

Native Language Identification

Université Paris-Saclay
M2-ISD

Groupe 08
Malik DOUFENE
Thiziri BOUCHAREB

Introduction

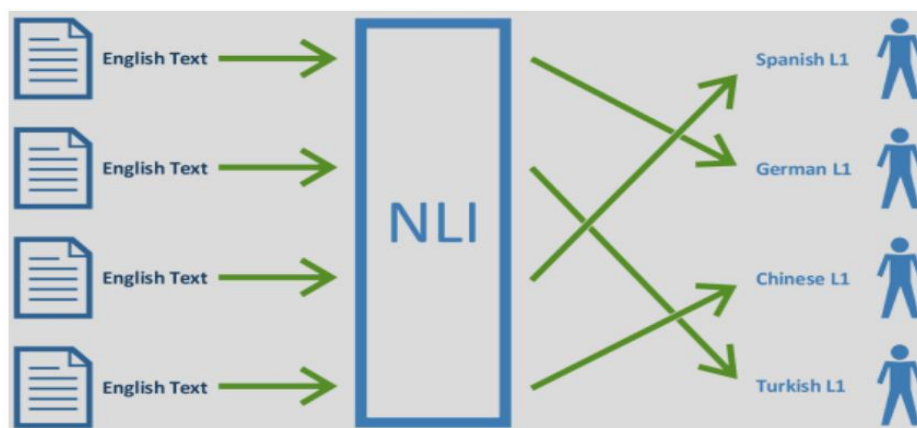
L'identification de la langue maternelle (NLI) revêt une importance capitale dans le domaine du traitement automatique du langage naturel (TALN). Elle désigne la capacité à déterminer la langue maternelle d'une personne à partir de ses écrits ou de ses discours. La NLI joue un rôle essentiel dans diverses applications, notamment la traduction automatique, l'adaptation de contenu, et la personnalisation des interactions linguistiques.

Data

Dans ce contexte, ce projet se concentre sur l'analyse de textes écrits en anglais, issus des expressions écrites de l'examen du TOEFL, en vue de leur classification selon la langue maternelle de leurs auteurs.

Le fichier source est au format .txt et contient des données étiquetées. Les cinq premiers caractères de chaque texte représentent la langue native de l'auteur.

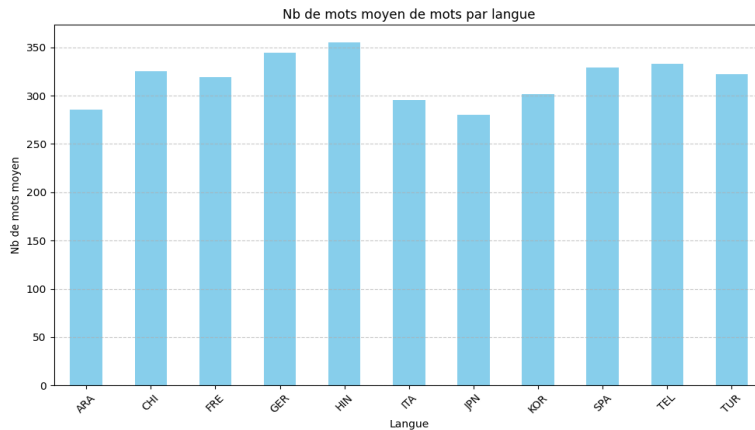
Il contient un total de 9900 échantillons, chacun étant associé à l'une des 11 langues suivantes : Allemand, Turc, Chinois, Telugu, Arabe, Espagnol, Hindi, Japonais, Coréen, Français et Italien.



