Natural Language Processing
Final Project

Prédiction du chef de ménage

réalisé par

BALLU Camille

camille.ballu@ensae.fr

Année académique 2023/2024

1 Présentation de la tâche

Les données sont extraites du projet Socface. Son objectif est de rassembler et de traiter par reconnaissance automatique de l'écriture les listes nominatives manuscrites des recensements de 1836 à 1936. Produites tous les cinq ans, ces listes sont organisées dans l'espace et résument les informations du recensement, en listant chaque individu avec certaines de ses caractéristiques, par exemple son nom, son année de naissance ou sa profession. Le projet Socface vise à produire une base de données de tous les individus ayant vécu en France entre 1836 et 1936. L'objectif de ce projet est de mettre au point un modèle efficace pour prédire si un individu est chef de ménage dans la base de données Socface et d'analyser les conditions de son application à l'ensemble du corpus La mention de chef de ménage dans les données permet entre autres de grouper les individus par ménage : tous les individus non chefs peuvent être associés au ménage du chef les précédant.

2 Présentation et description des données

La base de données est un échantillon extrait de la base de données Socface transmis sous format json dont l'ouverture s'effectue à partir de la reconnaissance d'un certain nombre de tokens correspondant aux caractéristiques transmises au cours du recensement par chaque individu. L'échantillon contient 20729 lignes une fois nettoyé des lignes ne contenant aucune information et des lignes ne contenant pas d'information quant au "link", à savoir quant au statut de la personne au sein de la famille. La base de données des chefs de famille contient 4619 lignes. La base de données requiert assez peu de nettoyage pour l'exploiter. La première opération de nettoyage consiste à l'ouverture du fichier à identifier les nouvelles pages scannées grâce aux liens d'images contenus dans le fichier json pour remplacer toutes les cellules qui contiennent le terme "idem" par celles qui les précèdent dans la colonne à l'exception de celles qui sont présentées en début de page (dans la mesure où les pages sont transmises dans le désordre). La deuxième opération de nettoyage consiste à homogénéiser les cas de non-réponse ou de réponse "néant" en tant que NaN.

2.1 Statistiques descriptives

Les statistiques descriptives conduites sur le dataset des chef de famille permettent de mettre en évidence certaines tendances marquées. Les chefs de famille sont en très grande majorité des hommes (Hommes à 78.9%, Veufs à 8% et Garçons à 2.4%, il est intéressant de noter également que le statut civil pour lequel les femmes sont les plus représentées dans le dataset des chefs de famille avec 8% est "Veuve"), autour de la cinquantaine (l'âge moyen du dataset est de 50 ans, l'âge médian de 49) nés en grande majorité entre 1850 et 1900. Les prénoms qui reviennent le plus souvent parmi les chefs de famille sont Jean, Pierre, Marie et Louis qui sont également les prénoms les plus donnés dans le dataset originel, et l'occupation très fortement majoritaire parmi les chefs de famille est celle de cultivateur (qui correspond aux personnes ayant reporté dans "occupation" soit "cultivateur", soit "cult", soit "cultivat"). La deuxième occupation est celle de journalier suivie de propriétaire.

