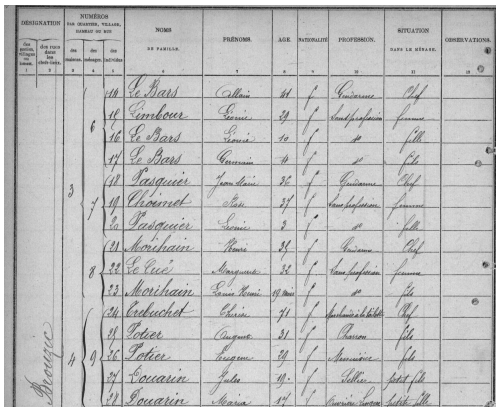
École Nationale de la Statistique et de l'Administration Economique

Natural Language Processing

Prédiction automatique du sexe



Arman Akgönül

Encadrant : Christopher Kermorvant

5 avril 2024

1 Présentation de la tâche et des données

1.1 Prédiction automatique du sexe

Notre objectif est de prédire le sexe de chaque personne décrite dans des manuscrits du XIXeme siècle. Pour ce faire, nous avons différentes informations comme son prénom, sa profession ou encore sa relation au chef de famille (épouse, fille, frère...)

Les manuscrit étant très nombreux, leur transcription numérique est faite automatiquement ce qui pose des problèmes dans certains cas à cause d'une mauvaise lecture des modèles d'OCR.

1.2 Données disponibles

1.2.1 Transcriptions manuelles

Pour créer et entrainer un modèle qui parviendra à determiner le sexe des personnes décrites dans les manuscrits à l'aide des différentes informations, nous disposons d'un faible nombre de données "annotées". C'est à dire que nous avons une base de données ou la transcription a été faite manuellement en plus de la transcription automatique ce qui a permis d'annoter les données et donc d'écrire si chaque personne est un homme ou une femme selon les informations disponibles.

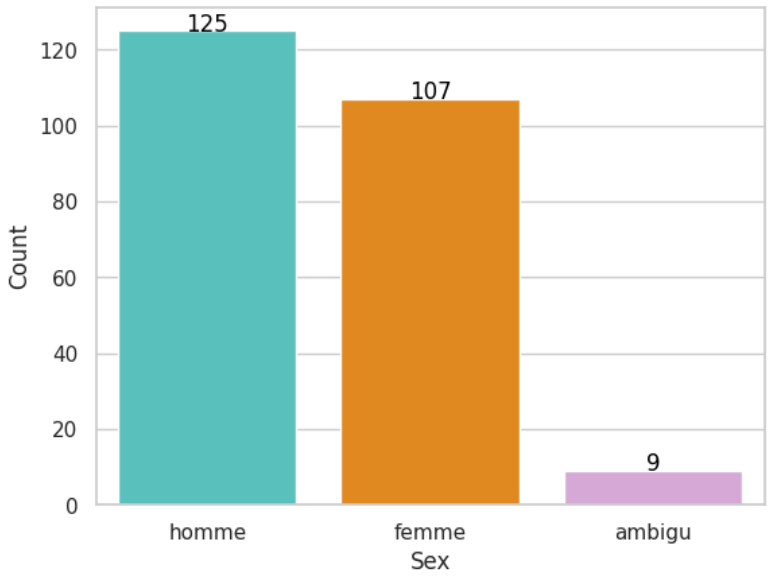


FIGURE 1 – Histogramme des labels

On voit ici que dans les données annotées nous avons 125 personnes annotées "hommes", 107 "femmes" et 9 personne "ambiguës" car les informations disponibles ne sont pas assez précises pour déterminer le sexe de la personne. En tout nous avons donc seulement 241 données labéllisées et 232 effectives si l'on exclut les labels ambigus.

1.2.2 Fichier de prénoms

Par ailleurs, nous avons également accès à une base de données qui indique, pour 6946 prénoms différents, la fréquence d'attribution à un homme et à une femme.

Par exemple ici, pour 4 prénoms choisis aléatoirement dans la base de données, on voit que le prénom 'Jérosme' est reservé aux hommes et "Aveline" aux femmes. En revanche, le prénom "Chery" s'il est principalement porté par des hommes est parfois attribué à des femmes (12.5%). Enfin, le prénom "Olésime" semble assez mixte car il n'est porté que dans 54.5% des cas par des femmes.

