RAPPORT DU PROJET D'EXTRACTION D'INFORMATIONS : DÉTECTION DE LANGUES

Définitions, problématique et cadre théorique

La **détection de langue** est le processus d'identification automatique de la langue. Ici on s'intéressera à la **détection de la langue d'un texte** (donc à l'écrit).
Cette tâche est essentielle dans de nombreuses applications, telles que le traitement automatique des langues (TAL), les moteurs de recherche, les systèmes de traduction automatique, et les interfaces utilisateur multilingues.
La détection de langue sert de point de départ pour d'autres tâches de traitement de texte, permettant de sélectionner les outils linguistiques appropriés pour l'analyse et le traitement ultérieur.

Ainsi, on considère **en entrée un segment de texte** qui peut être un mot, une phrase, un paragraphe ou un document complet, et e**n sortie une étiquette de langue** qui indique dans quelle langue le texte d'entrée est écrit.

La détection de langue doit pouvoir faire face à des **difficultés spécifiques** qui ne doivent pas l'empêcher d'avoir de bons résultats. Ces défis incluent :
- des **textes courts** (comme des mots individuels ou de courtes phrases) qui fournissent moins d'informations contextuelles, rendant la détection de langue plus difficile
- des **langues similaires**, qui partagent des racines linguistiques communes ou qui ont des similarités lexicales et qui peuvent donc être difficiles à distinguer
- des **textes multilingues** (des textes qui contiennent des portions écrites dans plusieurs langues)
- une **orthographe** qui peut parfois être **non standard**, comme les variations orthographiques, l'argot, et les erreurs typographiques qui peuvent compliquer la tâche.

Ici on s'intéressera en particulier à la **détection de langue pour des textes monolingues**.

La détection de langue repose sur plusieurs concepts et techniques en traitement automatique des langues et apprentissage automatique.

1. **Approches basées sur des règles, dictionnaires, lexiques :**

   Utilisent des règles linguistiques définies manuellement pour identifier la langue d'un texte. Cela peut inclure des règles basées sur des motifs de caractères, des bigrammes, et des trigrammes fréquents dans certaines langues.
   Utilisent des dictionnaires de mots spécifiques à chaque langue pour comparer les mots présents dans le texte avec ceux des dictionnaires. La langue du texte est déterminée en fonction de la correspondance maximale avec les mots du dictionnaire.

2. **Approches Statistiques :** cette méthode repose sur l'analyse statistique des caractéristiques linguistiques du texte, telles que la distribution des lettres, des mots et des séquences de caractères. Des modèles probabilistes sont ensuite utilisés pour attribuer une probabilité à chaque langue candidate, et la langue avec la probabilité la plus élevée est sélectionnée comme langue détectée.

Bag of Words (BoW) **:** Représente un texte comme un ensemble de mots ou de n-grammes sans tenir compte de l'ordre des mots. Un modèle de classification est ensuite entraîné sur ces représentations pour prédire la langue.
TF-IDF (Term Frequency-Inverse Document Frequency) **:** Une amélioration du modèle BoW qui pondère les mots en fonction de leur fréquence dans un document par rapport à leur fréquence dans l'ensemble du corpus.
Modèles de N-grammes **:** Utilisation de séquences de n caractères (ou mots) pour capturer les modèles linguistiques. Les modèles statistiques peuvent ensuite être appliqués pour estimer la probabilité qu'un texte appartienne à une langue donnée.

3. **Approche basée sur les réseaux de neurones** :

   Cette méthode utilise des réseaux de neurones artificiels pour apprendre des représentations vectorielles des mots ou des caractères dans différentes langues, puis utilise ces représentations pour prédire la langue du texte en fonction de ses caractéristiques linguistiques.

   On utilise LSTM, CNN, ou des modèles Transformer (comme BERT) pour la classification de texte.

Ces **différentes approches** peuvent être utilisées individuellement ou **combinées** pour améliorer la précision de la détection de la langue. Le choix de la méthode dépend souvent de la nature du texte à traiter, de la disponibilité des ressources linguistiques et des contraintes de performance.

État de l'art

Avant d'entrer véritablement dans le vif du sujet et de parler des données et modèles que nous avons utilisés, voyons tout d'abord l'état de l'art dans le domaine afin d'encore mieux situer le contexte.

État de l'art : Identification de langue avec la tokenisation de modèle de mélange gaussien

Dans le cadre de notre étude sur l'identification automatique de la langue dans les documents, nous nous sommes penchés sur les travaux récents dans ce domaine, notamment l'article intitulé "Language Identification using Gaussian Mixture Model Tokenization". Cet article propose une approche innovante en utilisant les modèles de mélanges gaussiens (GMM) pour la tokenisation, une méthode qui se distingue par son efficacité même en l'absence de transcriptions phonétiques détaillées.

Le contexte de cette recherche est marqué par un besoin croissant de systèmes capables de fonctionner efficacement avec des ressources linguistiques limitées, surtout dans les applications multilingues comme les services d'information téléphoniques internationaux. Les méthodes traditionnelles, bien que performantes, dépendent fortement de bases de données phonétiquement transcrites, qui ne sont pas disponibles pour toutes les langues. La méthode proposée par les auteurs, qui utilise des GMM pour la tokenisation, permet de surmonter cette limitation en minimisant les discordances potentielles qui pourraient survenir avec les systèmes basés sur la reconnaissance phonétique.

Au niveau des expérimentations, l'article détaille l'application de cette méthode sur un corpus varié incluant des conversations téléphoniques en 12 langues différentes. Le système utilise un tokeniseur GMM qui assigne les vecteurs de caractéristiques acoustiques à des partitions spécifiques de l'espace acoustique. Ce processus forme une base pour le développement de modèles de langue bigrammes qui prévoient la probabilité d'un token donné suivant un autre. Les expériences ont testé l'efficacité des modèles GMM avec différentes tailles de modèles de mélange, allant de 64 à 512, où il a été observé que l'augmentation de la taille du modèle améliore les performances jusqu'à un certain point.

Les résultats de ces expérimentations ont révélé que l'approche basée sur GMM, bien qu'initialement moins performante que certains systèmes traditionnels, offre des taux d'erreur compétitifs avec une réduction significative des coûts computationnels. L'utilisation de classificateurs de l'arrière-plan pour combiner les scores des modèles de langue a également montré une amélioration notable de la précision, soulignant l'efficacité de cette méthode dans des environnements où les ressources linguistiques sont limitées.

La pertinence de cet article pour notre projet de détection de langue dans un document est considérable. En effet, l'approche de tokenisation basée sur GMM offre une flexibilité notable pour l'identification de langues dans des contextes où les données peuvent être hétérogènes ou partiellement disponibles. Cette méthode pourrait être adaptée pour analyser les caractéristiques textuelles des documents, offrant ainsi une solution viable pour les applications nécessitant la détection de la langue dans des documents numériques. Elle ouvre également la voie à des applications plus larges telles que la traduction automatique et le résumé automatique dans des environnements multilingues.

En conclusion, les travaux présentés dans cet article constituent une base solide pour notre projet et suggèrent des pistes de recherche prometteuses pour améliorer la détection de langue dans les documents numériques, en rendant les systèmes plus flexibles et moins dépendants des ressources linguistiques préexistantes.