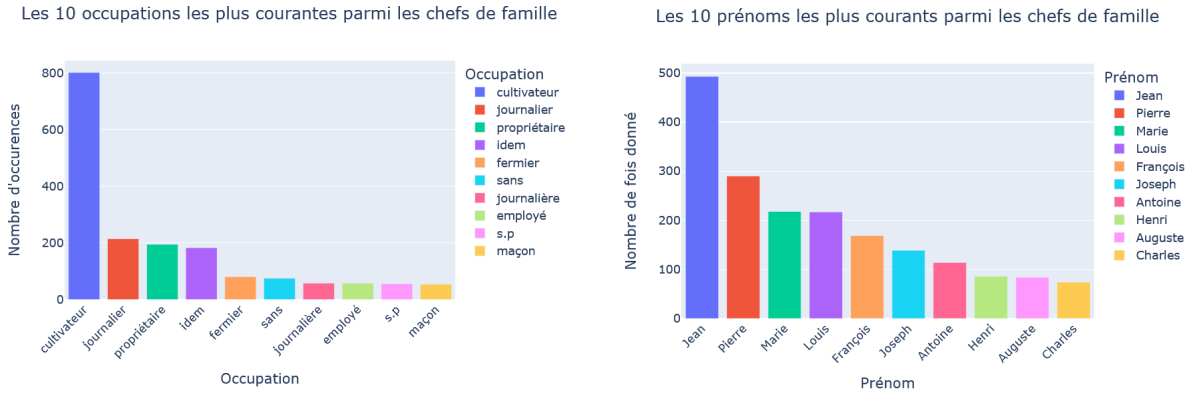

FIGURE 1 – Les 10 occupations et prénoms les plus observés parmi les chefs de famille

Les Wordclouds effectués sur les prénoms, les occupations, les noms de famille et le statut civil permettent de souligner ces tendances.

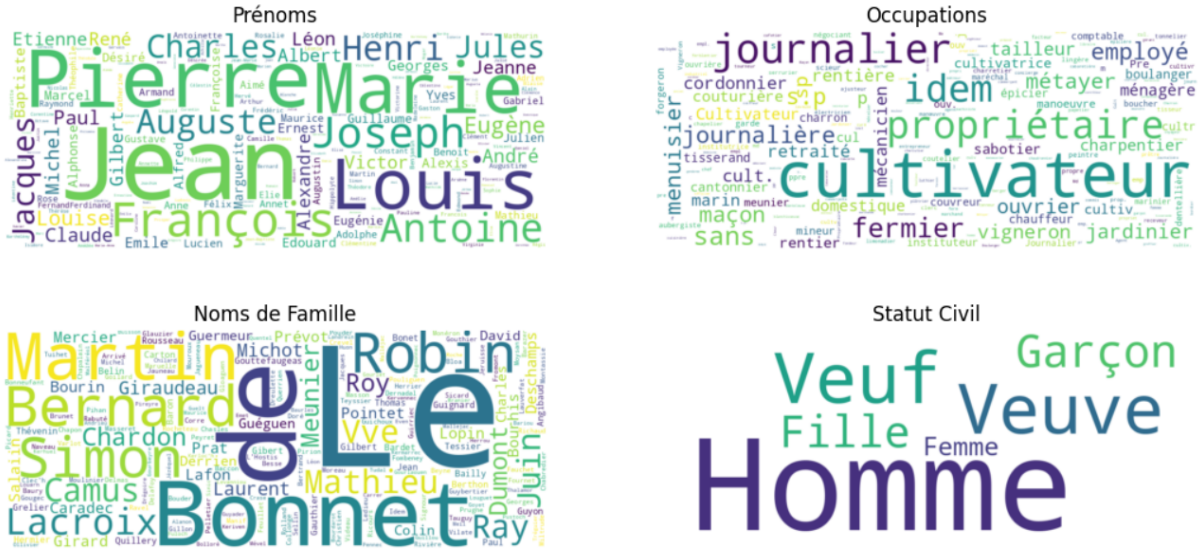

FIGURE 2 – Wordclouds sur les prénoms, occupations, noms de famille, et statuts civil au sein des chefs de famille

2.2 Taille des données cibles et données externes utilisables

La taille de l'extrait de la base de données transmis pour la tâche paraît assez pertinente pour entraîner et tester le modèle au vu de la distribution des chefs parmi le total d'individus recensés. Le modèle qui sera présenté ci-dessous devrait pouvoir être testé sur l'intégralité des données scannées au vu de ses performances sur GPU, que ce soit au vu de sa complexité, des caractéristiques retenues pour l'entraînement du modèle (toutes comme il sera présenté ci-dessous) et des performances souhaitées (environ 0,9 ou plus d'Accuracy).