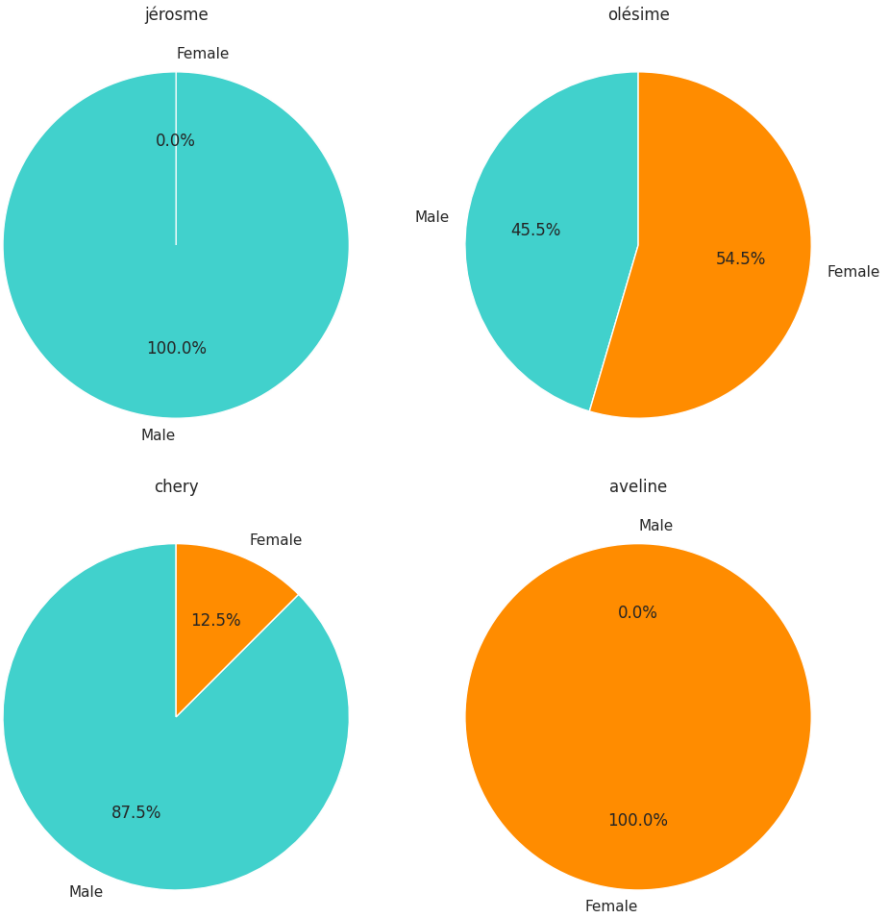


FIGURE 2 – fréquence des sexes

2 Analyse

Se pose maintenant la question de comment effectuer cette tâche. La première chose qui vient à l'esprit lorsqu'on parcourt les données, c'est que la tâche semble facilement réalisable par un humain pour différentes raisons : notre connaissance des prénoms mais également simplement par lien logiques, s'il est écrit que la personne est la fille du chef de famille, alors il est évident que c'est une femme. La difficulté réside principalement dans les erreurs de transcription automatique. D'ailleurs, la labellisation manuelle passant d'abord par une transcription manuelle témoigne de cela. Il y à donc deux moyen pour un humain de déterminer le sexe de la personne, par des règles de logique et d'habitude, et par l'esprit de généralisation et de correction des erreurs de transcription : s'il est écrit "épouxe" au lieu de "épouse", notre cerveau va immédiatement comprendre de quoi il s'agit, ce qui n'est pas forcément le cas d'une machine.

L'autre difficulté réside dans le très faible nombre de données à notre disposition. Si l'on met de coté les labels ambigus, nous avons seulement 232 données labbéllisées effectives qui doivent nous servir et d'entraînement et de validation.

Pour toutes ces raisons, nous avons opté pour différentes approches pour tenter de réaliser cette tache qui sont présentés si dessous.

