Évaluation des réponses courtes à l'aide de la similarité de phrases

Michel C. Desmarais[1], Ovide Bertrand Kuichua Kamdem[1], and Arman Bakhtiari[1]

Polytechnique Montréal
{michel.desmarais, ovide.kuichua, arman.bakhtiari}@polymtl.ca

Résumé Cet article propose une approche à l'évaluation automatique des réponses courtes (Automatic Short Answer Grading, ASAG) basée sur l'utilisation d'encodeur de phrases pour le calcul de similarité des réponses et une fonction de régression non linéaire pour la prédiction basée sur quelques réponses évaluées. Le contexte d'utilisation est celui un correcteur qui évalue manuellement une quinzaine de réponses, tandis que le reste est noté automatiquement. Ce contexte est relativement fréquent et correspond aux situations où un certain nombre de réponses d'élèves doivent être examinées avant que le correcteur soit suffisamment à l'aise pour attribuer une note finale. Les résultats montrent que l'approche proposée atteint une précision de notation comparable à celle des modèles de transformeurs affinés (*fine-tuned)*, qui sont beaucoup plus gourmands en calcul et plus complexes à utiliser pour l'entraînement, ce qui les rend moins attrayants pour de nombreux contextes d'utilisation. L'approche proposée a l'avantage d'être particulièrement facile et efficace à déployer, tout en offrant une bonne précision.

Keywords : Short Answer Grading (SAG)

1 Introduction

Comme toute application informatique, la conception d'un outil d'aide à la correction doit tenir compte du contexte d'utilisation. Ainsi, un contexte typique de correction d'un examen est qu'il comporte quelques dizaines de copies. Le correcteur corrige initialement une partie des copies afin d'affiner les critères de correction et "calibrer" la sévérité. Cette étape est particulièrement importante si l'enseignant doit fournir une ligne directrice à d'autres correcteurs qui l'assistent. La majorité des questions ne sont pas tirées d'anciens examens et la tâche implique quelques dizaines de copies à noter.

On ne peut donc pas se reposer sur un corpus étendu de réponses antérieures.

Ce contexte constitue le cadre général pour l'approche de correction automatique des réponses courtes proposée dans ce texte, une tâche connue en anglais sous le nom de *Automatic Short Answer Grading, ASAG*. Contrairement à plusieurs approches populaires en ASAG, ce contexte ne se prête pas particulièrement bien à l'entraînement de modèles de langue, notamment à l'affinage fin (*fine-tuning*) qui nécessite un volume relativement important de données. Par exemple, avec les données que nous utilisons, Zhu et al. [23] affirme que "l'ensemble de données Mohler ne contient que 2273 paires de réponses, ce qui est trop peu pour les modèles d'apprentissage profond" et ils ont dû recourir une technique d'enrichissement des données pour amener ce jeu de données à 3300 réponses. De plus, l'affinage fin de LLM demeure relativement coûteux en temps de calcul. Ces exigences rendent l'approche d'affinage fin moins adaptée au contexte d'utilisation de notre étude.

Par contre, ce contexte se prête très bien à une approche de similarité de texte entre une réponse étudiante et la réponse de référence. Les grands modèles de langue (LLM) préentrainés sont rapides et relativement efficaces pour effectuer un tel calcul de similarité. Nous proposons une approche qui utilise un transformeur récent et un modèle de régression non linéaire simple pour calculer le score à chaque question. La régression est paramétrée à partir d'une quinzaine de réponses que le correcteur doit fournir au préalable. C'est donc une approche semi-automatisée que nous avançons, mais qui permet d'assister un enseignant dans la correction dans la mesure où les scores calculés sont fiables.

Les résultats obtenus avec un corpus de données très utilisé pour évaluer la performance de modèles pour ASAG, les données de Mohler [16], se comparent en termes de précision à ceux l'état de l'art obtenus avec des modèles de transformeurs affinés. L'approche proposée est particulièrement simple à développer et repose sur des calculs largement plus efficaces en termes temps que les modèles affinés.