État de l'art : automatique détection and language identification of multilingue documents

Dans le cadre de notre projet sur la détection et l'identification de la langue, nous nous sommes penchés sur les travaux déjà réalisés dans ce domaine, notamment l'article intitulé « Automatic Detection and Language Identification of Multilingual Documents ». On y introduit une méthode capable de détecter qu'un document est multilingue, d'identifier les langues présentes et d'estimer leurs proportions relatives. Les chercheurs démontrent l'efficacité de leur méthode sur des données synthétiques, ainsi que sur des documents multilingues réels collectés sur le web.

Contexte :

Définition du contexte et explication des domaines et des enjeux : L'identification de la langue est la tâche consistant à détecter automatiquement la ou les langues présentes dans un document en fonction du contenu du document. Dans cet article est abordé le problème de la détection des documents contenant du texte dans plusieurs langues (documents multilingues).

La détection de la langue est une tâche fondamentale dans le traitement automatique des langues, avec des applications dans la recherche d'informations, la classification de documents, la traduction automatique, etc. Dans un monde de plus en plus connecté, où les données textuelles sont disponibles

dans de nombreuses langues, la capacité à identifier automatiquement la langue dans laquelle un document est rédigé revêt une importance croissante pour de nombreuses applications.

Objectif principal du travail et attentes au niveau des résultats :

L'objectif principal du travail est de proposer une méthode efficace pour détecter automatiquement les langues présentes dans des documents multilingues. Les auteurs cherchent à améliorer les performances par rapport aux travaux antérieurs en développant une approche basée sur un modèle de mélange génératif supervisé. Ils s'attendent à ce que leur méthode surpasse les approches existantes en termes de précision, de rappel et de F-mesure, tout en étant capable d'estimer avec précision les proportions relatives des langues dans les documents multilingues.

Les auteurs espèrent également que leur méthode sera applicable à un large éventail de langues et de types de documents, ce qui en fera une solution polyvalente pour la détection de la langue dans divers contextes. Enfin, ils visent à rendre leur méthode facilement accessible en fournissant une implémentation de référence et un ensemble de données synthétiques pour faciliter son adoption par la communauté de recherche et d'application.

Expérimentations et méthodes :

Les auteurs utilisent un modèle de mélange génératif supervisé (méthode probabiliste), inspiré des algorithmes de modélisation de sujets (dans ce contexte, les auteurs s'inspirent de ces algorithmes pour développer un modèle similaire qui peut identifier les langues présentes dans un document multilingue plutôt que les sujets).

Ils combinent ce modèle avec une représentation de document basée sur des recherches antérieures en identification de langue pour des documents monolingues. Cette méthode implique des étapes telles que la segmentation du document en fonction de la langue, l'extraction de fonctionnalités linguistiques, la modélisation de mélange pour chaque langue candidate, et l'estimation des proportions relatives des langues dans le document. On peut noter que la méthode utilisée ne nécessite pas de connaissances spécifiques sur les langues traitées telles que des lexiques ou des règles grammaticales. Elle se base plutôt sur des modèles statistiques appris à partir de données d'entraînement.

Aucune autre ressource linguistique externe n'est mentionnée comme étant nécessaire à la méthode. Les expérimentations ont été menées sur deux ensembles de données : ALTW2010, un ensemble de données synthétique de 10 000 documents bilingues, et WIKIPEDIA MULTI, un ensemble de données contenant un mélange de documents monolingues et multilingues générés à partir d'extraits de pages Wikipedia. ALTW2010 est principalement utilisé pour évaluer la capacité des systèmes à identifier correctement les langues dans les documents, tandis que WIKIPEDIA MULTI est utilisé pour tester les performances sur des documents multilingues réels.

Evaluation et résultats :

Dans l'article, plusieurs mesures d'évaluation sont employées pour évaluer les performances de la méthode proposée.
Précision (P) : Mesure la proportion de résultats positifs corrects parmi l'ensemble des résultats positifs prédits par le modèle.

Rappel (R) : Mesure la proportion de résultats positifs corrects parmi tous les résultats positifs réels dans les données.
Score F (F-score) : Représente la moyenne harmonique de la précision et du rappel. C'est une mesure combinée de la précision et du rappel qui tient compte à la fois des faux positifs et des faux négatifs.

Ces mesures sont calculées à la fois au niveau du document et au niveau de la langue, ce qui permet une évaluation complète des performances du modèle dans différentes situations. Quant à savoir si les résultats obtenus sont meilleurs que ceux des travaux antérieurs, l'article compare les performances de la méthode proposée avec celles d'autres systèmes existants sur différents ensembles de données, à la fois synthétiques et réels. Dans certains cas, les résultats obtenus avec la méthode proposée surpassent ceux des travaux antérieurs, tandis que dans d'autres cas, la méthode propose une alternative efficace avec des avantages spécifiques dans certains aspects de la détection de la langue.
Par exemple, dans l'expérience sur ALTW2010, la méthode proposée dépasse les approches existantes en termes de précision, de rappel et de score F. Cependant, dans l'expérience sur WIKIPEDIA MULTI, bien que les performances de la méthode proposée soient excellentes, elles peuvent différer selon les aspects évalués et les ensembles de données utilisés.

Pertinence de l'article par rapport à notre projet :
Cet article est pertinent pour notre projet sur la détection de la langue pour plusieurs raisons. Approche novatrice : L'article présente une méthode novatrice pour l'identification de la langue dans des documents multilingues en utilisant un modèle de mélange génératif. Cette approche offre une perspective intéressante pour améliorer la précision et la robustesse de la détection de la langue (perspective qu'on pourrait utiliser ? A voir, mais c'est dans tous les cas intéressant d'être au courant de ce qui se fait ou peut se faire).

Performance évaluée :
L'étude évalue la performance de la méthode proposée à l'aide de métriques standard telles que la précision, le rappel et le score F, ainsi que la corrélation entre les proportions prédites et réelles des langues dans les documents. Ces évaluations fournissent une validation empirique de l'efficacité de l'approche dans divers contextes, ce qui pourrait nous aider à évaluer et à comparer les performances de différentes méthodes que l'on pourrait utiliser dans
notre projet.

Données synthétiques et réelles :
L'article utilise à la fois des données synthétiques et des données réelles pour évaluer la méthode proposée. Cela permet de démontrer l'efficacité de l'approche dans des scénarios contrôlés ainsi que dans des situations plus proches des applications réelles, ce qui peut être intéressant pour notre travail (pas sûr de travailler sur les deux types de données, mais toujours pertinent au cas où l'on voudrait comparer).

Perspectives d'amélioration :
L'article identifie également plusieurs pistes de recherche futures, telles que l'exploration de la segmentation de documents par langue, le réglage des paramètres du modèle et l'extension à l'identification de langues "inconnues". Ces suggestions pourraient nous inspirer si l'on souhaite développer de nouvelles approches ou améliorer les méthodes existantes dans notre projet (c'est surtout un idéal à viser, mais pas sûr qu'on puisse le faire
actuellement).

Cet article fournit donc une contribution significative à la détection de la langue en proposant une méthode novatrice, en l'évaluant de manière approfondie et en identifiant des pistes de recherche futures. Ces éléments peuvent enrichir notre projet en nous fournissant des idées, des méthodes et des résultats pour aborder efficacement la détection de la langue dans divers contextes.

