Actes de JEP-TALN-RECITAL 2024
Toulouse, France, 8 au 12 juillet 2024

Trois méthodes Sorbonne et SNCF pour la résolution de QCM (DEFT2024)

Tom Rousseau[1], Marceau Hernandez[2], Iglika Stoupak[2], Angelo Mendoca-Manhoso[1], Andrea Blivet[1], Chang Liu[1], Toufik Boubehziz[3], Corina Chutaux[2], Gaël Guibon[4], Gaël Lejeune[2], Luce Lefeuvre[1]

(1) SNCF DTIPG prenom.nom@sncf.fr, ext.andrea.blivet,chang.liu@sncf.fr
(2) STIH, CERES/Sorbonne Université prenom.nom@sorbonne-universite.fr
(3) LIG/Université de Grenoble Alpes toufik.boubehziz@univ-grenoble-alpes.fr
(4) Université de Lorraine, CNRS, LORIA gael.guibon@loria.fr

RÉSUMÉ

Cet article décrit la participation de l'équipe Sorbonne-SNCF au Défi Fouille de Textes 2024, se concentrant sur la correction automatique de QCM en langue française. Le corpus, constitué de questions de pharmacologie, a été reformulé en assertions. Nous avons employé des techniques avancées de traitement du langage naturel pour traiter les réponses. Trois approches principales, Nachos_LLM, TTGV_byfusion, et TTGV_ollama_multilabel, sont présentées avec des scores EMR respectifs de 2.94, 4.19 et 1.68. Les résultats obtenus montrent des niveaux de précision différents, en soulignant les limites des approches multi-étiquettes. Des suggestions d'amélioration incluent l'ajustement des modèles de langage et des critères de classification.

ABSTRACT

This article describes the participation of the Sorbonne-SNCF team in the 2024 Text Mining Challenge, focusing on the automatic correction of French MCQs. The corpus, consisting of pharmacology questions, was reformulated into assertions. We employed advanced natural language processing techniques to evaluate the responses. Three main approaches, Nachos_LLM TTGV_byfusion, and TTGV_ollama_multilabel, are presented with EMR scores of 2.94, 4.19, and 1.68, respectively. The results show varying levels of accuracy, highlighting the limitations of multi-label approaches. Suggestions for improvement include adjusting language models and classification criteria.

MOTS-CLÉS : QCM, assertions, plongements, similarité, n-grammes, Grand Modèle de Langue.

KEYWORDS: MCQ, assertions, embeddings, similarity, n-grams, LLM.

1 Contexte et état de l'art en correction automatique de QCM

Nous présentons dans cet article la participation de l'équipe Sorbonne-SNCF au Défi Fouille de Textes 2024 (Labrak *et al.*, 2022), union de l'équipe SPQR et de l'équipe TTGV, qui avaient participé au DEFT 2023. Les méthodes mises en place résultent ainsi des expériences des deux équipes. La tâche de question-réponse (Q/R) est assez classique en Traitement Automatique du Langage (TAL) et est étroitement liée à des méthodes d'apprentissage utilisées dans l'enseignement, telles que les QCM

(Questions à Choix Multiples) ou la compréhension écrite (Touissi *et al.*, 2022; Soares *et al.*, 2021). Elle se compose traditionnellement de trois sous-tâches : la classification des questions, la recherche d'information et l'extraction de réponses. Selon le type de réponse requis, les questions peuvent être factuelles, de liste, de définition et questions complexes (Kolomiyets & Moens, 2011). La dernière catégorie est particulièrement difficile car elle implique généralement la fusion ou le post-traitement d'éléments textuels récupérés. La mesure la plus courante pour évaluer un système de Q/R est l'Exact Match Ratio (EMR) ; c'est à dire la proportion de réponses entièrement correctes.

Dans les années 2000, les techniques courantes pour résoudre la tâche de Q/R comprenaient le marquage de parties du discours, la reconnaissance d'entités nommées, les ontologies telles que WordNet pour fournir des synonymes de termes et les comparaisons textuelles basées sur la similarité cosinus. L'apprentissage automatique était souvent appliqué à des sous-tâches telles que la classification du type de question impliqué. Par la suite, les réseaux de neurones comme le LSTM ont fourni des solutions de pointe. Plus récemment, des modèles de langage tels que BERT ont été affinés pour la tâche de Q/R, créant ainsi des systèmes solides et potentiellement spécifiques à un domaine.