2 Travaux portant sur la correction automatique

Les dernières années ont été particulièrement fertiles dans le domaine de l'ASAG grâce aux avancées des grands modèles de langage, les LLM qui reposent sur l'architecture de transformeur [21].

La très grande majorité des des travaux récents se reposent sur les modèles de langage basés sur l'architecture transformeurs. Ces approches adoptent souvent un modèle pré-entrainés pour la vectorisation de texte. En vectorisant le texte de la réponse attendue et de celle de l'étudiant, un calcul de la similarité sémantique fournit un indicateur de la validité de la réponse [7–9, 11]. Amur et al. [3] ont effectué une revue des nombreuses études qui ont adopté cette approche pour le calcul de la similarité de texte en général, dont les applications pour l'ASAG. L'approche proposée ici s'inscrit dans cette catégorie, avec de meilleurs résultats qui se comparent plutôt à ceux des approches qui adoptent l'affinement.

En effet, plusieurs études récentes utilisent une approche d'affinement (*fine-tuning*) des modèles transformeurs pour la classification [10, 12, 13, 23]. Gard et al. [12] utilisent des paires réponses, la réponse attendue et la réponse étudiante, pour l'affinement du transformeur et obtiennent un résultat qui dépasse les autres approches avec une erreur quadratique moyenne (EQM) de 0.732 pour les données de Mohler [16] que nous utilisons dans la présente étude. Zhu et al. [23] effectuent un affinement avec une petite partie des réponses étudiantes pour obtenir le meilleur score de corrélation Pearson avec les mêmes données, 0.897 [1].

Parmi les plus récents travaux, on retrouve ceux qui utilisent l'IA générative pour demander un score directement à l'IA. Les résultats de Chang et al. [6] et de Grévisse [14] démontrent une nette amélioration entre les versions précédentes des IA génératives comme ChatGPT, ce qui suggère que cette approche offre une avenue pro-

1. À l'instar de plusieurs autres études, nous omettons ici celle de [20] dont les résultats pour les données Mohler sont de 0.949 pour la corrélation et de 0.04 pour l'erreur quadratique. Bien qu'impressionnants, ces résultats ne semblent pas comparables. Car, comme la corrélation des scores entre les deux correcteurs de ce jeu de données est de 0.597 et qu'un calcul de l'EQM interjuge donnerait 1.7, un facteur de 42 (1.7/0.04), les performances de 0.949 et 0.04 sont peu vraisemblables et nous présumerons qu'une erreur s'est glissée dans l'article.

metteuse, mais les auteurs de ces deux études s'entendent sur la nécessité de maintenir une supervision de l'évaluateur. Tobbler et al. [19] présentent un outil basé sur l'IA générative conçu pour noter les réponses des étudiants en les comparant à des réponses de référence. D'autre part, nous n'avons pas identifié d'études permettant de comparer les performances de cette approche et celles alternatives. En particulier, il n'existe pas à l'heure actuelle de score de performance de l'IA générative pour les données de Mohler qui permettrait une comparaison. Néanmoins, il semble acquis que l'IA générative puisse fournir une rétroaction utile [2] et qu'elle soit aussi utilisable pour enrichir des approches alternatives [5, 17, 22].

Mentionnons finalement l'étude de Agarwal et al. [1] qui utilise une approche d'apprentissage profond avec graphes et obtient de très bons résultats d'erreur quadratique de 0.76 avec les données de Mohler. L'approche nécessite cependant la création d'une structure de graphe basée sur l'analyse sémantique des phrases (AMR, [4]) et une phase d'entraînement avec GPU qui est vraisemblablement peu adaptée au contexte d'utilisation de notre étude.

Ce survol des travaux portant sur la correction automatique démontre que des percées importantes ont été réalisées dans la dernière décennie, surtout grâce aux grands modèles de langage. Cependant, les meilleures performances sont obtenues à partir de modèles qui nécessitent une phase de préentrainement coûteuse et qui s'avèrent peu adaptés au contexte d'utilisation de notre étude.