1. Analyse Exploratoire des Données

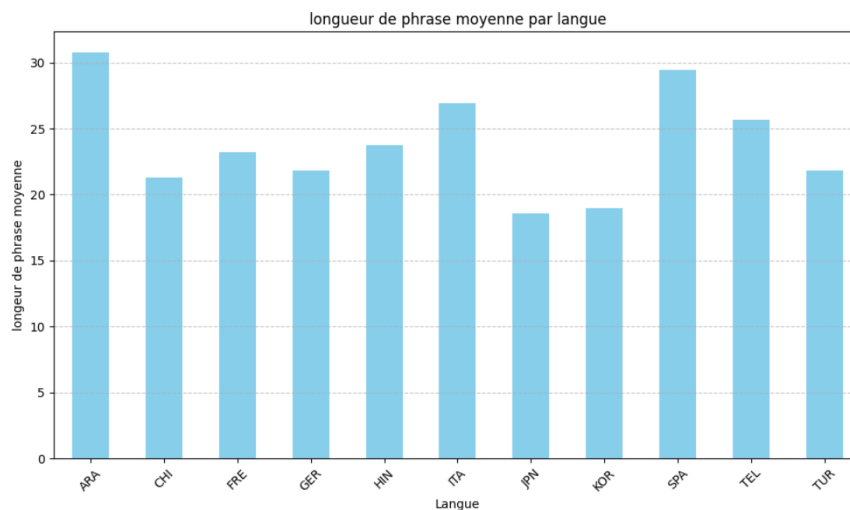
Les données brutes restent inexploitable prises telle quelles, donc on peut essayer d'en tirer quelques nouvelles features discriminatoires :

a. Nombre de mots moyens écrits par langue:



b. Nombre de mots par phrase en moyenne par langue :

Les langues peuvent avoir des tendances différentes en termes de longueur des phrases, certaines étant connues pour des phrases plus courtes et d'autres pour des phrases plus longues. Cette caractéristique peut aider à capturer ces variations.



c. Autres

Caractéristiques Linguistiques Basées sur le Texte Brut

TF-IDF Vectorization:

Cette technique va capturer la distribution des mots dans chaque phrase tout en atténuant l'importance des mots courants qui apparaissent fréquemment dans toutes les langues.

Analyse des N-grammes de Mots:

Les n-grammes de mots permettent de capturer les relations syntaxiques et sémantiques entre les mots, ce qui peut révéler des motifs linguistiques spécifiques à chaque langue.

Analyse des N-grammes de Caractères:

Les n-grammes de caractères sont utiles pour capturer les particularités orthographiques et phonétiques de chaque langue, notamment les schémas de lettres et de sons qui peuvent être distinctifs.

Caractéristiques Linguistiques Basées sur la Structure des Phrases

Erreurs Linguistiques Communes:

Les erreurs linguistiques récurrentes peuvent indiquer des caractéristiques spécifiques à chaque langue, telles que des règles grammaticales particulières ou des difficultés orthographiques.

Mots les Plus Utilisés par Langue:

Les mots les plus courants dans une langue peuvent fournir des indices sur ses caractéristiques linguistiques distinctives, telles que son vocabulaire de base, ses expressions idiomatiques et ses thèmes culturels prédominants.

2. Travail réalisé

Nous avons testé plusieurs approches

2.1. Modèles basiques sur données brutes :

2.1.1. Choix du tokenizer et du model de base

On a hésité entre les deux approches Bag Of Words et TF IDF, que nous avons utilisé associé avec 03 mêmes modèles, à savoir Naive Bayes, SVM (Machine à Vecteurs de Support), Logistic Regression et Random Forest, et les résultats de précision générale sont les suivants :

| | SVC | Logistic Regression | Random Forest |
|--------------|------|---------------------|---------------|
| Bag Of Words | 0.36 | 0.37 | 0.30 |
| TF-IDF | 0.62 | 0.64 | 0.47 |

Les résultats démontrent que TF-IDF surpasse significativement BoW dans tous les modèles testés (SVM, Régression logistique et Forêt aléatoire). Cette supériorité peut être attribuée à la capacité de TF-IDF à prendre en compte à la fois la fréquence et l'importance relative des mots dans le corpus, ce qui aide à capturer les caractéristiques discriminantes pour la classification. Parmi les modèles, SVM et la régression logistique montrent les meilleures performances avec TF-IDF, tandis que la Forêt aléatoire obtient des résultats inférieurs aux autres modèles, probablement en raison de sa nature d'agrégation des prédictions. En résumé, TF-IDF est préférable à BoW pour l'identification de la langue maternelle, et SVM ainsi que la régression logistique sont les modèles les plus efficaces pour ce problème.

