Correction automatique d'examens écrits par approche neuronale profonde et attention croisée bidirectionnelle

Yanis Labrak, Philippe Turcotte, Richard Dufour, Mickael Rouvier

Correction automatique d'examens écrits par approche neuronale profonde et attention croisée bidirectionnelle

Yanis Labrak[1] Philippe Turcotte[1] Richard Dufour[2] Mickael Rouvier[1]
(1) Avignon Université (LIA), 339 Chemin des Meinajaries, 84911 Avignon, France
(2) Nantes Université (LS2N), 2 Chemin de la Houssinière, 44300 Nantes, France
yanis.labrak@univ-avignon.fr, philippe.turcotte@alumni.univ-avignon.fr,
richard.dufour@univ-nantes.fr, mickael.rouvier@univ-avignon.fr

RÉSUMÉ
Cet article présente les systèmes développés par l'équipe LIA-LS2N dans le cadre de la campagne d'évaluation *DEFT 2022* (Grouin & Illouz, 2022). Nous avons participé à la première tâche impliquant la correction automatique de copies d'étudiants à partir de références existantes. Nous proposons trois systèmes de classification reposant sur des caractéristiques extraites de plongements de mots contextuels issus d'un modèle BERT (CamemBERT). Nos approches reposent sur les concepts suivants : extraction de mesures de similarité entre les plongements de mots, attention croisée bidirectionnelle entre les plongements et fine-tuning (affinage) des plongements de mots. Les soumissions finales comprenaient deux systèmes fusionnés combinant l'attention croisée bidirectionnelle avec nos classificateurs basés sur BERT et celui sur les mesures de similarité. Notre meilleure soumission obtient une précision de 72,6 % en combinant le classifieur basé sur un modèle CamemBERT affiné et le mécanisme d'attention croisée bidirectionnelle. Ces résultats sont proches de ceux obtenus par le meilleur système de cette édition (75,6 %).

ABSTRACT
Deep Neural Networks and Bidirectional Cross-Attention for Automatic Answer Grading.

This paper presents the systems developed by the LIA-LS2N team as part of *DEFT 2022* (Grouin & Illouz, 2022), a French text-mining challenge. We participated in the first task involving automatic marking of student answers based on existing reference information such as the question's text, teacher's answer and student's answer. We propose three classification systems relying on BERT contextual word embeddings (CamemBERT) to create various features. The features used by our systems include either : similarity measures between vectors, bidirectional cross-attention between embeddings or fine-tuned embeddings. The final submissions included two fusion models, combining the bidirectional cross-attention system with our BERT-based and similarity-based classifiers. Our best submission integrates the BERT-based classifier and bidirectional cross-attention, reaching a 72.6% precision. These results are close to those achieved by the best system for this edition (75.6%).

MOTS-CLÉS : Attention Croisée Bidirectionnelle, BERT, Transformers, Correction Automatique.

KEYWORDS: Bidirectionnal Cross-Attention, BERT, Transformers, Short Answer Grading.

1 Introduction

Le DÉfi Fouille de Textes (DEFT) est une campagne d'évaluation annuelle francophone. L'édition de cette année propose deux tâches d'évaluation automatique de réponses d'étudiants, une tâche

de base et une tâche continue où l'ont requête une base de données afin d'effectuer des prédictions itératives. Notre équipe s'est intéressée à la première tâche, qui consiste à prédire des notes d'après une référence existante. Ces tâches sont en continuité avec les tâches 2 et 3 introduites initialement dans l'édition 2021.

Le système ayant obtenu les meilleurs résultats lors de l'édition 2021 appliquait un Random Forest sur des caractéristiques de soft cardinalité ou *soft cardinality* en anglais (Jimenez *et al.*, 2015; Suignard *et al.*, 2021). Dans notre cas, nous proposons trois systèmes de classification reposant sur des caractéristiques extraites de plongements de mots contextuels issus de la version française de BERT (Devlin *et al.*, 2018), CamemBERT (Martin *et al.*, 2020), qui s'appuie initialement sur l'architecture Transformers (Vaswani *et al.*, 2017). Nos approches reposent sur les concepts suivants : extraction de mesures de similarité entre les plongements de mots, calcul de l'attention croisée bidirectionnelle entre les plongements de mots, ainsi que, le fine-tuning (affinage) des plongements de mots.