3 Experimentation de différents modèles

3.1 Baseline : Zero-shot Classification

Notre première tentative est la plus naïve et nous servira de baseline pour la suite. Un LLM en Zero-shot est il capable de réaliser cette tâche ? C'est à dire, sans entrainement spécifique à la tâche, un modèle de type

BERT utilisant la techologie transformers, est il capable de classifier homme/femme les personnes décrites dans les manuscrits à partir des transcriptions automatiques ?

Ici, le problème de peu de données ne se pose donc pas, nous pouvons utiliser les 232 données comme validation sachant que nous n'en utilisons aucune pour l'entraînement du modèle. Il y à seulement 3 variables ici.

- Le choix du modèle : il existe une multitude de LLM utilisant les transformers, nous allons donc tester plusieurs modèles connus pour leur efficacité pour ce genre de tâche de classification en français
- Le choix de l'hypothèse : Nous devons faire une requète au LLM du type "Selon ces informations, la personne est de sexe ". Modifier cette phrase peut conduire à des résultats différents, nous avons donc tester une variété de formulations différentes.
- L'ordre des labels : Etonnemment, l'ordre des labels (homme, femme) ou (femme, homme) semble avoir un impact sur les prédictions des modèles, nous avons donc tester les deux ordres.

La formulation qui semblait la plus efficace s'est avéré être la plus directe "Cette personne est de sexe .". Sous cette formulation nous avons testé 5 modèles différents pour déterminer la meilleure baseline possible

- "bert-base-multilingual-cased" : BertForSequenceClassification
- "camembert/camembert-base"
- "flaubert/flaubert_large_cased"
- "facebook/bart-large"
- "camembert/camembert-large"

Modèle	Rappel	F1-Score	Précision
BertForSequenceClassification	0.70	0.71	0.7112
Camembert-base	0.50	0.35	0.5388
Flaubert_large_cased	0.51	0.50	0.5000
Facebook/bart-large	0.58	0.58	0.5776
Camembert-large	0.47	0.44	0.4569

Table 1 – Résumé des résultats pour différents modèles LLM

On observe dans les résultats que "bert-base-multilingual-cased" semble être le modèle le plus adapté et le meilleur pour réaliser la tâche que nous voulons faire. A partir de maintenant, nous devons donc obtenir une meilleur accuracy que **0.711**.

3.2 Modèles de RegEx

3.2.1 Utilisation du prénom

L'utilisation d'un ReGex astucieux permet d'extraire de la transcription automatique le prénom de la personne décrite. Empriquement, nous remarquons que cette donnée est quasiment toujours présente et est donc une base de prédiction assez fiable.

Une fois le prénom récupéré, nous appliquons un algorithme à trois étapes

- Etape 1 : On regarde si le prénom est présent dans la base de données des prénoms et on lui attribut le sexe majoritairement lié à ce prénom (exemple : Jérosme –> homme)
- Etape 2 : Si le prénom n'est pas présent dans la base de données des prénoms, on utilise la fonction gender.detector du package Gender.guesser qui nous donne le sexe probable associé à un prénom
- Etape 3 : Si le prénom n'est pas non plus reconnu par le gender.detector, on en conclut qu'il y a une erreur dans la transcription. Par exemple dans nos données, "Gilbert" est devenu "Gilbeup". Dans ce cas, grâce au package fuzzywuzzy, on va chercher le prénom "le plus proche" dans la base de données des prénoms

et lui attribuer le sexe associée. Par exemple, fuzzywuzzy va comprendre que le prénom le plus proche de Gilbeup est Gilbert et va donc lui attribuer le sexe masculin.

Avec une telle approche, nous obtenons une accuracy de **0.957** ce qui est donc largement au dessus de notre baseline.

3.2.2 Utilisation de toutes les informations

En suivant la même logique, on peut tâcher d'améliorer le modèle précédent en utilisant toutes les informations à notre disposition et pas seulement le prénom. En effet un second ReGex nous permet d'extraire la profession et la relation au chef de famille de la personne.

En appliquant une logique similaire de proximité linguistique et visuelle, on est capable de repérer les mots qui donnent explicitement le sexe de la personne comme fille, frère, ou épouse mais également les versions dégradés par la transcription automatique.