Pour s'assurer des résultats précédents :

Nous avons mis en place des pipelines pour trois modèles, chacune de ces pipelines intègre une étape de **vectorisation des mots** avec TF-IDF pour représenter les données textuelles. Ensuite, des **grilles de recherche** sont définies pour ajuster les hyperparamètres spécifiques à chaque modèle. Les modèles sont ensuite entraînés et évalués en utilisant la **validation croisée** sur l'ensemble d'entraînement.

```
Meilleurs paramètres pour Naive Bayes: {}
Score sur l'ensemble de test pour Naive Bayes: 0.4791666666666667
Rapport de classification pour Naive Bayes:
      precision    recall  f1-score   support

   ARA      0.94      0.26      0.41       170
   CHI      0.51      0.48      0.50       175
   FRE      0.89      0.39      0.54       193
   GER      0.27      0.95      0.42       172
   HIN      0.59      0.73      0.65       176
   ITA      0.48      0.77      0.59       164
   JPN      0.86      0.26      0.40       186
   KOR      0.89      0.17      0.28       201
   SPA      0.56      0.31      0.39       196
   TEL      0.83      0.60      0.70       179
   TUR      0.42      0.67      0.51       168

 accuracy          0.50       1980
 macro avg      0.66      0.51      0.49       1980
 weighted avg    0.67      0.50      0.49       1980
```

```
Meilleurs paramètres pour SVM: {'clf_c': 10, 'clf_kernel': 'linear'}
Score sur l'ensemble de test pour SVM: 0.6649366223961156
Rapport de classification pour SVM :
      precision    recall  f1-score   support

   ARA      0.64      0.70      0.67       194
   CHI      0.72      0.72      0.72       200
   FRE      0.68      0.75      0.71       193
   GER      0.85      0.85      0.85       235
   HIN      0.65      0.70      0.68       196
   ITA      0.78      0.78      0.78       187
   JPN      0.68      0.67      0.67       181
   KOR      0.68      0.66      0.67       187
   SPA      0.68      0.65      0.67       207
   TEL      0.78      0.70      0.74       207
   TUR      0.77      0.74      0.75       213

 accuracy          0.72       2200
 macro avg      0.72      0.72      0.72       2200
 weighted avg    0.72      0.72      0.72       2200
```

```
Meilleurs paramètres pour Random Forest: {'clf_max_depth': 20, 'clf_n_estimators': 200}
Score sur l'ensemble de test pour Random Forest: 0.5195707070707071
Rapport de classification pour Random Forest:
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| ARA | 0.50 | 0.42 | 0.46 | 170 |
| CHI | 0.53 | 0.39 | 0.45 | 175 |
| FRE | 0.58 | 0.49 | 0.53 | 193 |
| GER | 0.53 | 0.66 | 0.59 | 172 |
| HIN | 0.44 | 0.56 | 0.49 | 176 |
| ITA | 0.43 | 0.71 | 0.53 | 164 |
| JPN | 0.48 | 0.67 | 0.56 | 186 |
| KOR | 0.65 | 0.41 | 0.50 | 201 |
| SPA | 0.56 | 0.31 | 0.40 | 196 |
| TEL | 0.50 | 0.65 | 0.56 | 179 |
| TUR | 0.61 | 0.41 | 0.49 | 168 |
| accuracy | | | 0.51 | 1980 |
| macro avg | 0.53 | 0.52 | 0.51 | 1980 |
| weighted avg | 0.53 | 0.51 | 0.51 | 1980 |

2.2. LLM

Ces nouveaux modèles de langue à grande échelle ont établi de nouvelles avancées dans diverses tâches de traitement automatique du langage naturel (NLP).

Compte tenu de leurs capacités de compréhension avancées, ils promettent d'améliorer significativement les performances de notre système. Nous avons tenté d'intégrer BERT et GPT dans notre approche, mais ces modèles exigent des ressources de calcul importantes pour l'entraînement. Malheureusement, cette contrainte nous a empêché d'expérimenter pleinement cette approche, le programme plante à chaque tentative avec l'erreur "not enough memory".

