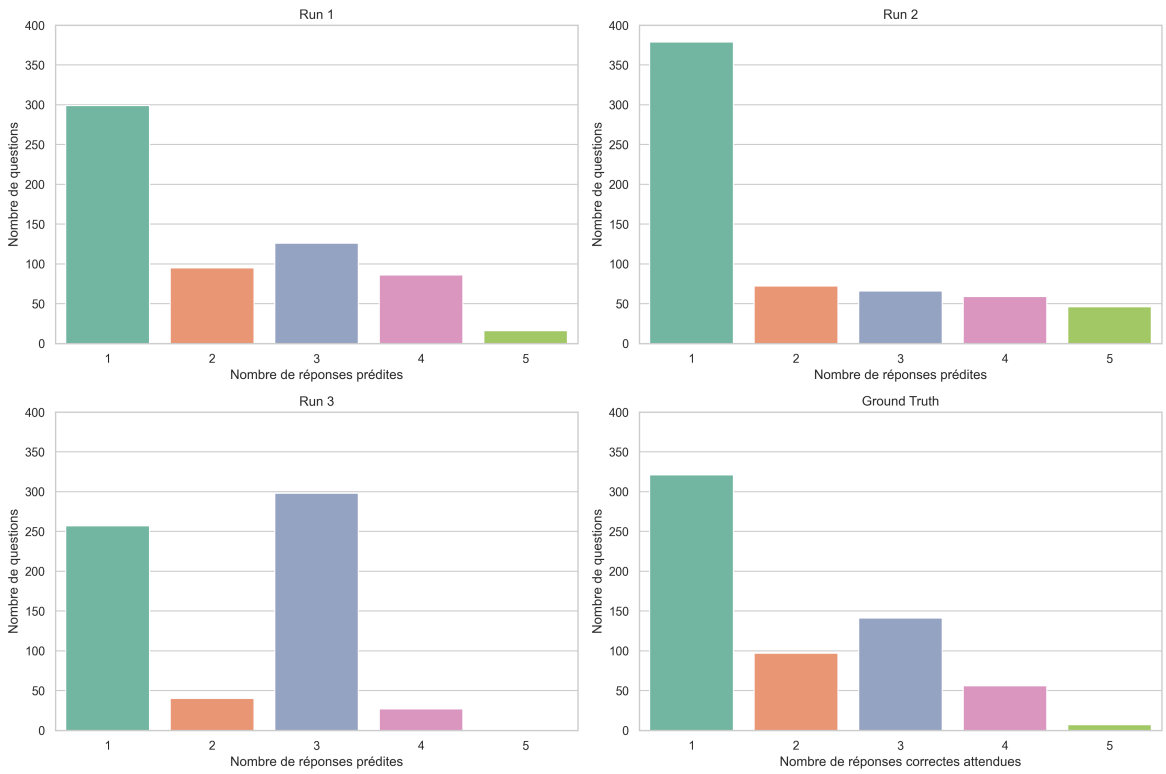


FIGURE 7 – Distribution du nombre de réponses prédites ou attendues

score EMR global. Cependant, on peut également déduire que ce modèle a du mal à prédire le bon nombre de réponses puisque la distribution finale du nombre de réponse prédite par question est assez différente de la distribution du jeu de test.

Enfin la run 1 reproduit correctement la distribution du nombre de réponses à prédire par question, elle a cependant un EMR global légèrement moins bon que la run 2. Il semblerait que le modèle de la run 1 soit meilleur pour prédire le nombre de réponse à prédire, mais plus faible pour trouver les bonnes réponses.

Avec ces résultats, on peut imaginer qu'une approche ensembliste pourrait permettre de rassembler les points forts de chaque run, qui semble souffrir de failles différentes. Il semblerait également que la seconde tâche aurait pu aider nos modèles. En effet, dans certaines de nos runs les modèles font de grandes erreurs dans le nombre de réponse à prédire avant même de trouver les bonnes réponses. Ainsi, jouer avec le seuil ou utiliser un modèle pour prédire le nombre de réponses correctes a priori pour permettre d'augmenter les performances.

# 6 Conclusion et perspectives

Notre participation au DÉfi Fouille de Textes (DEFT) 2023 nous donne des conclusions intéressantes. Les scores relativement faibles de nos systèmes, bien que d'approches différentes, semblent montrer qu'il est difficile d'obtenir de bonnes performances pour cette tâche sans utilisation de bases de connaissances externes. Ni les connaissances apprises à l'aide du jeu d'entraînement de FrenchMedMCQA, ni celles apprises lors du pré-entraînement de nos modèles ne semblent suffisantes. Les

jeux de pré-entraînement de nos modèles sont variés, avec des jeux spécialisés pour CamemBERT-bio et DrBERT, ou des jeux de très grande taille comme avec Flan-T5 (Chung *et al.*, 2022). Ces types de modèle montrent de bonnes performances sur un certain nombre de tâches biomédicales sans utilisation de bases de connaissances externes (Lehman *et al.*, 2023).

En effet, les questions de ce jeu de données sont très spécifiques et basées sur de la connaissance. Il serait alors intéressant d'explorer par la suite des méthodes qui exploitent des bases de connaissances externes.