L'avènement des grands modèles de langage (LLM), qui peuvent être considérés comme des bases de connaissances, a conduit à d'importants changements quantitatifs et qualitatifs dans les systèmes de Q/R. Zhuang *et al.* (2023) évaluent la famille ChatGPT sur plusieurs ensembles de données questions-réponses complexes, révélant le potentiel du modèle à surpasser certains modèles de pointe conçus spécifiquement pour la tâche. L'inconvénient majeur résidant alors dans le post-traitement permettant que les résultats du LLM permettent l'application d'une évaluation telle que l'EMR.

Le défi DEFT (Défi Fouille de Textes), organisé ette année par les Universités d'Avignon d'Aix-Marseille et de Nantes, vise à motiver et évaluer le développement de systèmes question-réponse de pointe spécialement adaptés à la langue française en fournissant des ensembles de données d'entraînement et de test et en évaluant les performances des participants sur la base de EMR et, éventuellement, de mesures supplémentaires (Labrak *et al.*, 2023a). Les modèles les plus performants du défi 2023 ont utilisé des LLM. Concrètement, l'équipe gagnante a affiné un modèle LLaMa grâce à la méthode d'adaptation LoRA. Néanmoins, il est important de noter que la tâche privilégie l'explicabilité des systèmes conçus et l'utilisation de corpus prédéfinis en langue française.

2 Description des données du défi et des méthodes développées

Le corpus FrenchMed MCQA est constitué de QCM provenant d'examens de pharmacologie français. Ces questions et réponses ont été élaborées manuellement par des experts médicaux et sont destinées à des étudiants de niveau universitaire. À chaque question est associée : un identifiant, cinq réponses possibles étiquetées de A à E, la ou les réponse(s) correcte(s) et le nombre de réponses correctes.

ID	Question	Réponses
...	Parmi les bactéries suivantes, une seule ne peut généralement pas être responsable d'une méningite aiguë, laquelle ?	a. Haemophilus influenzae b. Streptococcus pneumoniae **c. Neisseria gonorrhoeae** d. Tétracycline e. Mycobacterium tuberculosis

TABLE 1 – Exemple de question extraite du corpus d'entraînement (la bonne réponse est en gras)

2.1 Reformulation des réponses au QCM en assertions

Pour résoudre automatiquement les QCM, nous commençons par reformuler les questions et réponses en assertions. L'objectif est de faciliter la comparaison avec des bases de connaissances ou des modèles de langue. Les données d'entraînement sont reformulées en concaténant une question Q avec ses bonnes réponses R_i, où $i = \{a, b, c, d, e\}$ et n est le nombre de réponses correctes. L'ensemble $Q + R_i$ est reformulé en assertions à en traitant séparément les réponses affirmatives et négatives.

2.1.1 Reformulation des réponses affirmatives

Dans cette première stratégie, nous procédons par l'élimination des modules récurrents avec des expressions régulières puis le traitement des questions par morceaux suivant la classification présentée par Bezançon *et al.* (2023) où chaque question est composée de trois parties séparées par des virgules : $[P_1], [P_2], [P_3]$ où P_i pour $i = 1, 2, 3$ est une partie.
Le traitement de chaque question se fait en plusieurs phases, en commençant par **l'extraction des termes médicaux** et la détection de la négation. L'extraction est réalisée en comparaison avec une liste de mots courants en français. Ensuite, les **éléments lexicaux récurrents sont détectés**. Il s'agit des éléments nécessitant un traitement particulier, tels que "parmi", "concernant", ou "indiquer". Si le morceau ne contient pas de terme médical, il est éliminé. La dernière étape consiste en **la concaténation** de la question traitée avec une réponse proposée en fonction des éléments lexicaux récurrents détectés.

À l'issue de ces opérations, nous créons également la liste des termes contenus dans chaque phrase obtenue, pour cette sélection de termes seules les questions affirmatives sont considérées, comme dans la table 2 ci-dessous.

Phrase	**Termes**
Neisseria gonorrhoeae est une bactérie une seule ne peut généralement pas être responsable d'une méningite aiguë.	aiguë, bactérie, méningite

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

Le jeu de données de test est également reformulé, où chaque question est concaténée sous forme de phrase avec ses cinq réponses possibles. Nous ajoutons un champ pour les termes correspondants et un autre booléen qui indique si la question est affirmative ou négative. Après traitement, les premières réponses d'une question deviennent ainsi :

2.1.2 Reformulation des réponses négatives

Pour ne conserver que les questions négatives, nous filtrons celles contenant les mots : fausse, faux, inexacte, inexact, incorrect, incorrecte. Ensuite, nous extrayons le sujet de chaque question. Si la question se termine par un point d'interrogation, le sujet est le premier verbe rencontré, identifié avec le modèle pré-entraîné "fr_core_news_lg" de SpaCy. Nous cherchons des tokens de type `nmod` ou `obj` et identifions les tokens liés pour déterminer le sujet. Si la question ne se termine pas par " :", nous ajoutons ce caractère, découpons la question en segments avec " ?" comme délimiteur, puis nous