3 Évaluation par similarité sémantique et lissage gaussien d'un échantillon de réponses

Comme plusieurs autres avant elle, l'approche proposée repose principalement sur le calcul de la similarité sémantique avec des plongements obtenus par des transformeurs, mais elle se distingue par la présupposition que le correcteur fournira au préalable quelques réponses scorées pour chacune des questions pour guider le processus de correction. Cette contrainte semble raisonnable du fait que cette étape permet de mieux jauger la difficulté de la question et d'établir ainsi un calibrage de la rigueur de la correction. Elle permet aussi de vérifier au préalable si la réponse anticipée est bien la seule bonne réponse avant de procéder à une correction.

L'approche proposée consiste donc à simplement calculer la similarité cosinus entre les plongements des réponses attendues-étudiantes et à utiliser les scores observés pour définir une fonction non linéaire du calcul de la note sur une échelle de [0–5]. Une fonction non linéaire s'avère utile ici comme on peut constater plus loin à la question 1.4 de la figure 2 où des scores de 5/5 sont observés à partir d'une similarité supérieure à 0.6, alors que sous ce seuil la variabilité est plus forte. Une fonction linéaire influencerait négativement les scores des réponses dont la similarité est supérieure à 0.6 alors qu'une fonction non linéaire sera moins sensible aux erreurs créées par ce type de distribution.

Pour définir cette fonction, un lissage gaussien (donc non linéaire) est calculé avec les données disponibles pour définir un ensemble de points de prédiction. Une interpolation linéaire est ensuite faite pour déterminer la valeur finale d'une mesure de similarité entre deux points. À noter qu'il serait aussi possible de définir une fonction de régression gaussienne, mais l'approximation par interpolation s'avère efficace et efficiente, et congruente avec le principe de simplicité promu avec l'approche proposée.

D'autre part, il importe de remarquer qu'une courbe de lissage est calculée par question. En effet, la distribution des scores est très différente d'une question à l'autre (voir la section 4 qui suit et la figure 2). C'est du moins le cas pour les données de Mohler et il n'y a aucune raison de croire que ce soit une exception. Cette observation motive la création d'un modèle par question plutôt qu'un modèle général pour l'ensemble de l'examen.

4 Données

Le jeu de données Mohler [16] est une référence pour l'évaluation des systèmes de correction automatique des réponses courtes (ASAG) [3, 11]. Il se compose de 80 questions et de 2 273 réponses d'étudiants, chacune notée sur une échelle de 0 à 5 par deux enseignants. Les données proviennent d'un cours d'introduction à l'informatique d'une université au Texas. La longueur moyenne des réponses de référence est de 15 à 20 mots. Quelques exemples sont fournis au tableau 1.

À noter que nous considérons que chaque question de ce corpus possède une et une seule bonne réponse. Il peut y avoir une certaine flexibilité dans formulation de la réponse, comme c'est le cas pour la première question du tableau 1 portant sur le rôle d'un prototype, mais nous considérons néanmoins ici qu'il n'y a qu'une seule bonne réponse. Une approche plus sophistiquée s'imposerait s'il devait y avoir plusieurs bonnes réponses.

Tableau 1. Exemples de questions-réponses du corpus de Mohler [16]

Question	**Réponse**	
	de référence	**de l'étudiant**
What is the role of a prototype program in problem solving ?	To simulate the behaviour of portions of the desired software product.	High risk problems are address in the prototype program to make sure that the program is feasible. A prototype may also be used to show a company that the software can be possibly programmed.
How many constructors can be created for a class ?	Unlimited number.	as many as the user decides to use.
How do you delete a node from a binary search tree ?	Find the node, then replace it with the leftmost node from its right subtree (or the rightmost node from its left subtree).	You must first traverse the tree to find the appropriate value. Then you must make sure that the node is a leaf node. If it is, then you can delete the pointer to that specific node.
What are the main operations associated with a stack ?	push and pop	pop, push

La figure 1 illustre la distribution du score des 2 273 réponses selon leur similarité avec leur réponse attendue respective. On constate qu'il y a un grand nombre de réponses qui sont de 5/5 et relativement peu de réponses sous le score de 4/5. Cependant, la distribution par question est fort variable comme on le voit à la figure suivante.

