

CONSERVATOIRE NATIONAL DES ARTS ET MÉTIERS

le cnam

FOUILLE DE DONNÉES MASSIVES

RCP 216

Projet

Benoît MAYER

12 avril 2021

Sommaire

1	Introduction	3
2	Analyse exploratoire	4
2.1	Présentation des données	4
2.2	Visualisation des données	5
3	Pré-traitements	6
3.1	Calcul de caractéristiques	6
3.2	Correction grammaticale et orthographique	6
3.3	Nettoyage des données	7
3.3.1	Introduction	7
3.3.2	Assemblage	7
3.3.3	Transformation en jetons	7
3.3.4	Normalisation	8
3.3.5	Lemmatisation	8
3.3.6	Suppression des mots vides	8
3.4	Représentations vectorielles	8
3.5	Vérification de la multicolinéarité	10
3.5.1	Corrélation	10
3.5.2	Facteur d'Inflation de la Variance (VIF)	10
3.5.3	Analyse	10
3.6	Conclusion	10
4	Modélisation	11
4.1	Introduction	11
4.2	Échantillonnage	11
4.3	Régression linéaire multiple	12
4.3.1	Régression linéaire multiple globale	12
4.3.2	Régression linéaire multiple par sujet	12
4.4	Régression à vecteurs de support	13
4.5	Réseaux de neurones	13
4.6	Synthèse	14
5	Analyse des résultats	15
5.1	Comparaison par rapport aux correcteurs humains	15
5.2	Génération automatique d'une rédaction	15
6	Conclusion	17

1. Introduction

La correction automatique par ordinateur de rédactions d'étudiants est un sujet séduisant qui a dû faire rêver beaucoup de professeurs face à une pile de copies à corriger ! Dès les années 60, Ellis Batten Page, ancien professeur d'anglais au lycée, a théorisé un projet appelé *Project Essay Grade (PEG®)*[5] qui continue d'être développé et qui est à présent utilisé dans plus d'un millier d'écoles américaines [3].

Les avancées technologiques récentes et la popularité de nouvelles techniques en traitement automatique des langues ont permis de rendre ces systèmes toujours plus efficace. En 2012, une compétition organisée sur *Kaggle* met en lumière ce champ de l'intelligence artificielle : le challenge *Automated Essay Scoring* sponsorisé par la *Hewlett Foundation*[7]. Plus de 200 équipes y ont participé, et un concours parallèle a été organisé entre 9 éditeurs d'outils spécialisés dans ce domaine. PEG®, le programme précédemment cité, y a obtenu le meilleur score.

L'objectif de cette compétition était de déterminer si une correction par ordinateur est à présent aussi fiable que celle d'un humain, selon les propos d'un des organisateurs[1]. Nous allons essayer de répondre à notre tour à cette question en appliquant au jeu de données, disponible sur *Kaggle*, certains des modèles étudiés dans les unités d'enseignement RCP 216 - Ingénierie de la fouille et de la visualisation de données massives et STA 211 - Entreposage et fouille de données.

Nous allons tout d'abord procéder à une exploration des données avant d'effectuer des traitements préalables à une analyse et de créer des représentations vectorielles numériques de chacun des textes en décrivant le plus précisément possible les modèles utilisées, *Word2Vec* et *Glove*. Nous appliquerons ensuite des méthodes usuelles d'apprentissage statistique, des régressions linéaires, des régressions à vecteurs de support et des réseaux de neurones, et nous commenterons les résultats obtenus.

2. Analyse exploratoire

2.1. Présentation des données

Les données utilisées pour cette étude sont issues du concours *Automated Essay Scoring* organisé sur Kaggle^[7]. Le jeu de données comporte 12 978 rédactions d'étudiants américains, traitant de 8 sujets différents :

- Sujet 1 : L'impact de l'utilisation des ordinateurs sur la société
- Sujet 2 : La censure dans les bibliothèques
- Sujet 3 : Les aventures d'un cycliste dans le désert
- Sujet 4 : Les mémoires d'une jeune fille émigrée du Viet Nam
- Sujet 5 : L'arrivée aux États-Unis depuis Cuba de Narciso Rodriguez
- Sujet 6 : Les difficultés d'atterrissement des dirigeables sur l'Empire State Building
- Sujet 7 : La patience
- Sujet 8 : Les bénéfices du rire

Les textes sont anonymisés en utilisant une méthode appelée *Named Entity Recognizer*^[6] qui remplace chaque nom propre, chiffre ou date par un identificateur au format @TYPE1.

