GenderedNews

AGUIAR Mathilde - HAJJI Oumaima - SIDIBE Rokiatou dite Rose

17 mars 2022

1 Abstract

The equal representation of the genders in the media has been an issue that's been discussed by a lot of researchers and is a subject worth examining and looking into. For this precise reason, our project, GenderedNews, was brought to life to explore gender bias in french publications (articles from websites and newspapers) and analyze how it changes depending on the subject and the source.

GenderedNews could be seen as a pipeline starting from the retrieval of the articles, the processing of these articles to get the information regarding the representation rates for the genders, and the graphical representation of the results on the website.

2 Introduction

La représentation équitable des hommes et des femmes est une problématique qui existe depuis toujours, l'élément important qui fait ressortir notre problématique est une réelle prise de conscience de ce déséquilibre. La volonté étant alors d'observer la part et l'implication des hommes et des femmes dans les médias écrits sur des sujets importants (tels que l'économie, le sport, la politique, la religion, la culture et bien d'autres), de mesurer le biais de genre dans ces médias, et de trouver des propositions pour l'expliquer. En mettant en lumière ce biais, on espère sensibiliser le public et accroître la représentation des femmes dans les médias.

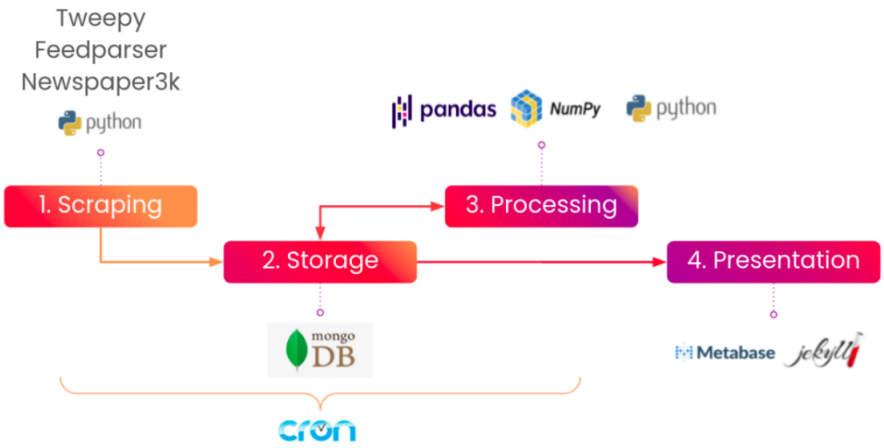
3 État de l'art

3.1 Avant de commencer :

Avant d'entrer dans le vif du sujet nous avons tout d'abord réalisé un état de l'art du projet. Pour cela nous avons compris et testé les fichiers sur le repository Gitlab. Les élèves précédent ayant réalisé une très bonne documentation, nous avons pu vite nous approprier la structure du projet. De plus, nous avons eu accès au rapport de stage de Nhat Quang Ho, un INFO5 de l'année dernière qui a fait son stage sur ce projet.

3.2 Précédente architecture

Nous avons voulu conserver au mieux l'architecture précédente et les technologies utilisées. C'est pourquoi notre nouvelle architecture (cf 4.1.4) ressemble fortement à la précédente.



3.2.1 Scraping

Dans cette partie, on utilise les librairies Python Tweepy, Feedparser ainsi que Newspaper3k pour récupérer les tweets émis par les comptes tweeter des médias que nous avons sélectionné. Une fois les tweets récupérés, on extrait les liens des articles cités dans le tweet et nous allons récupérer le contenu de ses articles.

3.2.2 Storage

Une fois nos articles de presse parsés, nous allons maintenant les stocker dans une base de données MongoDB. Chaque article est stocké en y incluant son contenu, son titre, sa source, son lien, et l'heure et la date de parution.

3.2.3 Processing

Une fois stockés, nous devons réaliser les calculs de taux de masculinité selon les *mentions* et les *citations*.

Pour cela la partie processing se divise en 2 grandes parties *mentions* et *citations*.

Les **mentions** : elles correspondent au nombre de fois où l'un prénom désignant une personne est employé par une autre personne tierce. Par exemple "Nous allons effectuer la critique du livre écrit par madame Virginia Woolf", représente une mention d'une personne de sexe féminin.

Les **citations** sont les propos rapportés, le plus souvent par un journaliste, qui concerne une personne. Les propos rapportés peuvent être à la fois entre guillemets ou bien introduits par certains termes tels que "selon, d'après, etc.".

Là où les mentions se basent plus sur de la détection puis analyse de prénoms, les citations requièrent une compréhension de la phrase plus complexe car la sémantique ainsi le contexte sont souvent plus compliqué à saisir. Dans la solution existante on utilise un modèle entier.

