Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles
Avignon, France, 27 juin au 1er juillet 2022

Stylo@DEFT2022 : Notation automatique de copies d'étudiant·e·s par combinaisons de méthodes de similarité

Ibtihel Ben Ltaifa[1] Toufik Boubehziz[1] Andrea Briglia[1] Corina Chutaux[1]
Yoann Dupont [2] Carlos-Emiliano González-Gallardo[3]
Caroline Koudoro-Parfait[1, 2, 4] Gaël Lejeune[1, 3]
(1) Sens Texte Informatique Histoire (STIH), Sorbonne Université, France
(2) Observatoire des Textes et des Connaissances (ObTIC), Sorbonne Université, France
(3) Laboratoire Informatique, Image et Interaction (L3i), La Rochelle Université, France
(4) Sorbonne Center for Artificial Intelligence (SCAI), Sorbonne Université, France
[ibtihel.ltaifa, toufik.boubehziz, andrea.briglia, corina.chutaux,
yoann.dupont, caroline.parfait, gael.lejeune]@sorbonne-universite.fr,
carlos.gonzalez_gallardo@univ-lr.fr

RÉSUMÉ

Cet article présente la participation de l'équipe STyLO (STIH, L3I, OBTIC) au DÉfi Fouille de Textes 2022 (DEFT 2022). La tâche proposée consiste en une évaluation automatique des questions à réponses courtes (EAQRC) de devoirs d'étudiant·e·s avec le corrigé de l'enseignant comme ressource pour chaque question. Nous exploitons dans notre approche une combinaison de différentes méthodes de représentation des données (corrigés et réponses) : mots, n-grammes de caractères (avec et sans frontières de mots), *word pieces] et sentence embeddings ainsi que de différents algorithmes pour calculer la note (régression linéaire et régression logistique). Les méthodes sont évaluées en termes d'exactitude et de corrélation de Spearman.*

ABSTRACT

Stylo@DEFT2022 : Automatic short answer grading by combinations of similarity methods

This paper presents the participation of the STyLO team at DÉfi Fouille de Textes 2022 (DEFT 2022). The proposed task consists in an automatic evaluation of short open-ended questions based on a few answers (models) already corrected by the teacher for each question. A combination of different text representation methods is proposed : words, character n-grams (with and without word boundaries), word pieces and sentence embeddings as well as different regression models (linear regression and logistic regression) for scoring responses are performed. These methods are evaluated in terms of accuracy and Spearman correlation.

MOTS-CLÉS : évaluation automatique des réponses courtes, n-grammes de caractères, word pieces, sentence embeddings, régression linéaire, régression logistique, Question à réponse ouverte courte, QROC, Évaluation automatique des questions à réponses courtes, EAQRC.

KEYWORDS: automatic short answer grading, character n-grams, word pieces, sentence embeddings, linear regression, logistic regression, ASAG, Open-ended question.

Atelier DÉfi Fouille de Textes (DEFT), pages 11–22.

1 Le défi de la correction Automatique de réponses d'étudiant·e·s

Dans la continuité de l'édition 2021 (Grouin *et al.*, 2021), cette nouvelle édition du Défi Fouille de Textes (DEFT) a porté sur la correction automatique de copies électroniques d'étudiant·e·s (du type questionnaire Moodle) et présente deux tâches. Une première tâche de prédiction des notes attribuées aux réponses des étudiant.e.s à chaque question d'après une référence existante, à l'instar des tâches proposées pour DEFT 2021. Puis, une tâche continue qui consiste à prédire les notes en interrogeant itérativement le serveur. Nous avons décidé de participer à la première tâche : prédictions apprises d'après une référence. Au travers de ces expériences, nous explorons différentes techniques d'extraction des traits lexicaux, de calcul de similarité (Bray-Curtis et Cosinus), de régression linéaire ou logistique, de classification et réseau de neurones. Les techniques sont évaluées en termes d'exactitude et de score de corrélation de Spearman.

La correction automatique peut s'apparenter à une question de recherche de similarité, c'est-à-dire qu'il s'agit de chercher comment, dans le cas où la réponse n'est pas strictement la même que le corrigé, la réponse donnée mérite tout ou partie des points. En ce sens, on peut trouver une filiation entre les éditions 2021 et 2022 du défi et l'édition 2020. Les expérimentations de l'édition DEFT 2020 ont eu pour but de résoudre les tâches suivantes : (I) similarité sémantique entre phrases et (II) sélection de phrases parallèles. Des modèles d'apprentissage profond multilingues, comme MUSE (Lample *et al.*, 2018) et M-BERT (Devlin *et al.*, 2018), ont été utilisés pour la tâche 1, par (Teissèdre *et al.*, 2020), leurs résultats ont montré des performances plutôt moyennes par rapports aux autres systèmes présentés. Néanmoins, ils ont obtenu les meilleurs résultats pour la tâche 2 en utilisant le modèle MUSE, alors même que le système n'exploitait pas les données d'entraînement. Des méthodes fondées sur des mesures de similarité n'impliquant pas d'apprentissage profond aveint également été proposées (Buscaldi *et al.*, 2020).