Une première méthode serait d'injecter dans les instructions que l'on donne à notre modèle de langue, un contexte pertinent vis-à-vis de la question. Ce contexte pourrait alors être extrait depuis un corpus biomédical dans laquelle on aurait fait une recherche sémantique ou une recherche par mots clés type MeSH, et ainsi donner les connaissances nécessaires pour répondre à la question (Noh & Kavuluru, 2018). Cette méthode n'assure cependant pas que le contexte contient les informations nécessaires pour répondre, mais seulement des phrases sémantiques proches de la question dans un corpus donné.

Une autre approche serait de reconnaître les différents types de question. En effet certaines sont basées sur du calcul numérique. On pourrait alors détecter le type de question et faire appel par la suite à un agent spécialisé pour ce problème. Schick *et al.* (2023) montrent qu'un modèle de langue génératif est capable d'identifier différents problèmes et d'utiliser d'autres agents pour y répondre. Cela demande cependant de trouver à l'avance les différents types de questions et de construire ou identifier un agent pertinent pour chacun d'entre eux.

Il est également possible d'utiliser une base de connaissances externe pour générer de nouvelles questions (Sileo *et al.*, 2023). Cela nous permettrait d'avoir un jeu d'entraînement plus conséquent d'une part, et d'autre part d'encoder des connaissances externes directement dans le jeu d'entraînement.

En conclusion, pour améliorer les performances de réponse automatique à des questionnaires à choix multiples en pharmacologie, il serait intéressant d'explorer des méthodes qui exploitent des bases de connaissances externes, en générant de nouvelles questions, en injectant du contexte pertinent dans les instructions données à un modèle génératif ou en construisant et en appelant des agents pertinents pour chacun des types de questions possibles. Ces pistes de recherche offrent des perspectives intéressantes pour de futurs travaux dans le domaine.

# Références


AKIBA T., SANO S., YANASE T., OHTA T. & KOYAMA M. (2019). Optuna : A next-generation hyperparameter optimization framework. *CoRR*, **abs/1907.10902**.

CHUNG H. W., HOU L., LONGPRE S., ZOPH B., TAY Y., FEDUS W., LI E., WANG X., DEHGHANI M., BRAHMA S., WEBSON A., GU S. S., DAI Z., SUZGUN M., CHEN X., CHOWDHERY A., NARANG S., MISHRA G., YU A., ZHAO V., HUANG Y., DAI A., YU H., PETROV S., CHI E. H., DEAN J., DEVLIN J., ROBERTS A., ZHOU D., LE Q. V. & WEI J. (2022). Scaling instruction-finetuned language models. DOI : 10.48550/ARXIV.2210.11416.

LABRAK Y., BAZOGE A., DUFOUR R., DAILLE B., GOURRAUD P.-A., MORIN E. & ROUVIER M. (2022). FrenchMedMCQA : A French multiple-choice question answering dataset for medical domain. In *Proceedings of the 13th International Workshop on Health Text Mining and Information Analysis (LOUHI)*, p. 41–46, Abu Dhabi, United Arab Emirates (Hybrid) : Association for Computational Linguistics.

LABRAK Y., BAZOGE A., DUFOUR R., ROUVIER M., MORIN E., DAILLE B. & GOURRAUD P.-A. (2023). Drbert : A robust pre-trained model in french for biomedical and clinical domains.

LEHMAN E., HERNANDEZ E., MAHAJAN D., WULFF J., SMITH M. J., ZIEGLER Z., NADLER D., SZOLOVITS P., JOHNSON A. & ALSENTZER E. (2023). Do we still need clinical language models ?

MARTIN L., MULLER B., JAVIER ORTIZ SUÁREZ P., DUPONT Y., ROMARY L., VILLEMONTE DE LA CLERGERIE E., SAGOT B. & SEDDAH D. (2020). Les modèles de langue contextuels Camembert pour le français : impact de la taille et de l'hétérogénéité des données d'entrainement. In C. BENZITOUN, C. BRAUD, L. HUBER, D. LANGLOIS, S. OUNI, S. POGODALLA & S. SCHNEIDER, Éds., *JEP-TALN-RECITAL 2020 - 33ème Journées d'Études sur la Parole, 27ème Conférence sur le Traitement Automatique des Langues Naturelles, 22ème Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues*, p. 54–65, Nancy, France : ATALA. HAL : hal-02784755.

NOH J. & KAVULURU R. (2018). Document Retrieval for Biomedical Question Answering with Neural Sentence Matching. *Proc Int Conf Mach Learn Appl*, **2018**, 194–201.

RAFFEL C., SHAZEER N., ROBERTS A., LEE K., NARANG S., MATENA M., ZHOU Y., LI W. & LIU P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, **21**(140), 1–67.

SCHICK T., DWIVEDI-YU J., DESSÌ R., RAILEANU R., LOMELI M., ZETTLEMOYER L., CANCEDDA N. & SCIALOM T. (2023). Toolformer : Language models can teach themselves to use tools.

SILEO D., UMA K. & MOENS M.-F. (2023). Generating multiple-choice questions for medical question answering with distractors and cue-masking.

TOUCHENT R., ROMARY L. & VILLEMONTE DE LA CLERGERIE E. (2023). CamemBERT-bio : Un modèle de langue français savoureux et meilleur pour la santé. working paper or preprint.

WANG L., LI R., YAN Y., YAN Y., WANG S., WU W. & XU W. (2022). Instructionner : A multi-task instruction-based generative framework for few-shot ner.