L'article est organisé de la manière suivante. La section 2 résume brièvement la tâche et le protocole expérimental. Ensuite, la section 3 présente les approches de fusion proposées ainsi que le prétraitement des données textuelles. Les expériences sont présentées dans la section 4 en même temps que les résultats, avant de conclure et de donner quelques perspectives dans les sections 5 et 6.

2 Description de la tâche

Pour cette édition DEFT, nous avons choisi de nous concentrer sur la tâche de base à savoir l'évaluation automatique de copies d'après une référence existante. L'objectif de cette tâche consiste à prédire la note d'une réponse d'un étudiant à une question (comprise entre 0 et 1) à partir d'éléments textuels tels que : la question, la réponse de l'étudiant ainsi que la réponse de l'enseignant. Cette tâche peut être modélisée comme un problème de régression ou de classification. Le Tableau 1 donne des exemples de questions. Le Tableau 2 donne quant à lui illustre les réponses d'étudiants avec les notes associées.

Id	Question	Correction enseignant (+ suggestion de notation)
1021	Quelle fonction PHP permet d'exécuter une requête sur une base de données PostgreSQL ? 	pg_query pas tenu compte des arguments dans la notation
2005	À quoi sert la balise <span> ? 	Elle permet de définir une partie de texte à laquelle on pourra par exemple appliquer un style ou un comportement particulier. 0.5 pour "La balise <span> sert à définir une partie du texte. ;" 0.5 aussi si parle de partie du document et pas du texte (si pas clair que inline et pas block)
2046	Quel problème d'accessibilité peut se poser lorsqu'un utilisateur agrandit la taille des caractères du texte ? 	perte de contenu, par exemple recouvrement de contenus (violation du principe 1 de wcag ie perceptible)

TABLE 1 – Exemple tiré du corpus DEFT de question ainsi que des corrections des enseignants.

Id	Note	Étudiant	Réponse de l'étudiant
1021	1.0	student122	$reponse=pg_query($connection, $requete);\n Il faut préalablement avoir effectué la connection avec la base : \n $connection=pg_connect($nombase);\n\n
2005	0.0	student10	la balise sert a créer des sections.
2046	0.5	student102	Certains caractères peuvent dépasser du cadre de la page web.
2040	0.8	student111	<!ELEMENT ue (#PCDATA)> <!–toujours pareil, ici on pourrait avoir autre chose que #PCDATA–>\n <!ATTLIST ue id ID #REQUIRED>\n <!ATTLIST ue lang CDATA>
2042	0.3	student92	<code>\n <enseignant>\n <nom> Snow</nom>\n <prenom>John</prenom>\n <bureau>IS</bureau>\n <statut>dead</statut>\n </enseignant>\n </code>

TABLE 2 – Exemple tiré du corpus DEFT de réponse d'étudiant avec les notes associées.

2.1 Évaluation

Pour l'évaluation de cette tâche, les participants pouvaient soumettre un maximum de trois soumissions sur une période de deux jours après réception des données d'évaluation. Les résultats obtenus n'ont été révélés qu'après la période d'évaluation. Afin d'évaluer nos systèmes lors de la phase d'entraînement et de validation, nous avons utilisé les métriques de précision, rappel et F-Mesure. Dans le cadre du défi, la métrique retenue pour l'évaluation des soumissions est la précision (ou *precision* en anglais).