```
File c:\Users\bouch\anaconda3\lib\site-packages\torch\nn\modules\sparse.py:162, in Embedding.forward(self, input)
    161 def forward(self, input: Tensor) -> Tensor:
--> 162     return F.embedding(
    163         input, self.weight, self.padding_idx, self.max_norm,
    164         self.norm_type, self.scale_grad_by_freq, self.sparse)

File c:\Users\bouch\anaconda3\lib\site-packages\torch\nn\functional.py:2233, in embedding(input, weight, padding_idx, max_norm, norm_type,
scale_grad_by_freq, sparse)
    2227 # Note [embedding_renorm set_grad_enabled]
    2228 # XXX: equivalent to
    2229 # with torch.no_grad():
    2230 #     torch.embedding_renorm_
    2231 # remove once script supports set_grad_enabled
    2232 _no_grad_embedding_renorm_(weight, input, max_norm, norm_type)
-> 2233 return torch.embedding(weight, input, padding_idx, scale_grad_by_freq, sparse)

RuntimeError: [enforce fail at alloc_cpu.cpp:80] data. DefaultCPUAllocator: not enough memory: you tried to allocate 15571353600 bytes.
```

Votre session a planté après avoir utilisé toute la mémoire RAM disponible. ✕

Dans une tentative de pallier cette limitation, nous avons réduit l'ensemble de données à seulement 100 lignes par groupe de langues. Cependant, même avec cette réduction, le programme a nécessité une heure pour s'exécuter, et la précision obtenue était de 0%.

✓ 62m 43.5s

Some weights of the model cl

- This IS expected if you a
- This IS NOT expected if y

Some weights of BertForSequ

You should probably TRAIN t

[c:\Users\TBB1\Anaconda\lib\](#)

warnings.warn(

100%

{'loss': 2.4205, 'learning_

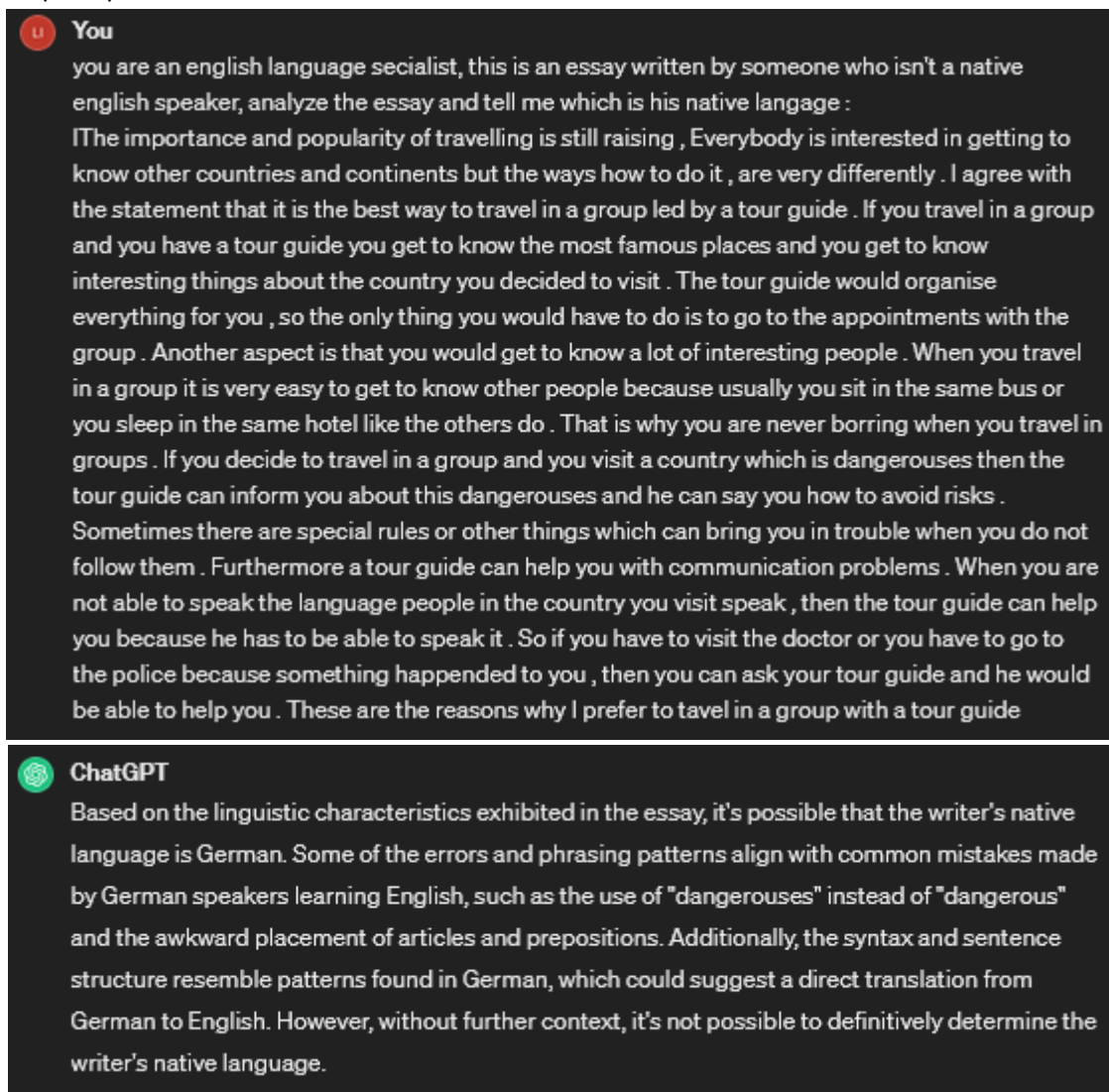
{'train_runtime': 3646.4556

100%

Validation Accuracy: 0.0

Remarque : Nous avons essayé d'utiliser GPT 3.5 en Zero Shot, c'est-à-dire sans fine tuning, et en utilisant l'api ouvert d'OpenAI, sur le peu de données qu'on a pu testé à cause des restrictions de la version gratuite nous avons été assez surpris de l'exactitude des prédictions, mais en plus du bémol de la limite des requêtes permises par heure, on peut aussi tombé sur d'autres pépins comme les réponses non précises de GPT quand la distinction de la langue devient challengeant, c'est-à-dire qu'il ne donne pas de réponse ou bien dis que l'auteur est english-native