Plusieurs approches ont été proposées dans le cadre de DEFT 2021 qui portent sur deux tâches concernant l'évaluation automatique des questions courtes liées au domaine de l'informatique. Les approches proposées s'appuient sur l'extraction des caractéristiques, à la fois des réponses et des questions, en utilisant principalement des techniques du TAL. Ces méthodes, fondées sur l'utilisation d'un algorithme d'apprentissage automatique, ont été utilisées pour prédire la note à partir de l'extraction de traits appartenant à différentes catégories : traits lexicaux, syntaxiques et sémantiques (Galhardi *et al.*, 2018). La majorité des approches proposées pour la résolution de cette tâche utilisent des modèles impliquant les formes directement identifiables dans les données : des descripteurs de types n-grammes (de carctères ou de mots) (Dupont *et al.*, 2021) éventuellement combinés des méthodes de pondération (TF-IDF) (Süzen *et al.*, 2020). On observe également l'utilisation de redescriptions via des méthodes de plongement de mots (par exemple FastText) (Poulain & Connes, 2021). Il s'agit de calculer les similarités lexicales entre la question et les réponses qui lui sont associées. Pour ce faire, sont couramment utilisées les mesures de similarité globales sur des vecteurs (sacs de mots ou sacs de *features*) comme les distances cosinus, Euclidienne et de Hamming.

Nous présentons une étude détaillée de méthodes basées sur les modèles de traitements lexicaux et les mesures de similarité. Dupont *et al.* (2021) ont proposé une méthode de correction fondée sur des similarités de chaînes de caractères. Les auteurs se sont basés sur un calcul de similarité entre la correction et la réponse de l'étudiant. Ils ont testé différents types de caractéristiques pour vectoriser : des mots (uni-grammes et bi-grammes) et des chaînes de caractères (restreintes à l'intérieur des mots ou libres). Une comparaison est faite en utilisant différentes mesures de similarité afin d'analyser leur influence sur la qualité des résultats. Ainsi, dans cet article, il apparaît que le coefficient de Dice (Dice,

1945) a donné les meilleurs résultats globaux, tandis que la distance cosinus a tendance à bien prédire les notes, mais est moins performante en terme d'écart avec la note recherchée. Wang *et al.* (2021) ont principalement utilisé des traits basés sur la similarité cosinus entre les réponses et la référence. Les auteurs ont analysé plusieurs types de vecteurs comme vecteurs de sac de mots (occurrences de mots, vecteurs tf-idf), embeddings non contextualisés, embeddings contextualisés et embeddings de phrases ajustés sur des corpus multilingues. Dans Poulain & Connes (2021), les auteurs ont abordé le problème comme une tâche de régression. Ils se sont appuyés sur l'extraction de traits lexicaux joints à une méthode de sélection des traits et une méthode d'augmentation des données textuelles. Dans Suignard *et al.* (2021), les auteurs ont exploré des différents types de similarité et ont défini des seuils de similarité pour affecter les notes en trois classes prédéfinies. Dans cette tâche, d'autres approches sont basées sur l'extraction des traits syntaxiques en utilisant les étiquetages morphosyntaxiques pour la détection de signes de ponctuation, par exemple Dupont *et al.* (2021). Enfin, les approches basées sur des traits sémantiques (Cartier, 2018) semblent rarement utilisés pour cette tâche.

2 Description des jeux de données

La tâche de base demandée dans le cadre de cette édition du défi est une tâche de poursuite automatique de l'évaluation de réponses d'étudiant·e·s à partir de premières corrections par l'enseignant pour chaque énoncé. Le corpus d'entraînement de DEFT 2022 se compose des corpus d'entraînement et de test de l'édition DEFT 2021. Le nouveau corpus de test 2022`https://deft.lisn.upsaclay.fr/2022/` est similaire du point de vue du contenu à celui de 2021. Les tableaux 1 et 2 représentent respectivement des exemples de questions et de réponses à la tâche de base.

ID question	Note	N°	Question	Correction enseignant
1001	1	1	Qu'est-ce que le World Wide Web ?	système hypertexte fonctionnant sur internet = une des applications d'internet, comme courrier électronique, messagerie instantanée...