Par ailleurs nous avons travaillé uniquement sur la partie *mention*, la partie *citations* nécessite une plus grande puissance de calcul et de se connecter à un server du LIG nous avons uniquement récupéré les résultats directement depuis la base de données et nous n'avons pas exécuté les calculs.

3.2.4 Presentation

Cette partie correspond au front-end du site Web. Il a été développé en utilisant un template **Jekyll** auquel on a incorporé des graphiques, réalisés avec Metabase, représentant les différentes mesures effectuées dans la partie précédente.

4 Contributions

4.1 Remplacement de la technologie Metabase

4.1.1 La technologie actuelle

Dans le cadre du projet Gendered News, nous avions parmi nos tâches principales la migration de la technologie Metabase qui présente quelques contraintes pour les perspectives d'évolution du projet vers une nouvelle technologie de business intelligence plus adéquate permettant la visualisation de nos données et qui permettrait ainsi plusieurs avantages notamment l'ajout de nouvelles sources de médias afin d'enrichir la solution.

Metabase est un outil de Business Intelligence très accessible aux utilisateurs métiers qui peuvent construire facilement des dashboards. Il peut également être utilisé par les data analystes/scientistes pour faire de l'exploration autour de la donnée, fonctionnant avec de nombreuses bases de données différentes avec des connecteurs disponibles en standard (MySQL, Postgres, BigQuery, MongoDB, Google Analytics, Snowflake, etc.)

Metabase présentait quelques limites pour l'expansion du nombre de sources pour le projet car la technologie ne permettait pas une différentiation unique des différentes entrées sur les graphes notamment en matière de code couleur, ce qui dans un soucis d'ergonomie aurait altéré l'expérience utilisateur.

4.1.2 Les alternatives

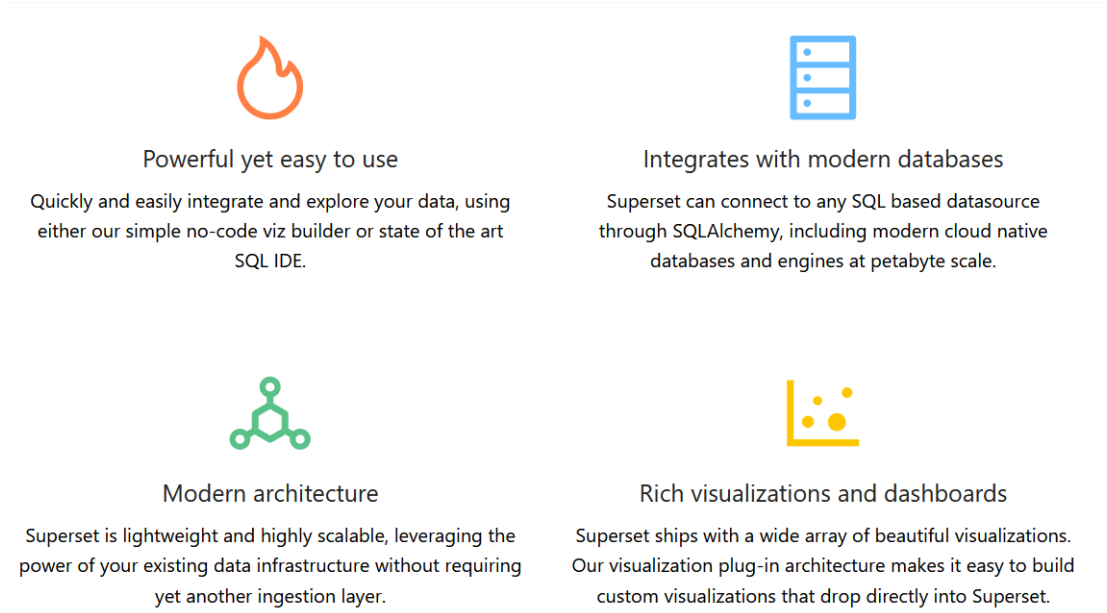
Afin d'effectuer ce changement, nous avons étudié trois solutions alternatives, Apache Superset, Kibana et Grafana, le document : Etude des alternatives pour Metabase retrace leurs descriptions.

Chacune de ses solution étant open source et présentant des avantages considérable pour le projet,, nous avons décider de partir sur la solution Apache Superset car c'est celle qui nous semblait la plus appropriée du fait de sa forte communauté, de la renommée de la solution mais également de la possibilité d'ajout de nombreuses sources, ce qui nous permettra ainsi de faire évoluer le projet Gendered News en l'enrichissant d'avantage.