La figure 2 illustre les distributions du score par similarité pour quelques questions individuelles et d'environ 30 étudiants. Comme on le constate, ces distributions varient considérablement d'une ques-

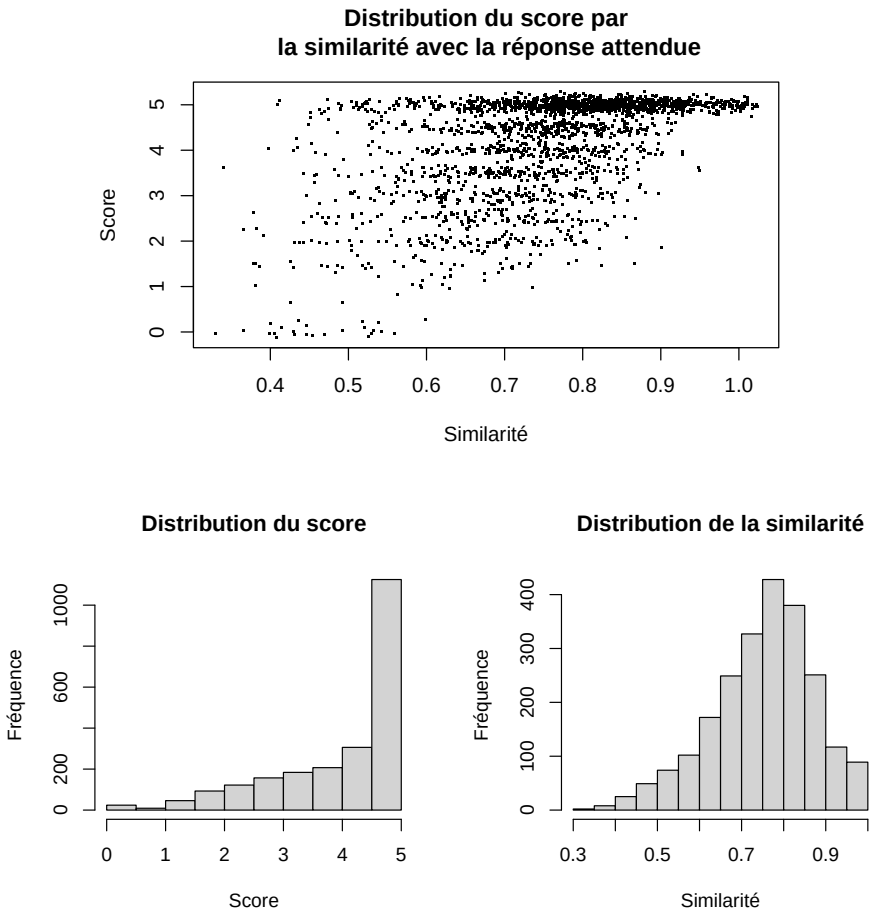

Figure 1. Figure du haut : distribution du score par la similarité entre la réponse de l'étudiant et celle attendue. Un bruit gaussien a été appliqué afin de discerner les points qui se superposent. Figure du bas : histogrammes des valeurs du score et de la similarité.

