Notation automatique de réponses courtes d'étudiants : présentation de la campagne DEFT 2022

Cyril Grouin Gabriel Illouz
Université Paris-Saclay, CNRS, LISN, 91400 Orsay, France
{prenom.nom}@lisn.upsaclay.fr

RÉSUMÉ

La correction de copies d'étudiants est une tâche coûteuse en temps pour l'enseignant. Nous proposons deux tâches d'attribution automatique de notes à des réponses courtes d'étudiants : une tâche classique d'entraînement de système et d'application sur le corpus de test, et une tâche d'amélioration continue du système avec interrogation d'un serveur d'évaluation. Les corpus se composent de réponses courtes d'étudiants à des questions en programmation web et bases de données, et sont anonymes. Quatre équipes ont participé à la première tâche. Les meilleures précisions de chaque équipe varient de 0,440 à 0,756 pour une précision moyenne de 0,542 et une médiane de 0,524. En raison de la complexité de la deuxième tâche, une seule équipe a participé, mais les résultats soumis ne sont pas exploitables.

ABSTRACT

Automatic grading of students' short answers : presentation of the DEFT 2022 challenge.

Evaluating student short answers is a time-consuming task for the teacher. In this challenge, we propose two tasks for automatically assigning grades to short student answers : a classic system training and application task on the test corpus, and a continuous system improvement task with questioning of an evaluation server. The corpora consist of short answers from students to questions in web and database programming, and are anonymous. Four teams participated in the first task. Each team's best accuracy ranges from 0.440 to 0.756 with an average accuracy of 0.542 and a median of 0.524. Due to the complexity of the second task, only one team participated, but the submitted results can not be used in our evaluation process.

MOTS-CLÉS : Correction automatique, réponses courtes d'étudiants, campagne d'évaluation.

KEYWORDS: Automatic grading, student short answer, challenge.

1 Introduction

La correction de copies d'étudiants est une tâche chronophage, quel que soit le niveau d'expérience de l'enseignant. Si les questionnaires à choix mutiples (QCM) permettent une correction automatique, les questions appelant des réponses en langue naturelle nécessitent encore une évaluation humaine. Chaque enseignant élabore des stratégies de correction pour accélérer la correction ou pour s'assurer d'un traitement équitable entre élèves, par exemple en corrigeant les copies question par question. Pourtant, cette stratégie se révèle également coûteuse en temps dans la mesure où des réponses identiques, appelant la même note, seront quand même évaluées les unes après les autres. Dans la suite de l'édition 2021 du défi fouille de texte (Grouin *et al.*, 2021), nous proposons deux tâches autour de l'attribution automatique de notes à des réponses courtes d'étudiants sur des questions de

programmation web. L'objectif vise à élaborer des méthodes automatiques ou semi-automatiques d'attribution de notes à des réponses d'étudiants, en vue d'aider le travail de correction des enseignants. Une nouveauté a été introduite avec une tâche expérimentale qualifiée de tâche « continue » (voir section 3.2) qui permet aux participants d'alterner entre interrogation du serveur d'évaluation pour obtenir la note d'un étudiant à une question et dépôt d'une soumission sur le serveur de manière itérative, dans l'objectif d'interroger le serveur de manière pertinente pour constituer un modèle le plus efficace possible. L'objectif d'un enseignant peut être de 80% de réponses bien corrigées. Pour atteindre cet objectif, la question que se pose un enseignant est de savoir combien de réponses doivent-être corrigées.

La campagne a été lancée le 24 février. 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 deux mois, du 1er mars au 30 avril. La phase de test a été scindée en deux avec une période différente par tâche (du 2 au 3 mai pour la tâche de base, du 4 au 9 mai pour la tâche continue). Contrairement aux éditions précédentes, les participants n'ont pas eu le choix des dates pour la phase de test. Quatre équipes se sont inscrites, et sont allées jusqu'au terme de la campagne, dont deux équipes académiques (une équipe jointe LIA-LS2N et une équipe Sorbonne Université sous le nom « STyLO » — issue de l'équipe « Queer » de l'année précédente —, et deux équipes industrielles (EDF R&D et l'équipe SNCF R&D sous le nom « TGV »).