4.1.3 Superset

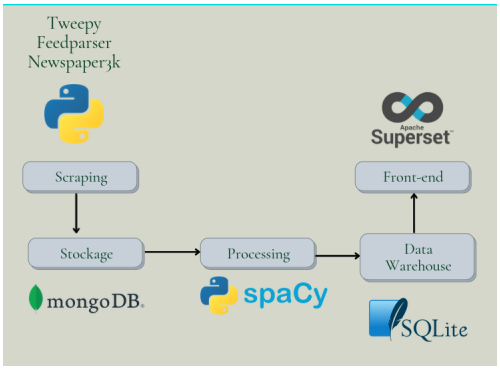
Apache Superset est un outil de visualisation et d'exploration de données, créée à l'origine par Airbnb, qui l'ayant cédé à la fondation Apache, est devenue open-source.
Superset fonctionne en tant qu'application Web sur les principaux navigateurs internet, c'est un logiciel développé en Python et qui utilise la librairie Flask comme framework Web.

Il s'agit d'un projet open source qui évolue et grandit très rapidement, la communauté est très réactive. Ses points forts sont les nombreuses bases de données que l'on peut connecter, et des visuels variés, faciles à utiliser, il est très simple de créer des tableaux de bord. Il faut toutefois connaître le SQL pour pouvoir pleinement exploiter cet outil de visualisation. Ce logiciel open source est une bonne alternative à d'autres solutions comme Metabase, en fonction de nos critères de sélection.

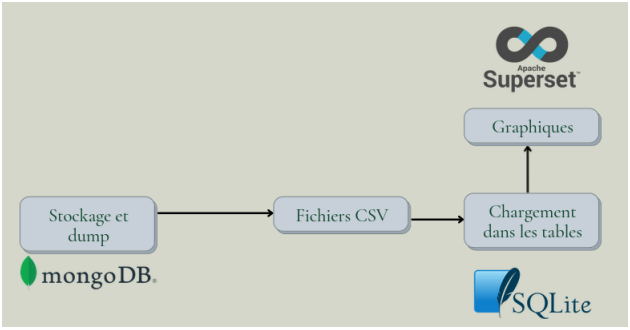


4.1.4 MongoDB vers SQLlite

Malgré ses nombreux avantages, le seul inconvenient de la solution Apache Superset est qu'il ne prend pas en charge les bases de données NoSQL, comme MongoDB, dont notre projet est contitué. Nous avons donc trouvé une solution intermédiaire afin de pouvoir récupérer et transmettre nos données de notre base de donnée MongoDB vers une base de données relationnelle SQLite qui ensuite pourra être configurée sur Superset.



Pour ce faire on utilise on rajouter une dernière étape à notre **main**. Une fois tous les calculs de masculinité effectués, et leurs résultats stockés dans une base Mongodb dédiée, nous récupérons ces résultats sous forme de fichiers CSV. Ceux-ci sont alors automatiquement chargés dans les tables de notre base de données SQLite, puis retransmises à Superset qui met automatiquement à jour les graphiques que nous avons définis préalablement.



4.2 Ajout des nouvelles sources

Une source représente un média existant, l'ajout de nouvelles sources s'est réalisé en trois grandes étapes, la création d'un fichier par source en se basant sur un template existant, le parcours des sites des différentes sources afin de recueillir leurs catégories principales et enfin la création d'un script qui nous a permis d'observer les catégories réelles des articles des différentes sources pendant quelques jours et ainsi d'adapter les nôtres dans le but d'avoir une homogénéité plus optimale des catégories sur le site de Genderednews.

Les articles sont récupéré à travers le feed twitter des différents médias. Les catégories prédéfini dans le projet pour créer une homogénéité sont :

INTERNATIONAL, FRANCE, SOCIETE, ECONOMIE, POLITIQUE, DEBATS ET OPINIONS, EDUCATION, RELIGION , CULTURE, SPORT, SANTE, NUMERIQUE , ENTREPRISES, PEOPLE, FAIT DIVERS, SCIENCE ET ENVIRONNEMENT , INDEFINI

Ainsi, nous devons classé toutes celles provenant des différents médias suivant ces grandes catégories.

Les porteurs de Projet, nous ont fournis une liste de différentes sources de médias qu'ils souhaitaient ajouter, avec des priorité différentes, nous avons ainsi pu ajouter toutes celles de priorité importante, au total 16 nouvelles sources qui pourront être intégré aux visualisations du site web du projet.

— 20minutes.fr
— Actu.fr
— France Inter
— France24.fr
— Franceinfo.fr
— L'Express
— L'Humanité
— L'Opinion
— La Voix du Nord
— Le Dauphine Libéré
— Le Monde diplomatique
— Le Point
— Le Télégramme
— Marianne
— Ouest France
— Sud Ouest

Le script d'observation des catégories des articles des différentes sources est un avantage considérable car les futures équipes qui travailleront sur le projet pourront facilement d'utiliser afin de déterminer les catégories réelles des sources qu'ils voudront ajouter plus tard.