Exemple de prompt :



You

you are an english language secialist, this is an essay written by someone who isn't a native english speaker, analyze the essay and tell me which is his native langage :

!The importance and popularity of travelling is still raising , Everybody is interested in getting to know other countries and continents but the ways how to do it , are very differently . I agree with the statement that it is the best way to travel in a group led by a tour guide . If you travel in a group and you have a tour guide you get to know the most famous places and you get to know interesting things about the country you decided to visit . The tour guide would organise everything for you , so the only thing you would have to do is to go to the appointments with the group . Another aspect is that you would get to know a lot of interesting people . When you travel in a group it is very easy to get to know other people because usually you sit in the same bus or you sleep in the same hotel like the others do . That is why you are never borring when you travel in groups . If you decide to travel in a group and you visit a country which is dangerouses then the tour guide can inform you about this dangerouses and he can say you how to avoid risks . Sometimes there are special rules or other things which can bring you in trouble when you do not follow them . Furthermore a tour guide can help you with communication problems . When you are not able to speak the language people in the country you visit speak , then the tour guide can help you because he has to be able to speak it . So if you have to visit the doctor or you have to go to the police because something happended to you , then you can ask your tour guide and he would be able to help you . These are the reasons why I prefer to tavel in a group with a tour guide

ChatGPT

Based on the linguistic characteristics exhibited in the essay, it's possible that the writer's native language is German. Some of the errors and phrasing patterns align with common mistakes made by German speakers learning English, such as the use of "dangerouses" instead of "dangerous" and the awkward placement of articles and prepositions. Additionally, the syntax and sentence structure resemble patterns found in German, which could suggest a direct translation from German to English. However, without further context, it's not possible to definitively determine the writer's native language.