2 Corpus

Les données proviennent des cours de « programmation web » et « base de données », récoltées sur plusieurs années. Bien que l'objectif final soit similaire sur les deux tâches, les corpus de questions sont spécifiques à chaque tâche. Ils se composent d'une liste de questions, et pour chacune des questions, d'une liste des réponses correspondantes. Les corpus d'entraînement de cette année se composent des corpus d'entraînement et de test de DEFT 2021. En revanche, les corpus de test sont inédits. Chaque corpus a été anonymisé, les identités des étudiants étant remplacées par un identifiant unique (jusqu'à 122 étudiants différents). En fonction de la provenance des données (deux cours avec des promotions différentes par année), le nombre de réponses par question peut varier. En cas d'absence de réponse de la part d'un étudiant, la mention « NO_ANS » est renseignée comme réponse. En revanche, aucune correction orthographique n'est appliquée sur les réponses d'étudiants.

Le tableau 1 présente deux exemples de questions issues du corpus d'entraînement. La question 2032 appelle une réponse en langue naturelle tandis que la question 2034 attend du code informatique. La correction de l'enseignant peut s'accompagner de commentaires pour faciliter l'attribution des notes. Les données provenant de l'interface Moodle, des balises de mise en forme restent présentes (<p> et <br>) tandis que les balises de code informatique sont représentées par les entités HTML correspondantes (< et > pour les chevrons ouvrant et fermant).

Le code informatique de la question 2034 doit se lire : <code> <ue id=PW2> </code> tandis que la réponse de l'enseignant sera lue : <ue id="PW2"> suivie d'un commentaire sur les notes à attribuer.

Le tableau 2 fournit les réponses de cinq étudiants aux questions 2032 et 2034 du tableau 1 ainsi que les notes de référence fournies par l'enseignant. Les notes sont normalisées entre 0 (absence de réponse ou réponse incorrecte) et 1 (bonne réponse), avec au maximum deux décimales en cas de réponse partiellement correcte.

Question	Note	Id	Intitulé	Correction et commentaire enseignant
2032	1	32	Quel est l'intérêt d'utiliser du code AJAX ?	 Permet l'échange de données avec le serveur sans mise à jour complète de la page Ok pour permet de màj une partie de la page sans avoir à la recharger complètement.
2034	1	34	Modifiez le code XML ci-dessous pour le rendre valide : <code> <ue id=PW2> </code> ?	 <ue id="PW2"> 1 si guillemets ajoutés 0 sinon 0.5 si transformé id en sous-élément (mais pas si supprimé la notion d'id)

TABLE 1 – Exemples de questions avec correction et commentaire de l'enseignant

Question	Note	Identifiant	Réponse étudiant
2032	1	student106	Il permet de mettre à jour dynamiquement la page sans avoir à recharger la page entière.
2032	0	student107	NO_ANS
2032	0.5	student108	AJAX permet de modifier en temps réel une page, sans avoir à faire appel au serveur. Par exemple, on peut changer le contour d'un bouton lorsque la souris passe dessus.
2032	0.2	student109	Cela permet d'appeler des scripts dans la page web
2032	1	student12	Le code AJAX permet d'actualiser une partie d'une page web sans avoir à recharger toute la page.
2034	1	student106	<code>\n <ue id="PW2">\n </code>
2034	1	student107	<code>\n <ue id="PW2"></ue>\n </code>
2034	0	student108	<code>\n <ue id=PW2> </ue>\n </code>
2034	0	student109	<code>\n <ue id=PW2></ue>\n </code>
2034	0.5	student12	<code>\n <ue>\n <id>PW2</id>\n </ue>\n </code>

TABLE 2 – Exemples de réponses d'étudiants aux questions 2032 et 2034 et notes de l'enseignant

Plusieurs difficultés apparaissent à la lecture de ces exemples :

- si les notes 0 ou 1 sont conformes aux commentaires laissés en complément de la réponse, un travail d'interprétation est nécessaire pour les notes intermédiaires. Cependant, la majorité des notes attribuées sur le corpus d'entraînement (77,1%) concerne les notes 0 et 1, tandis que les notes intermédiaires sont minoritaires avec une sur-représentation de la note 0,5 (11,8%).
- sur les questions qui attendent une réponse en langue naturelle (question 2032), la mention « NO_ANS » suffit pour attribuer la note 0. Sur les questions de code informatique (question 2034), c'est parce que le code est reproduit à l'identique que les étudiants 108 et 109 obtiennent la note 0. Une note nulle ne dépend donc pas uniquement d'une absence de réponse.