A travers l'observation des catégories, nous avons également pu détecter un problème dans les méthodes de récupération des catégories des articles à partir des meta data, en effet dans l'ancienne

version, les catégories étaient uniquement indexer dans le champ "section" renvoyé dans les meta data, sauf que pour une partie de nos nouvelles sources, la structuration de leur meta data étaient différentes donc ne contenait pas de champ "section", induisant donc automatiquement la mise à Indéfini de la catégorie.
Nous avons pu corriger ce problème en extrayant la catégorie dans le lien de l'article, dans les cas où elle ne peut être directement récupérée des meta data, ce qui nous permet ainsi d'améliorer grandement l'homogénéisation de ces dernières.

4.3 Amélioration de la reconnaissance des prénoms avec un algorithme NER

4.3.1 Notre but :

Dans les travaux réalisés précédemment, il existe déjà une implémentation pour les traitement NER, c'est l'étape que nous avons nommé "processing" plus haut (cf 4.1.4) . Cependant celle-ci comporte quelques défauts. L'un des principaux défaut est la confusion au niveau des noms propes de villes/lieux-dit/marques, etc. et les prénoms portant le même nom que ces derniers, ex : la ville de "Paris" et le prénom anglais "Paris". Un autre problème délicat sont les noms possédant les 2 genres, ex : le prénom "Camille" peut se référer à une femme ou un homme, nous avons donc besoin du contexte pour comprendre s'il s'agit d'une femme ou d'un homme. Ces problématiques se réfèrent plus généralement au domaine de **named-entity disambiguation (NED)** ou **named-entity linking** (cf. Article Wikipedia)

4.3.2 Pipeline SpaCy :

Une des approches que nous avons expérimenté est de créer une pipeline SpaCy customisée. Pour ce faire on commence par collecter des données annotées que l'équipe GETALP ont mise à notre disposition. Ces données annotées ont été annotées grâce à l'outil collaboratif d'annotation **Doccano**. Un énorme avantage de Doccano est que les données récupérées sont téléchargées au format **jsonl**. C'est le format qu'exploite les pipelines Spacy, nous n'avons donc pas à reformater nos données.

Ensuite nous instancions une pipeline Spacy dédiée au NER (Named Entity Recognition). Nous lui précisons les labels que nous souhaitons utiliser : **Féminin, Masculin**. Avec ces labels et ces données nous allons commencer un entraînement, **fine-tuning**, de la pipeline. Enfin nous allons lui donner un nouveau jeu de données inconnues et lui demander de faire des prédictions sur ce jeu de données. Le but étant que lors du parsing d'un texte, notre pipeline soit capable de déterminer si le nom propre rencontré soit tout d'abord un nom de personne et si cette personne est de sexe féminin ou masculin.

4.3.3 Implémentation d'un algorithme NER simple

Une deuxième piste, et notre approche implémentée dans le notebook **NER_mentions.ipynb**, était d'utiliser un modèle SpaCy de la reconnaissance des entités pour la langue française. Ce modèle nous a permis d'étiqueter nos données et de récupérer les entités les plus importantes dans notre cas : les personnes.

Après avoir étiqueté les données, nous avons effectué un filtrage afin de ne garder que les mots ayant comme label **PER**. Ensuite, nous avons implémenté une fonction qui s'appuie sur un fichier csv contenant des milliers de prénoms et leur genre associé (**prenoms-clean.csv**).

Nous avons gardé les informations dans un dictionnaire afin de pouvoir l'utiliser dans les partie restantes du projet, notamment lors de l'évaluation des performances.

5 Evaluation

5.1 Evaluation des performances de l'algorithme TALN

Après avoir testé notre algorithme sur des jeux de données différents (notamment un fichier contenant 101 documents déjà annotés sur l'outil **docanno**), et afin de l'évaluer, nous avons implémenté plusieurs fonctions qui calculent la performances de l'algorithme suivant des différentes métriques.

5.1.1 Introduction du fichier de données utilisé pour l'évaluation

Comme mentionné dans le paragraphe précédent, nous avons utilisé un jeu de données que l'on a récupéré depuis la page **docanno** du projet. Ce fichier a été annoté manuellement pour assigner un genre pour chaque prénom trouvé.

Ce fichier est de forme *jsonl* et contient 101 différents documents. Pour chaque document, on peut récupérer une liste des labels ('Feminin' ou 'Masculin') et la position du mot étiqueté (le prénom).