Quelques exemples donnés dans la description du challenge :

- "I attend Springfield School..." -> "...I attend @ORGANIZATION1"
- "once my family took my on a trip to Springfield." -> "once my family took me on a trip to @LOCATION1"
- "John Doe is a person, and so is Jane Doe. But if I talk about Mr. Doe, I can't tell that's the same person." -> "...@PERSON1 is a person, and so is @PERSON2. But if you talk about @PERSON3, I can't tell that's the same person."
- "...my phone number is 555-2106" -> "...my phone number is @NUM1"

Le fichier de données est composé de 28 colonnes :

- *essay_id* : la clé primaire de la table
- *essay_set* : le sujet de la rédaction
- *essay* : le contenu de la rédaction, codé en Windows-1252
- 25 colonnes de scores, dont 22 sont disponibles pour un nombre très restreint de rédactions (maximum 2292/12978). Les trois restantes sont données pour toutes les rédactions sauf pour deux que nous excluons :
 - *rater1_domain1*
 - *rater2_domain1*
 - *domain1_score* : à l'exception du sujet 8, cette colonne est, en fonction des sujets, la moyenne ou la somme des deux précédentes.

Ces trois derniers scores sont ceux que nous conservons pour notre modélisation. Le score global *domain1_score* sera notre variable objectif, et *rater1_domain1* ainsi que *rater2_domain1* des variables de comparaisons utilisées dans la partie 5.1.

Le tableau ci-dessous résume la distribution des valeurs de *domain1_score* :

Sujet	Min	Max	Nb	Unique	Moyenne	Écart-Type
1	2	12	1783	11	8.53	1.54
2	1	6	1800	6	3.42	0.77
3	0	3	1726	4	1.85	0.82
4	0	3	1771	4	1.43	0.94
5	0	4	1805	5	2.41	0.97
6	0	4	1800	5	2.72	0.97
7	2	24	1569	23	16.06	4.59
8	10	60	723	34	36.95	5.75

Nous constatons que les modes de notation sont très variables. Les notes du sujet 3 vont de 0 à 3, tandis que le sujet 8 utilise une échelle de notation de 10 à 60. Logiquement les notes moyennes et les écarts types sont très différents les uns des autres.

Le sujet original de Kaggle traitait ce problème comme une classification, en utilisant toutefois comme mesure d'évaluation le Kappa de Cohen à pondération quadratique afin de pénaliser l'ampleur des désaccords en fonction de l'échelle ordinaire des scores.

Pour cette étude, nous choisissons de traiter ce sujet comme un problème de régression. Nous centrons et réduisons les scores pour pouvoir les comparer entre les sujets, et nous utilisons l'erreur quadratique moyenne comme fonction objectif :

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

où :

- \hat{Y}_i est le score prédit par notre modèle
- Y_i est le score vraiment obtenu par la rédaction i

2.2. Visualisation des données



FIGURE 1 – Visualisation des mots les plus fréquents par sujet

Dans les nuages de mots ci-dessus, la taille des mots représente leur fréquence dans le corpus de texte, une fois les mots vides supprimés.

Nous constatons que les mots les plus fréquemment utilisés correspondent au champ lexical de chacun des sujets et sont très différents d'un sujet à l'autre. Il faudrait donc traiter les rédactions sujet par sujet plutôt que dans leur globalité, c'est ce que nous essaierons de faire dans la partie 4.3.2.

3. Pré-traitements

3.1. Calcul de caractéristiques

Les identificateurs créés par la méthode d'anonymisation sont enlevés faute de pouvoir en obtenir une représentation vectorielle.

Avant de procéder à cette suppression, nous ajoutons des colonnes de caractéristiques comptant le nombre d'identificateurs par types, pour chacune des rédactions.

Type	Identificateurs	Colonne
Nom d'organisation	@ORGANIZATION	nb_organization
Mot en majuscule	@CAPS	nb_caps
Nom de personne	@PERSON et @DR	nb_person
Nom de lieu	@LOCATION, @CITY et @STATE	nb_location
Nom de devise	@MONEY	nb_money
Durée temporelle	@TIME	nb_time
Date	@DATE et @MONTH	nb_date
Chiffre	@PERCENT et @NUM	nb_percent

TABLE 1 – Colonnes de compte des identificateurs

Nous créons également une colonne comptant le nombre de mots de chaque rédaction, *nb_words*.

3.2. Correction grammaticale et orthographique

Les rédactions écrites par les étudiants présentent souvent des erreurs de grammaire ou d'orthographe. Nous utilisons donc un outil de vérification pour les identifier, les compter et les corriger.

L'intérêt est double :

- l'orthographe et la grammaire peuvent être prises en compte dans la notation des rédactions, même si ce n'est expressément indiqué que dans les guides de correction des sujets 2, 7 et 8,
- une mauvaise orthographe peut gêner le travail d'analyse lexicale entrepris dans les parties suivantes.