Pour trouver des données externes qui pourraient être exploitées dans le cadre de la tâche, il faudrait s'orienter vers des données supplémentaires de recensement (enregistrements de recensement de différentes périodes ou régions par exemple), des données démographiques complémentaires (distribution des âges à l'époque, taille des ménages, niveaux de revenus et niveaux d'éducation par exemple) ou encore des données géospatiales (localisation, les caractéristiques du quartier et les classifications urbaines/rurales) ou historiques (changements sociétaux, aux tendances économiques et aux changements politiques). On peut espérer dans le cas français trouver certaines de ces données de façon publique sur des sites d'agences gouvernementales, d'instituts de recherche ou de portails de données ouvertes. Le site internet FranceArchives, portail national des archives françaises, peut à cet égard contenir des données intéressantes, notamment dans la section Généalogie et famille qui contient un panorama de documents archivés d'Etat civil qu'il s'agisse de registres paroissiaux et d'état civil, des recensements de la population, des registres matricules, de délibérations communales ou départementales, des cartes et plans, de photographies et cartes postales ou de documents audiovisuels.

3 Analyse des modèles comparatifs

Afin d'affiner le choix du modèle dans le cadre de la tâche à effectuer, quelques tests ont été réalisés en termes de représentations Bag-of-Words et TF-IDF en utilisant deux algorithmes différents pour des tâches de classification : le classificateur Multinomial Naive Bayes (MultinomialNB) et le classificateur RandomClassifier.

Les résultats obtenus avec le classificateur MultinomialNB, basé sur le théorème de Bayes, sont dans l'ensemble pas bons, ou moins bon qu'avec l'usage de CAMEMBERT quand nous le verrons par la suite. Le

classificateur est en effet principalement utilisé pour les tâches de classification avec des caractéristiques discrètes et adapté aux problèmes de classification de textes où les caractéristiques représentent des nombres ou des fréquences d'occurrences ce qui n'est pas le cas ici. Il fonctionne bien avec les ensembles de données contenant un nombre relativement important de caractéristiques.

	Bag of Words	**TF-IDF**
F1-score	0.676	0.727
Accuracy	0.882	0.891

Table 1 – Métriques de performances pour le MultinomialNB

Les résultats obtenus avec le Radomclassifier (un classificateur utilisé comme point de référence pour comparer les performances d'autres classificateurs) sont également mitigés. Il attribue de manière aléatoire des étiquettes sans tenir compte des caractéristiques ou des modèles présents dans les données. Il est généralement utilisé pour évaluer si un classificateur plus sophistiqué est nettement plus performant que le hasard.

	TF-IDF	**Fast-Text**
F1-score	0.611	0.725
Accuracy	0.864	0.887

Table 2 – Métriques de performances pour le RandomClassifier

4 Expérimentation

4.1 Tokenization et préparation des données

Le processus de tokenization consiste à convertir un texte brut en un format adapté à l'alimentation d'un modèle de réseau de neurones. La tokenization est ici effectuée à l'aide du tokenizer Camembert, en particulier CamembertTokenizerFast, adapté de RobertaTokenizer et XLNetTokenizer, entrâiné sur un large corpus de textes français et qui est optimisé pour la vitesse. La tokenization est effectuée au niveau du mot, ce qui signifie que le texte est divisé en mots individuels ou tokens. Le tokenizer traite le texte d'entrée et le segmente en mots sur la base des espaces et des signes de ponctuation. Les integrer labels sont ensuite convertis en OneHot Econders. Cela garantit que chaque label est représenté sous la forme d'un vecteur binaire dont un seul élément (correspondant à la classe) est 1 et les autres sont des 0. Le OneHot Encoding permet de prévenir des biais d'ordonnancement en représentant chaque classe de manière indépendante et de calculer certaines fonctions de perte couramment utilisées dans les tâches de classification, telles que la cross-entropy loss. Cela permet aussi de mieux distinguer les classes dans le calcul du gradient. Les train, test et validation datasets sont crées en utilisant la custom Dataset class de Pytorch.