Native Language Identification with Large Language Models

Dans le cadre de notre étude sur l'identification automatique de la langue dans les documents, nous nous sommes penchés sur les travaux récents dans ce domaine, notamment l'article intitulé 'Native Language Identification with Large Language Models'. Le texte discute de l'application des grands modèles de langue (LLMs), comme GPT-4, dans l'identification de la langue maternelle (NLI), qui consiste à prédire la première langue d'un écrivain sur la base de ses écrits dans une seconde langue, par exemple l'anglais. Les expériences montrent que les LLMs, en particulier GPT-4, atteignent une grande précision dans la classification NLI sans nécessité de classes prédéfinies, offrant des avantages potentiels pour l'apprentissage des langues, la linguistique judiciaire et l'explicabilité des décisions des modèles. Les principales découvertes de l'article incluent l'application réussie de GPT-4 établissant un nouveau record de performance avec une précision de 91.7% sur l'ensemble de tests TOEFL11 dans un cadre sans exemple préalable.

Les LLMs comme GPT-4 démontrent une compétence en identification de la langue maternelle (NLI) sans être limités à un ensemble de classes connues, offrant de nouvelles possibilités pour des applications réelles dans des domaines tels que l'acquisition d'une seconde langue et la linguistique judiciaire. De plus, les implications pratiques de l'utilisation de LLMs comme GPT-4 pour le NLI incluent la capacité à fournir des explications pour l'identification de la langue basée sur des erreurs orthographiques, des motifs syntaxiques et des motifs linguistiques traduits. La méthode décrite dans l'article implique la conduite d'expériences dans un cadre sans exemple préalable, où les modèles sont évalués sans données d'entraînement spécifiques pour l'identification de la langue maternelle (NLI). Elle est également renforcée par une analyse comparative contre des modèles établis d'interférence linguistique, soutenue par des recherches linguistiques et des expériences passées. Bien que les évaluations initiales puissent parfois nécessiter un raffinement supplémentaire, l'application systématique de ces étapes permet une
prédiction robuste et informée sur la langue maternelle de l'auteur basée sur les traces linguistiques qu'il laisse dans son écriture en anglais. Les chercheurs utilisent des modèles de langue de haute performance (LLMs) tels que GPT-3.5 et GPT-4, accédant à l'ensemble de données TOEFL11 pour l'évaluation. Cet ensemble de données comprend des essais en anglais écrits par des locuteurs natifs de 11 langues différentes. Dans leurs expériences, ils se concentrent sur la performance de classification NLI en comparant à quel point différents LLMs peuvent prédire avec précision la langue maternelle des auteurs des textes sans avoir été spécifiquement formés pour cette tâche. Ils évaluent la performance des modèles en utilisant la précision comme métrique d'évaluation principale et comparent leurs résultats avec des études précédentes pour mesurer les améliorations réalisées avec des LLMs comme GPT-3.5 et GPT-4. En général, la méthode implique l'utilisation des connaissances préexistantes issues de la préformation standard dans ces modèles avancés de langue pour déduire la langue maternelle d'un auteur sur la base de ses écrits dans une seconde langue sans aucun ajustement ou données d'entraînement spécifiques aux tâches NLI.

Prédire la langue maternelle de quelqu'un qui parle anglais comme seconde langue implique d'examiner attentivement comment il utilise l'anglais. En remarquant des erreurs spécifiques et des motifs inhabituels dans leur écriture, nous pouvons deviner leur première langue. Les résultats de l'étude montrent que les modèles GPT, en particulier GPT-4, sont très compétents dans la classification de l'identification de la langue maternelle (NLI). Cela démontre que les grands modèles de langue (LLMs) peuvent exceller dans le NLI sans classes prédéfinies et offrir des avantages potentiels pour l'apprentissage des langues et la linguistique judiciaire. L'étude souligne comment les LLMs avancés ont le potentiel d'améliorer les tâches NLI et de fournir un raisonnement explicable pour leurs décisions. Cet article est particulièrement utile pour nous de développer un système simple qui détecte automatiquement la langue d'entrée, car il met en évidence l'efficacité des grands modèles de langue, comme GPT-4, dans l'identification de la langue sans configurations complexes ni données d'entraînement spécifiques. L'utilisation de ces modèles permet de simplifier le développement de systèmes de détection linguistique, réduisant la nécessité de vastes bases de données linguistiques ou de règles algorithmiques complexes. En intégrant les capacités des LLMs, les développeurs peuvent créer
des systèmes qui détectent automatiquement et avec précision la langue des textes entrants en se basant sur l'analyse contextuelle et la reconnaissance de motifs linguistiques, facilitant ainsi l'implémentation dans diverses applications, de la traduction automatique aux interfaces utilisateur multilingues.

Dans le cadre du projet de détection de langue dans les documents, les approches présentées dans nos trois articles offrent des perspectives complémentaires et innovantes. L'article d'Anissa propose une méthode de tokenisation basée sur des modèles statistiques non supervisés pour des documents monolingues, ce qui est utile pour une identification rapide et efficace de la langue principale dans un contexte où peu de données linguistiques sont disponibles. L'article de Nicolas, quant à lui, étend cette approche à des documents multilingues en utilisant des modèles probabilistes supervisés pour évaluer les proportions des différentes langues présentes, permettant ainsi une analyse plus fine et détaillée des contenus mixtes. Enfin, l'article de Yu-Chieh apporte une dimension supplémentaire en utilisant des caractéristiques linguistiques et la structure syntaxique pour identifier la langue maternelle des auteurs à partir de leur écriture en anglais, ce qui pourrait être adapté pour améliorer la précision des techniques de détection de langue en général. Ensemble, ces méthodes pourraient être intégrées pour créer un système robuste et adaptatif, capable de traiter efficacement aussi bien les documents monolingues que multilingues, enrichissant ainsi notre capacité à analyser divers types de contenus textuels. En outre, ces trois articles nous fournissent chacun une méthode qu'on peut mettre en place ou adapter dans le cadre de notre projet. Nous pourrions également utiliser ces méthodes pour les comparer entre elles afin de déterminer lesquelles sont les plus efficaces selon les ressources ou le contexte, et comprendre pourquoi c'est le cas, ce qui enrichirait encore davantage notre compréhension et notre efficacité dans le domaine de la détection de langue.

Données et Corpus

Nous avons un corpus pour chaque langue :

Concernant le corpus pour le **français**, il provient d'un projet universitaire créé par l'Université Toulouse II, et plus précisément par le département de Recherche et Développement pour l'Apprentissage des Connaissances (REDAC).
Ils ont rassemblé un grand nombre d'articles à partir de Wikipédia, qui couvrent une grande variété de sujets, ce qui rend ce corpus très riche et diversifié et donc parfait pour la détection de langue dans plus ou moins n'importe quel contexte.

Le corpus est disponible en ligne, sur le site web du département REDAC. Il est important de noter que ce corpus est régulièrement mis à jour avec de nouveaux articles de Wikipédia, ce qui le rend encore plus précieux pour nos recherches.

En utilisant ce corpus, nous pouvons explorer et analyser le langage utilisé dans ces articles, ce qui peut nous aider à mieux comprendre comment les informations sont présentées et partagées dans ce contexte particulier.

Concernant le corpus **anglais**, il vient d'une source très connue et fiable : Wikimedia, l'organisation qui gère Wikipedia. Plus précisément, nous avons téléchargé le corpus à partir du site web Wikimedia Dumps, où ils fournissent des copies régulières du contenu de Wikipedia.
On peux télécharger le corpus en anglais, en date du 1er mai 2024. Ce corpus est très volumineux, car il contient des millions d'articles de Wikipedia. (99,83Giga de texte)

Cependant, avant de pouvoir utiliser ce corpus, nous devions le nettoyer et le découper en plusieurs morceaux moins volumineux. En effet, Le corpus est à l'origine au format XML, ce qui signifie qu'il contient non seulement le texte des articles, mais aussi beaucoup d'informations structurelles et de métadonnées inutiles pour notre projet.