TABLE 1 – Un exemple de question de la tâche de base, extraits du site DEFT 2022

ID question	Note	ID étudiant	Réponse étudiant
1001	0,5	student101	Ce sont les pages web accessible par tout navigateur.\n
1001	0	student108	Un réseau mondial \n
1001	1	student3	C'est le systeme hypertexte qui sert à consulter des documents et des pages hébergés sur le réseau internet\n
1001	0	student49	NO_ANS

TABLE 2 – Des exemples de réponses à la tâche de base, extraits du site DEFT 2022

L'évaluation automatique des copies d'étudiant·e·s repose sur la corrélation entre la grille de notation et la correction fournie par l'enseignant. Cependant, durant l'exploitation et l'analyse de la base de données, on s'est aperçu que la correction de l'enseignant comportait des incohérences dues à la subjectivité du correcteur. Ainsi ont été observées les conclusions suivantes :

Les réponses attendues aux questions (ex. Q2008, Q2016) ainsi que les grilles de critères ne sont pas claires, voire inintelligibles dans certains cas. Par exemple, pour la question 2016, la grille de notation est la suivante :

> **À quoi sert le code suivant d'une feuille de style ?**
>
> ```
>  p::first-line{
> font-variant: small-caps; }
> ```
>
> **Corrigé :**
>
> Met la première ligne de chaque paragraphe en petites majuscules.
>
> Presque ok si ne savent pas ce qu'est small caps mais ont compris le reste.
>
> 0 si pas dit qu'il s'agit de la première ligne des paragraphes.

Cette formulation avec des expressions adverbiales d'approximation est sans doute plus adaptée comme consigne à un correcteur humain que comme support à l'apprentissage automatique. C'est un des grands enjeux de ce sujet, comment rendre des instructions de correction traitables par la machine.

Pour ce qui est de la notation, *1p* correspond majoritairement à la réponse attendue par l'enseignant. Il est donc aisé de distinguer les valeurs correctes des mauvaises (1 ou 0). En revanche, il y a un problème avec les réponses notées *0,5p* et *0,7p*, très souvent des réponses similaires et parfois identiques ont reçu des notes différentes. Par exemple, ligne 829, 813, 808 (pour ne noter que quelques occurrences), il y a une incohérence dans la notation : des réponses quasi identiques sont notées une fois avec *0.5p* et après avec *0,7p*. Ce même problème a été identifié entre le *0p* et le *0,2p*. Le professeur accorde *0,2p* par indulgence et subjectivité. Le cas le plus évident est la question 2021 que nous décrivons ci-dessous.

> **Qu'affiche le code suivant ?**
>
> ```
> <code>
>     <?php
>         $eleves[]="Jean Bleu";
>         $eleves[]="Jacques Blanc";
>         $eleves[]="Paul Rouge";
>         echo $eleves[1];
>     ?>
> <code>
> ```
>
> **Corrigé :**
>
> ```
> Jacques Blanc
> ```

Cependant, beaucoup d'étudiants ont répondu Jean Bleu et ont obtenu *0,2p* au lieu de *0p*. Cette inconsistance de notation s'étend sur la plupart des réponses notées avec *0,2p*. La notation de l'enseignant est beaucoup moins précise quand les étudiants rédigent des réponses longues et il arrive souvent dans ce cas de figure que des réponses similaires ne soient pas notées de la même façon et que l'enseignant décide de donner *1p* là où avant il avait donné *0,7p* voire *0,5p*.

S'aligner sur les notes de l'enseignant dans l'automatisation de la correction, c'est emporter une part de sa subjectivité dans l'automatisation de la correction.

3 Méthodes de similarité pour la correction automatique

Dans cette tâche, nous proposons une approche basée sur l'usage de plusieurs méthodes de similarité pour la correction automatique. Ainsi, nous utilisons un modèle de classification et deux modèles de régression qui exploitent différentes mesures de similarités entre la réponse de l'étudiant et la correction. Plus précisément, nous construisons un vecteur où chaque dimension est une similarité calculée sur une représentation particulière des textes. Nous utilisons deux mesures de similarité (Bray-Curtis et Cosinus) que nous appliquons sur différentes représentations vectorielles des textes : représentations en mots, en n-grammes de caractères, en en *word pieces* ou encore des *sentence embeddings*. Nous présentons chaque étape successivement.

3.1 Extraction de caractéristiques

Nous avons utilisé quatre types de caractéristiques pour la vectorisation des réponses d'étudiant·e·s puisque les représentations en mots, *Bag of Words*, s'avèrent insuffisamment souples pour traiter les problèmes de variation des formes, ce qui n'est que partiellement réglé par la lemmatisation ou la correction automatique sur des données textuelles standards sera *a fortiori* problématique lorsque l'on va traiter du code informatique. Nous avons donc ajouté différentes représentations avec des chaînes de caractères mots ou non-mots. Nous avons également utilisé une représentation en *WordPiece* (Kudo, 2018) afin de fournir des sous-chaînes pertinentes en plus des n-grammes de caractères. Nous avons utilisé plus précisément les *word pieces* fournies par `CamemBERT` (Martin *et al.*, 2019). Enfin, des vecteurs contextualisés Sentence-BERT. Nous avons testé ces différentes manières de vectoriser. Par la suite, nous avons utilisé des différentes mesures de similarité afin de comprendre l'influence de différents types de caractéristiques sur la qualité des résultats.