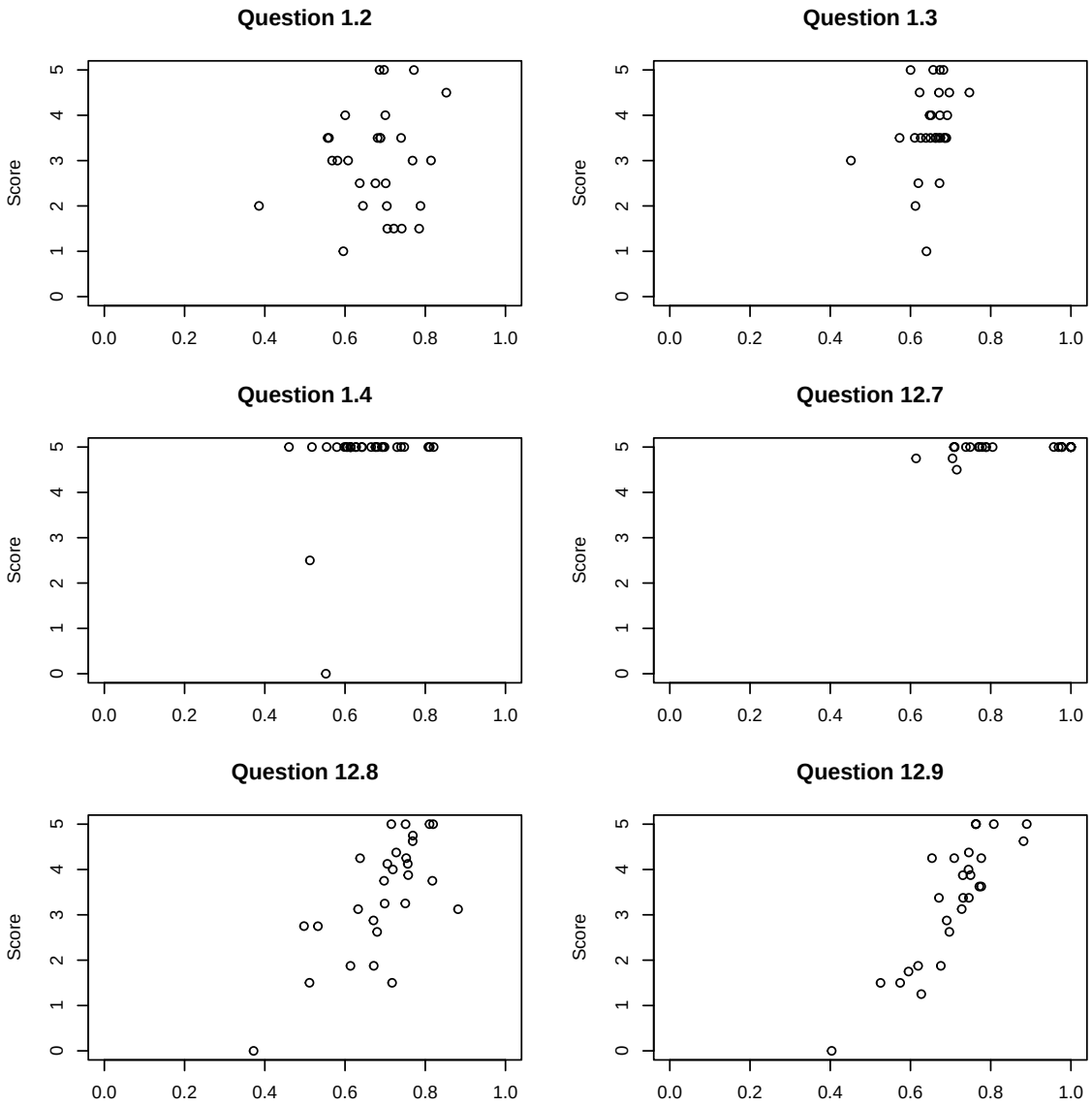

Figure 2. Distribution du score par la similarité (en x) de quelques questions.

tion à l'autre. Par exemple, les questions 1.2 et 1.3 comportent un faible taux de corrélation, contrairement à 12.9. Les questions 1.4 et 12.7 ont, quant à elles, un très haut taux de réussite qui rend le calcul de la corrélation très variable sur la base d'un échantillonnage. Compte tenu des particularités de chaque distribution, il s'avère plus efficace de modéliser une distribution propre à une question que pour l'ensemble.

5 Méthodologie

La tâche consiste à calculer un score à partir d'une mesure de similarité. Chacune des 2 273 réponses étudiantes est associée à une réponse attendue. Nous avons utilisé un modèle de langue pour obtenir un encodage pour chaque réponse sous la forme d'un vecteur dont la longueur varie typiquement de quelques centaines à quelques milliers de chiffres pour les plus grands modèles. Nous effectuons par la suite un calcul du cosinus entre les vecteurs qui sert de mesure de similarité. Aucun prétraitement des données n'est réalisé.