5.1.2 Évaluation par document

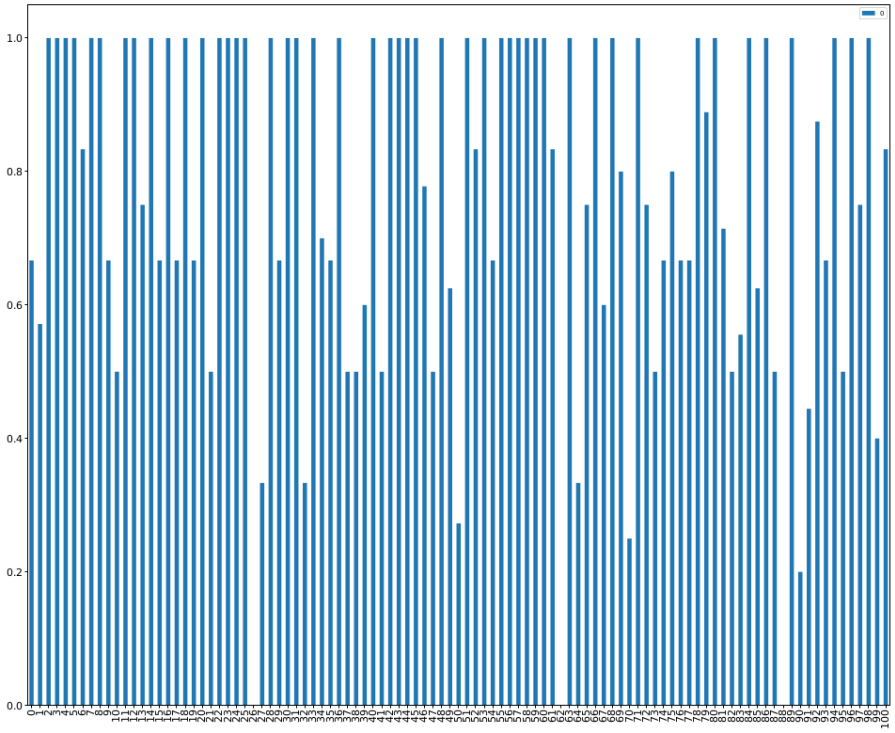
Afin d'évaluer par document, nous avons exécuté l'algorithme sur les données et stocké les résultats sous forme d'un dictionnaire où l'identifiant du fichier est la clé et la liste des prénoms reconnus est la valeur.

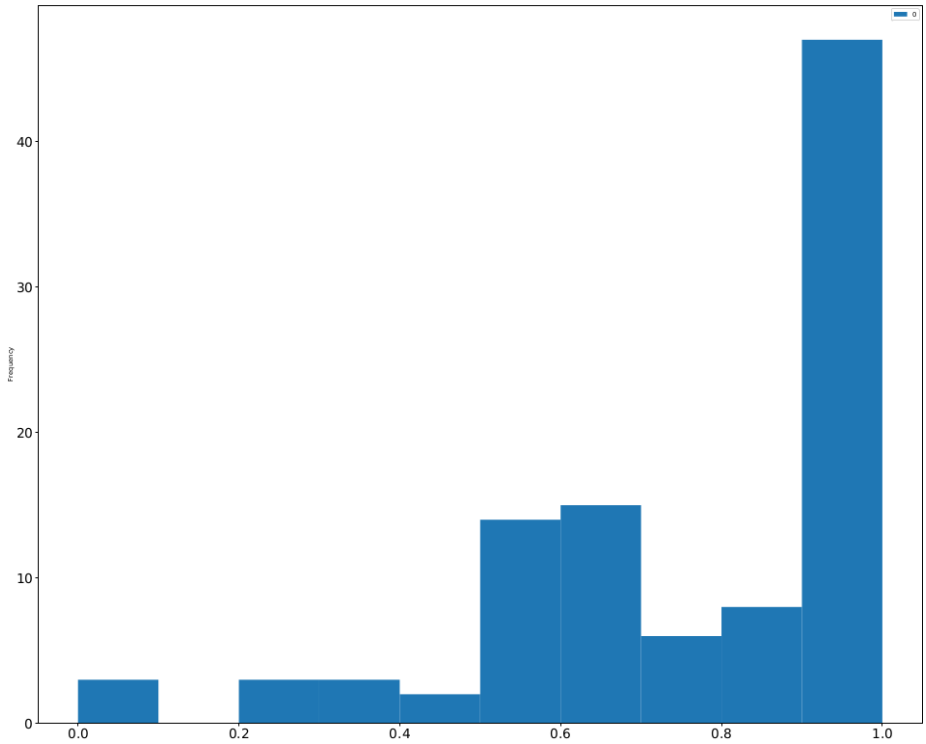
Nous avons aussi créé un autre dictionnaire contenant tous les prénoms reconnu dans les données initiales (celles récupérée depuis **docanno**).