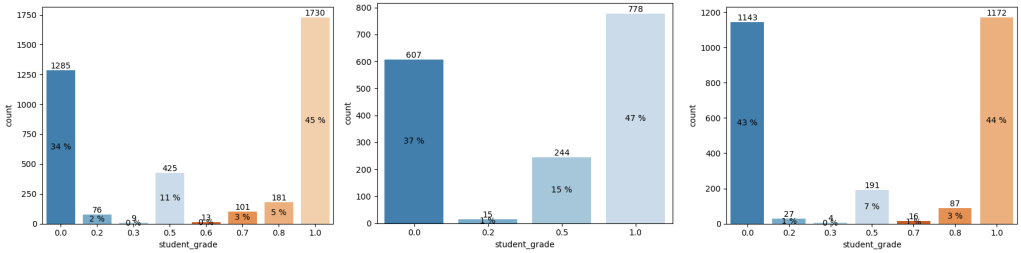
2.2 Corpus

Le corpus est composé de questions, de réponses d'étudiants et de réponses des enseignants. L'ensemble des données sont extraites de questionnaires électroniques de type *Moodle* et portent sur des énoncés en informatique, plus particulièrement de programmation web. Deux ensembles de questions et réponses nous ont été fournis, l'un pour l'entraînement et l'autre pour l'évaluation. Chaque ensemble est formé d'une paire de fichiers textes de format tabulaire (*i.e.* séparés par des tabulations) et ne contenant pas de ligne d'en-tête pour les colonnes. Les éléments dans ces deux fichiers sont liés par un numéro unique associé à chaque question. Nous avons formé un corpus de validation en retirant 15 % du corpus d'entraînement afin d'évaluer chaque cycle d'apprentissage. Le Tableau 3 donne le nombre de questions et réponses sur les corpus d'entraînement, de validation et d'évaluation.

	Corpus		
Champ	**Entraînement**	**Validation**	**Évaluation**
Questions uniques	50	48	21
Réponses enseignant uniques	50	48	21
Réponses étudiant uniques	3 247	573	1 644

TABLE 3 – Distribution de questions et réponses parmi les corpus.

La Figure 1 donne la distribution des notes sur les différents corpus. On constate que dans le corpus d'apprentissage, les notes des étudiants sont comprises entre 0 et 1 avec un pas de 0,1. Pour modéliser cette tâche, nous avons opté pour une méthode de classification sur 11 classes, de 0 à 1 avec un pas de 0,1.



Entraînement Validation Évaluation

FIGURE 1 – Distribution des notes sur les différents corpus.

3 Méthodologie

3.1 Pré-traitements

Les données textuelles fournies contenaient des balises HTML de mise en page, ainsi que du code faisant partie de la question et/ou réponse.

Les pré-traitements suivants ont donc été appliqués sur ces données textuelles :

1. Transformation des lettres en minuscule.
2. Suppression des espaces en début et/ou fin de texte.
3. Effacement des caractères `\n` et `\t`.
4. Suppression des balises HTML superflues.
5. Décodage des entités nommées HTML, telles que "` `", "`<`" ou "`>`".
6. Ajout d'un espace avant et après toute ponctuation ou tout symbole.
   — Cette règle n'est pas appliquée aux nombres décimaux tels que les suggestions de notes.
7. Remplacement des espaces successifs par un seul espace.

Certaines réponses officielles de questions contenaient également des suggestions de notation suivant le format "`<p>NOTE EXPLICATION</p>`". Le format des notes suggérées a été standardisé pour avoir un entier avant et après le séparateur de décimale. De plus, tous les séparateurs de décimale ont été transformés en point ("."). Nous avons ensuite créé une version du corpus avec et une autre sans ces suggestions de notation. Dans la version avec les suggestions, nous les concaténons à la réponse de l'enseignant avec un jeton spécial ("__PTS__").

Finalement, pour le système de similarités (voir section 3.2.1) et celui basé sur les plongements lexicaux CamemBERT (voir section 3.2.3), nous avons combiné la réponse de l'enseignant à celle de l'étudiant pour le vecteur d'entrée puisque cette combinaison obtenait les meilleurs résultats lors de nos analyses. Ces résultats sont disponibles dans le Tableau 3.1.

Combinaison	Précision	Rappel	F1-score
$R_{etudiant} + R_{enseignant} + Q$	59,26%	60,77%	59,94%
$R_{etudiant} + R_{enseignant}$	**66,08%**	**65,57%**	**65,30%**
$R_{etudiant} + Q$	52,24%	46,35%	45,53%

TABLE 4 – Résultats obtenus pour chaque combinaison en vecteur d'entrée pour notre classifieur basé sur les plongements lexicaux CamemBERT, où $R_{etudiant}$ est la réponse de l'étudiant, $R_{enseignant}$ est la réponse de l'enseignant et Q est le texte de la question.