N-grammes sans frontières de mots (`char`) : Dans cette approche, un n-gramme désignera une séquence de N caractères consécutifs (Jalam, 2003) où les unigrammes sont des mots uniques, bigrammes des séquences de deux mots, trigrammes des séquences de trois mots et ainsi de suite. Dans notre cas, nous nous sommes basés sur les trigrammes et quadrigrammes. Cette technique de représentation capture les connaissances des séquences de caractères les plus fréquentes (affixes et mots grammaticaux par exemple) sans prendre en considération les frontières de mots.

N-grammes avec frontières de mots (`char_wb`) : Cette technique de représentation prend en considération le début et la fin du mot. Dans le cas de frontière de mot, les caractères entre les frontières peuvent être des mots (tokens), de la ponctuation ou des blancs situés entre deux mots. Ces n-grammes de caractères restreints à l'intérieur des mots (on parle parfois de *sub-word level features*) sont des descripteurs un peu plus « linguistiquement motivés »des n-grammes libres (voire par exemple (Labeau & Allauzen, 2017; Lejeune & Cartier, 2017). Ces représentations n'exploitant pas de ressource externes, elles sont strictement endogènes au sens de (Vergne, 2003), nous avons complété avec des représentations sous forme de *word pieces*.

Sous-chaînes de mots (`WordPiece`) : Pour traiter efficacement les mots rares, nous utilisons des unités de sous-mots (également appelées "wordpieces"). WordPiece est un algorithme de tokénisation de sous-mots utilisé dans le Traitement automatique des langues (TAL) (Wu *et al.*, 2016). L'utilisation de WordPiece donne un bon équilibre entre la flexibilité des caractères uniques et l'efficacité des mots complets pour le décodage, et évite également le besoin de traitement spécial des mots inconnus.

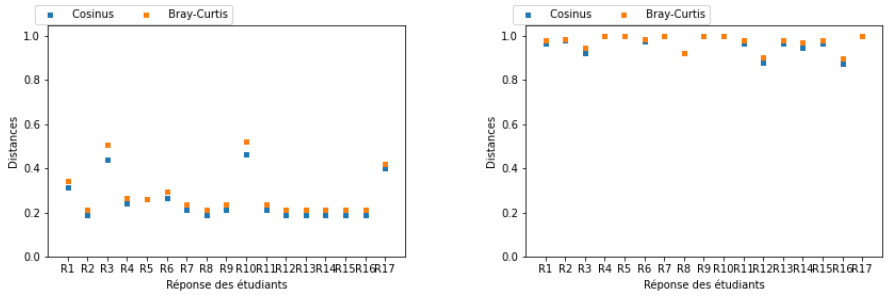
Vecteurs contextualisés SentenceBERT (`sbert`) : Les modèles *tranformers* pré-entrainés comme BERT (Bidirectional Encoder Representations from Transformers) permettent d'acquérir une connaissance plus fine de la langue. BERT est un modèle de représentation du langage basé sur une architecture à réseaux de neurones qui repose sur le principe des transformateurs bidirectionnels (Teissèdre *et al.*, 2020). En effet, l'entraînement bidirectionnel de modèle BERT donne une connaissance plus profonde du contexte de la phrase et du flux linguistique, comparé aux modèles de langue basés sur un contexte unidirectionnel comme dans les modèles RNN (réseaux de neurones récurrents) (Granet *et al.*, 2017). Les modèles à base de Transformers pré-entrainés du type BERT (Devlin *et al.*, 2018) et plus précisément BERT multilingue (MULTI-BERT) ont montré leur efficacité dans plusieurs tâches de Traitement automatique des langues (TAL) (Dor *et al.*, 2020). Sentence-BERT [1] (Reimers & Gurevych, 2019) est une adaptation du modèle BERT qui utilise des structures de réseau siamois et triplet (Chopra *et al.*, 2005) pour dériver des incorporations de phrases sémantiquement significatives. BERT est capable d'apprendre les contextes entre les mots d'un texte ainsi que les sous-mots de gauche à droite et de droite à gauche. Dérivé de (MULTI-BERT) et de Sentence-BERT, le modèle Sentence Multilingual que nous utilisons mesure la similarité textuelle sémantique entre des paires de phrases. Plus précisément, nous appliquons deux modèles pré-entraînés (`paraphrase-multilingual-MiniLM-L12-v2` [2] et `bert-base-multilingual-cased` [3]) pour le calcul de la similarité sémantique entre des paires de phrases. Le modèle `paraphrase-multilingual-MiniLM-L12-v2` [4] du package Sentence Transformer offre une bonne qualité et rapidité. Avant d'appliquer les deux modèles pré-entraînés, une étape de préparation de texte est nécessaire. Une représentation vectorielle correspondant aux différents symboles (tokens) des phrases d'entrée est faite. Ceci permet la comparaison entre phrases à l'aide de la similarité cosinus.