Ensuite, nous avons évalué la performance en calculant le nombre des prénoms reconnus par les deux et qui ont le même label.

Nous avons défini trois possibilités :
- Performance = 1 : si les deux reconnaissent les mêmes prénoms (même nombre + mêmes labels)
- Performance = 0 : si notre algorithme reconnaît un prénom qui n'est pas existent dans l'autre dictionnaire.
- Performance = *len(prenoms_en_commun) / len(prenoms_dictionnaire_docanno)* : on calcule le taux de reconnaissance de l'algorithme

Pour avoir une représentation visuelle du résultat, nous avons aussi dessiné des graphes pour montrer la performance de l'algorithme pour chaque document :





5.1.3 Évaluation pour tous les documents

Afin d'avoir la performance de l'algorithme sur la totalité des données, nous avons calculé la moyenne des performances de chaque document.

Nous avons trouvé comme valeur : 0.7726083322617976

5.1.4 Évaluation de la reconnaissance par genre

Dans cette partie, nous avons essayé de tester la performance de notre algorithme lors de l'assignation de genre afin de voir si un prénom est plus susceptible d'être assigné à un label plus que l'autre.

Pour ce faire, nous avons parcouru les prénoms dans le dictionnaire découlant de l'algorithme en comparant leurs labels à ceux trouvés dans le dictionnaire **docanno**. Nous avons calculé le nombre d'erreurs d'assignation et affiché le type d'erreur (exemple : 'Femme' au lieu de 'Homme') et les prénoms qui ont été mal-étiquetés.

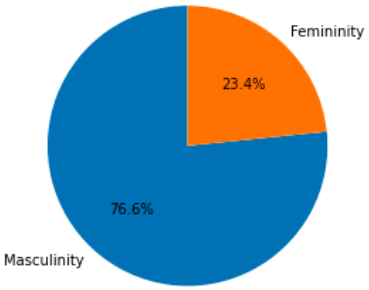
```
The error femme -> homme: 0
The error homme -> femme: 1
The error femme -> epicene: 0
The error homme -> epicene: 11
here are the names with the error (label_in_docanno, label_in_algo)
{'sameh': ('Homme', 'Epicene'), 'george': ('Homme', 'Epicene'), 'dany': ('Homme', 'Epicene'), 'billy': ('Epicene', 'Homme'), 'alex': ('Epicene',
'Homme'), 'harmony': ('Homme', 'Femme')}
```

Comme vous pouvez voir, la majorité des erreurs parvenaient des prénoms qui sont plutôt épicènes.

5.1.5 Calcul du taux de masculinité / féminité

Une autre manière d'évaluer notre algorithme est d'effectuer un calcul des taux de masculinité et de féminité sur l'ensemble des documents. Pour ce faire, nous avons calculé le nombre d'apparition des labels 'Homme' et 'Femme' dans le dictionnaire qui a découlé de l'application de notre algorithme.

```
Mentions masculinity rate in the documents:  0.766260162601626
Mentions femininity rate in the documents:  0.23373983739837398
```



5.1.6 Test sur des prénoms mixtes

Nous avons aussi testé notre implémentation sur des phrases contenants des prénoms mixtes tels que : Dominique, Claude, Sacha ...

Cela nous a permis de vérifier le biais de notre algorithme qui a étiqueté la majorité des prénoms mixtes en tant que masculin.

```
{ 0: [],
1: [('alexis', 'Homme')],
2: [('claude', 'Homme')],
3: [('maxine', 'Femme'), ('maxime', 'Homme')],
4: [('frédérique', 'Femme')],
5: [('sacha', 'Homme')],
6: [('jackie', 'Homme')]}
```

5.2 Evaluation des catégories définies pour les articles

Afin de pouvoir visualiser les catégories des articles, nous avons implémenté un script **check_articles_categories** qui récupère **N** articles de la base de données et effectue un filtrage de leurs méta-données afin de stocker les informations les plus utiles dans un fichier *csv*.

Les informations importantes que nous avons stockées sont :
- Le titre de l'article.
- L'url de l'article.
- La date de publication de l'article.
- La source de l'article.
- La catégorie de l'article dans la source.
- La catégorie assignée à l'article par nos scripts.