4.2 Modèle CamembertForSequenceClassification

Le modèle ici utilisé est le modèle basé sur CAMEMBERT pour la classification des séquences (CamembertForSequenceClassification). Il est initialisé à partir de poids pré-entraînés en utilisant ('camembert-base'). Le modèle est déplacé vers le dispositif approprié (GPU) en utilisant model.to(device). L'optimiseur AdamW est utilisé pour l'apprentissage avec un taux d'apprentissage de 5e-5. La boucle d'apprentissage s'exécute pendant 3 époques. À chaque époque, les données d'apprentissage sont itérées par lots à l'aide d'un DataLoader. Les paramètres du modèle sont mis à jour en fonction de la perte calculée à l'aide de la rétropropagation. Après l'apprentissage, le modèle passe en mode évaluation à l'aide de model.eval().

AdamW est un algorithme d'optimisation qui s'appuie sur l'optimiseur Adam, Adaptive Moment Estimation. AdamW apporte une modification à l'algorithme Adam original (un algorithme d'optimisation adaptatif couramment utilisé pour la formation des réseaux neuronaux et qui maintient des taux d'apprentissage par paramètre, adaptant les taux d'apprentissage en fonction des estimations des premiers et seconds moments des gradients et intègre des mécanismes de correction des biais pour atténuer les biais dans les estimations) afin de tenir compte de la décroissance des poids (une technique utilisée pour empêcher l'ajustement excessif en ajoutant un terme de pénalité à la fonction de perte qui diminue les poids importants), qui est une forme de régularisation couramment utilisée dans l'apprentissage des réseaux de neurones.

4.3 Résultats

En entraînant le modèle sur 8 variables (age, civil_status, occupation, firstname, lob, nationality, employer et surname_household), les résultats obtenus sont très satisfaisants :

- **Accuracy** : 0.9520651190835092
- **F1 Score** : 0.9523701748961737

Le modèle s'exécute sous GPU en un peu plus de 4 minutes. La matrice de confusion est présentée ci-dessous. Le modèle est très performant pour les vrais positifs et vrais négatifs et moins performant sur les faux négatifs (66) mais surtout sur les faux positifs (93). Ce qui veut dire que le modèle a une tendance à assigner un statut de chef à des individus qui ne le sont pas. L'observation des tableaux de faux positifs et de faux négatifs semble montrer que les faux positifs sont bien souvent des hommes (ou du moins des personnes portant un prénom d'homme car le statut civil est bien souvent manquant) pour lesquels peu d'informations annexes sont transmises ou qui occupent des fonctions d'agriculteur ou de cultivateur et que les faux négatifs sont davantage des femmes ou des hommes (encore une fois le statut civil est manquant la plupart du temps) qui n'occupent pas des fonctions d'agriculteur et de cultivateur. Il est intéressant de noter la présence d'un certain nombre de personnes portant le prénom ambidextre de "Claude" dans le dataset des faux positifs ce qui pourrait suggérer qu'il est intéressant de coupler l'analyse effectuée dans cette tâche avec l'analyse de la première tâche consistant à prédire le genre des individus.