ID	Phrase	Termes	Négation
...	a. Dans les conditions physiologiques le pH le plus élevé est mesuré dans le suc gastrique.	*ph, physiologiques, gastrique , suc*	faux
	b. Dans les conditions physiologiques le pH le plus élevé est mesuré dans la bile vésiculaire.	*ph, physiologiques, vésiculaire*	
	c. Dans les conditions physiologiques le pH le plus élevé est mesuré dans le suc pancréatique.	*ph, physiologiques, pancréatique, suc*	

TABLE 3 – Exemples de réponses pour une question donnée

découpons le dernier segment avec " :". Nous appliquons ensuite une série de règles pour rendre la phrase syntaxiquement correcte.

Nous créons une liste des valeurs possibles (A à E) enlevons les réponses valides (champ "correct_answers") pour obtenir les réponses invalides. En travaillant sur les questions négatives, nous rendons négatives les réponses valides pour obtenir une liste d'assertions vraies sur le sujet. Nous utilisons le modèle pré-entraîné pour traiter les réponses valides, identifiant le premier verbe ou auxiliaire rencontré pour l'intégrer dans une structure négative ("ne ... pas"). Si la phrase ne contient pas de verbe, nous ajoutons une négation ("n'est pas"). Enfin, nous assemblons le sujet, le groupe verbal reconstitué et la réponse pour obtenir des assertions vraies, qu'elles soient négatives ou positives.

2.1.3 Optimisation des assertions

À partir des traitements effectués ci-dessus, nous obtenons une grande partie d'assertions reformulées. Cependant, nous remarquons tout de même la présence de bruit, notamment des adjectifs tels que « vrai », « exact », « correct », « juste » ainsi que des pronoms tels que « quel » et « lequel », enfin les noms tels que « proposition », « affirmation », « réponse » utilisés pour formuler des interrogations.

Les assertions ont été traitées avec des expressions régulières afin d'en supprimer les modules empêchant la bonne construction syntaxique basé sur la procédure décrite dans la section 2.1. Après observation des assertions syntaxiquement incorrectes, 9 modules ont été retirés. Ils sont du type : « Quelles sont les propositions vraies », « Celle qui est vraie/exacte/juste », « Parmi les propositions suivantes », « Laquelle ou lesquelles de ces propositions/affirmations est vraie/exacte/juste ». Ci-dessous un exemple d'assertion syntaxiquement incorrecte avec ces termes, avant traitement (1) et après (2) :

(1) Concernant la chimioprophylaxie anti-paludeenne laquelle ou lesquelles est / sont vraie(s) la Malarone® (Atovaquone + Proguanil) est photosensibilisante

(2) Concernant la chimioprophylaxie anti-paludeenne, la Malarone® (Atovaquone + Proguanil) est photosensibilisante

Certaines questions ne sont pas détectées et traitées par les expressions régulières mentionnées ci-dessus, notamment les questions posées par le pronom interrogatif « quel ». La séquence dirigée par ce pronom peut se trouver dans le sujet, comme dans l'exemple (3) mais également dans le complément d'objet ou dans le complément circonstanciel :

(3) Quel signe clinique n'est jamais retrouvé lors d'une intoxication aigue à la cocaïne ?

Pour reformuler ces questions en assertions, nous avons opté pour le remplacement direct des séquences de mots dont le pronom « quel » est la tête par les réponses correspondantes. Cette opération consiste à identifier l'arbre syntaxique ayant pour sommet « quel », en utilisant l'analyse des dépendances syntaxiques fournie par SpaCy.

2.1.4 Limites des assertions reformulées

Certaines assertions demeurent syntaxiquement incorrectes et sont imparfaites, bien que les informations clés de la question et de la réponse soient présentes. Cela est dû à la variété des formulations des questions, qui demanderait de nombreuses opérations de pré-traitement pour prendre en compte tous les cas possibles.

2.2 Description de la méthode Nachos_LLM

Principe : Dans cette approche, nous considérons que chaque paire de question et de réponse possible issue des QCM forme une assertion a à tester : vraie ou fausse. L'évaluation logique repose alors sur une comparaison avec une base B de connaissances médicales et pharmaceutiques, essentiellement construite à partir du corpus biomédical *NACHOS*(Labrak *et al.*, 2023b) : a est alors considérée vraie lorsqu'il existe une affirmation $b \in B$ « suffisamment proche » de a ; autrement, a est considérée fausse.