Pour nettoyer le corpus, nous allons utiliser un programme informatique qui va parcourir tous les fichiers XML et extraire uniquement le texte des articles.

Une fois le nettoyage terminé, nous avons donc un corpus propre et prêt à l'emploi, composé exclusivement du texte des articles de Wikipédia.

Concernant le corpus pour le **chinois**,

Pour préparer notre corpus chinois, nous commençons par consulter le site https://dumps.wikimedia.org/zhwiki/ afin de vérifier la dernière mise à jour et de télécharger le fichier « zhwiki-latest-pages-articles.xml.bz2 », qui pèse environ 2 Go. Concernant la conversion du format XML en texte, bien que plusieurs méthodes soient suggérées en ligne, nous avons opté pour une solution spécifique. Initialement, nous avons envisagé d'utiliser genism.corpora avec WikiCorpus pour cette conversion, mais avons renoncé à la lemmatisation car elle n'est plus fonctionnelle. À la place, nous utilisons un script disponible sur https://github.com/txtcn/wiki, basé sur celui du site https://spaces.ac.cn/archives/4176, également dérivé de genism.corpora.wikicorpus, mais sans la lemmatisation. Après la conversion du fichier XML en texte, nous procédons à la division du fichier en 29 parties, chacune d'environ 100 Mo. Une fois le nettoyage terminé, nous disposons ainsi d'un corpus chinois structuré à partir des articles de Wikipédia.

Méthodes et expérimentations

Yu-Chieh a tenté une approche basée sur des **réseaux de neurones** avec utilisation de **Transformers**, mais sans succès.

Anissa et Nicolas quant à eux, se sont intéressés aux **approches statistiques**, à travers des modèles comme le **modèle de mélange Gaussien** (Gaussian Mixture Model ou GMM), les **arbres de décisions**, les f**orêts aléatoires**, les **modèles multinomiaux utilisant le théorème de Bayes**, et les **machines à vecteur de support** (Support Vector Machine ou SVM).
Ainsi, avec ces différents modèles, nous pourrons faire des comparaisons pour savoir le(s)quel(s) s'en sorte(nt) le mieux et pourquoi.

Modèle de mélange Gaussien (Gaussian Mixture Model ou GMM)

GMM est un **modèle probabiliste** largement utilisé pour la modélisation de données. Il est particulièrement utile pour la **classification** et la **modélisation de densité de probabilité**.

Mélange de distributions gaussiennes :
Le modèle de mélange gaussien suppose que les données sont générées à partir d'un mélange de plusieurs distributions gaussiennes (ou normales). Chaque distribution gaussienne dans le mélange représente une composante du modèle, et chaque composante capture une partie de la variabilité des données.

Paramètres du modèle :
Pour chaque composante du mélange, le modèle de mélange gaussien possède plusieurs paramètres, notamment la moyenne, la covariance et la proportion de la composante dans le mélange. La moyenne et la covariance définissent la forme de la distribution gaussienne, tandis que la proportion détermine la pondération de chaque composante dans le mélange.

Estimation des paramètres :
L'estimation des paramètres du modèle de mélange gaussien se fait généralement à l'aide de l'**algorithme d'espérance-maximisation (EM).** Cet algorithme est un algorithme itératif qui alterne entre les étapes d'espérance (E-step) et de maximisation (M-step) pour trouver les meilleurs paramètres du modèle qui maximisent la vraisemblance des données observées.

Utilisation du modèle :
Une fois que les paramètres du modèle ont été estimés, le modèle de mélange gaussien peut être utilisé pour **estimer la probabilité qu'une nouvelle observation appartienne à chaque composante du mélange**.

Modèle basé sur des machines à vecteurs de support (SVM) (modèle d'apprentissage supervisé) avec SVC (Support Vector Classifier)

Support Vector Machine (SVM) est un **algorithme d'apprentissage automatique supervisé** utilisé pour la **classification** et la **régression.** L'objectif principal de SVM est de **trouver un hyperplan dans un espace multidimensionnel** qui **sépare** les exemples de **différentes classes** avec la marge maximale, c'est-à-dire la distance maximale entre les exemples les plus proches de chaque classe et l'hyperplan.

SVM (Support Vector Machine) :
C'est l'algorithme de base qui peut être utilisé pour la **classification binaire** et la **régression**. Il est capable de trouver l'hyperplan optimal dans l'espace des caractéristiques pour séparer les exemples de différentes classes.

SVC (Support Vector Classifier) :
C'est la **version de SVM pour la classification**. En utilisant l'algorithme d'optimisation d'un problème quadratique (QP), SVC cherche à trouver l'hyperplan qui maximise la marge tout en minimisant la classification incorrecte. Il peut gérer à la fois des problèmes de classification binaire et multiclasse.

LinearSVC (Linear Support Vector Classification) :
LinearSVC est une implémentation de SVM pour la **classification avec un noyau linéaire**. Contrairement à SVC, LinearSVC utilise une méthode d'optimisation différente basée sur l'optimisation de l'erreur de classification. Cela le rend **plus rapide que SVC** pour des ensembles de données de grande taille, mais il ne supporte **que la classification binaire**.

Autrement dit, SVM, SVC et LinearSVC sont tous des algorithmes basés sur le principe des machines à vecteurs de support pour la classification, mais ils diffèrent dans leurs implémentations spécifiques et leurs capacités. SVM est plus général et peut être utilisé pour la classification et la régression, tandis que SVC et LinearSVC sont spécifiquement conçus pour la classification. SVC peut gérer à la fois la classification binaire et multiclasse, tandis que LinearSVC est limité à la classification binaire mais est généralement plus rapide pour de grands ensembles de données.

Considérant que la tâche de **détection de langue** relève de la **classification binaire** (un texte est d'une langue ou d'une autre dans les textes qu'on estime monolingues) on décide d'utiliser **LinearSVC**.

Modèle basé sur MultinomialNB (Multinomial Naive Bayes)

Multinomial Naive Bayes (MNB) est un **modèle probabiliste** largement utilisé pour la **classification de textes avec des fonctionnalités discrètes** (telles que les mots comptés). Il s'agit d'une extension du classificateur Naive Bayes standard, qui est basé sur le théorème de Bayes et suppose l'indépendance conditionnelle des caractéristiques.

Dans le contexte de la détection de la langue, MNB peut être utilisé pour prédire la langue d'un texte donné en **calculant la probabilité que ce texte appartienne à chaque langue possible, puis en choisissant la langue avec la probabilité la plus élevée**.

Apprentissage :
Pendant la phase d'apprentissage, le modèle MNB est entraîné sur un ensemble de données contenant des exemples de **textes étiquetés avec leurs langues correspondantes**.
Le modèle apprend les **distributions de probabilité des mots** pour chaque langue.

Prédiction :
Pour prédire la langue d'un nouveau texte, le modèle calcule la **probabilité conditionnelle que le texte appartienne à chaque langue possible**, en utilisant les distributions de probabilité apprises pendant la phase d'apprentissage. Il applique ensuite le théorème de Bayes pour **estimer la probabilité a posteriori de chaque langue sachant le texte**, et **choisit la langue avec la probabilité la plus élevée comme prédiction**.

MNB est particulièrement efficace pour la classification de textes, en particulier lorsque les fonctionnalités sont des comptages de mots ou d'autres caractéristiques discrètes, et lorsque les hypothèses d'indépendance conditionnelle des caractéristiques sont raisonnables pour le problème donné. Il est également robuste aux données manquantes et fonctionne bien même avec des ensembles de données de petite taille.