La première rédaction (*essay_id 1*) est la suivante :

Dear local newspaper, I think effects computers have on people are great learning skills/affects because they give us time to chat with friends/new people, helps us learn about the globe(astronomy) and keeps us out of troble! Thing about! Dont you think so? How would you feel if your teenager is always on the phone with friends! Do you ever time to chat with your friends or buisness partner about things. Well now - there's a new way to chat the computer, theirs plenty of sites on the internet to do so : @ORGANIZATION1, @ORGANIZATION2, @CAPS1, facebook, myspace ect. Just think now while your setting up meeting with your boss on the computer, your teenager is having fun on the phone not rushing to get off cause you want to use it. How did you learn about other countrys/states outside of yours? Well I have by computer/internet, it's a new way to learn about what going on in our time! You might think your child spends a lot of time on the computer, but ask them so question about the economy, sea floor spreading or even about the @DATE1's you'll be surprise at how much he/she knows. Believe it or not the computer is much interesting then in class all day reading out of books. If your child is home on your computer or at a local library, it's better than being out with friends being fresh, or being perpressured to doing something they know isnt right. You might not know where your child is, @CAPS2 forbidde in a hospital bed because of a drive-by. Rather than your child on the computer learning, chatting or just playing games, safe and sound in your home or community place. Now I hope you have reached a point to understand and agree with me, because computers can have great effects on you or child because it gives us time to chat with friends/new people, helps us learn about the globe and believe or not keeps us out of troble. Thank you for listening.

Onze fautes y sont identifiées :

Erreur trouvée	Correction apportée	Commentaire
troble	trouble	Mauvaise orthographe
Dont	Don't	Pas d'apostrophe pour la contraction
buisness	business	Mauvaise orthographe
facebook	Facebook	Pas de majuscule à un nom propre
myspace	MySpace	Pas de majuscule à un nom propre
ect.	act.	Mauvaise orthographe
country's	countries	Forme plurielle incorrecte
perpressured	per pressured	Mauvaise orthographe
isnt	isn't	Pas d'apostrophe pour la contraction
fobidde	forbidden	Mauvaise orthographe
troble	trouble	Mauvaise orthographe

TABLE 2 – Corrections apportées à la première rédaction

Nous remarquons que toutes les fautes identifiées sont légitimes mais que pour deux d'entre elles la correction apportée est inadéquate :

- *ect.* : la forme correcte est *etc.*, l'abréviation de *Et cetera*.
- *perpressured* : l'étudiant entendait sûrement écrire *peer pressured*, la pression sociale.

Nous constatons aussi que certaines fautes ont été oubliées, par exemple :

- *theirs plenty of sites* au lieu de *there are plenty of sites*
- *If your child is home* au lieu de *If your child is at home*

Néanmoins, la pertinence des erreurs trouvées sur cet échantillon et sur les autres qui ont été testés est suffisamment bonne pour permettre d'améliorer le jeu de données.

3.3. Nettoyage des données

3.3.1 Introduction

Pour obtenir de chacun des textes une représentation vectorielle numérique qui soit facilement exploitable dans des modèles d'apprentissage statistique, nous allons tout d'abord effectuer différentes étapes de nettoyage des données :

- Assemblage 3.3.2
- Transformation en jetons 3.3.3
- Normalisation 3.3.4
- Transformation en lemmes 3.3.5
- Suppression des mots vides 3.3.6

3.3.2 Assemblage

Les caractères d'espacement inutiles pour l'analyse du texte, tels que les retours à la ligne, sont supprimés et chaque texte est transformé en un document structuré prêt pour les traitements ultérieurs.

3.3.3 Transformation en jetons

Chaque texte est découpé en jetons, c'est à dire ici en mots. La division est effectuée en fonction des caractères d'espacement.

Exemple (tronqué) du code brut de la structure obtenue :

```
[Row(annotatorType='token', begin=0, end=3, result='Dear', metadata='sentence' : '0', embeddings=[]),
Row(annotatorType='token', begin=5, end=9, result='local', metadata='sentence' : '0', embeddings=[]),
Row(annotatorType='token', begin=11, end=19, result='newspaper', metadata='sentence' : '0', embeddings=[]),
Row(annotatorType='token', begin=20, end=20, result='.', metadata='sentence' : '0', embeddings=[]),
...]
```

```
Row(annotatorType='token', begin=1816, end=1818, result='you', metadata='sentence' : '0', embeddings=[]), Row(annotatorType='token', begin=1820, end=1822, result='for', metadata='sentence' : '0', embeddings=[]), Row(annotatorType='token', begin=1824, end=1832, result='listening', metadata='sentence' : '0', embeddings=[]), Row(annotatorType='token', begin=1833, end=1833, result='.', metadata='sentence' : '0', embeddings=[])]
```

3.3.4 Normalisation

Tous les caractères autres que les lettres sont supprimés et les lettres majuscules sont transformées en minuscules.

3.3.5 Lemmatisation

Chaque mot est remplacé par sa racine. Un verbe est par exemple pris à l'infinitif, *forbidden* devient *forbid*, et chaque mot au singulier, *country* pour *countries*.