3.2 Mesures de similarité

Nous appliquons deux mesures de similarité sur les différentes représentations vectorielles des textes générées à partir de quatre différents types de caractéristiques : mots, n-grammes de caractères (avec et sans frontières de mots), word pieces et sentence embeddings. Pour ce faire, nous avons utilisé les métriques Cosinus et Bray-Curtis. La similarité Cosinus permet de déterminer la proximité entre deux vecteurs d'interprétation dans l'espace défini (Claveau, 2012).C'est une mesure classique largement utilisée dans la communauté du TAL bien que d'autres métriques comme la distance de Jaccard (Jaccard, 1901), de Dice (Dice, 1945) mais aussi Bray-Curtis (Bray & Curtis, 1957) puissent apporter de meilleurs résultats dans certains cas (Buscaldi *et al.*, 2020).

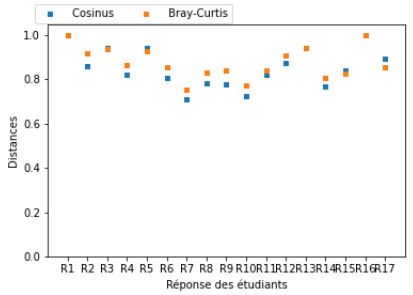
Les résultats pour la similarité Cosinus apparaissent souvent comme étant les meilleurs, néanmoins, comme le remarque (Koudoro-Parfait *et al.*, 2021) la mesure cosinus semble avoir tendance à sous-estimer la distance entre les vecteurs comparés et donc potentiellement à sur-estimer la similarité entre la réponse de l'étudiant et le corrigé. Ce qui amène l'idée de combiner différentes mesures de similarité pour assurer une certaine complémentarité.

1. https://www.sbert.net/
2. https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v
3. https://huggingface.co/bert-base-multilingual-cased
4. https://www.sbert.net/docs/pretrained_models.html

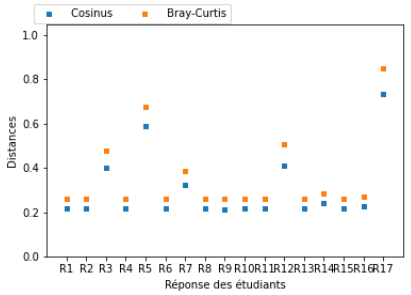


(a) Réponses à la question 1005

(b) Réponses à la question 1006



(c) Réponses à la question 1013



(d) Réponses à la question 1019

FIGURE 1 – Résultats obtenus en utilisant la similarité Cosinus et la similarité de Bray-Curtis au grain caractère, exemples extraits des matrices d'entraînement des modèles de prédiction de notes pour quatre questions différentes.

3.3 Protocole expérimental

Les scores de similarité ont été utilisé par la suite pour entraîner des modèles qui prédisent une note pour une question donnée. Pour cela, nous nous sommes basés sur une notation de prédiction par apprentissage. On parle de « prédiction par apprentissage » quand il s'agit de prédire une variable numérique (dans notre cas la note de l'étudiant·e), à partir d'une ou plusieurs autres variables numériques (les scores de similarité présentés sous forme de matrice de similarité). Nous utilisons trois méthodes de notation différentes : notation par régression, notation par classification et notation par réseau de neurones. Dans le cadre du développement de nos méthodes, nous les avons évaluées en utilisant les données de test du Deft 2021 (DEFT-TEST-2021) comme données de développement. Le Tableau 3 montre les résultats obtenus en termes de précision (P) et de score de corrélation de Spearman (r_s).

	M1	M2	M3
Précision (P)	0,642	**0,678**	0,653
Corrélation de Spearman (r_s)	0,633	0,543	**0,644**

TABLE 3 – Résultats de nos méthodes entrainées sur DEFT-TRAIN-2021 et évaluées DEFT-TEST-2021

Pour chaque méthode, nous avons donc testé différents types de vectorisation avec différentes mesures de similarité. Chaque représentation est testée avec différentes valeurs de N et différentes mesures de distances.

Représentation	Longueurs	Distances	# Dimensions
`char`			16
`char_wb`	[3 : 3], [3 : 4],[3 : 5] . . .[4 : 6]	Cosinus et Bray-Curtis	16
`WordPiece`			16

TABLE 4 – Les 48 dimensions utilisés pour la représentation de la similarité (nous y ajoutons deux représentations issues de `Sentencebert` pour aboutir à 50 dimensions)

Pour chaque réponse, nous obtenons un vecteur dont chaque dimension représente une distance mesurée sur une représentation particulière. La représentation finale est un vecteur de distances qui sera exploité par des méthodes de *machine learning*. Ce vecteur comporte 48 dimensions issues de représentations en chaînes de caractères (cf. Tableau 4) auxquelles nous ajoutons deux similarités obtenues par SentenceBert.