Le tableau 2 rapporte différentes statistiques des modèles pour l'encodage de phrases que nous avons étudiés. Notre choix s'est porté sur `mxbai-embed-large` dont la corrélation entre les similarités et les scores est la plus élevée. Ce modèle est performant. Il encode une centaine de questions en légèrement plus d'une seconde avec un processeur M4 par Apple.

L'objectif est d'obtenir une fonction de régression fournissant un score prédit pour une valeur de similarité entre la réponse étudiante et la réponse attendue. Cette fonction est repose sur un échantillon de 15 réponses pour lesquelles un score est obtenu par le correcteur. Plusieurs variantes ont été testées avec des résultats relativement semblables. La version utilisée ici consiste à calculer une somme pondérée des 15 scores fournis en fonction d'une distance Gaussienne de similarité avec la réponse pour laquelle on désire prédire un score. Une fonction est définie pour chaque question.

Le choix des 15 réponses est basé sur un échantillonnage uniforme des scores : une fois les réponses ordonnancées selon la similarité, la sélection est faite sur les rangs impairs.

Le code et les données utilisées l'expérimentation sont accessibles du lien `https://osf.io/rsg2w/?view_only=d5e180fe1463474f894e4065e3356b99`.

Tableau 2. Corrélations entre les réponses étudiantes et de référence pour trois modèles. Les autres informations rapportées sont le nombre de paramètres (en millions), la quantification, la taille des plongements (*embeddings*), le temps de traitement moyen pour 100 questions et l'année d'introduction.

Modèle	Corrélation	# Paramètres (m)	Quan.	Vec.	Temps	Année
all-minilm-l6-v2	0.482	22.6	–	384	0.5s	2021
bge-m3	0.517	576.6	16	1024	1.2s	2024
mxbai-embed-large	0.523	334	16	1024	1.2s	2024

6 Résultats

Le tableau 3 rapporte la performance du modèle proposé, simsemlis, en termes de corrélation et d'erreur quadratique moyenne (EQM). Ces résultats sont mis en comparaison avec ceux des approches de l'état de l'art [1,12,17,23] et de deux autres modèles basés ceux-ci sur des approches plus classiques et dites *bag-of-words* [15,18]. Les meilleures performances sont indiquées en caractère gras.

Tableau 3. Performance des modèles. Voir [1] pour une liste plus exhaustive de comparaisons ([1], tableau 1).

Approche	**Type**	**Corrélation**	**EQM**
simsemlis (notre approche)	Sim. sém.	0.623	0.755
État de l'art			
Garg et al. (2022) [12]	Rég. BERT + Sém. sim.	0.777	**0.732**
Agarwal et al. (2022) [1]	Graphe + transformeur	nd	0.762
Ouahrani et al. (2024) [17]	IAG+Sim. sém.	0.735	0.779
Zhu et al. (2022) [23]	BERT + LSTM.	**0.897**	0.827
Approches classiques			
Sultan et al. (2016) [18]	BOW	0.592	0.887
Mohler et al. (2011) [15]	BOW	nd	0.999

L'approche proposée se situe donc au deuxième rang des meilleures approches en termes d'EQM, mais au dernier ou avant dernier rang quant à la corrélation parmi les approches de l'état de l'art.

7 Discussion

On constate que les meilleurs performances diffèrent selon qu'on retient la corrélation ou l'EQM. Comme on peut le voir à la figure 2 des questions comme 1.4 et 12.7 peuvent avoir une corrélation faible, car presque tous les scores sont de 5/5, mais l'erreur quadratique peut être faible si la fonction du score a une courbe qui reflète bien la distribution. C'est particulièrement le cas pour la question 12.7 où la fonction de lissage gaussien. C'est aussi le cas pour 1.4, mais on remarque que deux réponses sont faibles (2.5 et 0) et que dans cette même zone de similarité on retrouve aussi des scores de 5.