3.3.6 Suppression des mots vides

Les mots très fréquents mais peu discriminants comme les articles sont éliminés. La liste exacte des mots supprimés est la suivante :

i, me, my, myself, we, our, ours, ourselves, you, your, yours, yourself, yourselves, he, him, his, himself, she, her, hers, herself, it, its, itself, they, them, their, theirs, themselves, what, which, who, whom, this, that, these, those, am, is, are, was, were, be, been, being, have, has, had, having, do, does, did, doing, a, an, the, and, but, if, or, because, as, until, while, of, at, by, for, with, about, against, between, into, through, during, before, after, above, below, to, from, up, down, in, out, on, off, over, under, again, further, then, once, here, there, when, where, why, how, all, any, both, each, few, more, most, other, some, such, no, nor, not, only, own, same, so, than, too, very, s, t, can, will, just, don, should, now, i'll, you'll, he'll, she'll, we'll, they'll, i'd, you'd, he'd, she'd, we'd, they'd, i'm, you're, he's, she's, it's, we're, they're, i've, we've, you've, they've, isn't, aren't, wasn't, weren't, haven't, hasn't, hadn't, don't, doesn't, didn't, won't, wouldn't, shan't, shouldn't, mustn't, can't, couldn't, cannot, could, here's, how's, let's, ought, that's, there's, what's, when's, where's, who's, why's, would

À ce stade, l'exemple de la première rédaction est devenu :

dear, local, newspaper, think, effect, computer, people, great, learn, skill, affect, give, time, chat, friend, new, people, help, learn, globe, astronomy, keep, trouble, thing, dont, think, feel, teenager, always, phone, friend, ever, time, chat, friend, business, partner, thing, well, new, way, chat, computer, plenty, site, internet, facebook, myspace, act, think, set, meet, boss, computer, teenager, fun, phone, rush, get, cause, want, use, learn, country, state, outside, well, computer, internet, new, way, learn, go, time, might, think, child, spend, lot, time, computer, ask, question, economy, sea, floor, spread, even, youll, surprise, much, know, believe, computer, much, interest, class, day, read, book, child, home, computer, local, library, well, friend, fresh, per, pressure, something, know, isnt, right, might, know, child, forbid, hospital, bed, driveby, rather, child, computer, learn, chat, play, game, safe, sound, home, community, place, hope, reach, point, understand, agree, computer, great, effect, child, give, time, chat, friend, new, people, help, learn, globe, believe, keep, trouble, thank, listen

3.4. Représentations vectorielles

Le but de cette étape fondamentale est de transformer les listes de lemmes, obtenues précédemment, en une représentation sous forme de vecteurs numériques, représentation facilement exploitable par des modèles d'apprentissage statistique.

Le choix a été fait d'utiliser des vecteurs de dimension 300 obtenus grâce à deux méthodes différentes : un modèle *Word2Vec* entraîné sur le jeu de données étudié, et un modèle *GloVe* déjà pré-entraîné.

Nous obtenons une vectorisation des mots, que nous sommes pour en obtenir une au niveau des rédactions.

Vectorisation Word2Vec Le modèle *Word2Vec* (méthode Skip-gram) correspond à un réseau de neurones à trois couches, ayant :

- En couche d'entrée, un vecteur en *encodage 1 parmi n (one-hot)*, de même taille que le nombre total de mots différents dans le dictionnaire, ou au moins dans notre corpus de textes. Chaque dimension correspond à un mot, elles valent donc toutes zéro sauf celle qui correspond au mot considéré.
 - Une couche cachée ayant pour nombre de neurones la valeur de la dimension de la représentation vectorielle que nous recherchons.
 - En couche de sortie, un autre vecteur dictionnaire, de même taille que le vecteur d'entrée, où à chaque dimension est affectée une probabilité : celle que le mot correspondant à la dimension soit dans le voisinage du mot d'entrée.
- La fonction d'activation utilisée pour la couche de sortie est la fonction Softmax et il n'y pas de fonction d'activation pour la couche cachée, le modèle effectue donc simplement un produit matriciel :

$$y = \text{softmax}(\mathbf{W}_2^\top \mathbf{W}_1^\top x)$$

où :

- x est le vecteur d'entrée de taille T , le nombre de mots de notre corpus, où un élément correspondant au mot considéré vaut 1 et tous les autres 0
- y est le vecteur de sortie, décrivant la probabilité que chacun des mots soit dans le voisinage du mot d'entrée
- \mathbf{W}_1 et \mathbf{W}_2 , matrices de poids, respectivement de dimensions $T \times 300$ et $300 \times T$
- **softmax**, la fonction $\frac{e^{z_i}}{\sum_j e^{z_j}}$, qui permet de transformer les éléments d'un vecteur z en loi de probabilité, i.e. compris entre 0 et 1 et dont la somme vaut 1.

L'étape d'entraînement du modèle consiste à trouver les matrices \mathbf{W}_1 et \mathbf{W}_2 de façon à ce que le vecteur de probabilité y de trouver un mot dans le voisinage d'un terme corresponde au mieux à celui qui est observé.

Nous pouvons réécrire cette formule sous la forme suivante :

$$p(w_j|w_i) = \frac{\exp(u_{w_j}^\top v_{w_i})}{\sum_{w=1}^T \exp(u_w^\top v_i)}$$

où :

- u_{w_j} est la représentation de sortie du mot w_j
- v_{w_i} est la représentation d'entrée du mot w_i

Dans le cas de notre jeu de données, si nous cherchons par exemple les 5 plus proches voisins de mot *bike*, nous obtenons le résultat suivant :

Mot	Similarité
Mountain	0.679528
Ride	0.678682
Tire	0.658973
Wood	0.643557
Trail	0.640353

TABLE 3 – 5 plus proches voisins de *bike* pour le modèle *Word2Vec*

Nous constatons que les 5 mots les plus probables dans le voisinage du mot *bike* appartiennent tous au champ lexical du vélo tout-terrain (*mountain bike* en anglais), ce qui est l'objet du sujet 3.