Cette approche soulève des questions pratiques, auxquelles nous chercherons à répondre : (I) Comment transformer une paire de question-réponse en assertion, (II) Comment comparer deux affirmations de ce genre, (III) Comment construire une base de connaissances adaptée à ce mécanisme, (IV) Comment tester une affirmation issue du QCM.

Méthodologie : La reformulation des paires de question-réponse en assertions est détaillée en 2.1. Pour faciliter la comparaison des assertions entre elles, nous avons choisi de les représenter par des plongements textuels, pour lesquels des mesures de similarité conventionnelles existent (notamment la distance euclidienne et la similarité cosinus, que nous avons utilisées). Le modèle permettant d'obtenir ces plongements est détaillé dans la suite. La base B de connaissances médicales que nous avons utilisée a été construite par plongement textuel des affirmations nous avons considérées comme vraies : les documents figurant dans NACHOS et les affirmations vraies tirées des QCM disponibles (train, dev et test). Ainsi, B prend la forme d'une matrice $M \in \mathbb{R}^{N \times D}$, où N est le nombre de documents pris en compte et D la taille (fixe) de leur plongement textuel. L'évaluation des assertions telles que nous les construisons repose alors sur les étapes suivantes :

1. Transformation d'une paire (q, r) de question-réponse en assertion a, prenant la forme d'un document de T tokens w_t : $a = (w_t)_{1 \leqslant t \leqslant T}$;
2. Calcul du plongement textuel $x = f(a)$ du document par application d'un modèle de plongement f à ses tokens ;
3. Recherche du plus proche voisin $x^\star$ de x dans B : $x^\star = \underset{y \in B}{\operatorname{argmin}} \operatorname{dist}(x, y)$ et $d^\star = \operatorname{dist}(x, y)$
4. Décision sur la valeur logique de a en fonction de $d^\star$: si $d^\star >$ seuil (pour un seuil fixé à l'avance), alors a est jugée vraie ; sinon, fausse.

Répondre à une question d'un QCM consiste alors à évaluer par cette méthode les assertions correspondant aux cinq réponses possibles. Un cas limite peut toutefois se produire : celui où les cinq

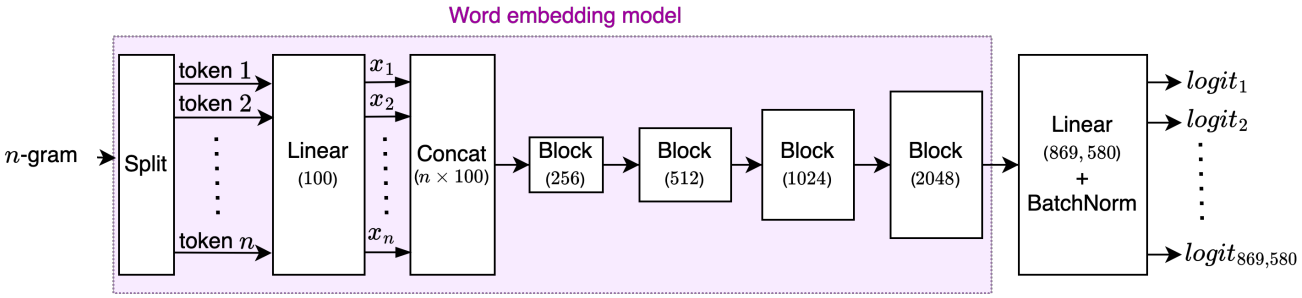

FIGURE 1 – Modèle de langue entraîné (~2 milliards de paramètres). « Block » correspond à une couche linéaire, suivie d'une couche de normalisation par batch (BatchNorm1d) et d'une activation ReLU. Les plongements textuels par mots sont obtenus par la partie du modèle encadrée en mauve, sans la couche "split" : la couche linéaire est commune à tous les mots (tokens) ; la sortie du dernier bloc « block » correspond à un plongement ($\in \mathbb{R}^{2048}$).

réponses sont jugées fausses – alors qu'au moins une est attendue. Dans cette éventualité, nous assouplissons notre règle de décision pour accepter une unique réponse : celle, parmi les cinq options, de plus petite distance à B (qui dépasse donc notre seuil).