La question se pose donc si c'est l'EQM ou la corrélation qui devrait être retenue comme mesure de performance étant donné l'écart relativement important du rang des meilleures approches. Étant donné que les distributions de scores par question s'éloignent fortement de la distribution gaussienne (ou normale) et qu'il y a parfois un grand nombre de scores égaux (surtout pour la valeur de 5) et que ce n'est pas le cas des scores estimés, le choix de l'EQM s'impose. Nous avançons en fait que la corrélation seule n'est pas une mesure adéquate puisqu'elle est invariable à la translation et à la mise l'échelle. Par exemple, en supposant des valeurs observées de $[0, 1, 2]$, des valeurs prédites de $[3, 4, 5]$ donneraient une corrélation de 1 qui omet l'erreur de biais. Des valeurs prédites de $[4.0, 4.1, 4.2]$ (une mise à l'échelle) donneraient quant à elles une corrélation de 1. On voit bien ici comment la corrélation peut s'avérer trompeuse et c'est pourquoi nous adopterons uniquement l'EQM. De plus, l'EQM reflète davantage ce que l'utilisateur désire : un faible écart entre les valeurs prédites et observées.

En ne retenant que l'EQM, la méthode simsemlis se situe au deuxième rang de l'état de l'art. Compte tenu qu'elle est beaucoup moins exigeante en temps de calcul et en simplicité que les approches d'affinage de transformer [1,12,23], elle semble donc un choix valable.

Mentionnons cependant que l'approche de paraphrasage de la réponse attendue [17] se compare probablement à simsemlis en termes de simplicité puisqu'elle ne nécessite pas d'affinage. Le principe consiste à créer plusieurs versions de réponses attendues avec l'IA générative afin d'améliorer l'évaluation d'une réponse étudiant. Mais en ce qui

concerne la présente étude, la performance EQM demeure à l'avantage de simsemlis.

8 Conclusion

Nous proposons une méthode qui offre l'avantage d'une performance près de l'état de l'art en termes de précision. Elle est aussi très peu gourmande en termes de ressources de calculs comparativement aux autres approches de l'état de l'art qui nécessitent un affinement, et elle est d'une grande simplicité à implémenter. Cependant, elle nécessite une correction manuelle d'une quinzaine de réponses par question.

Cependant, une analyse de la distribution des scores par question suggère que l'étape de correction manuelle pour calibrer la correction ou pour affiner un modèle est peut-être incontournable.

Mais que la correction manuelle d'une partie des réponses soit incontournable ou non, la responsabilité d'avoir une correction fiable repose toujours dans les mains de l'invidu qui corrige. Pour cette raison, une prochaine étape de cette étude encore exploratoire est de déterminer dans quelle mesure l'approche est en mesure d'identifier les évaluations automatiques qui sont fiables de celles qui nécessiteraient une vérification.

Références

1. Agarwal, R., Khurana, V., Grover, K., Mohania, M., Goyal, V. : Multi-relational graph transformer for automatic short answer grading. In : Carpuat, M., de Marneffe, M.C., Meza Ruiz, I.V. (eds.) Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies. pp. 2001–2012. Association for Computational Linguistics, Seattle, United States (Jul 2022). https://doi.org/10.18653/v1/2022.naacl-main.146, https://aclanthology.org/2022.naacl-main.146/
2. Aggarwal, D., Bhattacharyya, P., Raman, B. : "I understand why I got this grade" : Automatic short answer grading with feedback (2024), https://arxiv.org/abs/2407.12818
3. Amur, Z.H., Kwang Hooi, Y., Bhanbhro, H., Dahri, K., Soomro, G.M. : Short-text semantic similarity (stss) : Techniques, challenges and future perspectives. Applied Sciences **13**(6), 3911 (2023)
4. Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., Schneider, N. : Abstract meaning representation for sembanking. In : Proceedings of the 7th linguistic annotation workshop and interoperability with discourse. pp. 178–186 (2013)