Par ailleurs, puisque le travail d'évaluation a été réalisé par des humains, des erreurs restent possibles (fatigue, mauvaise compréhension de la réponse d'un étudiant, etc.).

3 Tâches

Deux tâches sont proposées autour de la notation automatique de réponses d'étudiants : la première tâche consiste classiquement à prédire des notes à partir d'une référence (tâche de base, section 3.1) tandis que la seconde tâche est dynamique et repose sur une interrogation continue du serveur d'évaluation pour produire un modèle efficace et en affinant les prédictions (tâche continue, section 3.2). Alors que les systèmes élaborés sur la tâche de base permettent d'attribuer automatiquement des notes, les systèmes de la tâche continue ont vocation à aider l'enseignant en lui proposant une organisation des corrections, avec pour objectif, soit de minimiser le temps passé en correction, soit de proposer une correction partielle automatique jusqu'à atteindre un niveau d'erreurs non corrigées acceptable et que l'enseignant aurait à vérifier pour attribuer la bonne note correspondante.

Pour la phase de test, si les jeux de questions/réponses sont distincts pour les deux tâches, nous avons imposé l'ordre des tâches — d'abord la tâche de base puis la tâche continue — pour éviter que le corpus de test de la tâche continue (récupérable avec les notes de référence par interrogation du serveur, voir section 3.2) ne soit utilisé pour enrichir le corpus d'apprentissage de la tâche de base.

3.1 Tâche de base

Présentation Cette tâche est qualifiée de « base » dans la mesure où elle revient à produire un système ou à entraîner un modèle statistique sur les données d'entraînement, puis à appliquer ce système ou ce modèle sur les données de test pour prédire les notes de chaque réponse d'étudiants. Le corpus d'entraînement se compose des questions et réponses notées par l'enseignant (cf. tableaux 1 et 2, provenant des corpus d'entraînement et de test des deux tâches de DEFT 2021, soit un total de 88 questions et 6620 réponses d'étudiants), tandis que le corpus de test se compose des questions et réponses non notées (nouveau corpus de 24 questions et 2640 réponses d'étudiants). Les participants ont été autorisés à soumettre jusqu'à trois sorties de système pour évaluer les performances de leur système sous différentes configurations (voir section 4.1).

Évaluation Les résultats produits sont évalués au moyen d'une précision (classement officiel) et d'une corrélation de Pearson (formule 1) comme utilisée dans des travaux proches en anglais (Mohler & Mihalcea, 2009; Dzikovska *et al.*, 2013; Burrows *et al.*, 2015; Mizumoto *et al.*, 2019), avec Cov(X,Y) la covariance des variables X et Y, et σX et σY les écarts-types de ces variables.

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

Baseline Nous reprenons la baseline développée en 2021. Ce système décompte les mots communs (de plus de quatre caractères, mis en minuscules) entre la réponse de l'étudiant et la question/réponse de l'enseignant. Ce décompte est rapporté au nombre de mots conservés dans la question et la réponse de l'enseignant pour produire un score normalisé avec une valeur de 1 si le score est supérieur ou égal à 0,5, une valeur de 0,5 si supérieur ou égal à 0,4 et la conservation des autres valeurs.

3.2 Tâche continue

L'objectif poursuivi dans cette tâche expérimentale consiste à mettre en place des systèmes ou des modèles intelligents en s'appuyant sur des exemples représentatifs des réponses produites par les étudiants, de manière à limiter le nombre de réponses à corriger. Des réponses identiques, voire similaires, devraient recevoir automatiquement la même note. Le corpus d'entraînement reprend les 50 questions (3820 réponses) du corpus d'entraînement DEFT 2021 (tâche 2) également utilisé sur la tâche de base, alors que le corpus de test est nouveau et différent de celui de la tâche de base (25 questions, 2750 réponses, 110 étudiants).