Vectorisation Glove Le modèle *GloVe* (*Global Vectors for Word Representation*) utilise une méthode différente pour obtenir une vectorisation similaire.

Une matrice de cooccurrence des mots \mathbf{X} est tout d'abord établie sur l'ensemble du corpus de texte. Cette matrice est prise en compte dans le modèle précédent pour aboutir à la fonction objectif suivante :

$$\bar{J} = \sum_{i=1}^T \sum_{j=1}^T f(\mathbf{x}_{w_i w_j})(u_{w_j}^\top v_{w_i} - \log \mathbf{x}_{w_i w_j})^2$$

où f est une fonction de pondération pour limiter l'impact sur le modèle des cooccurrences trop fréquentes.

Plutôt qu'entraîner ce modèle sur notre jeu de données, nous utilisons des représentations vectorielles issus d'un entraînement sur un ensemble de textes très large : tous les articles de *Wikipedia* et quatre ans de fils de presse issus des fichiers *Gigaword*.

En cherchant à nouveau les 5 mots les plus proches de *bike*, nous trouvons à présent des synonymes plus généraux :

Mot	Similarité
Bicycle	0.925
Rides	0.871
Bikes	0.847
Ride	0.840
Cart	0.796

TABLE 4 – 5 plus proches voisins de *bike* pour le modèle *GloVe*

3.5. Vérification de la multicolinéarité

3.5.1 Corrélation

En calculant une matrice de corrélation pour les caractéristiques obtenues, nous constatons que les coefficients de corrélation supérieurs en valeur absolue à 0.7 représentent une proportion élevée de l'ensemble :

- 4.42% pour la vectorisation *Word2Vec*,
- 9.37% pour la vectorisation *GloVe*.

3.5.2 Facteur d'Inflation de la Variance (VIF)

Les caractéristiques issues des vectorisations *Word2Vec* et *GloVe* ont toutes un facteur d'inflation de la variance (VIF) supérieur à 5.

3.5.3 Analyse

De nombreuses variables sont corrélées entre elles et ont un facteur d'inflation de la variance (VIF) supérieur à 5, nous sommes donc dans un contexte de forte multicolinéarité. Nous ne choisirons donc que des modèles qui permettent de pallier à cette situation.

3.6. Conclusion

Au terme de toutes ces étapes, nous obtenons un fichier de 12 976 lignes et de 311 colonnes numériques que nous pouvons à présent facilement utiliser dans des modèles d'apprentissage statistique.

4. Modélisation

4.1. Introduction

Dans cette partie, nous allons appliquer trois types de modèles sur les données précédemment obtenues :

- Des régressions linéaires multiples [4.3](#),
- Des régressions à vecteurs de support [4.4](#),
- Des réseaux de neurones [4.5](#).

4.2. Échantillonnage

Avant l'échantillonnage, nous découpons le jeu de données par sujet et par note obtenue en groupes de taille égale :

Note Sujet \	Très mauvaise	Mauvaise	Moyenne	Bonne	Très bonne	Toutes
1	356	357	356	357	357	1783
2	359	360	360	360	360	1799
3	345	345	345	345	346	1726
4	354	354	354	354	355	1771
5	361	361	361	361	361	1805
6	360	360	360	360	360	1800
7	313	314	314	314	314	1569
8	144	145	144	145	145	723
Tous	2592	2596	2594	2596	2598	12976

TABLE 5 – Comptes des rédactions par sujets et par groupes de notes

Note Sujet \	Très mauvaise	Mauvaise	Moyenne	Bonne	Très bonne	Toutes
1	-1.327	-0.343	-0.175	0.516	1.326	0.0
2	-1.259	-0.537	-0.035	0.754	1.073	-0.0
3	-1.179	-0.884	0.186	0.460	1.413	0.0
4	-1.396	-0.459	-0.116	0.605	1.363	-0.0
5	-1.420	-0.421	-0.113	0.609	1.345	-0.0
6	-1.472	-0.444	0.288	0.309	1.319	0.0
7	-1.480	-0.503	0.067	0.542	1.369	0.0
8	-1.382	-0.536	0.041	0.531	1.336	-0.0
Tous	-1.362	-0.513	0.015	0.542	1.315	-0.0

TABLE 6 – Notes moyennes des rédactions par sujets et par groupes de notes

Nous procédons ensuite à un échantillonage de 70% de l'ensemble de données pour constituer notre jeu d'entraînement et des 30% restants pour effectuer les tests.