Modèle basé sur DecisionTreeClassifier (Arbres de Décision)

DecisionTreeClassifier est un **algorithme d'apprentissage automatique** largement utilisé pour la **classification**. Il appartient à la famille des **arbres de décision**, qui sont des **modèles prédictifs utilisés pour représenter les décisions sous forme d'arbre**. Chaque nœud de l'arbre représente une caractéristique (ou un attribut), chaque branche représente une décision basée sur cette caractéristique, et chaque feuille représente le résultat de la décision.

Construction de l'arbre :
L'algorithme commence par diviser l'ensemble de données d'entraînement en sous-ensembles plus petits en sélectionnant la caractéristique qui fournit la meilleure division selon certains critères de qualité (comme l'indice de Gini ou l'entropie). Cette division se fait de manière récursive jusqu'à ce qu'une condition d'arrêt soit atteinte, par exemple lorsque tous les exemples dans un sous-ensemble appartiennent à la même classe, ou lorsque la profondeur maximale de l'arbre est atteinte.

Prédiction :
Une fois que l'arbre de décision est construit, il peut être utilisé pour prédire la classe cible (ou l'étiquette) pour de nouvelles instances en les traversant de haut en bas selon les règles de décision apprises pendant la phase de construction. Chaque instance est attribuée à la classe majoritaire dans le nœud feuille correspondant.

Les avantages des arbres de décision sont leur facilité d'interprétation et leur capacité à gérer à la fois des données numériques et catégoriques. De plus, ils **n'exigent généralement pas beaucoup de prétraitement des données**, comme la normalisation des caractéristiques. Cependant, ils ont **tendance à être sensibles au surajustement (overfitting)** lorsqu'ils sont trop profonds et qu'ils captent des détails spécifiques aux données d'entraînement qui ne généralisent pas bien aux nouvelles données. Cela peut être atténué en utilisant des techniques de régularisation ou en limitant la profondeur de l'arbre.

Modèle basé sur RandomForestClassifier (Forêts Aléatoires)

RandomForestClassifier est un **algorithme d'apprentissage supervisé** qui fait partie de la famille des **méthodes ensemblistes**, plus précisément des méthodes de **forêts aléatoires (Random Forests)**. Il est largement utilisé pour la **classification**, en particulier pour les problèmes dans lesquels les données présentent une **grande dimensionnalité et une complexité élevée**.

Création d'un ensemble d'arbres de décision :
L'algorithme crée un **ensemble (ou une forêt) d'arbres de décision pendant la phase d'entraînement**. Chaque arbre est construit à partir d'un **échantillon aléatoire** de l'ensemble de données d'entraînement (bootstrap) et utilise un **sous-ensemble aléatoire des caractéristiques** disponibles à chaque division de l'arbre.

Entraînement des arbres de décision :
Chaque **arbre est entraîné de manière indépendante sur son échantillon** bootstrap. L'objectif est de créer des arbres qui sont à la fois **diversifiés et précis**.

Prédiction par vote majoritaire :
Lors de la prédiction, **chaque arbre de la forêt donne une prédiction de classe pour une nouvelle instance**. La classe prédite est déterminée par un vote majoritaire parmi tous les arbres de la forêt. C'est-à-dire que la **classe finale attribuée à l'instance est celle qui a reçu le plus grand nombre de votes parmi tous les arbres**.

RandomForestClassifier présente plusieurs avantages :

- Il réduit le risque de surajustement (overfitting) par rapport à un seul arbre de décision en combinant les prédictions de plusieurs arbres.
- Il est capable de gérer des ensembles de données avec de grandes dimensions et des caractéristiques hautement corrélées.
- Il fournit des mesures d'importance des caractéristiques, qui peuvent être utiles pour l'analyse des données.

Cependant, il peut être plus coûteux en termes de temps de calcul et de mémoire par rapport à un seul arbre de décision, en raison de la création et de l'entraînement de multiples arbres. De plus, l'interprétation des résultats peut être plus complexe en raison de la présence de multiples arbres dans la forêt.

Modèle de NLI(Native Language Identification)

L'expérience visait à développer des modèles pour détecter la langue d'un texte donné et déterminer si les textes en anglais sont écrits par des locuteurs natifs ou non natifs, en utilisant des données textuelles en anglais, français et chinois.
Les textes ont été traités dans un tableau de données avec des étiquettes de langue correspondantes, et le texte anglais a été divisé en phrases étiquetées comme "Natif" ou "Non-Natif". Des n-grammes de caractères et des caractéristiques TF-IDF ont été extraits à l'aide de CountVectorizer et TfidfVectorizer, respectivement, et deux modèles MultinomialNB ont été entraînés pour la détection de la langue et la détection natif/non-natif. Les deux modèles ont démontré une grande précision et des performances robustes, les rapports de classification montrant des métriques fiables. L'expérience a démontré avec succès l'efficacité de ces méthodes, suggérant que les travaux futurs pourraient inclure l'extension du jeu de données et l'exploration de modèles plus avancés.

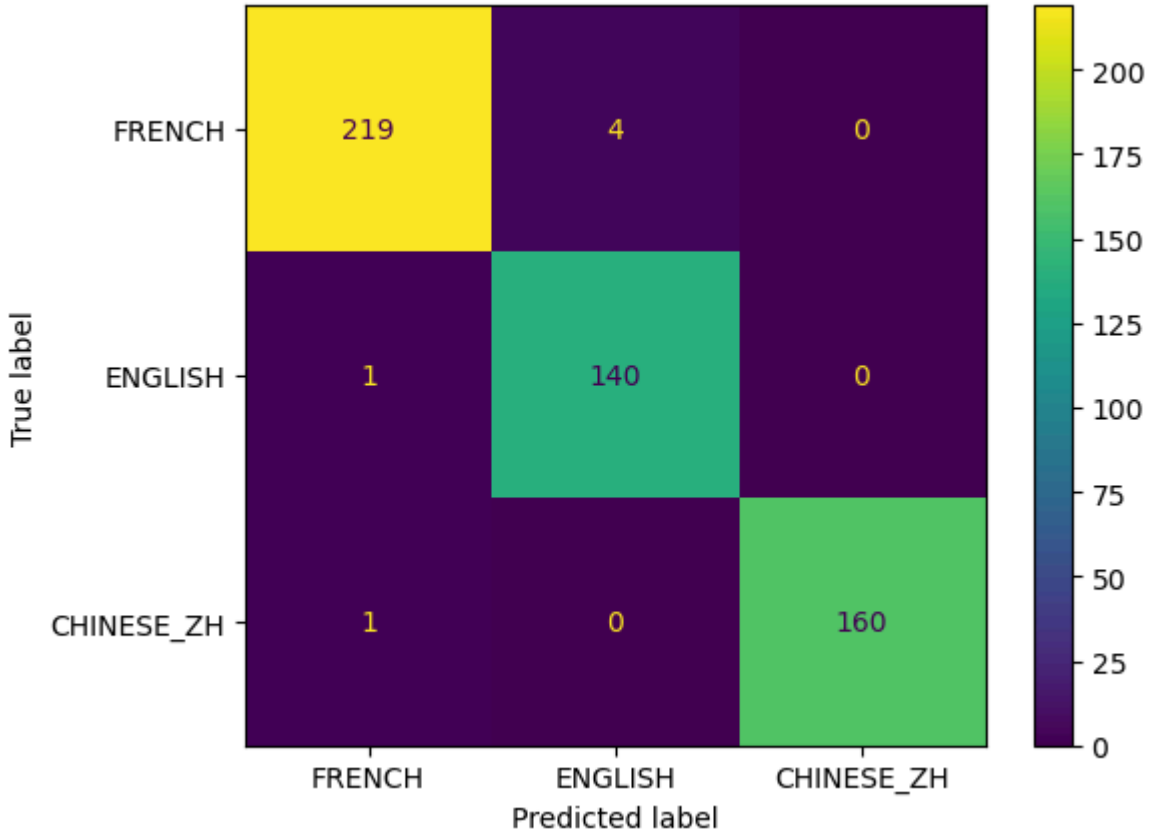
Évaluation et résultats

Modèle de mélange Gaussien (Gaussian Mixture Model ou GMM)