Exemple À la question 2017 (*« Le code PHP est-il exécuté sur la machine cliente ou sur le serveur web ? »*), quarante-trois étudiants sur cent seize ont formulé exactement [1] la même réponse (*« le code PHP est exécuté sur le serveur web »*), vingt ont effectué une réponse plus courte (*« sur le/un serveur web »*), et douze ont répondu avec une reprise pronominale (*« il est exécuté sur le serveur web »*), soit 64,7% de la promotion qui aura répondu correctement avec des réponses très proches ; inversement, six étudiants ont répondu de manière erronée (*« le code PHP est exécuté sur la machine cliente »*), soit 5,2% de l'effectif, et trois n'ont pas répondu. Les autres étudiants ont correctement répondu, avec soit des variantes dans leurs réponses (oubli du terme « web » pour qualifier le serveur), soit des précisions que l'enseignant n'a pas pénalisées (*« ... c'est le javascript qui est exécuté sur la machine cliente », « ... et non sur la machine cliente », « ... le client ne reçoit que le résultat du script »,* etc.). L'outil idéal pour l'enseignant serait celui qui regroupe les réponses d'étudiants similaires et propose à la correction une seule réponse représentative de chaque groupe, puis qui attribue la même note à tous les autres étudiants du groupe de réponses (voir figure 1) comme suggéré par Basu *et al.* (2013). Sur la question 2017, en corrigeant les réponses « serveur web » et « machine cliente », l'enseignant aurait, avec seulement deux copies, déjà corrigé un peu plus des deux tiers de la promotion (81 réponses d'étudiants).

Déroulement Dans cette optique, nous avons déployé un serveur d'évaluation sécurisé (accès protégé et nombre limité d'opérations sur la base de données) permettant aux participants :

- de demander la note de référence d'un étudiant à une question
- de soumettre un fichier de prédictions de notes pour les étudiants à cette question
- puis de recommencer avec pertinence sur la même question ou sur une nouvelle question [2]

Bien qu'il soit possible d'accéder à toutes les notes de référence, l'objectif reste une interrogation raisonnée du serveur pour construire un modèle efficace. Dans cette perspective, nous interdisons la demande d'une nouvelle note de référence tant que la soumission de prédictions n'a pas été faite [3]. En revanche, un système peut interroger le serveur de manière itérative pour obtenir toutes les notes de référence. Dans ce cas, l'évaluation mettra en évidence une progression linéaire des réponses correctes, et pénalisera ce type de soumissions.

1. Modulo les fautes d'orthographe et de frappe telles que : *excuté, exécutée, exécuter, éxécuté, exucuté, serveru,* etc.
2. Sur la question 2017, une stratégie consisterait à demander la note d'un étudiant ayant répondu « *le code PHP est exécuté sur le serveur web* », à attribuer la note renvoyée par le serveur (1) à tous les étudiants ayant répondu de manière similaire et une note fictive aux autres étudiants, puis à demander la note d'un étudiant ayant répondu « *le code PHP est exécuté sur la machine cliente* » et attribuer la note retournée (0) pour toutes les réponses similaires, et de poursuivre l'interrogation pour les réponses dont la note n'a pas encore été demandée, ou d'appliquer une notation automatique des réponses restantes.
3. Cette contrainte vise à éviter la récupération de la référence, ce qui reviendrait à travailler sur la tâche de base.