Nous vérifions que chacune des catégories définies précédemment a toujours une pondération équivalente dans cet ensemble et que notre échantillonage est donc suffisamment équilibré :

Note Sujet \ Note	Très mauvaise	Mauvaise	Moyenne	Bonne	Très bonne	Toutes
1	249	272	255	238	259	1273
2	249	267	248	247	260	1271
3	245	242	248	241	252	1228
4	246	250	253	242	238	1229
5	243	251	250	265	261	1270
6	243	251	255	257	242	1248
7	208	209	207	214	223	1061
8	108	102	96	98	99	503
Tous	1791	1844	1812	1802	1834	9083

TABLE 7 – Comptes des rédactions par sujets et par groupes de notes dans l'échantillon d'entraînement

4.3. Régression linéaire multiple

4.3.1 Régression linéaire multiple globale

Comme montré en 3.5.3, les données présentent une forte multicolinéarité. Pour pallier à cette situation, nous utilisons une régression Ridge.

En effectuant cette régression linéaire sur l'échantillon d'entraînement, nous obtenons les résultats suivants :

Vectorisation	MSE entraînement	MSE test
Word2Vec	0.59	0.60
GloVe	0.49	0.57

TABLE 8 – Résultats des régressions linéaires multiples

L'erreur quadratique moyenne est presque identique pour les deux modèles de vectorisation, avec un léger avantage pour *GloVe*

4.3.2 Régression linéaire multiple par sujet

En décomposant l'échantillon d'entraînement par sujet et en effectuant une régression linéaire sur chaque nouvel ensemble, nous obtenons les résultats suivants :

Sujet	Vectorisation	MSE entraînement	MSE test
1	Word2Vec	0.30	0.29
2	Word2Vec	0.45	0.48
3	Word2Vec	0.46	0.46
4	Word2Vec	0.31	0.36
5	Word2Vec	0.27	0.27
6	Word2Vec	0.28	0.32
7	Word2Vec	0.43	0.42
8	Word2Vec	0.49	0.53
1	GloVe	0.25	0.41
2	GloVe	0.34	0.60
3	GloVe	0.35	0.65
4	GloVe	0.22	0.48
5	GloVe	0.19	0.38
6	GloVe	0.21	0.42
7	GloVe	0.31	0.60
8	GloVe	0.17	1.35

TABLE 9 – Résultats des régressions linéaires multiples par sujet

Avec la vectorisation *Word2Vec*, nous constatons que nous obtenons pour tous les sujets des erreurs quadratiques moyennes égales ou inférieures à celles que nous avions en considérant l'échantillon d'entraînement dans sa globalité. La moyenne des erreurs est de 0.39, très inférieure à celle obtenue précédemment, 0.60.

À l'inverse les résultats sont moins bons pour la vectorisation *GloVe*, 0.61 en moyenne contre 0.57 précédemment. L'erreur est très élevée sur l'échantillon de test surtout pour le sujet 8, alors même qu'elle est très faible sur l'échantillon d'entraînement. Il est probable qu'il y ait un surentraînement, d'autant plus que nous disposons de moitié moins de rédactions pour ce sujet que pour les autres.

4.4. Régression à vecteurs de support

Les classificateurs à vaste marge ou machines à vecteurs de support, *Support Vector Machine (SVM)* en anglais peuvent être également utilisés en régression. Ces modèles sont alors appelés *Support Vector Regression (SVR)*, ce que nous traduisons en régression à vecteurs de support.

Ces méthodes sont réputées pour rester performantes en présence de forte multicolinéarité [4], comme ici.

Nous utilisons uniquement un noyau linéaire, en faisant varier la taille de la marge.

Vectorisation	C	MSE entraînement	MSE test
Word2Vec	0.00001	0.92	0.93
Word2Vec	0.0001	0.89	0.89
Word2Vec	0.001	0.80	0.81
Word2Vec	0.01	0.70	0.71
Word2Vec	0.1	0.63	0.63
Word2Vec	10	1.61	1.66
Word2Vec	100	4.90	5.08
Word2Vec	1000	1.75	1.70
Word2Vec	10000	1.91	1.98
GloVe	0.00001	0.69	0.74
GloVe	0.0001	0.58	0.63
GloVe	0.001	0.55	0.61
GloVe	0.01	0.52	0.58
GloVe	0.1	0.52	0.59
GloVe	10	2.38	2.46
GloVe	100	1.00	1.05
GloVe	1000	2.21	2.24
GloVe	10000	4.26	4.44

TABLE 10 – Résultats des régressions à vecteurs de support en fonction de C

Nous constatons que nous obtenons pour les deux vectorisations les meilleurs résultats pour une marge de taille moyenne ($C = 0.1$).

4.5. Réseaux de neurones

Nous utilisons une structure extrêmement simple de réseaux de neurones : deux couches entièrement connectées dont nous allons faire varier la taille.