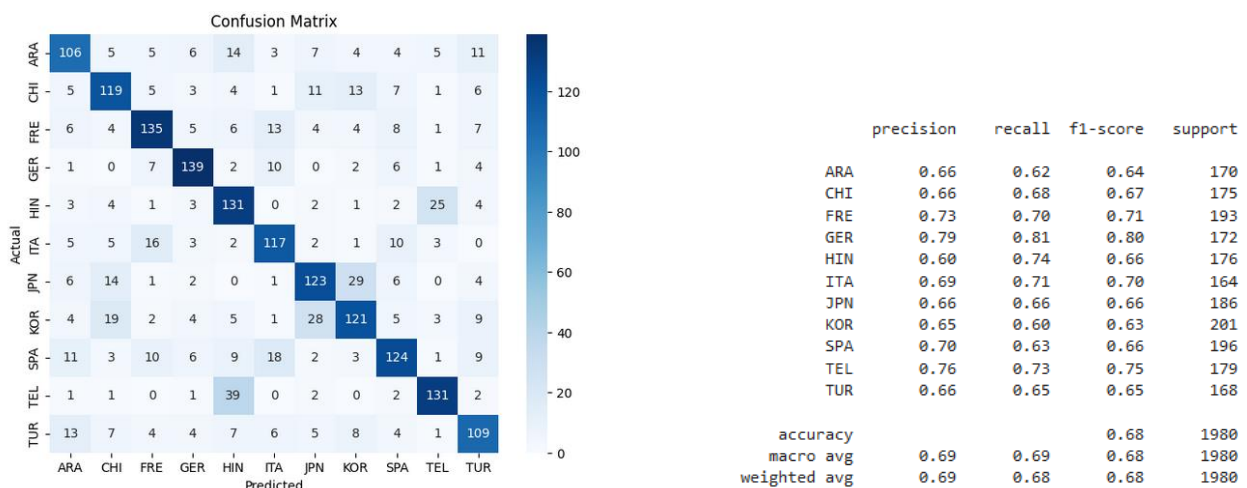
2.3. Ensemble Classifier

L'ensemble learning, plus précisément le vote majoritaire (ou hard voting), où plusieurs modèles individuels sont combinés pour produire une prédiction finale. Cette approche est considérée intéressante pour plusieurs raisons :

1. **Amélioration de la robustesse** : En combinant plusieurs modèles, on peut compenser les faiblesses individuelles de chaque modèle. Par exemple, un modèle peut être plus performant pour certaines langues tandis qu'un autre peut être meilleur pour d'autres. En combinant leurs prédictions, on obtient un modèle plus robuste et généralisable.

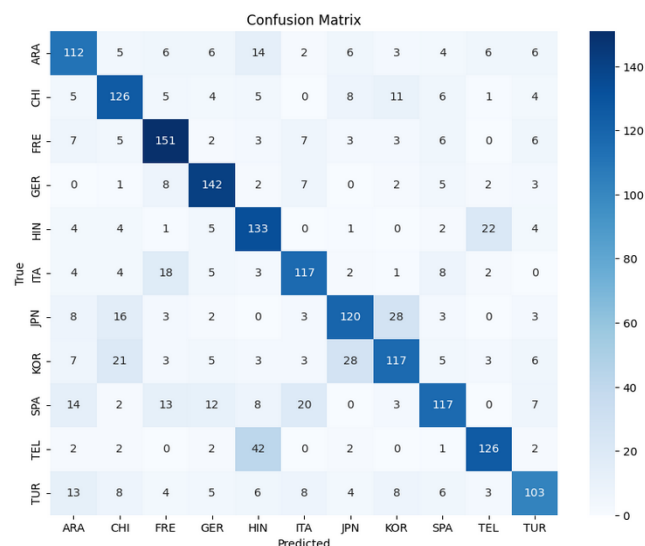
2. **Réduction du surapprentissage** : L'utilisation de plusieurs modèles différents peut aider à réduire le surapprentissage (overfitting) en introduisant de la diversité dans les prédictions. Cela peut conduire à des modèles qui généralisent mieux sur de nouvelles données.
3. **Meilleure performance globale** : L'ensemble learning a souvent démontré une performance supérieure à celle des modèles individuels dans de nombreux scénarios. En exploitant la sagesse collective des modèles, on peut souvent obtenir de meilleurs résultats de prédiction.
4. **Flexibilité et diversité** : L'approche d'ensemble learning permet d'utiliser différents types de modèles, ce qui peut conduire à une diversité de perspectives et d'approches dans la prise de décision. Par exemple, dans le code fourni, on utilise à la fois des SVM avec différents noyaux et une régression logistique, ce qui apporte une diversité dans les stratégies de classification.
5. **Facilité d'implémentation** : Les méthodes d'ensemble comme le vote majoritaire sont relativement simples à mettre en œuvre, surtout avec des bibliothèques de machine learning comme scikit-learn en Python. Cela permet une utilisation pratique même pour des praticiens moins expérimentés.

2.3.1. SVC linear + SVC rbf + Logistic Regression:



2.3.2. Bagging classifieur :