5. Carpenter, D., Min, W., Lee, S., Ozogul, G., Zheng, X., Lester, J. : Assessing student explanations with large language models using fine-tuning and few-shot learning. In : Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024). pp. 403–413 (2024)
6. Chang, L.H., Ginter, F. : Automatic short answer grading for Finnish with ChatGPT. In : Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 23173–23181 (2024)
7. Condor, A., Litster, M., Pardos, Z. : Automatic short answer grading with sbert on out-of-sample questions. International Educational Data Mining Society (2021), https://educationaldatamining.org/EDM2021/virtual/static/pdf/EDM21_paper_149.pdf
8. Del Gobbo, E., Guarino, A., Cafarelli, B., Grilli, L. : GradeAid : a framework for automatic short answers grading in educational contexts—design, implementation and evaluation. Knowledge and Information Systems **65**(10), 4295–4334 (2023), https://rdcu.be/d5aDn
9. Divya, A., Haridas, V., Narayanan, J. : Automation of short answer grading techniques : Comparative study using deep learning techniques. In : 2023 Fifth International Conference on Electrical, Computer and Communication Technologies (ICECCT). pp. 1–7 (2023). https://doi.org/10.1109/ICECCT56650.2023.10179759, https://ieeexplore.ieee.org/document/10179759
10. Firoozi, T., Bulut, O., Epp, C.D., Naeimabadi, A., Barbosa, D. : The effect of fine-tuned word embedding techniques on the accuracy of automated essay scoring systems using neural networks. Journal of Applied Testing Technology pp. 21–29 (2022)
11. Gaddipati, S.K., Nair, D., Plöger, P.G. : Comparative evaluation of pretrained transfer learning models on automatic short answer grading. arXiv preprint arXiv :2009.01303 (2020)
12. Garg, J., Papreja, J., Apurva, K., Jain, G. : Domain-specific hybrid BERT based system for automatic short answer grading. In : 2022 2nd International Conference on Intelligent Technologies (CONIT). pp. 1–6. IEEE (2022)
13. Ghavidel, H.A., Zouaq, A., Desmarais, M.C. : Using BERT and XLNET for the automatic short answer grading task. In : CSEDU (1). pp. 58–67 (2020)
14. Grévisse, C. : Llm-based automatic short answer grading in undergraduate medical education. BMC Medical Education **24**(1), 1060 (2024)
15. Mohler, M., Bunescu, R., Mihalcea, R. : Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In : Lin, D., Matsumoto, Y., Mihalcea, R. (eds.) Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies. pp. 752–762. Association for Computational Linguistics, Portland, Oregon, USA (Jun 2011), https://aclanthology.org/P11-1076/
16. Mohler, M., Mihalcea, R. : Text-to-text semantic similarity for automatic short answer grading. In : Lascarides, A., Gardent, C., Nivre, J. (eds.) Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009). pp. 567–575. Association for Computational Linguistics, Athens, Greece (Mar 2009), https://aclanthology.org/E09-1065/
17. Ouahrani, L., Bennouar, D. : Paraphrase generation and supervised learning for improved automatic short answer grading. International Journal of Artificial Intelligence in Education pp. 1–44 (2024)

18. Sultan, M.A., Salazar, C., Sumner, T. : Fast and easy short answer grading with high accuracy. In : Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies. pp. 1070–1075 (2016)
19. Tobler, S. : Smart grading : A generative ai-based tool for knowledge-grounded answer evaluation in educational assessments. MethodsX **12**, 102531 (2024)
20. Tulu, C.N., Ozkaya, O., Orhan, U. : Automatic short answer grading with semspace sense vectors and malstm. IEEE Access **9**, 19270–19280 (2021)
21. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I. : Attention is all you need. Advances in neural information processing systems **30** (2017)
22. Yoon, S.Y. : Short answer grading using one-shot prompting and text similarity scoring model (2023), `https://arxiv.org/abs/2305.18638`
23. Zhu, X., Wu, H., Zhang, L. : Automatic short-answer grading via BERT-based deep neural networks. IEEE Transactions on Learning Technologies **15**(3), 364–375 (2022)