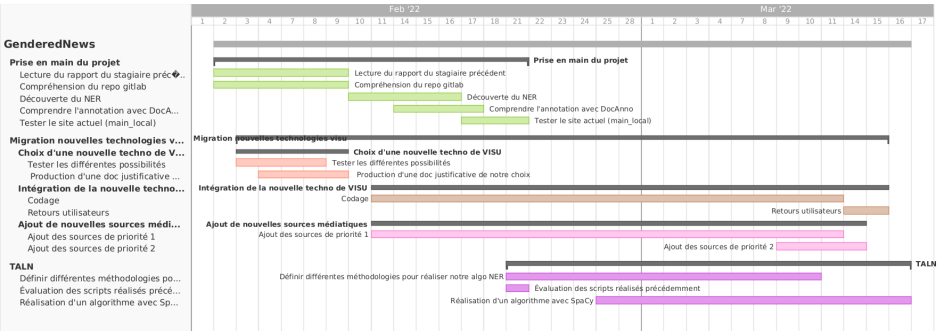
Ce script nous a permis de trouver des problèmes lors de l'assignation des catégories et de corriger ces soucis.

6 Gestion de projet

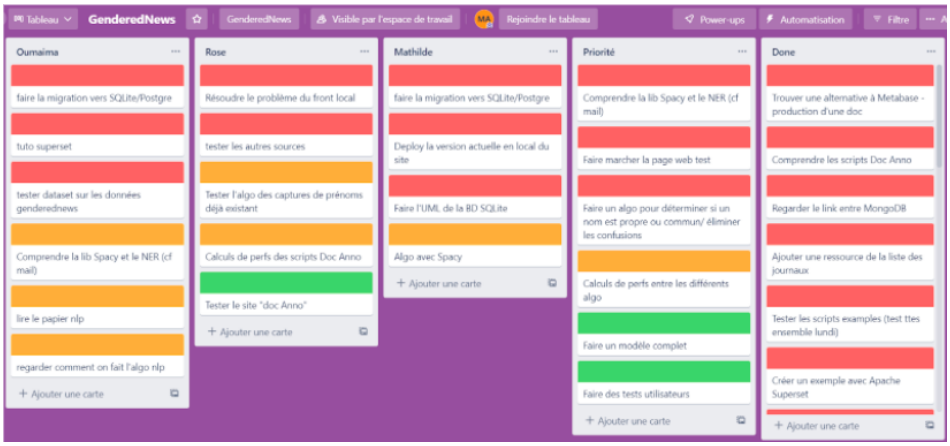
Nous avons suivi la méthode AGILE afin de bien mener notre projet ; Même avant le début du premier sprint, nous avions déjà décidé les rôles des membres de l'équipe. Au fur et à mesure que le projet avançait, et pour chaque sprint, nous avions défini des tâches et des objectifs courts et atteignables à

réussir suivant des délais.

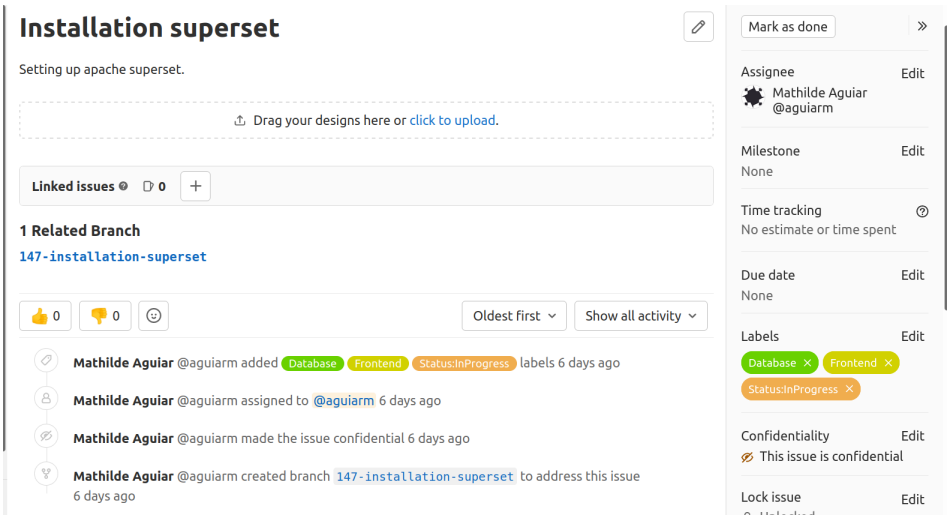
Au début du projet nous avions réalisé un diagramme de Gantt prévisionnel pour planifier dans le temps les tâches à réaliser.