Dans ce test nous avons utilisé la même instance de SVC couplé avec du TF-IDF mais chacune s'ait entraîné sur un train set différent, toujours avec un vote sur la prediction final.



- Analyse : Le modèle semble bien performant pour certaines langues comme la classe 3, où la plupart des prédictions sont correctes.
- Cependant, il existe des confusions entre certaines langues, notamment entre la classe 1, 5, 7 et 10, où les prédictions incorrectes sont plus fréquentes.
- Les langues qui sont souvent confondues ont probablement des caractéristiques linguistiques similaires, ce qui rend la distinction plus difficile pour le modèle.
- Malgré ces confusions, le modèle parvient à correctement classer un pourcentage significatif des exemples, comme en témoigne l'accuracy globale d'environ 71.1%.

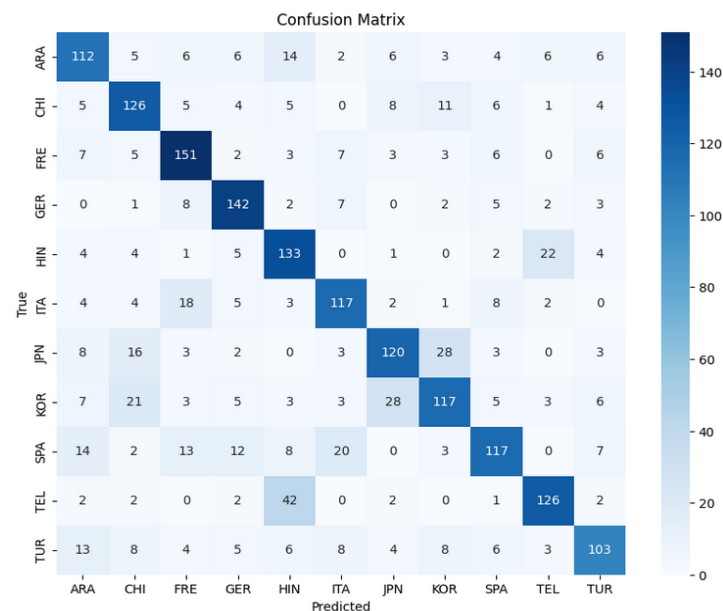
En résumé, bien que le modèle puisse confondre certaines langues, il parvient néanmoins à bien classer une proportion importante des données. Cependant, des efforts supplémentaires peuvent être nécessaires pour améliorer sa capacité à distinguer les langues qui sont souvent confondues.

2.3.3. SVC + multiples preprocessing features :

Dans ce test nous avons utilisé des instances SVC entraîné sur des différentes features vectoriser avec de TF-IDF (word unigram, word bigram, word trigram, caractere Ngram),

2.4. Model hiérarchique (Model final utilisé lors de l'examen) :

Après avoir entraîné et tester notre meilleur model trouvé jusqu'à présent (SVC avec TFIDF et word Bigram), on s'intéresse aux erreurs qu'il a pu commettre pour distinguer des classes de langues similaires ou difficile a distinguer pour notre model.



On remarque qu'on peut classifier les langues de la manière suivante :

Indien: HIN, TEL

Romance: ITA, FRE, SPA

Asian: KOR, JPN, CHI

Et enfin german, Turkish et arabe chacune seule

Donc, en se basant sur ca nous avons entraîné notre modèle de base choisi à prédire ces 05 classes là au lieu des 11 initiales et les résultats sont les suivants :

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| Arabic | 0.90 | 0.50 | 0.64 | 170 |
| Asian | 0.87 | 0.93 | 0.90 | 562 |
| German | 0.89 | 0.79 | 0.84 | 172 |
| Indian | 0.90 | 0.97 | 0.93 | 355 |
| Romance | 0.81 | 0.93 | 0.87 | 553 |