D’après la matrice de confusion, un résultat équilibré pratiquement parfait pour le chinois (160 trouvés et corrects sur 161 et 1 chinois non trouvé (étiquetté en français) et aucun bruit) et l'anglais (140 trouvés et corrects sur 141 et 1 anglais non trouvé étiquetté en français et aucun bruit), un peu moins bien pour le français mais qui reste excellent (219 trouvés et corrects pour 4 non trouvés et étiquetté en anglais, et 1 anglais et 1 chinois étiquetté avec erreur en français).
Globalement d’excellents résultats donc, même avec des langues très éloignées des langues occidentales qu’on a plus l’habitude de manipuler avec ces outils. Ici précisément, on peut même remarquer que la précision et le rappel pour le chinois sont supérieures à celle du français et de l’anglais.

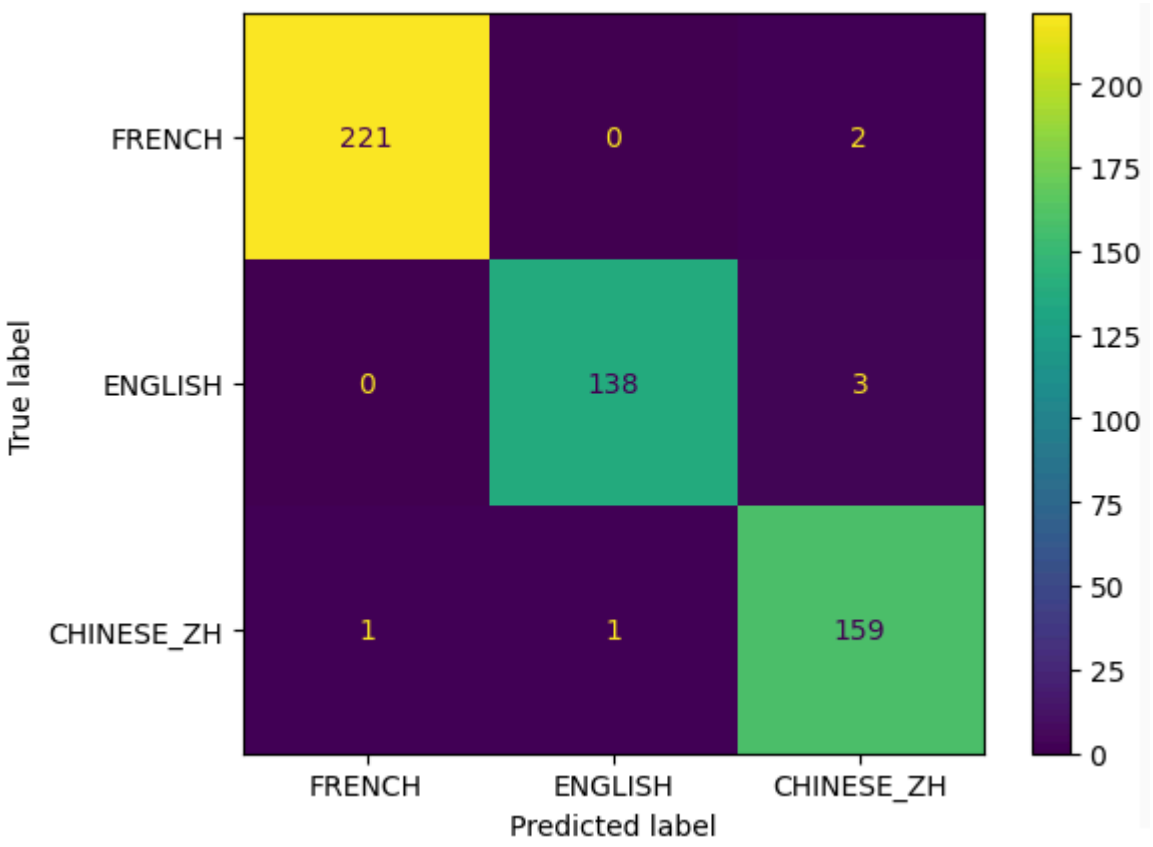
```
Précision FRENCH: 0.9909502262443439
Précision ENGLISH: 0.9722222222222222
Précision CHINESE_ZH: 1.0

Rappel FRENCH: 0.9820627802690582
Rappel ENGLISH: 0.9929078014184397
Rappel CHINESE_ZH: 0.9937888198757764
```



Modèle basé sur des machines à vecteurs de support (SVM) (modèle d'apprentissage supervisé) avec SVC (Support Vector Classifier)

D'après sa matrice de confusion, résultats avec SVC équilibrés : très peu d'erreurs que ce soit pour le français (2 français ont été prédit comme du chinois et 1 chinois a été prédit comme du français), l'anglais (2 anglais prédit comme du chinois et 1 chinois prédit comme de l'anglais) ou le chinois.
Qualité des résultats à peu près équivalente (mais un peu moins bien quand même) au modèle de mélange Gaussien (GMM) vu précédemment.



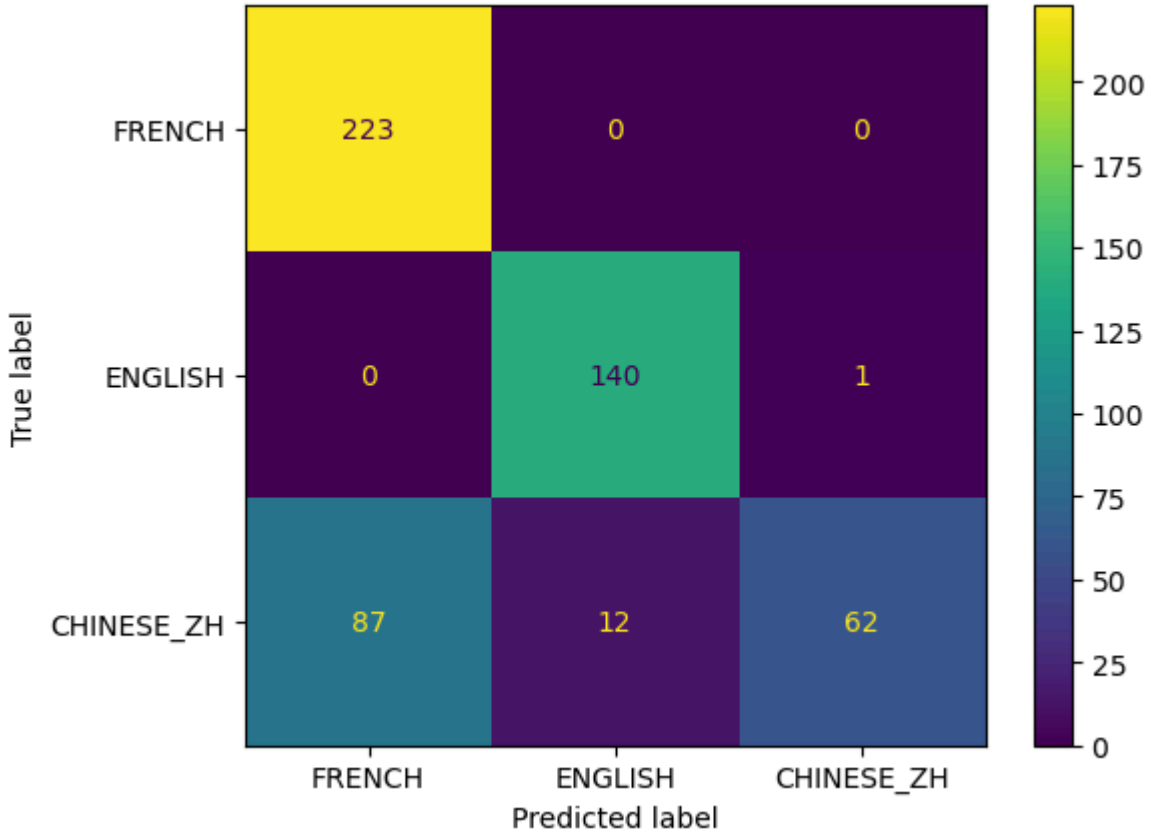

```
#Score avec BoW (Bag of Words)
print(clf_svc_bow.score(X_test_bow, labels_test))
```

```
0.9866666666666667
```