Modèle de plongement textuel : Nous avons choisi de représenter les assertions, aux nombres de tokens inégaux, par un plongement textuel de taille fixe. Pour une assertion $a = (w_t)_{1 \leqslant t \leqslant T}$, nous calculons les plongements $(x_t)_{1 \leqslant t \leqslant T} = (f(w_t))_{1 \leqslant t \leqslant T} \in \mathbb{R}^{\mathbb{D}}$ (de taille fixe D) des tokens qui la composent, puis nous calculons leur moyenne arithmétique $x = \frac{1}{T} \sum_{t=1}^{T} x_t$: le vecteur $x \in \mathbb{R}^{\mathbb{D}}$ correspond au plongement de a. Le modèle de plongement f que nous utilisons est inspiré de word2vec. Dans son mode CBOW (*Continuous Bag-of-Words*), le réseau de neurones sous-tendant word2vec est entraîné sur une tâche de langue classique : prédire un mot compte tenu de ceux qui l'entourent. Apprendre cette tâche a entraîné un bénéfice collatéral : le réseau de neurones en question a appris un plongement textuel (par token) qui s'est révélé utile à d'autres tâches que celle d'origine. Suivant cette approche, cherchant à obtenir des plongements adaptés au contexte pharmaceutique, nous avons conçu notre propre modèle de langue. Différentes architectures ont été explorées : des réseaux de neurones *feed-forward*, *fully-connected* de profondeurs et de largeurs différentes. La tâche consiste à prédire le mot suivant un n-gramme donné. Le temps étant trop contraint pour explorer davantage de pistes, nous n'avons utilisé que des trigrammes ($n = 3$) et n'avons pas mené d'hyperoptimisation automatique. La taille du vocabulaire (dictionnaire construit sur notre base de connaissance et sur les assertions vraies du jeu de train) est de 869 580 tokens. Le jeu de train fourni a été réparti entre jeu d'entraînement (80%) et jeu de validation (20%). Le modèle le plus performant que nous avons obtenu (validation accuracy $= 41\%$) compte environ 2 milliards de paramètres. Son entraînement, distribué sur quatre GPU A100 (80 Go de VRAM), a duré près de 75h (46 époques) et a consommé environ 49,44 kWh (GPU : 27 kWh ; CPU : 8,44 kWh ; RAM : 14 kWh) [1].

2.3 Description de la méthode TTGV_byfusion

Nous reprenons ici la méthode BYFUSION développée dans Bezançon *et al.* (2023). Ainsi, nous allons énoncer les éléments principaux de la méthode ici, pour plus de détails, se référer à l'article en

1. A titre indicatif, au tarif « Bleu » (réglementé) actuellement appliqué en France par EDF (0,2516 €/kWh, en option « Base » ; voir grille tarifaire 2024), cette consommation d'énergie aurait été facturée 12,44 €.

question. Cette méthode se base sur un calcul de similarité entre des vecteurs d'effectifs de chaînes de caractères, permettant ainsi de comparer les énoncés "synthétiques" (questions combinées aux réponses possibles) avec une base de données médicales de référence.

La base de données médicales a été constituée à partir du jeu de données *NACHOS*, en excluant les phrases contenant moins de 3 mots différents. Les mots de chaque phrase restante, répondant à des critères spécifiques [2], ont été stockés sous forme de listes. À partir de ces listes et des questions, une matrice d'occurrences terme-document a été générée. Cette matrice filtre les mots apparaissant au moins une fois dans les questions et deux fois dans l'ensemble du corpus, diminuant ainsi l'espace mémoire. Le corpus fourni pour le défi est ainsi traité ligne par ligne et vectorisé en n_grammes de caractères (avec $5 < N < 6$ et *char_wb*) de manière à constituer une matrice de référence.

Pour chaque question à résoudre, cinq énoncés synthétiques sont créés par concaténation des question et réponses. Ces énoncés sont également vectorisés en (5-6)-grammes de caractères, puis comparés à la matrice de référence. La proximité entre les énoncés et les phrases/paragraphes du corpus *NACHOS* est mesurée par similarité cosinus. L'hypothèse sous-jacente est que l'énoncé synthétique issu d'une bonne réponse sera proche de phrases pertinentes du corpus de référence.

Il s'agit ensuite de sélectionner parmi ces énoncés la ou les bonnes réponses. Nous avons sélectionné les réponses selon deux critères : (1) Similarité maximale : L'énoncé le plus similaire avec une des lignes du corpus, (2) Seuil de similarité : Tous les énoncés avec une similarité dépassant 0.7 [3] avec une des phrases du corpus de référence. Les réponses ainsi sélectionnées sont considérées comme les bonnes réponses à la question donnée. La première méthode nous permet d'obtenir la réponse la plus correcte, au détriment du rappel. Lorsque la seconde méthode nous assure alors un meilleur rappel.

En parallèle des améliorations apportées sur le temps d'exécution ainsi que sur le coût en ressources, en changeant les structures de données et en filtrant en amont pour limiter l'espace de description. Varier la longueur des n-grammes et se limiter aux n-grammes contenus dans les bornes des mots (*char_wb*) a permis d'améliorer l'exactitude.