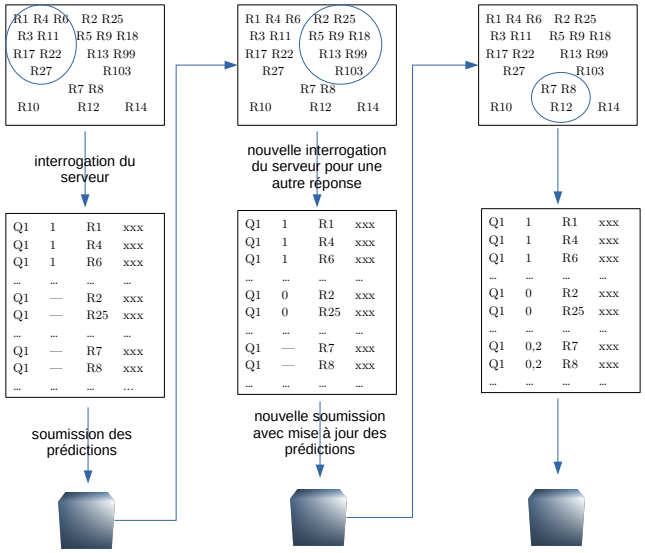


FIGURE 1 – Système idéal : pour une question, après avoir regroupé les réponses similaires, le système interroge le serveur pour obtenir la note d'une réponse, affecte cette note aux autres réponses du groupe, soumet les prédictions, puis effectue une nouvelle itération sur un autre groupe de réponses

Nous avons fourni aux participants trois scripts Python [4] permettant d'interroger le serveur, d'envoyer les notes et d'y déposer la soumission. Pour la phase d'entraînement uniquement, un quatrième script permet de vider la base de données en vue d'évaluer une nouvelle stratégie de récupération des notes et de mise à jour du système de prédiction. Pour la phase de test, l'ensemble des étapes précédemment décrites est proposé, à l'exclusion de la suppression du contenu de la base [5] : une seule stratégie d'interrogation du serveur est donc possible sur les données du test.

Motivations Contrairement à la tâche de base où nous évaluons un ensemble de prédictions sans connaître la stratégie de traitement des questions (vision « à plat »), la tâche continue devait nous permettre de comprendre la stratégie appliquée par les participants :

- dans quel ordre les questions ont-elles été abordées ? Y a t-il eu une étape de regroupement des questions par type de réponse attendu (langue naturelle, code informatique, réponse courte) ? Ou bien les demandes ont-elles été faites par groupes de réponses en prioritisant les grands ensembles de réponses similaires pour corriger un maximum de copies d'un coup ?
- combien de notes ont été demandées par question ? Le nombre demandé est-il sensiblement le même pour chaque question ? Les notes ont-elles été demandées pour les réponses isolées ou bien une stratégie de notation automatique a t-elle été appliquée dans ces cas ?
- observe t-on une évolution rapide des performances du système ? Est-il possible d'identifier un système permettant de minimiser les corrections à effectuer ?

4. https://deft.lisn.upsaclay.fr/2022/guide-deft2022-v2.pdf ; à charge aux participants d'adapter ces scripts pour leur chaîne de traitements.

5. Offrir aux participants la possibilité de vider la base revient à permettre de récupérer les notes de référence sur le corpus de test de manière itérative, puis de faire une soumission avec un maximum de « bonnes prédictions » après avoir vidé la base.

4 Résultats

Quatre équipes ont participé à la tâche de base (section 4.1) et une seule équipe a participé à la tâche continue (section 4.2).

4.1 Tâche de base

Les quatre participants ont chacun soumis trois fichiers de prédictions. Le tableau 3 présente les résultats (précision pour l'ensemble et moyenne par question des corrélations de Pearson) et le classement pour la tâche de base (RP=rang précision, classement officiel, RC=rang corrélation). Nous intégrons également dans ce tableau les résultats de notre baseline (section 3.1) et ceux d'un tirage aléatoire. Sur l'ensemble des soumissions effectuées par les participants et officiellement prises en compte pour l'évaluation, la précision moyenne est de 0,542 et la médiane de 0,524.

Equipe	Run	Précision	RP	Corrélation	RC
EDF Lab (Suignard *et al.*, 2022)	1	0,752	–	0,70	–
	2	0,756	1	0,70	–
	3	0,323	–	0,76	1
LIA-LS2N (Labrak *et al.*, 2022)	1	0,440	–	0,00	–
	2	0,404	–	0,00	–
	3	0,440	4	0,00	4
STyLO (Ben Ltaifa *et al.*, 2022)	1	0,512	–	0,63	2
	2	0,580	–	0,56	–
	3	0,641	2	0,51	–
TGV (Gaudray-Bouju *et al.*, 2022)	1	0,491	–	0,17	–
	2	0,536	–	0,54	3
	3	0,624	3	0,42	–
Baseline	–	0,522	–	0,37	–
Tirage aléatoire	–	0,380	–	0,47	–
LIA-LS2N, *hors compétition*	1	0,606	–	0,39	–
	2	0,726	–	0,69	–
	3	0,649	–	0,48	–