Modèle basé sur MultinomialNB (Multinomial Naive Bayes)

Les résultats obtenus avec le modèle MultinomialNB (Naive Bayes multinomial) sont décevants pour la langue chinoise, car il ne parvient pas à détecter la majorité des échantillons en chinois, les confondant souvent avec du français. Cependant, les performances du modèle sont excellentes pour l'anglais et parfaites pour le français.

De plus, les résultats globaux de ce modèle sont inférieurs à ceux des modèles précédents. Il est à noter que l'hypothèse d'indépendance conditionnelle des caractéristiques n'est pas respectée dans ce cas, ce qui pourrait expliquer en partie les résultats plus mitigés par rapport aux deux autres modèles.




```
#Score avec BoW (Bag of Words)
print(clf_multinb_bow.score(X_test_bow, labels_test))
```

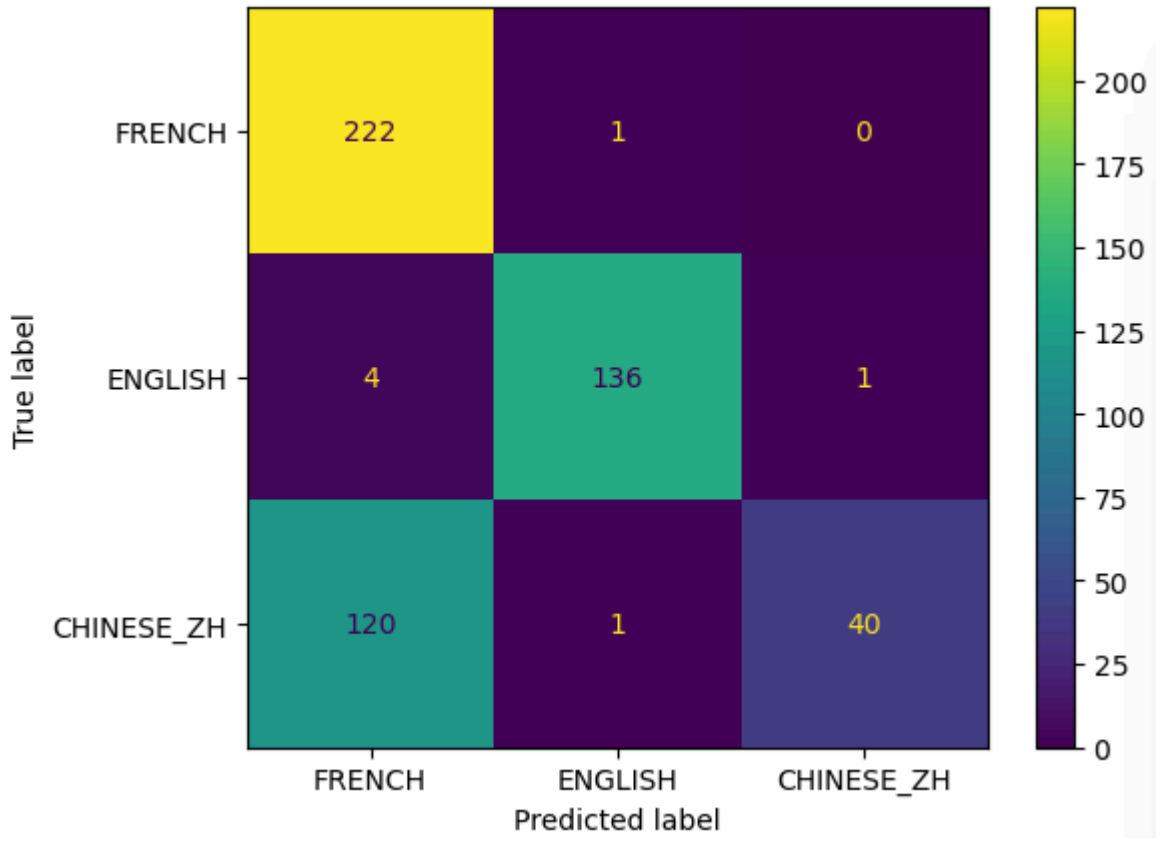
```
0.8095238095238095
```

Modèle basé sur DecisionTreeClassifier (Arbres de Décision)

Résultats catastrophiques avec un modèle basé sur DecisionTreeClassifier (Arbres de décision) pour le chinois (détection comme du français d'un très grand nombre de textes chinois), mais excellent pour l'anglais et pratiquement parfait pour le français.
Un score encore un peu plus bas.
Dans tous les cas, éviter d'utiliser ce modèle pour la détection de langues : on a beaucoup mieux !

```
#Score avec BoW (Bag of Words)
print(clf_decisiontree_bow.score(X_test_bow, labels_test))
```

```
0.758095238095238
```