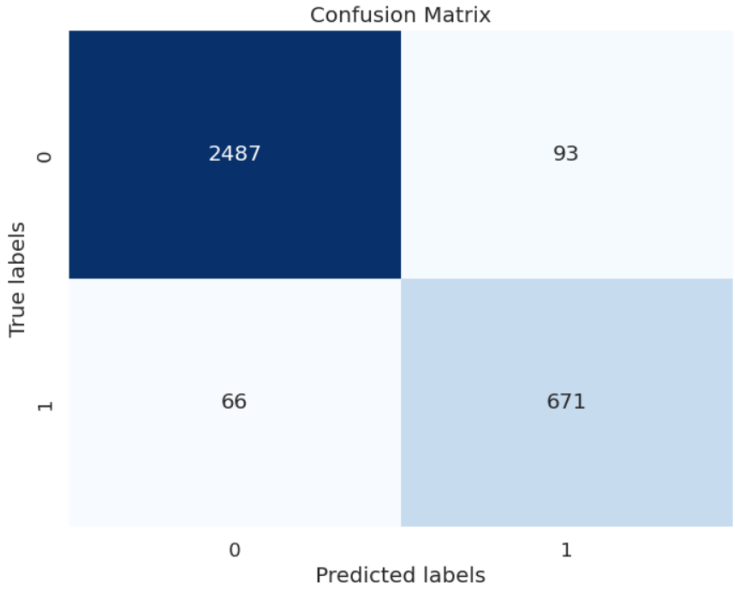

FIGURE 3 – Matrice de confusion du modèle

4.4 Jouer sur les paramètres

4.4.1 Paramètres d'entrée

Les résultats ne sont pas les mêmes en fonction du nombre de colonnes que l'on prend en compte pour l'apprentissage. En ne conservant que age,civil_status, occupation et firstname, l'accuracy est de 0.87 ce qui est inférieur au résultat obtenu lorsque nous incluons 8 colonnes. Toutefois le temps d'exécution du modèle est rallongé. C'est là tout l'enjeu de l'adaptation du modèle choisi à la très large base de données Socface.

4.4.2 Nombre d'époques

En augmentant le nombre d'époques à 5, le temps d'exécution est beaucoup plus long et passe à plus de 10 minutes. C'est ici qu'intervient l'arbitrage temps d'entraînement et taille du dataset. Avec 5 époques, l'Accuracy n'augmente que de 0.003 Les résultats étant déjà très satisfaisants pour 3 époques, il semble raisonnable de conserver ce paramètre.

- **Accuracy** : 0.9550798914681942
- **F1 Score** : 0.9559297565011058

4.4.3 Learning rate

L'augmentation du taux d'apprentissage (Learning Rate) en passant de 5e-5 à 1e-3 conduit à une convergence plus rapide, mais qui a de forts risques de dépasser le minimum. C'est le cas comme le met en évidence la matrice de confusion dans ce cas de figure, le modèle est incapable de reconnaître les positifs et les faux positifs.

4.4.4 Batch size

La modification du batch size peut avoir plusieurs effets, comme des temps d'apprentissage plus rapides par époque lorsque le batch size augmente car davantage de données sont traitées en parallèle. Lorsque le batch size diminue toutefois cela peut produire davantage de bruit pendant l'entraînement ce qui peut aider le modèle à mieux se généraliser à des données inédites. Toutefois l'augmentation du batch size nécessite plus de mémoire. Si la mémoire du GPU est limitée, l'augmentation peut entraîner des erreurs de mémoire. Le test de passer à un batch size de 20 met en évidence des résultats en terme d'Accuracy et de F1-Score très similaires au modèle avec 16 batch.

- **Accuracy** : 0.9520651190835092
- **F1 Score** : 0.9521293510871894

5 Conclusion

Nous avons mis en évidence au cours de ce projet un modèle performant de prédiction des chefs de famille sur un extrait de la base de données Socface. L'enjeu est de pouvoir par la suite adapter ce modèle à l'intégralité des données qui sont collectées dans le cadre de ce projet pour pouvoir notamment faire des analyses à l'échelle du foyer. Le modèle tourne déjà rapidement mais le temps d'exécution pourrait encore être amélioré. Il serait par ailleurs intéressant comme évoqué de compléter la base par des données d'état civil qui pourraient contribuer à améliorer les performances de prédiction du modèle.

Références