Couche 1 \ Couche 2	10	25	50	100	200	300
10	0.65	0.65	0.67	0.64	0.65	0.63
25	0.96	0.65	0.64	0.67	0.64	0.64
50	0.87	0.72	0.65	0.66	0.64	0.64
100	0.67	0.66	0.68	0.63	0.66	0.67
200	0.65	0.65	0.69	0.64	0.64	0.65
300	0.64	0.65	0.64	0.64	0.63	0.63

TABLE 11 – MSE sur l'échantillon test des réseaux de neurones en fonction de la taille des couches 1 & 2 pour la vectorisation *Word2Vec*

Couche 1 \ Couche 2	10	25	50	100	200	300
10	0.59	0.66	0.65	0.59	0.59	0.66
25	0.78	0.74	0.59	0.60	0.62	0.59
50	0.84	0.64	0.60	0.59	0.74	0.68
100	0.61	0.60	0.73	0.64	0.63	0.72
200	1.15	0.65	0.70	1.07	0.62	0.77
300	0.68	0.68	0.74	1.32	0.64	0.6

TABLE 12 – MSE sur l'échantillon test des réseaux de neurones en fonction de la taille des couches 1 & 2 pour la vectorisation *GloVe*

Nous constatons que certaines configurations semblent être à proscrire, telles que 25 neurones en couche 1 et 10 neurones en couche 2 pour la vectorisation *Word2Vec* ainsi que 200 neurones en couche 1 et 10 neurones en couche 2 pour la vectorisation *GloVe*. Mises à part celles-ci, le nombre de neurones choisi a étonnamment peu d'impact sur l'erreur quadratique moyenne mesurée pour notre modèle.

4.6. Synthèse

Modèles	Word2Vec	GloVe
Régression linéaire	0.60	0.57
Régression linéaire par sujet	0.39	0.61
Régression à vecteurs de support	0.63	0.58
Réseaux de neurones	0.63	0.59

TABLE 13 – Meilleurs résultats obtenus pour chaque type de modèles, par vectorisation

Il est très surprenant de constater que nous obtenons le plus faible taux d'erreur sur nos prédictions de notes avec le modèle le plus simple que nous ayons utilisé : une régression linéaire avec une vectorisation *Word2Vec* entraînée uniquement sur le faible volume de données dont nous disposions.

5. Analyse des résultats

5.1. Comparaison par rapport aux correcteurs humains

L'une des comparaisons qui avait été faite lors du concours *Automated Essay Scoring* était d'opposer les résultats des modèles aux notes des correcteurs humains (les colonnes *rater1_domain1* et *rater2_domain1*) et non plus seulement à la moyenne (*domain1_score*).

Pour refaire cette évaluation dans le contexte d'un problème de régression, nous allons centrer et réduire les notes des colonnes *rater1_domain1* et *rater2_domain1* puis calculer leur erreur quadratique moyenne.

En réduisant et centrant les notes, nous supprimons potentiellement un biais : un des correcteurs est peut-être plus clément que l'autre, ou bien utilise une gamme de notes plus réduite.

Sujet	Moyenne de rater1	Écart-type de rater1	Moyenne de rater2	Écart-type de rater2	Moyenne de domain1	Écart-type de domain1
1	4.26	0.84	4.27	0.82	8.53	1.54
2	3.42	0.77	3.44	0.78	3.42	0.77
3	1.74	0.78	1.70	0.75	1.85	0.82
4	1.32	0.88	1.32	0.88	1.43	0.94
5	2.22	0.99	2.22	0.99	2.41	0.97
6	2.56	0.98	2.55	0.98	2.72	0.97
7	8.02	2.42	8.04	2.52	16.06	4.59
8	18.34	3.17	18.56	3.17	36.95	5.75

TABLE 14 – Moyenne et écart-types des colonnes *rater1*, *rater2* et *domain1*

Nous constatons que ce n'est pas le cas, les moyennes et écart-types par sujet de *rater1* et *rater2* sont tous presque égaux.

Sujet	MSE de rater1	MSE de rater2
1	0.56	1.84
2	0.37	1.68
3	0.46	1.36
4	0.30	0.81
5	0.49	1.53
6	0.45	1.61
7	0.56	2.02
8	0.74	2.52
Moyenne	0.47	1.60

TABLE 15 – MSE des correcteurs *rater1* et *rater2* par sujet

Le correcteur *rater1* a une erreur quadratique moyenne bien plus faible que le correcteur *rater2* : 0.47 contre 1.60. La moyenne des deux est donc de 1.04.

Tous nos modèles obtiennent des taux d'erreur plus faibles que cette moyenne. Celui qui a été le plus performant, la régression linéaire par sujet en utilisant une vectorisation *Word2Vec*, fait même bien mieux que le meilleur correcteur.

Au vu de ces résultats, nous pouvons supposer que les méthodes que nous avons utilisées, qui sont pourtant assez simples, réussissent mieux que les correcteurs humains à évaluer les rédactions des étudiants.

5.2. Génération automatique d'une rédaction

Il est possible d'utiliser un réseau de neurones pour générer automatiquement de nouveaux textes à partir d'un corpus. Nous allons utiliser pour cela une architecture déjà disponible[2] que nous entraînons uniquement avec les rédactions du sujet 1 qui ont obtenu de très bons résultats (10/12 et plus).