Notation par régression (M1) Il semble naturel de chercher, au-delà de trouver la bonne note, de minimiser l'écart avec la note attendue puisque toutes les erreurs ne se valent pas (il vaut mieux donner 0,5 à un étudiant qui mérite 0,75 plutôt que lui donner 0). Dans cette première méthode, nous avons dinc défini la tâche de notation comme une tâche de régression linéaire. Le classifieur est donc nourri Nous avons utilisé les traits suivants : `char`, `char_wb`, `WordPiece` et `sbert`. Nous avons alors obtenu une précision de 0,642 ainsi qu'une corrélation de Spearman de 0,633.

Notation par classification (M2) Afin de vérifier que notre hypothèse sur la pertinence de définir la tâche comme une tâche de régression, nous avons défini la tâche de notation comme une tâche de classification avec 3 classes (*0 , 0,5 , 1*). Nous avons utilisé un algorithme de régression logistique avec le *solver* Saga, ce qui donnait de meilleurs résultats que les autres algorithmes d'apprentissage que nous avons utilisé. Nous avons exploité les mêmes traits que pour notre première méthode, à ceci près qu'enlever les descripteurs de type `char` améliorait sensiblement les résultats. Nous avons alors obtenu une précision de 0,678 ainsi qu'une corrélation de Spearman de 0,543. Nous avons obtenu une meilleure précision (3,6 points) que pour la méthode 1, mais au détriment de la corrélation (-9 points). Si l'amélioration de la précision n'était pas attendue, en revanche la chute importante de la corrélation semblait accréditer la pertinence de traiter la tâche comme une question de régression.

Notation par réseau de neurones (M3) Dans cette méthode, nous avons fait recours à un modèle de réseau de neurones pour effectuer la notation. Le modèle est constitué de deux couches de neurones associées à une fonction d'activation de type sigmoïde. L'algorithme prend en entrée les mêmes traits de la première méthode et la note prédite en sortie. La précision du modèle est de 0,653 avec une corrélation de Spearman de 0,644.

4 Résultats et discussion

4.1 Analyse de nos résultats

La Figure 2 affiche les résultats de nos trois méthodes sur l'ensemble de données de test de la tâche 1, ainsi que les scores minimal (min), maximal (max), la moyenne et la médiane de l'ensemble des participants. Les résultats obtenus actuellement sont supérieurs à la moyenne et à la médiane des

résultats des autres équipes, ce qui laisse penser que notre méthode a un certain potentiel. Cependant, notre précision est assez nettement à celle de l'équipe arrivée en tête du défi. Notre approche exploite principalement la métrique cosinus, mais certaines expériences laissent penser que cette mesure sous-estimerait la distance entre deux groupes comparés. Nous envisageons qu'il serait intéressant d'appliquer d'autres mesures de distances à ces données afin d'obtenir des résultats plus fins.