3.2 Systèmes proposés

3.2.1 Vecteurs de similarités

Dans cette partie, nous avons extrait des caractéristiques fondées sur un ensemble de distances entre les deux plongements de mots CamemBERT (Martin *et al.*, 2020) issus des réponses d'étudiants et

d'enseignants dans le but de nous permettre ensuite de classifier les documents en fonction de leurs notes.

L'avantage de CamemBERT est que le modèle de langage a été pré-entraîné sur une très grande quantité de données textuelles en français dans le but de capturer le sens profond des mots et ainsi réduire la polysémie. CamemBERT fournit des performances à l'état de l'art pour la langue française sur diverses tâches similaires à celle que nous souhaitons résoudre (par exemple : classification, question-réponse, désambiguïsation, ...). Sa méthode d'entraînement est différente des autres représentations de mots comme `Word2Vec` (Mikolov *et al.*, 2013), `GloVe` (Pennington *et al.*, 2014), `FastText` (Bojanowski *et al.*, 2016) ou encore `Flair NLP` (Akbik *et al.*, 2019), car il permet d'obtenir une représentation des mots dynamique et entièrement bidirectionnelle à l'aide des Transformers et plus particulièrement du mécanisme de self-attention (Vaswani *et al.*, 2017).

Cependant, pour permettre une comparaison des plongements de mots entre eux à l'aide d'un calcul de distance, il nous faut des vecteurs mono-dimensionnels représentant le sens des phrases. Pour ce faire, nous avions le choix entre, appliquer une mise en commun des vecteurs des mots grâce à une minimisation (min-pooling), maximisation (max-pooling) ou moyenne (mean-pooling). Soit, de procéder à la sélection d'une des couches cachées du réseau Transformers (Jawahar *et al.*, 2019). Après analyse des résultats des quatre méthodes et de l'ensemble des couches cachées sur le corpus d'évaluation, nous avons décidé d'utiliser la représentation fournit par la huitième couche du modèle CamemBERT.

Concernant les distances sélectionnées, nous avons choisi d'utiliser les onze méthodes de calculs de distances suivantes :

- Cosine
- Euclidean
- Braycurtis
- Canberra
- Chebyshev
- Cityblock
- Correlation
- Euclidean
- Jensenshannon
- Minkowski
- Sqeuclidean

Ces méthodes de calcul de distances nous permettent, une fois leurs résultats concaténés, d'obtenir un vecteur de similarités. Ce vecteur nous sert ensuite à effectuer la classification en l'une des onze classes grâce à l'utilisation d'un perceptron multicouche (MLP).

3.2.2 Attention croisée bidirectionnelle

Pour ce système, nous avons utilisé les mêmes plongements de mots CamemBERT que ceux précédemment utilisés, à une condition près, nous n'appliquons pas de mise en commun des vecteurs des mots et gardons deux vecteurs bi-dimensionnels de taille 512 par 768. Une fois les vecteurs construits, nous les donnons en entrée du modèle qui va procéder de la sorte :

- Calculer l'attention croisée bidirectionnelle entre les deux vecteurs et conserver uniquement leurs états cachés (`hidden_state`).

- Puis, concaténer les deux sorties avec les vecteurs de chaque phrase passés préalablement dans un Bi-GRU, dans le but d'en extraire la dernière couche, qui synthétise au mieux le sens général de la phrase.

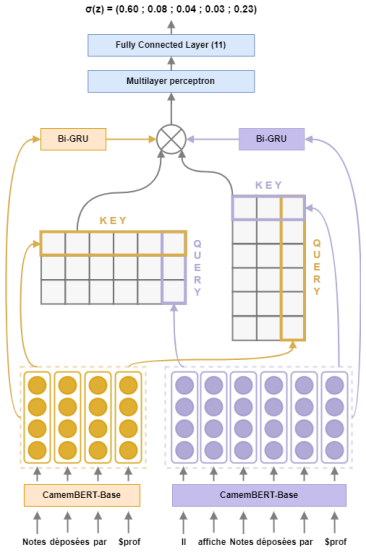


FIGURE 2 – Attention croisée bidirectionnelle + Bi-GRU.