TABLE 3 – Résultats et classement des équipes participantes à la tâche de base (RP=rang précision, classement officiel, RC=rang corrélation). Moyenne = 0,542, médiane = 0,524 (sur les soumissions officielles)

L'équipe LIA–LS2N s'est rendu compte d'un problème technique dans son système à l'occasion de la communication des résultats sur le phase de test. Après correction, les valeurs de précision des nouveaux résultats produits augmentent, comme renseignées au bas du tableau 3. Nous précisons que les valeurs de précisions moyenne et médiane correspondent à celles calculées à partir des soumissions officielles et non des soumissions corrigées de cette équipe.

Méthodes des participants Les participants ont utilisés des techniques variées pour cette tâche de prédiction des notes, en s'appuyant sur des comparaisons du contenu de la question, de la réponse de l'étudiant, et de la réponse attendue. Sur cette tâche, les techniques les plus simples se sont nettement démarquées des méthodes à base de plongements de mots.

Ainsi, Suignard *et al.* (2022) de l'équipe vainqueur (EDF R&D) ont utilisé des forêts d'arbres aléatoires pour classer les réponses parmi les trois valeurs de notes les plus répandues dans le corpus (0 0,5 et 1) ; les différentes soumissions reposent sur un nombre différent d'arbres (cent pour la première soumission, deux cents pour la deuxième) ; la troisième soumission repose sur cent arbres et une échelle à dix valeurs comprises entre 0,0 et 1,0 mais les résultats obtenus ont chuté.

Les plongements de mots du modèle CamemBERT (Martin *et al.*, 2020) ont été utilisés par plusieurs participants, avec un travail de comparaison de plusieurs calculs de similarité et d'affinage (*fine-tuning*) du modèle sur le corpus comme réalisé par Labrak *et al.* (2022) de l'équipe LIA–LS2N, en complément d'une comparaison des représentations du texte (mots, n-grammes de caractères, *word pieces, sentence embeddings*) comme fait par Ben Ltaifa *et al.* (2022) pour l'équipe STyLO.

Une première étape d'identification du type de la question au moyen d'un arbre de décision, entraîné sur une annotation manuelle du corpus, a été appliquée par Gaudray-Bouju *et al.* (2022) de l'équipe TGV pour déterminer le type de *features* à extraire (CamemBERT-NER, étiquettes morpho-syntaxiques avec Flair, pourcentage de code dans la réponse, etc.), suivi de l'utilisation d'hyperparamètre *gridsearch* pour les algorithmes de prédiction. L'identifiant numérique de l'étudiant a également été utilisé en complément des autres caractéristiques pour réaliser les prédictions. D'autres approches à base de tf-idf notamment ont également été tentées par cette équipe.

Nous retenons que les plongements obtiennent de bons résultats sur cette tâche, mais la taille limitée du corpus (bien que déjà importante) a permis aux méthodes plus simples d'obtenir de meilleurs résultats en considérant la tâche comme un problème de classification parmi trois valeurs seulement (0–0,5–1). Les approches à base de plongements auront cependant obtenu de meilleurs résultats si l'on considère que l'ensemble des valeurs de notes comprises entre 0,0 et 1,0 peut se rencontrer lors de l'évaluation, ce qui était le cas d'après les commentaires de l'enseignant sur certaines questions.

4.2 Tâche continue

Sur la tâche continue, nous n'avons reçu qu'une seule soumission. En raison d'une incompréhension sur le fonctionnement de la tâche, que nous avons probablement mal expliquée, les éléments soumis sur le serveur ne sont pas significatifs au regard des objectifs que nous avions fixés sur cette tâche.