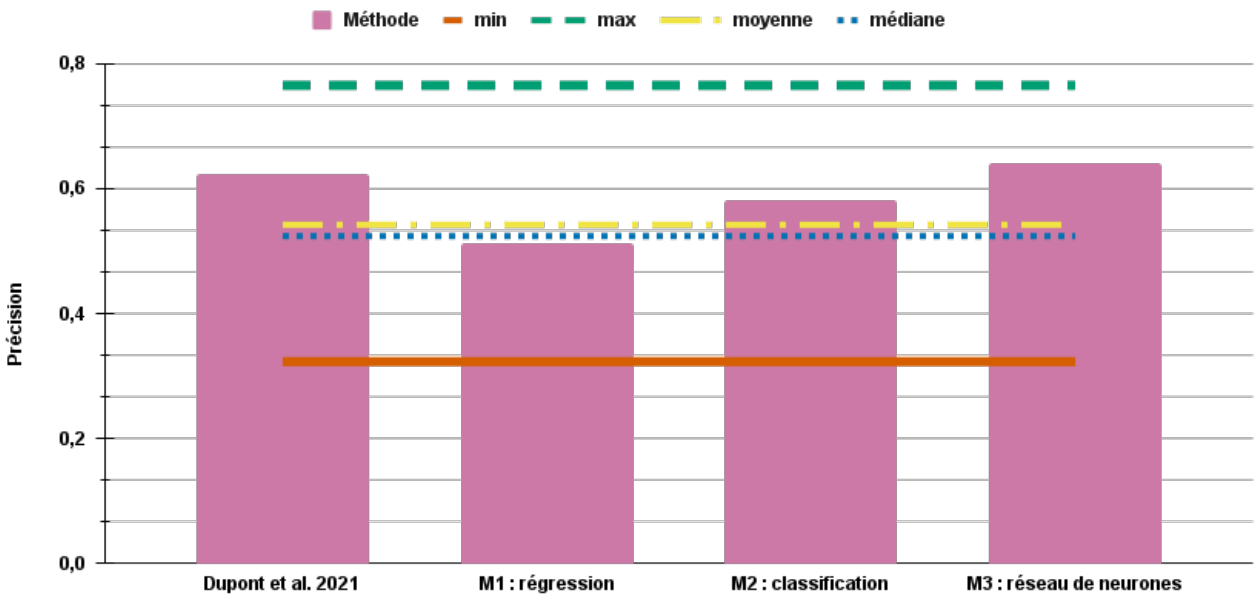


FIGURE 2 – Résultats des nos méthodes et statistiques générales

Comparativement à nos résultats de l'année précédente sur la tâche 2 (Dupont *et al.*, 2021), nous sommes passés, sur l'ensemble de test de 2021, d'une précision de $0,624$ à une précision de $0,678$ (M3), soit un gain de $5,4$ points de pourcentage. Nous avons donc pu améliorer notre précision de manière intéressante par rapport à notre dernière participation. En comparant les résultats, nous avons pu remarquer que nous avons principalement amélioré la notation des questions de code.

4.2 Discussion

Nous avons proposé ici une approche simple pour la correction de réponses d'étudiant·e·s étant donné une correction préalable. Si nous nous référons à notre précédente participation (Dupont *et al.*, 2021), notre score a augmenté de $5,4$ points sur l'ensemble de test de 2021. Néanmoins, les scores de ces deux éditions ne peuvent pas être comparés de manière aussi directe, étant donné que les ensembles de test utilisés sont différents. Nous avons pu améliorer nos performances depuis notre dernière contribution (Dupont *et al.*, 2021), même si ces améliorations demeurent marginales, surtout si l'on considère que cette dernière méthode était non supervisée. En perspective de ce travail, nous souhaiterions tester d'autres similarités que le cosinus, comme la distance de Jaccard ou l'indice de Dice par exemple, pour estimer la proximité. Nous visons aussi à utiliser d'autres modèles de sentence embedding comme le modèle : *Universal Sentence Embedding* (Cer *et al.*, 2018).

Remerciements

Ce travail a été partiellement soutenu par les projets ANNA (2019-1R40226) et TERMITRAD (2020-2019-8510010) financés par la Région Nouvelle-Aquitaine, France.

Références

BRAY J. R. & CURTIS J. T. (1957). An ordination of the upland forest communities of southern wisconsin. *Ecological monographs*, **27**(4), 326–349.

BUSCALDI D., FELHI G., GHOUL D., LE ROUX J., LEJEUNE G. & ZHANG X. (2020). Calcul de similarité entre phrases : quelles mesures et quels descripteurs ? In *6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Atelier DÉfi Fouille de Textes*, p. 14–25 : ATALA ; AFCP.

CARTIER E. (2018). Dynamique lexicale des langues : éléments théoriques, méthodes automatiques, expérimentations en français contemporain. *document inédit HDR*.

CER D., YANG Y., KONG S.-Y., HUA N., LIMTIACO N., JOHN R. S., CONSTANT N., GUAJARDO-CESPEDES M., YUAN S., TAR C., SUNG Y.-H., STROPE B. & KURZWEIL R. (2018). Universal sentence encoder. DOI : 10.48550/ARXIV.1803.11175.

CHOPRA S., HADSELL R. & LECUN Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, p. 539–546 : IEEE.

CLAVEAU V. (2012). Vectorisation, okapi et calcul de similarité pour le tal : pour oublier enfin le tf-idf (vectorization, okapi and computing similarity for nlp : Say goodbye to tf-idf)[in french]. In *Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 2 : TALN*, p. 85–98.

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*.

DICE L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, **26**(3), 297–302.

DOR L. E., HALFON A., GERA A., SHNARCH E., DANKIN L., CHOSHEN L., DANILEVSKY M., AHARONOV R., KATZ Y. & SLONIM N. (2020). Active learning for bert : an empirical study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 7949–7962.

DUPONT Y., GONZÁLEZ-GALLARDO C.-E., LEJEUNE G., MILLOUR A. & TANGUY J.-B. (2021). Queer@deft2021 : Identification du profil clinique de patients et notation automatique de copies d'étudiants. In *Actes de l'atelier Défi Fouille de Textes@TALN 2020 Classification de cas cliniques et correction automatique de copies d'étudiants. Atelier DÉfi Fouille de Textes*, p. 95–107, Lille, France : Association pour le Traitement Automatique des Langues. QUEER@DEFT2021 : Patients Clinical Profile Identification and Automatic Student Grading.

GALHARDI L. B., DE MATTOS SENEFONTE H. C., DE SOUZA R. C. T. & BRANCHER J. D. (2018). Exploring distinct features for automatic short answer grading. In *Anais do XV Encontro Nacional de Inteligência Artificial e Computacional*, p. 1–12 : SBC.

GRANET A., MORIN E., MOUCHÈRE H., QUINIOU S. & VIARD-GAUDIN C. (2017). Étude préliminaire de reconnaissance d'écriture sur des documents historiques. In *Rencontre des Jeunes Chercheurs en Recherche d'Information (RJCRI)*.

