OPENCLASSROOMS

Evaluating language knowledge of ELL students from grades 8-12

Project 8 : Participez à une compétition Kaggle

Mestapha Oumouni

Mentor : Samir Tanfous

OpenClassrooms,

Ingénierie Machine Learning

29 octobre 2022

Table des matières

Mise en contexte

Kaggle est une plateforme qui organise des compétitions en Data Science et qui récompense les meilleurs analystes internationaux. La mission est donc de chercher (selon un choix libre), et participer à une compétition réelle et en cours sur la plateforme. Tout en partageant les résultats obtenus avec la communauté de la plateforme.

1 Présentation de la compétition

L'écriture est une compétence fondamentale dans l'apprentissage d'une seconde langue. Malheureusement, c'est un petit nombre d'étudiants capables de se perfectionner, souvent parce que les tâches d'écriture sont rarement assignées à l'école.

Avec la croissance rapide de la population des étudiants apprenant l'anglais comme langue seconde, la tâche d'évaluation des compétences linguistiques est devenue de plus en plus fatigante pour les enseignants. Elle demande un grand effort de concentration afin d'évaluer correctement les différentes composantes mesurant le niveau linguistique d'un apprenant.

Les outils existants ne sont pas en mesure de fournir une rétroaction basée sur les compétences linguistiques de l'étudiant, ce qui entraîne une évaluation finale qui peut être biaisée au détriment de l'apprenant. La science des données peut être en mesure d'améliorer les outils de rétroaction automatisés pour mieux répondre aux besoins uniques de ces apprenants.

L'objectif de cette compétition est de développer un modèle de machine learning permettant d'évaluer les compétences linguistiques des apprenants de la langue anglaise (ELLs) de la 8e au 12e niveau. Une évaluation correcte, précise et non biaisée peut soutenir les étudiants dans leur parcours d'apprentissage. L'utilisation d'un ensemble de données d'essais de textes et évaluer avec soin par des enseignants de ELLs, peut aider à développer des modèles d'évaluation des compétences linguistiques.

Dans ce rapport, nous allons commencçer par présenter l'ensemble des données utilisé pour entrainer le modèle d'évaluation. Il s'agit d'un modèle de régression. Ensuite, nous effectuons une analyse exploratoire rapide. Nous considérons deux modèles de regression, le premier modèle baseline est le SVM regressor, le deuxième est basé sur le modèle Bert pré-entrainé. Finalement nous comparons les résultats des deux modèles suivant la précision et l'effort d'entrainement.

2 Préparation et analyse de la dataset

L'ensemble de données fourni pour cette compétition comprend des essais de textes des expressions écrites par des apprenants de la langue anglaise (ELLs) de la 8e à la 12e niveau. Les essais ont été notés selon six mesures analytiques :

— la cohésion
— la syntaxe
— le vacabulaire
— la phraséologie
— la grammaire
— les conventions

Chaque mesure qualifie une composante de la compétence de rédaction et d'assimilation de la langue. Des scores plus élevés correspondant à une plus grande maîtrise de cette mesure. Les valeurs de ces différentes composantes de mesures analytiques varient de 1 à 5 avec un incrément de 1/2.

L'ensembles des données contient 3911 textes. Il ne compte pas de valeurs nulles et de doublon. La Figure 1 montre une vue de la tête du dataset.

```
df_train.head(10)
```

	text_id	full_text	cohesion	syntax	vocabulary	phraseology	grammar	conventions
0	0016926B079C	I think that students would benefit from learn...	3.5	3.5	3.0	3.0	4.0	3.0
1	0022683E9EA5	When a problem is a change you have to let it ...	2.5	2.5	3.0	2.0	2.0	2.5
2	00299B378633	Dear, Principal\n\nIf u change the school poli...	3.0	3.5	3.0	3.0	3.0	2.5
3	003885A45F42	The best time in life is when you become yours...	4.5	4.5	4.5	4.5	4.0	5.0
4	0049B1DF5CCC	Small act of kindness can impact in other peop...	2.5	3.0	3.0	3.0	2.5	2.5
5	004AC288D833	Dear Principal,\r\n\r\nOur school should have ...	3.5	4.0	4.0	3.5	3.5	4.0
6	005661280443	Imagine if you could prove other people that y...	3.5	4.0	3.5	3.5	4.0	4.0
7	008DDDDD8E8D	I think it's a good idea for the estudnets to ...	2.5	2.5	2.5	2.5	2.5	2.0
8	009BCCC61C2A	positive attitude is the key to success. I agr...	3.0	3.0	3.5	3.5	3.0	3.0
9	009F4E9310CB	Asking more than one person for and advice hel...	3.0	3.0	3.5	2.5	3.0	2.5