Nous avons également essayé de limiter les fréquences documentaires minimum et maximum des n-grammes, d'utiliser différents outils de mesure de similarité, notamment Bray-Curtis, ou encore utiliser des pondérations (type Tf-Idf et BM25). Cependant, ces pistes n'ont pas abouti.

Cette méthode *TTGV_byfusion*, bien que simple et faiblement supervisée, semble ici atteindre un plafond de performance, avec les optimisations auxquelles nous avons pu penser.

2.4 Description de la méthode TTGV_ollama_multilabel

Dans la continuité de l'approche développée l'année dernière (Blivet *et al.*, 2023), nous avons mis en place une classification multi-étiquette avec, cette fois-ci, l'intégration d'assertions et l'usage d'un modèle de langage causal (CLM) pour l'espace latent. Nous avons ainsi été motivés par plusieurs hypothèses. Premièrement, **(hyp1)** nous estimons que l'affinage par classification d'un CLM pré-entraîné de taille réduite suffit pour obtenir de bons résultats pour une classification multiétiquette. Deuxièmement, **(hyp2)** nous voulons vérifier que la modification du contexte par le biais de l'amorce pour la tâche de classification améliorera la prédiction. Notamment en vérifiant la prise en compte des assertions (voir section 2.1.3) et de la négation ainsi que leur rôle le QCM.

2. pas de ponctuation, d'espaces seuls, de mots outils, ni de mots de 2 lettres ou moins
3. déterminé empiriquement depuis le jeu d'entraînement

Concrètement, nous avons utilisé Open Llama v2 (Geng & Liu, 2023), un CLM aux données d'entraînement connues ne dépassant pas 3 milliards de paramètres. Ce modèle possède cependant l'inconvénient d'être issu de la première version de Llama et souffre donc de la comparaison avec les dernières versions de celui-ci. Nous avons appliqué un affinage par réduction de l'espace latent à l'aide du LoRA (Hu *et al.*, 2021) dans une approche non quantifiée, l'approche *quantized* n'étant pas nécessaire compte tenu de la taille du modèle de langage. Pour affiner le modèle par un LoRA, nous avons appliqué la mise à jour sur les requêtes (*query*) et valeurs (*values*) avec un rang à 64 et un alpha à 16. D'autres configurations du rang et de l'alpha du LoRA ont été essayées (voir section 3) suivant ainsi plusieurs recommandations existantes (Hu *et al.*, 2021; Dettmers *et al.*, 2024). Le modèle utilisé consiste en une simple adjonction d'un classifieur multiétiquette à partir du dernier espace latent h_{t-1} du CLM en fin d'amorce comme ceci : $\mathcal{L} = BCE(\sigma(h_{t-1}))$. Le seuil choisi pour le système soumis est de 0.15, tandis que l'entraînement s'est fait sur 6 epochs par arrêt automatique en cas de non progression, soit ~40 minutes sur un GPU Nvidia 6000 (48 Go de RAM).

Contrairement à la première approche proposée (Section 2.2), nous n'avons pas exploité *NACHOS* pour cette approche. Afin de définir les hyperparamètres à utiliser, nous avons opté pour une approche empirique sur plusieurs valeurs pour le seuil (0.15, 0.3, 0.4, et 0.5). Les valeurs 0.5 et 0.4 se sont montrées très limitées puisqu'elles induisent au modèle une prédiction unique pour chaque question, transformant alors la tâche en multiclasse. Nous avons donc mis un seuil à 0.15 compte tenu de la timidité du modèle à prédire davantage de classes. Nous avons essayé des amorces avec et sans intégration du pré-traitement évoqué en section 2.1, donnant lieu à trois approches : (1) Trouver les bonne amorces pour chaque réponse possible à l'aide des mots clés extraits, puis prendre les assertions candidates ; (2) Trouver la bonne amorce pour chaque réponse possible à l'aide des mots clés extraits, puis prendre seulement les assertions inférieures à n mot, en considérant n = 5, 10, ou 15 afin de pallier la taille réduite du contexte de OpenLlama ; (3) Pas d'assertion. Pour sélectionner les assertions, nous avons utilisé les mots clés extraits précédemment. Dans tous les cas, nous avons intégré la valeur booléenne d'indication de négation dans la question. Enfin, nous avons également pris en compte le positionnement de ces assertions en tout début ou au milieu de l'amorce.