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 l'atelier Défi Fouille de Textes@TALN 2020 Classification de cas cliniques et correction automatique de copies d'étudiants. Atelier DÉfi Fouille de Textes*, p. 1–13, Lille, France : Association pour le Traitement Automatique des Langues. Clinical cases classification and automatic evaluation of student answers : Presentation of the DEFT 2021 Challenge.

JACCARD P. (1901). Distribution de la flore alpine dans le bassin des dranses et dans quelques régions voisines. *Bulletin de la Société vaudoise des sciences naturelles*, **37**, 241–272.

JALAM R. (2003). Apprentissage automatique et catégorisation de textes multilingues. *PhD Tesis, Université Lumiere Lyon*, **2**.

KOUDORO-PARFAIT C., LEJEUNE G. & ROE G. (2021). Spatial named entity recognition in literary texts : What is the influence of OCR noise ? In L. MONCLA, C. BRANDO & K. MCDONOUGH, Éds., *GeoHumanities@SIGSPATIAL 2021 : Proceedings of the 5th ACM SIGSPATIAL International Workshop on Geospatial Humanities, Beijing, China, November 2 - 5, 2021*, p. 13–21 : ACM. DOI : 10.1145/3486187.3490206.

KUDO T. (2018). Subword regularization : Improving neural network translation models with multiple subword candidates. *arXiv preprint arXiv :1804.10959*.

LABEAU M. & ALLAUZEN A. (2017). Character and subword-based word representation for neural language modeling prediction. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, p. 1–13, Copenhagen, Denmark : Association for Computational Linguistics. DOI : 10.18653/v1/W17-4101.

LAMPLE G., CONNEAU A., RANZATO M., DENOYER L. & JÉGOU H. (2018). Word translation without parallel data. In *International Conference on Learning Representations*.

LEJEUNE G. & CARTIER E. (2017). Character based pattern mining for neology detection. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, p. 25–30, Copenhagen, Denmark : Association for Computational Linguistics. DOI : 10.18653/v1/W17-4103.

MARTIN L., MULLER B., SUÁREZ P. J. O., DUPONT Y., ROMARY L., DE LA CLERGERIE É. V., SEDDAH D. & SAGOT B. (2019). Camembert : a tasty french language model. *arXiv preprint arXiv :1911.03894*.

POULAIN T. & CONNES V. (2021). Deft 2021 : évaluation automatique de réponses courtes, une approche basée sur la sélection de traits lexicaux et augmentation de données. In *Actes de l'atelier Défi Fouille de Textes@TALN 2020 Classification de cas cliniques et correction automatique de copies d'étudiants. Atelier DÉfi Fouille de Textes*, p. 31–40, Lille, France : Association pour le Traitement Automatique des Langues. DEFT 2021 : Automatic short answer grading, a lexical features selection and data augmentation based approach.

REIMERS N. & GUREVYCH I. (2019). Sentence-bert : Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, p. 671–688 : Association for Computational Linguistics.

SUIGNARD P., BENAMAR A., MESSOUS N., CHRISTOPHE C., JUBAULT M. & BOTHUA M. (2021). Participation d'edf r&d à deft 2021. In *Actes de látelier Défi Fouille de Textes@TALN 2020 Classification de cas cliniques et correction automatique de copies détudiants. Atelier DÉfi Fouille*

de Textes, p. 72–81, Lille, France : Association pour le Traitement Automatique des Langues. EDF R&D Participation to DEFT 2021.

SÜZEN N., GORBAN A. N., LEVESLEY J. & MIRKES E. M. (2020). Automatic short answer grading and feedback using text mining methods. *Procedia Computer Science*, **169**, 726–743.

TEISSÈDRE C., BELKACEM T. & ARENS M. (2020). Similarité sémantique entre phrases : apprentissage par transfert interlingue. In *6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Atelier DÉfi Fouille de Textes*, p. 97–107 : ATALA ; AFCP.

TEISSÈDRE C., BELKACEM T. & ARENS M. (2020). Similarité sémantique entre phrases : apprentissage par transfert interlingue.

VERGNE J. (2003). Un outil d'extraction terminologique endogène et multilingue. In *Actes de TALN 2003*, volume tome 2, p. 139–148, France. HAL : hal-00256093.

WANG X., LIU X. & YUE Y. (2021). Mesure de similarité textuelle pour l'évaluation automatique de copies d'étudiants. In *Actes de l'atelier Défi Fouille de Textes@TALN 2020 Classification de cas cliniques et correction automatique de copies d'étudiants. Atelier DÉfi Fouille de Textes*, p. 63–71, Lille, France : Association pour le Traitement Automatique des Langues. Textual similarity measurement for automatic evaluation of students' answers.

WU Y., SCHUSTER M., CHEN Z., LE Q. V., NOROUZI M., MACHEREY W., KRIKUN M., CAO Y., GAO Q., MACHEREY K. *et al.* (2016). Google's neural machine translation system : Bridging the gap between human and machine translation. *arXiv preprint arXiv :1609.08144*.