FIGURE 1 – Aperçu de la tête des données

3 Analyse exploratoire

Nous commençons par nettoyer les données en supprimant des retours à la ligne et des espaces supplémentaires. Ensuite, on visualise les distributions des six composantes d'évaluations. La Figure 2 montre que ces variables d'évaluation sont normalement distribuées.

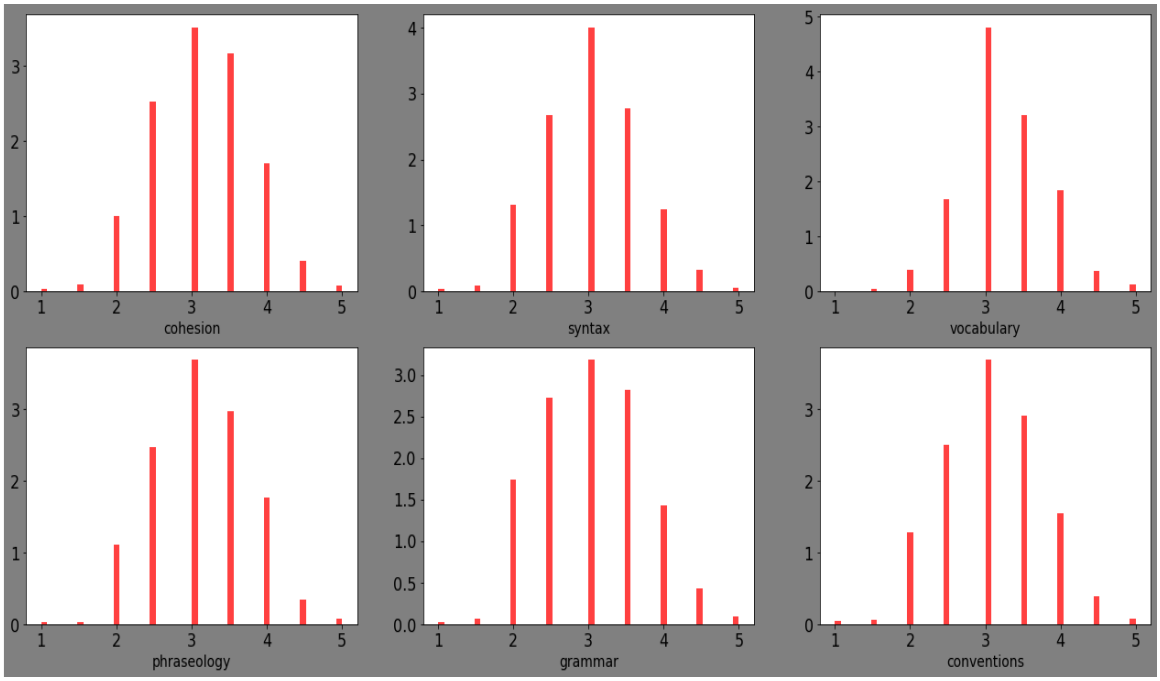

FIGURE 2 – Distrubution des scores des compétences

La figure 3 montre la matrice de corrélation des variables d'évaluation. On peut constater qu'elles sont positivement corrélées avec l'absence d'une corrélation significative, qui peut affecter un modèle de régression.