3 Résultats obtenus avec les différentes méthodes

Les résultats de l'approche multiétiquette sont loin d'être satisfaisants, comme nous pouvons le voir en Table 4. Nous pouvons également constater que, sur les données de validation, le rang utilisé pour le LoRA entraîne une légère amélioration de l'EMR. Dans les faits, un alpha à 16 avec un rang à 4 donne 28,9 en hamming score et 1,90 en EMR, contre 28,5 en hamming et 2.88 en EMR avec un rang à 64. Cela indique sans doute par là la nécessité d'appliquer un affinage plus catégorique du modèle pré-entraîné. Cela peut s'expliquer par la taille relativement petite du modèle, requérant moins de précaution quant à l'oubli catastrophique, l'un des effets du LoRA. Toujours dans cette optique, nous n'avons observé aucune amélioration significative en changeant la langue de l'amorce vers du français. C'est pourquoi nous avons décidé de conserver une amorce majoritairement en anglais en suivant l'hypothèse selon laquelle la taille du modèle implique des instructions de préférence en anglais, langue majoritaire des données de pré-entraînement.

Hypothèse 1 invalidée. Compte tenu des résultats obtenus aussi bien en validation qu'en test, l'approche multiétiquette invalide la première hypothèse (**hyp1**) pour l'affinage d'un modèle pré-entraîné, tandis que pré-entraînement sur NACHOS d'un modèle à l'architecture adaptée (Section 2.2)

démontre un plus grand potentiel avec moins de paramètres. Malgré tout, les résultats restent dans l'ensemble bas et invalident donc cette hypothèse.

Hypothèse 2 à confirmer. L'intégration dans l'amorce des assertions à l'aide des mots clés extraits semble rendre le modèle plus confus. La limite de cette approche semble tenir en deux facteurs : la taille réduite des paramètres et la fenêtre de contexte relativement petite compte tenu des modèles plus récents (mais dépassant la limite de 3 milliards de paramètres). Il convient donc de vérifier cette hypothèse sur les derniers modèles, tout en vérifiant l'impact de la position des informations additionnelles comme indiqué en section 2.4.

Nom de l'équipe	Hamming	EMR	Nom du système
SPQRR	38.07	2.94	Nachos_LLM
SPQRR	26.97	4.19	TTGV_byfusion
SPQRR	28.75	1.68	TTGV_ollama_multilabel

TABLE 4 – Résultats obtenus par l'équipe sur le jeu de test de DEFT 2024.

3.1 Résultats statistiques

Les prédictions des trois modèles proposés ont été comparées statistiquement aux réponses attendues afin de confronter les tendances. Nous proposons également de comparer les résultats des modèles d'un point de vue uniquement statistique afin de voir si des tendances proches de la référence peuvent être identifiées. Nous avons sélectionné le nombre de réponses prédites en moyenne par question, le nombres de réponses multiples et les taux de réponses comprenant les propositions a), b), c), d) et e).

Valeurs attendues	Nachos_LLM	BY_Fusion	Ollama
Nombre de réponses prédites (moyenne : *2,8*)	3,99	3,3	2,29
Réponses multiples (*384*)	319	141	311
Taux de réponses a) (*53,66%*)	60,37%	36,05%	89,09%
Taux de réponses b) (*50,52%*)	62,89%	34,59%	51,36%
Taux de réponses c) (*51,15%*)	59,33%	29,77%	21,59%
Taux de réponses d) (*45,91%*)	58,70%	34,38%	14,26%
Taux de réponses e) (*51,15%*)	61,01%	33,33%	7,76%

TABLE 5 – Comparaison des prédictions des différents modèles par rapport à la référence - (les données les plus proches de la référence sont soulignées) OU Valeurs (tendances) statistiques observées pour les trois modèles.

4 Discussion et perspectives d'amélioration

Analyse qualitative et quantitative des résultats. L'approche Nachos_LLM prédit 75% de réponses partiellement correctes, c'est-à-dire contenant au moins l'une des réponses correctes. 33% des réponses prédites par le modèle contiennent l'ensemble des cinq réponses, malgré le fait qu'il ne

s'agisse pas de la réponse correcte. Concernant les bonnes réponses complètes prédites par le modèle (17 cas), la répartition entre les questions laissant le choix d'une ou de plusieurs réponses et les questions contraignant une seule réponse vraie est proche (8/17). Quant au type de réponses attendues, la répartition est également similaire avec 9 cas où les réponses possibles sont des phrases complètes, et 8 cas où il s'agit de syntagmes nominaux.

Quant à l'approche TTGV_byfusion, 80% des réponses prédites sont partiellement correctes et contiennent au moins l'une des réponses correctes. 7% des prédictions du modèle contiennent l'ensemble des 5 réponses alors qu'il ne s'agit pas de la réponse correcte. Concernant les prédictions correctes du modèle (12 cas), 1/3 des prédictions concernent des questions proposant une ou plusieurs réponses possibles, et 2/3 restreignant à une seule réponse. Côté réponses, il s'agit de phrases complètes dans 2/3 des cas et de syntagmes nominaux dans 1/3 des cas.