En effet, le participant a intégré dans sa stratégie de prédiction le quatrième script qui autorise la suppression du contenu de la base de données. Parce que ce script a été volontairement rendu inopérant pour la phase de test (voir section 3.2), le participant a été contraint de revoir son système pendant cette phase. Une deuxième incompréhension concerne le nombre d'interrogations autorisées. Le participant a compris qu'il n'était possible d'accomplir qu'une seule demande de note sur l'ensemble du corpus et n'a donc soumis que le premier résultat de son système, alors que notre limite d'une soumission concernait l'ensemble du parcours du corpus de test. Il était donc possible de demander autant de notes que nécessaire par question, mais à l'issue du traitement du corpus, il n'était pas possible de vider la base pour recommencer une nouvelle stratégie (seule limite imposée). Nous avons cependant évalué le fichier de prédictions généré par le système du participant comme pour le tâche de base et calculé une précision de 0,416 et une corrélation négative (-0,02).

5 Conclusion

L'édition 2022 du défi fouille de texte (DEFT) a de nouveau été consacrée à la notation automatique de réponses courtes d'étudiants, dans la suite de ce qui a été proposé pour DEFT 2021. Une tâche classique de prédiction des notes a été proposée et a reçu l'attention de quatre équipes qui ont chacune soumise trois fichiers de prédictions. Les meilleures précisions de chaque équipe varient de 0,440 à 0,756, avec une précision moyenne de 0,542 et une médiane de 0,524. La seule différence par rapport à la tâche similaire proposée en 2021 concerne l'augmentation notable du nombre de questions et réponses fournies pour l'apprentissage cette année, avec 88 questions et 6620 réponses d'étudiants, contre 50 questions et 3820 réponses l'année précédente. Les corpus de test sont différents entre les deux éditions, mais restent similaires au niveau du contenu. Nous observons à la fois une amélioration globale des performances entre les deux éditions (les équipes ayant participé aux deux éditions ont vu leur précision sur leur meilleure soumission augmenter de 0,682 à 0,756 pour EDF R&D, et de 0,630 à 0,641 pour l'équipe QUEER en 2021 devenue en partie STyLO en 2022), ainsi que des écarts plus importants entre résultats des participants (les précisions variaient de 0,630 à 0,682 sur les meilleures soumissions de chaque équipe en 2021, et de 0,624 à 0,756 en 2022).

Si ce corpus a été utilisé pour de la correction automatique de réponses courtes d'étudiants, il peut également être employé en tant que corpus de reformulations validées comme étant équivalentes dans un contexte donné. Ceci pourrait constituer une autre piste de recherche pour les travaux de réécriture et de lisibilité.

Une deuxième tâche plus expérimentale consistait à interroger de manière pertinente un serveur d'évaluation pour obtenir les notes de référence, afin d'améliorer en continue son modèle de prédiction. En raison de la complexité technique pour intégrer l'interrogation raisonnée du serveur dans sa chaîne de traitements, malgré la mise à disposition de scripts, un seul participant a participé à cette tâche. Les résultats ne sont cependant pas exploitables et significatifs au regard de la définition de la tâche.

Références

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

BEN LTAIFA I., BOUBEHZIZ T., BRIGLIA A., CHUTAUX C., DUPONT Y., GONZÁLES-GALLARDO C.-E., KOUDORO-PARFAIT C. & LEJEUNE G. (2022). Stylo@DEFT2022 : Notation automatique de copies d'étudiant·e·s par combinaisons de méthodes de similarité. In *Actes de DEFT*, Avignon, France.

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

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

GAUDRAY-BOUJU V., GUETTIER M., LERUS G., GUIBON G., LABEAU M. & LEFEUVRE L. (2022). Participation de l'équipe TGV à DEFT 2022 : Prédiction automatique de notes d'étudiants à des questionnaires en fonction du type de question. In *Actes de DEFT*, Avignon, France.

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

LABRAK Y., TURCOTTE P., DUFOUR R. & ROUVIER M. (2022). Correction automatique d'examens écrits par approche neuronale profonde et attention croisée bidirectionnelle. In *Actes de DEFT*, Avignon, France.

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 *Proc of ACL*, p. 7203–7219.

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

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

SUIGNARD P., HUANG X. & BOTHUA M. (2022). Participation d'EDF R&D à DEFT 2022. In *Actes de DEFT*, Avignon, France.