Nous obtenons le texte suivant :

Dear mosse you have a great computer that connects and face and the people and to play the computer in a computer that is agrees to see, and they can also be able to play out with the are of the pain and the world because they can convenient the society. They are one of the most interacting to the Dear @CAPS1, @CAPS2 @CAPS3 @CAPS3 @CAPS1 for an internet and never don't have ever had a better are a great students and to access to all of the computer and instantly. The computer is better and prove the world and ever had to talk out the computer and technology is a week and the computer. The ho DDeNTH1 seem that the computer is the only because they're because they use the computer to socialize some others and is that and facebook in the reason about the powerposser to be and a positive effect on the video is really beneficial. As a lot of the internet and find the computer in the compute DDeNTH1 less the computer and main all the computer will be the easier and they are an exercise and computers is all computers are to make the computer. One of the @CAPS3 @CAPS3 the @ORGANIZATION1 people are a positive game and play and touch with computers and do you give the computer to the world "Dear @CAPS1 of the world that will play and help us and good they tell with anyone is staying to society. The @CAPS2 @CAPS3 and can communicate and be websites and the lives of that and @CAPS3 are learning to the @CAPS1 and far with the teen with health. See, it is a thing option to any do the soc e''' mess and more than the internet is a computer to use these people that we are all the victims or even all around the positive effect on the computer and quickley. The other people want to learn to the person and teacher that is even devices the time on the computer will be and instead of ways geeRe and how that are doing the computer in the @LOCATION2 and even learning to people the computer to stay in technology. When the computer is a family for a world. I really need to see the world because of a computer of my family can be to society. For example the computer seement espical on the Dear @CAPS1, @CAPS1, @CAPS1 @CAPS3 @PERCENT1 of @NUM1 out of the @CAPS2 or @CAPS2 and what we want to be away from the internet in teachers. The house the are more and files that you can be able to bring us to conclusion and with what the take more than the technology that we don't even were all an DDeNe, @CAPS1 spending time in other websites and sending the computer and email. In social coordination have a computer and far away for the @LOCATION1 both the positive effect on people and said they are not a great time to just be bad. For example, indoors will be expenses to the computer to say DDeNTH1 and the fact the day of @ORGANIZATION1 computers are a very computer. The computer and many family is so our data because they see what they can be able to the better and some and talking to anything that is an essay on the shop earny @ORGANIZATION2 and computers went on the computer result.

Les phrases générées respectent assez souvent la structure grammaticale anglaise, mais n'ont aucun sens.

Pourtant en appliquant de nouveau les mêmes pré-traitements, avec une vectorisation *GloVe*, et en utilisant le modèle de régression linéaire entraîné sur les rédactions du sujet 1 pour l'évaluer, nous obtenons un résultat de 0.93, presque un écart-type au dessus de la moyenne, soit 10/12 selon l'échelle utilisée dans les données sources. Une très bonne note !

6. Conclusion

Les modèles que nous avons développés obtiennent de très bons résultats en théorie, et d'après les mesures que nous avons utilisées, font même mieux que les correcteurs humains.

Ces notations sont pourtant avant tout basées sur la recherche de mots clefs, et comme montré en 5.2 il est très facile de duper ce système, avec un texte n'ayant aucun sens qu'un correcteur humain exclurait aussitôt. Il faut donc espérer qu'un modèle comme celui-ci ne soit pas utilisé en pratique !

Même avec des outils plus perfectionnés que ceux montrés dans cette étude, nous pouvons émettre certaines réserves sur les capacités actuelles des méthodes d'apprentissage statistique pour noter des rédactions d'étudiants. La difficulté est de réellement *comprendre* toutes les nuances de sens d'un texte pour pouvoir l'évaluer efficacement.

Références

- [1] *Man and machine : Better writers, better grades*. URL : http://www.uakron.edu/im/online-newsroom/news_details.dot?newsId=40920394-9e62-415d-b038-15fe2e72a677.
- [2] Woolf Max. *textgenrnn*. <https://github.com/minimaxir/textgenrnn/>. 2020.
- [3] *Measurement Incorporated : Automated Essay Scoring*. URL : <http://www.measurementinc.com/products-services/automated-essay-scoring>.
- [4] Ewa NOWAKOWSKA. « Modeling in a multicollinear setup : Determinants of SVR advantage ». In : *Model Assisted Statistics and Applications* 5 (nov. 2010), p. 219-233. DOI : [10.3233/MAS-2010-0167](https://doi.org/10.3233/MAS-2010-0167).
- [5] Ellis B. PAGE. « The Imminence of... Grading Essays by Computer ». In : *The Phi Delta Kappan* 47.5 (1966), p. 238-243. ISSN : 00317217. URL : <http://www.jstor.org/stable/20371545>.
- [6] Alan RITTER et al. « Named entity recognition in tweets : an experimental study ». In : EMNLP '11 (2011), p. 1524-1534. URL : <http://dl.acm.org/citation.cfm?id=2145432.2145595>.
- [7] THE WILLIAM AND FLORA HEWLETT FOUNDATION. *The Hewlett Foundation : Automated Essay Scoring*. 2012. URL : <https://www.kaggle.com/c/asap-aes/>.