3.2.3 CamemBERT fine-tuned

Pour ce dernier système, nous avons décidé d'entraîner un classifieur perceptron multicouche (Haykin, 1994) avec, en entrée, des plongements de mots `CamemBERT-base` obtenus à partir du corpus ne disposant pas des suggestions de notation de l'enseignant. Nous avons utilisé les frameworks Flair NLP (Akbik *et al.*, 2019) et HuggingFace Transformers (Wolf *et al.*, 2019) pour l'affinage des plongements de mots CamemBERT, qui nous permet d'obtenir des représentations plus adaptées à notre contexte bien particulier. Le classifieur est entraîné simultanément afin de classifier les documents en l'une des onze notes possibles. Le vecteur d'entrée du système est en fait une concaténation de la réponse de l'enseignant, la réponse de l'étudiant à une question précise, espacé d'un token spécial de séparation ("`[SEP]`"). Nous avons également implémenté une évaluation croisée avec cinq `K-folds` stratifiés afin de déterminer un indice de performance pour le système. Le modèle ayant obtenu le meilleur résultat a été retenu pour la fusion, présentée dans la section suivante.

3.3 Fusion des systèmes

Après avoir évalué les systèmes individuellement, nous avons constaté que certains systèmes étaient en capacité de mieux prédire certaines classes que d'autres. Il est donc possible que la fusion de ces systèmes amène à de meilleures performances globales. Nous avons donc proposé une approche pondérant manuellement les scores de probabilité en sortie de la `Softmax` en fonction de leurs classes de "prédilection" sur le corpus de validation afin d'obtenir un méta-système.

Le système proposé repose sur une architecture à 2 niveaux (figure 3). Le premier niveau consiste à obtenir différents points de vue d'une question-réponse en utilisant différents systèmes de prédiction de note. Nous proposons d'utiliser les 3 systèmes qui sont décrits plus en détails dans la partie 3.2. Le second niveau permet de combiner les systèmes du niveau 1. La combinaison des systèmes se fait au niveau des scores de probabilités. Ainsi, les scores donnés par les différents systèmes du premier niveau sont additionnés.

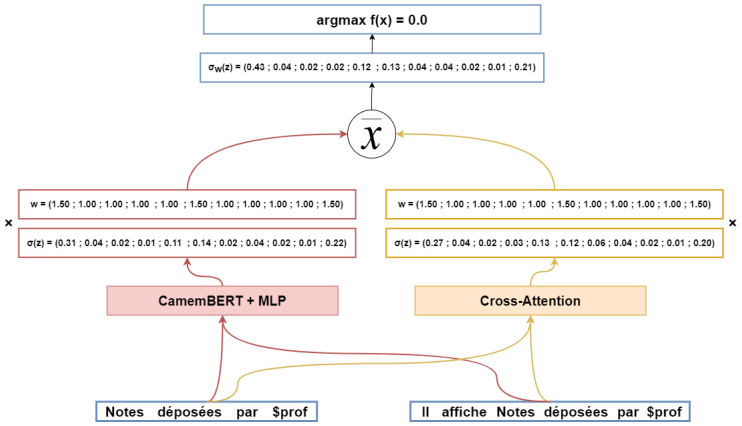


FIGURE 3 – Architecture CamemBERT et Attention croisée bidirectionnelle (Run 2).

4 Résultats

Dans le cadre de la campagne d'évaluation, nous avons proposé une baseline et deux systèmes issus de la fusion. Les résultats complets sont résumés dans le Tableau 5.

	Validation				Évaluation
Méthode	**Taux de succès**	**Précision**	**Rappel**	**F1-Mesure**	**Précision**
Best DEFT-2021	-	-	-	-	68,20 %
Best DEFT-2022	-	-	-	-	**75,60 %**
Run 1 : Similarities Features	65,63 %	86,63 %	65,63 %	71,80 %	60,60 %
Run 2 : CamemBERT + Attention croisée bidirectionnelle	**71,90 %**	**73,72 %**	**71,90 %**	**72,69 %**	**72,60 %**
Run 3 : Similarities Features + Attention croisée bidirectionnelle	66,06 %	87,48 %	66,06 %	72,17 %	64,90 %
Attention croisée bidirectionnelle	53,22 %	48,98 %	53,22 %	50,64 %	54,70 %
CamemBERT	71,47 %	68,79 %	71,47 %	69,83 %	69,70 %

TABLE 5 – Tableau des résultats.