Enfin, l'approche TTGV_ollama_multilabel totalise 72% de réponses partiellement correctes et aucun cas où toutes les réponses a, b, c, d et e sont prédites. Les 8 prédictions correctes du modèle sont réparties comme suit : 75% des prédictions concernent des questions proposant une ou plusieurs réponses possibles, et les 25% restants sont des prédictions restreignant à plusieurs réponses uniquement. Les réponses sont quant à elles réparties à égalité entre syntagmes nominaux seuls et phrases complètes.

Limites de l'approche multi-étiquette. Au cours de nos expériences, nous avons identifié plusieurs limites à l'utilisation d'OpenLlama en tant que représentation latente pour une classification multiétiquette. Premièrement, la taille de contexte réduite à 3 200 comparée à la norme actuelle des LLMs de 4 096, voire 8 000, explique en partie seulement les obstacles à l'intégration des assertions dans l'amorce. Deuxièmement, les assertions impliquent un comportement étrange du modèle, trop d'informations répétées amènent le modèle à ne prédire souvent qu'une seule et même classe. De plus, la position dans l'amorce de ces informations ajoutées se révèle tout particulièrement importante pour un usage en classification du modèle (Mao *et al.*, 2023), quelles soient en début d'amorce ou adjointes aux réponses possibles. La troisième limite est due à la taille limitée du modèle à 3 milliards de paramètres constituant la particularité de la tâche DEFT de cette année, rendant l'intégration d'informations supplémentaires difficile. Enfin, c'est surtout la sélection du seuil pour déterminer si une étiquette est à prendre en compte ou non qui a entraîné davantage de difficultés. Par rapport à l'approche multiétiquette appliquée l'année dernière (Blivet *et al.*, 2023), le modèle y est ici beaucoup trop sensible au seuil, n'atteignant que rarement un équilibre entre choix multiples et singuliers. En travaux futurs, il nous faudrait changer l'approche et remplacer la fonction sigmoïdale en sortie des logits pour appliquer une approche binaire répétée et, ainsi, éventuellement permettre au modèle de proposer des performances plus satisfaisantes.

Références

BEZANÇON J., BOUBEHZIZ T., CHUTAUX C., ZINE O., ACENSIO L., KOUDORO-PARFAIT C., BRIGLIA A. & LEJEUNE G. (2023). "spqr@deft2023 : Résolution automatique de qcm médicaux à partir de corpus de domaine et de mesures de similarité". In *Actes de CORIA-TALN 2023*.

BLIVET A., DEGRUTÈRE S., GENDRON B., RENAULT A., SIOUFFI C., BOUJU V. G., CERISARA C., FLAMEIN H., GUIBON G., LABEAU M. *et al.* (2023). Participation de l'équipe ttgv à deft 2023˜ : Réponse automatique à des qcm issus d'examens en pharmacie. In *Actes de CORIA-TALN 2023. Actes du Défi Fouille de Textes@ TALN2023*, p. 23–38.

DETTMERS T., PAGNONI A., HOLTZMAN A. & ZETTLEMOYER L. (2024). Qlora : Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, **36**.

GENG X. & LIU H. (2023). Openllama : An open reproduction of llama.

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*.

KOLOMIYETS O. & MOENS M.-F. (2011). A survey on question answering technology from an information retrieval perspective. *Information Sciences*, **181**(24), 5412–5434. DOI : 10.1016/j.ins.2011.07.047.

LABRAK Y., BAZOGE A., DAILLE B., DUFOUR R., MORIN E. & ROUVIER M. (2023a). Tâches et systèmes de détection automatique des réponses correctes dans des qcms liés au domaine médical : Présentation de la campagne deft 2023. In *Actes de CORIA-TALN 2023*, Paris, France : Association pour le Traitement Automatique des Langues (ATALA).

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. (2023b). Drbert : A robust pre-trained model in french for biomedical and clinical domains. *bioRxiv*, p. 2023–04.

MAO J., MIDDLETON S. E. & NIRANJAN M. (2023). Do prompt positions really matter ?

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*.

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

ZHUANG Y., YU Y., WANG K., SUN H. & ZHANG C. (2023). Toolqa : A dataset for llm question answering with external tools. In A. OH, T. NAUMANN, A. GLOBERSON, K. SAENKO, M. HARDT & S. LEVINE, Éds., *Advances in Neural Information Processing Systems*, volume 36, p. 50117–50143 : Curran Associates, Inc.