La distribution du nombre des phrases par texte est légerment asymétrique avec une moyenne de 19 phrases par texte (Figure 4). La distribution du nombre des mots par texte est asymétrique avec une moyenne de 430 mots par texte (Figure 5)

	cohesion	syntax	vocabulary	phraseology	grammar	conventions
cohesion	1.000000	0.695459	0.666151	0.690058	0.638689	0.666151
syntax	0.695459	1.000000	0.680562	0.725467	0.709525	0.700025
vocabulary	0.666151	0.680562	1.000000	0.735261	0.654852	0.664292
phraseology	0.690058	0.725467	0.735261	1.000000	0.719746	0.666842
grammar	0.638689	0.709525	0.654852	0.719746	1.000000	0.673301
conventions	0.666151	0.700025	0.664292	0.666842	0.673301	1.000000

FIGURE 3 – Matrice de corrélation des différentes compétences

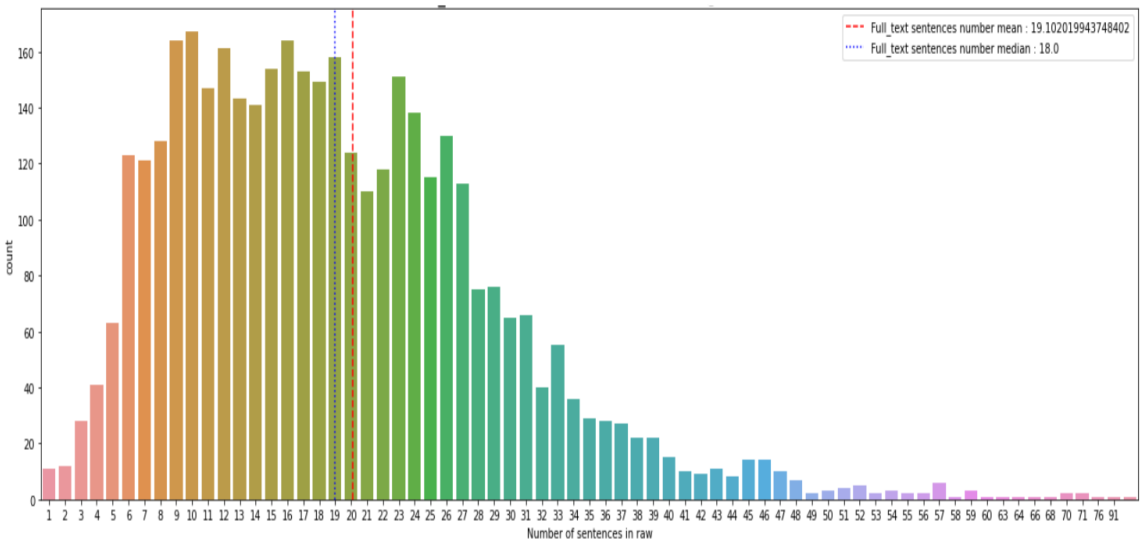

FIGURE 4 – distribution du nombre de phrases par texte

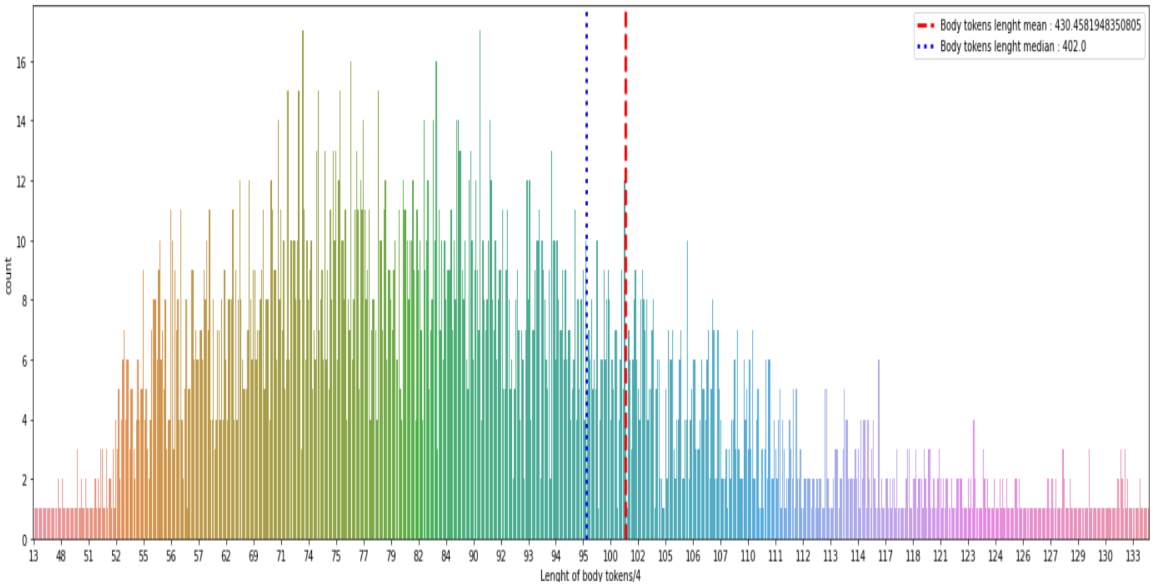

FIGURE 5 – distribution du nombre des mots par texte

4 Modèle de prédiction

Les données mises à disposition sont séparées en deux ensembles, un jeu pour entrainer et évaluer le modèle et un jeu de test de trois textes pour la soumision. Nous divisons le premier ensemble en 80% de données pour l'entrainement et 20% pour la validation. Nous considérons deux modèles de prédiction, on les compare suivant le score d'accuracy et le temps d'entrainement. Nousutilisons la métrique fournie par la compétition.

La qualité des prédictions sera quantifiée en utilisant la moyenne de la norme L2 de chaque composante (MCRMSE) :

$$\frac{1}{N}\sum_{j=1}^{N}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_{i,j}-\hat{y}_{i,j})^2}$$

où N est le nombre de compétances d'évaluation (ici 6 composantes), $y_{:,j}$ et $\hat{y}_{:,j}$ sont l'exacte et la prédiction valeur pour la j-ième compétence, respectivement.

4.1 SVM regressor