Run 1 : Vecteur de similarités (baseline) Le premier système fait office de *baseline*. Les systèmes proposés lors de l'édition 2021 utilisaient entre autres des caractéristiques de similarités. Nous avons donc extrait des plongements de mots CamemBERT à partir des documents comportant les suggestions de notes afin de calculer les coefficients de similarités entre les vecteurs. Nous utilisons 11 différentes méthodes pour calculer les distances qui formeront le vecteur d'entrée du système. Nous utilisons ensuite ces vecteurs de similarités pour entraîner un classifieur MLP.

Run 2 : CamemBERT + Attention croisée bidirectionnelle. Le système basé sur un modèle CamemBERT affiné et celui sur l'attention croisée bidirectionnelle obtenaient des scores plus élevés pour la classe $0,5$, malgré des performances généralement plus faibles pour 0 et 1. Nous avons alors évalué la complémentarité de ces systèmes et avons observé une amélioration globale des performances lors de leur fusion. Nous avons mis en place une architecture permettant de former un méta-système en partant des sorties softmax de chacun des modèles puis effectué une moyenne pondérée des vecteurs. Cette méthode de fusion permet d'adapter la pondération aux performances du modèle sur chacune des classes afin de tirer le meilleur compromis des deux modèles.

Run 3 : Vecteur de similarités + Attention croisée bidirectionnelle. Le troisième système est assez similaire au précédent. Nous souhaitions ici quantifier l'apport de la méthode d'attention croisée

bidirectionnelle au système d'extraction de similarités vu dans le run 14. Plus techniquement, le système de d'extraction de similarités avait tendance, durant la phase de validation, à bien mieux prédire les classes 0 et 1 avec un certain avantage pour la classe 0. Alors que le système, basé sur l'attention croisée bidirectionnelle, a quant à lui de meilleures capacités à prédire les classes $0,5$ et 1. À l'issu de la fusion des deux systèmes, nous avons gagné, sur la phase d'évaluation, $4,3\%$ de précision.

5 Discussion

Le méta-système de fusion `CamemBERT + Attention croisée bidirectionnelle` apportent de meilleurs résultats lors de l'évaluation finale, bien que les autres systèmes obtenaient de meilleurs résultats sur le corpus de validation. Sa précision de 72,6 % se rapproche du meilleur système de l'édition 2022, ce dernier ayant obtenu une précision de 75,6 % (+3 %). Ces résultats montrent l'existence d'une piste viable qui n'avait pas encore été explorée lors de l'édition 2021. En effet, les résultats du méta-système (72,6%) et du classifieur CamemBERT seul (69,7%) sont supérieurs au meilleur système de l'édition 2021 qui avait obtenu une précision de 68,2%. Les approches de l'édition 2021 calculaient la soft cardinalité ou la similarité entre des deux plongements de mots issus de modèles BERT, tandis que notre approche vise à entraîner un classifieur basé sur des plongements de mots CamemBERT affinés, ou *fine-tuned* en anglais, sur notre domaine d'application et à le fusionner avec un système d'attention croisée bidirectionnelle.

Nous avons été très surpris par les gains de performances apportés par l'attention croisée bidirectionnelle et sa complémentarité avec les autres systèmes. L'apport le plus marquant est celui avec l'extraction des vecteurs de similarités, où nous sommes partis de 60,6 % de précision sans attention croisée bidirectionnelle pour atteindre 64,9 % avec, soit un gain de tout de même de 4,3 points de précision. Concernant le deuxième run, l'attention croisée bidirectionnelle a apporté des gains plus faibles (+2,9 %), mais reste tout de même significatif.