Nous voulions aussi utiliser un algorithme qui repère les suffixes féminins comme dans "cuisinière" ou "employée" pour la profession, mais il semblerait que ces informations ne soient pas fiables car les données empiriques nous montrent que des "cuisinières" ou "couturières" sont en réalité des hommes. Nous n'avons donc pas appliqué cet algorithme.

En ajoutant cet algorithme avant celui des prénoms précédemment décrit, on atteint une accuracy de **0.97** ce qui est encore un peu mieux.

3.3 Modèles de machine learning

L'utilisation de modèles de machine learning est délicate du fait du très faible nombre de données à notre disposition. Pour autant, en appliquant une méthode de cross-validation leave one out, on peut espérer obtenir des résultats intéréssants. Le leave one out consiste à exclure une donnée, entraîner le modèle sur les 231 autres puis tester sur la donnée exclue. On fait ça pour les 232 données de sortes qu'elles auront toutes servi de test une fois et auront toutes servi d'entraînement 231 une fois. Comme précédemment, on essaie plusieurs modèles usuels pour tâcher d'obtenir la meilleure accuracy possible.

Modèle	Précision
Régression Logistique	0.8405
SVM	0.6940
KNN	0.5388
Arbre de Décision	0.8103
Forêt Aléatoire	0.8190
Boosting de Gradient	0.8750

Table 2 – Précision de différents modèles de machine learning

On observe cette fois que le meilleure modèle semble être le gradient boosting suivi de près par la régréssion logistique. Pour autant, une précision de 0.875 ne nous permet pas de battre les résultats précédemment obtenus.

3.4 Finetuning d'un LLM

Notre dernière approche est de loin la plus complexe. Elle consiste à reprendre le meilleur LLM de l'approche numéro 1, mais de cette fois finetuner le modèle pour qu'il soit précisement adapté à la tâche que nous souhaitons réaliser.

Toutefois, 231 données sont largement insuffisantes pour finetuner un modèle comme BERT sans compter la division nécessaire pour produire un dataset de validation.

Notre idée est la suivante, nous allons d'abord diviser nos 231 données en dataset de test et de validation (90/10). Ensuite, pour obtenir plus de données sur lesquelles BERT pourra s'entraîner, nous allons reprendre la base de données des prénoms et la modifier. Des colonnes 'firstname' et 'prob_male' dont on avait précédemment tiré la probabilité pour Jérosme d'être porté par un homme, nous créons la phrase "Le prénom "Jérosme" est généralement porté par "" ". Nous obtenons ainsi 6946 nouvelles phrases labellisées sur lesquelles BERT va également pouvoir s'entraîner à classifier. Nous concaténons ainsi ces nouvelles phrases labellisées avec nos

données restantes dans notre dataset de test, de sorte que nous avons maintenant plus de 7000 données sur lesquelles entraîner notre LLM.

Avec 3 époques et un batch size de 16, nous obtenons les résultats suivants qui sont au delà de toutes nos espérances :

Époque	Train Loss	Train Accuracy	Val Loss	Val Accuracy
1	0.032133293	0.9870	0.149047349	0.9583
2	0.006568685	0.9980	0.171601737	0.9583
3	0.003592518	0.9993	0.003489288	1.0000

TABLE 3 – Résultats du fine-tuning de BertForSequenceClassification

4 Conclusion

Voici un tableau récapitulatif de nos différentes approches avec les résultats correspondants.

Approche	Meilleure Valeur de Précision
Zero-Shot Classification	0.711 (BertForSequenceClassification)
ReGex et prénoms	0.970 (Prénoms et relations)
Machine Learning	0.8750 (Gradient Boosting)
Fine-tuning de BertForSequenceClassification	1.000

TABLE 4 – Récapitulatif des meilleures valeurs de précision pour différentes approches

La principale limite de notre travail réside évidemment dans le très faible nombre de données de validation dont nous disposons, et il ne serait pas du tout étonnant qu'on obtienne des résultats moins bons si l'on appliquait ces modèles à un ensemble de données plus large. Toutefois, nos résultats sont satisfaisant et permettent de répondre à la problématique initiale.