Nous commençons d'essayer l'algorithme classique SVM regressor pour entrainer les données textes (Figure 6). Nous utilisons le vectoriseur "Tfidfvectorizer" de sklearn afin de transformer les textes brutes en matrices de nombre réels. Les entrées de cette matrice représentent des fréquences pondérées des mots constituants le document.

```
grid.best_params_
```

```
{'C': 10, 'epsilon': 0.1, 'gamma': 1, 'kernel': 'rbf'}
```

```
svr_clf = SVR(**grid.best_params_)
error = []
for k in range(0,y_train.shape[1]):
  svr_clf.fit(X_train, y_train[:,k])
  rf_preds = svr_clf.predict(X_test)
  error.append(rmse(rf_preds,y_test[:,k],squared=False))
np.mean(error)
```

```
0.5490805237617936
```

FIGURE 6 – Extrait du code d'entrainement du modèle SVR

Les paramètres du modèle SVR ont été fine-tuné en utilisant l'estimateur GridSearchCV afin de trouver ceux qui minimisent l'erreur du modèle. Le score du modèle selon la métrique (MCRMSE) est d'ordre 0.549 ; avec un temps d'entrainement moins de 2min.

4.2 Bert pré-entrainé

Le modèle BERT [1,2] est an open-source machine learning framework pour le traitement du texte natural (NLP). Basé sur des transformers [3], il a été entrainé sur le corps de Wikipedia et peut-être fine-tuné pour un ensemble de problèmes liés aux traitement de texte. La technique utilisée par Bert consiste à masquer aléatoirement les mots dans

la phrase, puis essayer de les prédire.Cela signifie que Bert regarde dans les deux sens et qu'il utilise le contexte complet de la phrase, à gauche et à droite, afin de prédire le mot masqué. La représentation des entrées de BERT (Figure 7) sont formées par la superposition des prolongements suivants :

— Tokenisation des mots et ajout de tokens de début et de fin de phrase
— Marqueur ajouté à chaque phrase pour les distinguer
— Un marqueur de position est ajouté à chaque token (mots) pour indiquer sa position.