Enfin, notons que les résultats officiels obtenus par nos runs étaient considérablement plus bas que ceux obtenus sur le corpus d'évaluation final, soit 44,0 % pour le run1, 40,4 % pour le run 2 et 44,0 % pour le run 3. Suite à une analyse, nous avons remarqué que le chargement des données à prédire se faisait aléatoirement, ce qui explique pourquoi l'ordre des notes prédites ne correspondaient pas lors de notre soumission. Les résultats que nous avons présentés dans l'article ont été obtenus en désactivant le chargement aléatoire lors des prédictions et en exécutant les scripts d'évaluation fournis par les organisateurs de DEFT 2022.

6 Conclusion

Les performances de nos systèmes de base sont similaires aux résultats obtenus lors de l'édition 2021. Notre système de fusion CamemBERT couplé à l'attention croisée bidirectionnelle s'est approché du meilleur résultat obtenu cette année. Cette approche de classification basée sur les plongements de mots CamemBERT, nous semble donc pertinente à approfondir lors de la prochaine édition du défi. Il serait effectivement intéressant de combiner cette approche avec les caractéristiques de "soft cardinality" (Jimenez *et al.*, 2015) utilisés par le meilleur système de l'édition 2021 (Suignard *et al.*, 2021). L'architecture du système pourrait également être améliorée en remplaçant notre sélection manuelle des poids pour la moyenne pondérée par un modèle perceptron multicouche (Haykin, 1994). Nous espérions aussi étudier les performances des plongements de code source tels que `CodeBERT` par Microsoft (Feng *et al.*, 2020) puisque certaines réponses contiennent des extraits de code source.

Références

AKBIK A., BERGMANN T., BLYTHE D., RASUL K., SCHWETER S. & VOLLGRAF R. (2019). Flair : An easy-to-use framework for state-of-the-art nlp. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, p. 54–59.

BOJANOWSKI P., GRAVE E., JOULIN A. & MIKOLOV T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv :1607.04606*.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2018). Bert : Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv :1810.04805*.

FENG Z., GUO D., TANG D., DUAN N., FENG X., GONG M., SHOU L., QIN B., LIU T., JIANG D. & ZHOU M. (2020). Codebert : A pre-trained model for programming and natural languages. DOI : 10.48550/ARXIV.2002.08155.

GROUIN C. & ILLOUZ G. (2022). Notation automatique de réponses courtes d'étudiants : présentation de la campagne DEFT 2022. In *Actes de DEFT*, Avignon, France.

HAYKIN S. (1994). *Neural networks : a comprehensive foundation*. Prentice Hall PTR.

JAWAHAR G., SAGOT B. & SEDDAH D. (2019). What does BERT learn about the structure of language ? In *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy. HAL : hal-02131630.

JIMENEZ S., GONZALEZ F. A. & GELBUKH A. (2015). Soft Cardinality in Semantic Text Processing : Experience of the SemEval International Competitions. *Polibits*, p. 63 – 72.

MARTIN L., MULLER B., SUÁ REZ P. J. O., DUPONT Y., ROMARY L., DE LA CLERGERIE É., SEDDAH D. & SAGOT B. (2020). CamemBERT : a tasty french language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-main.645.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013). Efficient estimation of word representations in vector space.

PENNINGTON J., SOCHER R. & MANNING C. (2014). GloVe : Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1532–1543, Doha, Qatar : Association for Computational Linguistics. DOI : 10.3115/v1/D14-1162.

SUIGNARD P., BENAMAR A., MESSOUS N., CHRISTOPHE C., JUBAULT M. & BOTHUA M. (2021). Participation d'EDF R&D à DEFT 2021 (EDF R&D participation to DEFT 2021). In *Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles. Atelier DÉfi Fouille de Textes (DEFT)*, p. 72–81, Lille, France : ATALA.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER L. & POLOSUKHIN I. (2017). Attention is all you need.

WOLF T., DEBUT L., SANH V., CHAUMOND J., DELANGUE C., MOI A., CISTAC P., RAULT T., LOUF R., FUNTOWICZ M., DAVISON J., SHLEIFER S., VON PLATEN P., MA C., JERNITE Y., PLU J., XU C., SCAO T. L., GUGGER S., DRAME M., LHOEST Q. & RUSH A. M. (2019). Huggingface's transformers : State-of-the-art natural language processing. DOI : 10.48550/ARXIV.1910.03771.