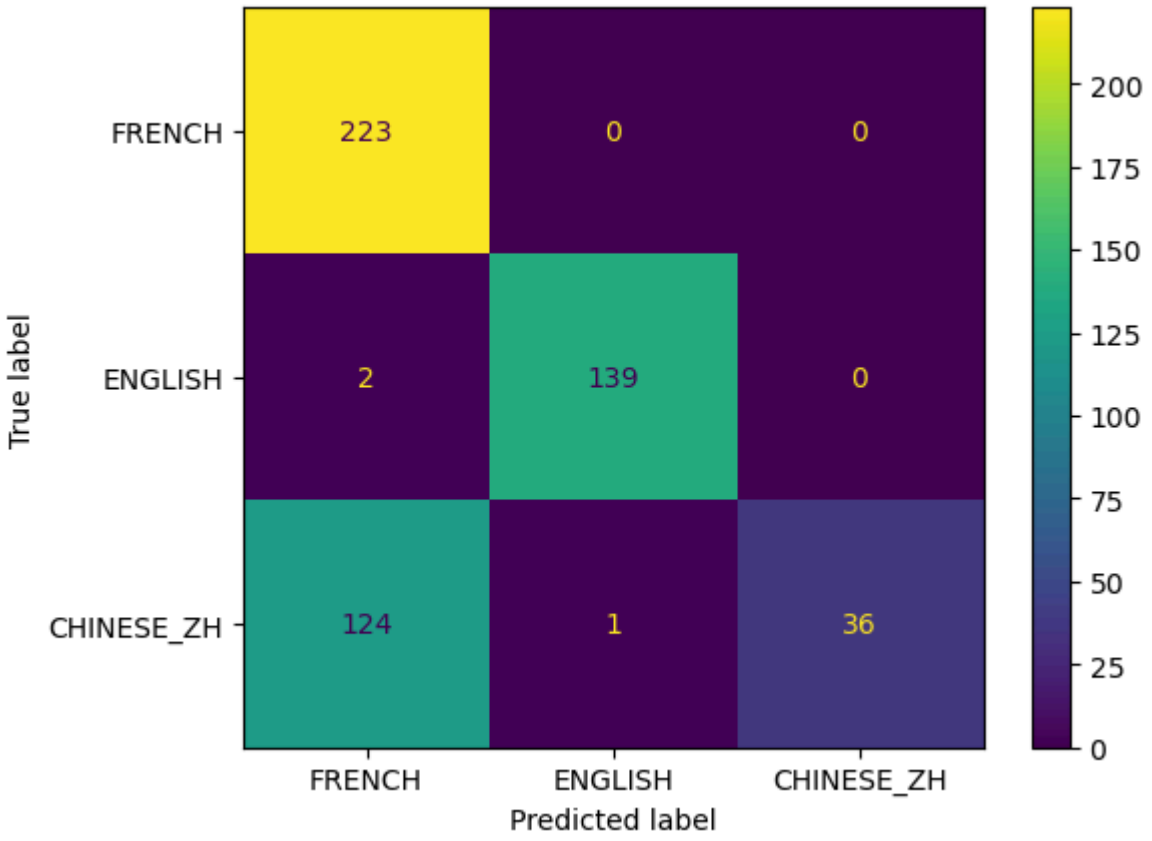
Modèle basé sur RandomForestClassifier (Forêts Aléatoires)

Résultats très nuls avec un modèle basé sur RandomForestClassifier (Forêts aléatoires) pour le chinois, mais excellent pour l'anglais et parfait pour le français.

On aurait pu s'attendre à des résultats meilleurs qu'avec les arbres de décisions vu précédemment, puisque les forêts aléatoires sont censées être de meilleures alternatives… Mais ce n'est pas le cas… C'est avec ce modèle ainsi que les arbres de décision qu'on a les plus faibles résultats pour les modèles statistiques…

```
#Score avec BoW (Bag of Words)
print(clf_randforest_bow.score(X_test_bow, labels_test))
```

```
0.758095238095238
```



Ainsi, pour un modèle statistique de détection de langues, mieux vaut s'orienter vers un GMM ou un modèle basé sur

```
Language Detection Accuracy: 1.0
Language Classification Report:
              precision    recall  f1-score   support

     Chinese       1.00      1.00      1.00         1
     English       1.00      1.00      1.00         1
      French       1.00      1.00      1.00         1

    accuracy                           1.00         3
   macro avg       1.00      1.00      1.00         3
weighted avg       1.00      1.00      1.00         3

English Native/Non-Native Detection Accuracy: 1.0
English Text Classification Report:
              precision    recall  f1-score   support

      Native       1.00      1.00      1.00         6
  Non-Native       1.00      1.00      1.00         4

    accuracy                           1.00        10
   macro avg       1.00      1.00      1.00        10
weighted avg       1.00      1.00      1.00        10
```

SVC, car ce sont ces modèles qui présentent les meilleures performances.

Précision Globale

Précision Globale: 100% pour la classification des langues et des natifs/non-natifs. Précision, Rappel, F1-Score: Chaque métrique a obtenu un score parfait de 1.00 dans toutes les catégories, indiquant des performances exceptionnelles du modèle.

Classification des Langues

Langues Évaluées: Chinois, Anglais, Français Support: Chaque langue avait un seul point de données contribuant aux métriques d'évaluation.

Classification des Textes en Anglais

Catégories: Natif et Non-Natif Performance: Détection parfaite avec une capacité égale à identifier les textes natifs et non-natifs. Support: Natif (6 instances), Non-Natif (4 instances).

Moyennes d'Évaluation

Moyenne Macro: Moyenne non pondérée à travers toutes les catégories. Moyenne Pondérée: Prend en compte le nombre d'instances dans chaque catégorie, fournissant une vue équilibrée de la précision du modèle.

Cependant, même si les résultats du modèle sont bons, ils ne peuvent pas être utilisés pour la détection de langue réelle en raison du manque de données et de l'absence de modèles neuronaux plus sophistiqués. La taille limitée du jeu de données réduit la capacité du modèle à généraliser à des textes nouveaux et variés. De plus, l'utilisation de modèles neuronaux plus avancés, tels que les réseaux de neurones profonds, pourrait améliorer la précision et la robustesse des résultats, permettant une meilleure application en conditions réelles.

Perspectives d'amélioration

Le projet a rencontré plusieurs défis, principalement dans la préparation et la gestion du corpus. Les efforts initiaux avec un ensemble de données limité ont conduit à l'expansion vers l'ensemble du corpus de Wikipédia(le corpus fait à main c'est ok), ce qui a introduit des difficultés opérationnelles en raison de sa taille, en particulier avec le corpus chinois et les problèmes d'encodage dans le corpus français. Développer un modèle de détection de langue en utilisant le Python de base était complexe, soulignant la nécessité d'utiliser des modèles avancés basés sur des transformateurs. Pour améliorer, il est essentiel de commencer par des ensembles de données plus petits, de standardiser l'encodage, de tirer parti des cadres de machine learning avancés et d'utiliser des modèles de transformateurs pré-entraînés pour améliorer l'efficacité et les performances.

Difficultés rencontrées pendant le projet

Le défi initial était de préparer le corpus. Nous avons d'abord sélectionné quatre sujets divers qui semblaient universellement pertinents pour l'entraînement de nos modèles. Cependant, après avoir consulté notre professeur, nous avons décidé d'élargir notre portée et d'utiliser l'ensemble du corpus de Wikipédia, y compris les versions en anglais, français et chinois.

Bien que les ensembles de données de Wikipédia soient complets et bien structurés, leur taille immense a posé des défis opérationnels significatifs. Par exemple, le corpus chinois complet à lui seul représente environ 2 Go. Même après les avoir segmentés en fichiers plus petits de 100 Mo, nous avons constaté que la performance du modèle ralentissait considérablement. De plus, le corpus français, encodé en Windows-1252, a présenté d'autres complications, entravant l'efficacité du traitement des données.

En outre, inspiré par une étude que j'ai consultée, j'ai tenté de développer un modèle de détection de langue qui identifie les caractéristiques linguistiques courantes telles que l'orthographe, la typographie, la syntaxe et la grammaire. Reproduire un modèle similaire en utilisant uniquement Python de base s'est avéré complexe.

L'article de référence utilisait des invites pour demander à ChatGPT d'identifier la langue maternelle d'une personne. Suivant cette approche, j'ai également sollicité ChatGPT pour comprendre ses mécanismes sous-jacents. Il est devenu évident que l'utilisation d'un modèle basé sur un transformateur est essentielle pour analyser et mettre en œuvre une détection de langue efficace.