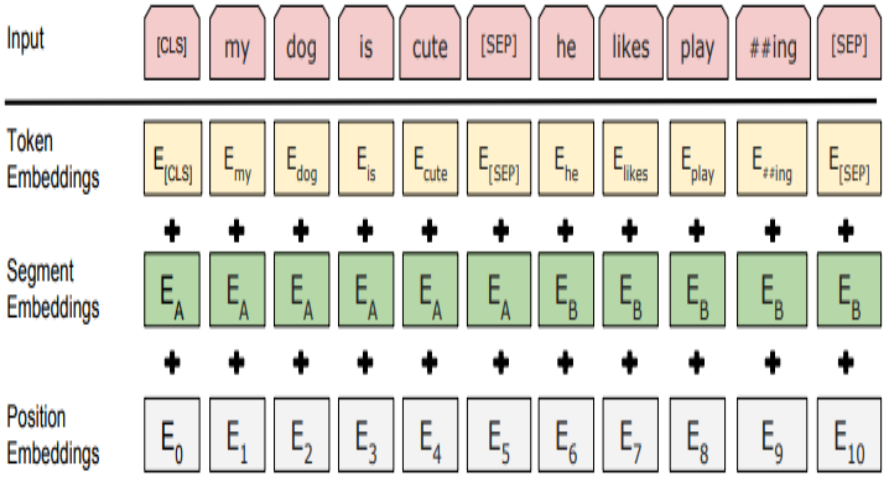

FIGURE 7 – Représentation d'entrées de BERT

Nous utilisons le modèle basé sur Bert pré-entrainé avec des couches congelées afin de réutiliser les features pré-entraînées sans les modifier. Ensuite, nous ajoutons des couches entraînables au-dessus des couches congelées pour adapter les features pré-entraînées à nos données (Figure 8 [4]). L'entrainement est réalisé à l'aide du GPU de Colab.

Le modèle construit améliore significativement le score de prédiction avec un important temps d'ajustement, en comparaison avec le modèle SVR. Le tableau 1 résume les résultas de la comparaison.

	SVR	BERT
score	0.549	0.47
temps d'ajustement	$\approx$ 2 (min)	59 (min)

TABLE 1 – Comparaison des résultats

```
bert_model = transformers.TFBertModel.from_pretrained("bert-base-uncased",
    attention_probs_dropout_prob=0,hidden_dropout_prob=0)
# Freeze the BERT model to reuse the pretrained features without modifying them.
bert_model.trainable = False

bert_output = bert_model.bert(
    input_ids, attention_mask=attention_masks, token_type_ids=token_type_ids)
sequence_output = bert_output.last_hidden_state
pooled_output = bert_output.pooler_output
# ---------------------Add trainable layers on top of frozen layers
bi_lstm = tf.keras.layers.Bidirectional(
tf.keras.layers.LSTM(128, return_sequences=True))(sequence_output)
# Applying hybrid pooling approach to bi_lstm sequence output.
avg_pool = tf.keras.layers.GlobalAveragePooling1D()(bi_lstm)
max_pool = tf.keras.layers.GlobalMaxPooling1D()(bi_lstm)
concat = tf.keras.layers.concatenate([avg_pool, max_pool])
dropout = tf.keras.layers.Dropout(0.2)(concat)
x = tf.keras.layers.Dense(512, activation="relu")(dropout)
x = tf.keras.layers.Dense(512, activation="relu")(x)
dropout = tf.keras.layers.Dropout(0.1)(x)
output = tf.keras.layers.Dense(6)(dropout)
model = tf.keras.models.Model(
    inputs=[input_ids, attention_masks, token_type_ids], outputs=output
```

FIGURE 8 – Extrait du modèle basé sur Bert prè-entrainé

5 Conclusion

Le modèle basé sur Bert pré-entrainé avec l'extraction des features est le modèle retenu pour l'évaluation des compétences linguistiques. Toutefois le score des prédictions peut être encore amélioré, en testant d'autres modèles pré-entrainé avec plus de couches denses et de technique de pooling (en moyenne ou pondéré). Aussi analyser l'impact de la taille des phrases, des couches denses et le taux d'apprentissage.

Références

[1] J.DEVLIN, M. CHANG, K. LEE AND K. TOUTANOVA, *BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding*; arXiv :1810.04805, 2018.

[2] `https://huggingface.co/docs/transformers/model_doc/bert`.

[3] `https://huggingface.co/docs/transformers/model_doc/bert`.

[4] `https://keras.io/examples/nlp/semantic_similarity_with_bert/`