| Turkish | 0.92 | 0.65 | 0.76 | 168 |
| accuracy | | | 0.86 | 1980 |
| macro avg | 0.88 | 0.80 | 0.82 | 1980 |
| weighted avg | 0.87 | 0.86 | 0.86 | 1980 |

- La classe "Arabic" a une précision de 0.90, ce qui signifie que 90% des prédictions de cette classe sont correctes. Cependant, le rappel est de 0.50, ce qui indique que le modèle ne parvient à identifier que la moitié des instances réellement arabes. Le score F1 est de 0.64.
- Pour la classe "Asian", la précision est élevée à 0.87, ce qui suggère que la plupart des prédictions pour cette classe sont correctes. Le rappel est également élevé à 0.93, ce qui signifie que le modèle récupère la plupart des instances réellement asiatiques. Le score F1 est de 0.90.
- La classe "German" a une précision de 0.89 et un rappel de 0.79, avec un score F1 de 0.84. Cela montre une performance généralement bonne, bien que le rappel soit légèrement inférieur à la précision.
- La classe "Indian" a des scores élevés dans toutes les métriques, avec une précision de 0.90, un rappel de 0.97 et un score F1 de 0.93. Cela indique que le modèle prédit avec précision et rappelle efficacement les instances de langue indienne.
- La classe "Romance" a une précision légèrement inférieure à 0.81, mais un rappel élevé de 0.93, conduisant à un score F1 de 0.87. Cela montre que bien que le modèle puisse avoir quelques erreurs de prédiction pour cette classe, il est bon à identifier la plupart des instances réellement romanes.
- La classe "Turkish" a une précision élevée de 0.92, mais un rappel plus faible de 0.65, conduisant à un score F1 de 0.76. Cela suggère que le modèle peut prédire avec précision les instances turques, mais qu'il en manque un certain nombre.

A partir de là, on a entraîné un nouveau model sur chacune des classes ayant plusieurs sous langues (Indian, Asian, Romance), et on selon le label initialement prédit on prédit avec le sous model associé au groupe de langue. On a choisi de donner une plus grande confiance au premier model qui prédit les groupe de langue de par ses résultats, en effet il n'y aura pas de recyclage, car chaque phrase prédite aura dans tout les cas un label final de ladite classe, on peut résumer le comportement de notre système par le schéma ci-dessous.