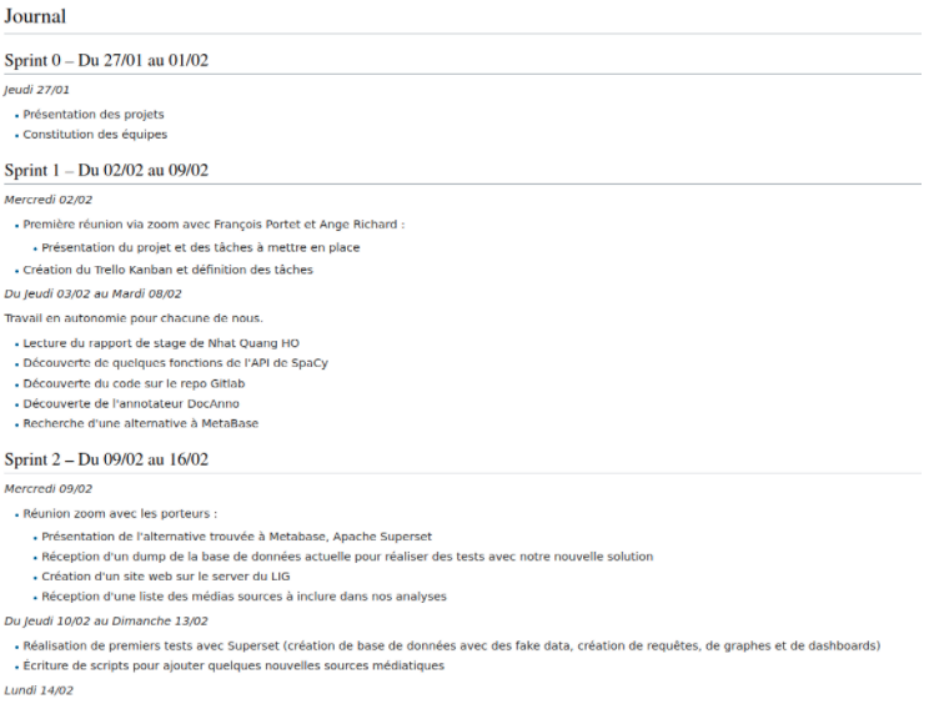
L'utilisation d'un Kanban board nous a permis de planifier les tâches et d'assigner des niveaux de priorité pour chacune sans perdre le côté visuel et simple d'un tableau de gestion. Cette démarche nous a permis de bien gérer toutes les facettes de notre projet et de s'adapter au flux variable des tâches.



Pour la partie technique, nous avons utilisé l'outil Git qui permet de gérer les versions du projet en s'appuyant sur Gitlab pour le côté collaboratif où on partageait nos travaux au fur et à mesure que le projet avançait. Nous avons en particulier utilisé les issues Gitlab pour définir, pour chaque tâche, une issue et une branche associée.



Nous avons aussi gardé une trace des sprints et des grandes tâches effectuées sur la fiche de suivi Wiki Air afin de permettre aux porteurs du projet de suivre notre avancement.



Journal

Sprint 0 – Du 27/01 au 01/02

Jeudi 27/01

- Présentation des projets
- Constitution des équipes

Sprint 1 – Du 02/02 au 09/02

Mercredi 02/02

- Première réunion via zoom avec François Portet et Ange Richard :
  - Présentation du projet et des tâches à mettre en place
- Création du Trello Kanban et définition des tâches

Du Jeudi 03/02 au Mardi 08/02

Travail en autonomie pour chacune de nous.

- Lecture du rapport de stage de Nhat Quang HO
- Découverte de quelques fonctions de l'API de SpaCy
- Découverte du code sur le repo Gitlab
- Découverte de l'annotateur DocAnno
- Recherche d'une alternative à MetaBase

Sprint 2 – Du 09/02 au 16/02

Mercredi 09/02

- Réunion zoom avec les porteurs :
  - Présentation de l'alternative trouvée à Metabase, Apache Superset
  - Réception d'un dump de la base de données actuelle pour réaliser des tests avec notre nouvelle solution
  - Création d'un site web sur le server du LIG
  - Réception d'une liste des médias sources à inclure dans nos analyses

Du Jeudi 10/02 au Dimanche 13/02

- Réalisation de premiers tests avec Superset (création de base de données avec des fake data, création de requêtes, de graphes et de dashboards)
- Écriture de scripts pour ajouter quelques nouvelles sources médiatiques

Lundi 14/02

7 Conclusion

La problématique de la représentation équitable des genres dans les médias est un sujet important qui nécessite une étude approfondie. C'est pour cette raison que ce sujet est influent puisqu'il ouvre des portes pour le calcul et l'analyse de ce biais.

Gendered News, étant l'un des projets les plus intéressants et enrichissants que l'on a eus, nous a permis de prendre en main des outils technologiques différents et d'apprendre comment gérer plusieurs parties du projet.

Pendant les 6 semaines de ce projet, nous avons acquis des connaissances dans une multitude de sujets techniques, tels que : Web scraping, la manipulation des bases de données (SQL et non SQL), et la reconnaissance des entités.

8 Glossaire

— **TALN** : Traitement automatique du langage naturel ; procédés informatiques mis en place pour exploiter les données contenant des langues écrites ou parlées par des êtres humains.
— **NLP** : Natural Language Processing, la traduction anglaise du terme *TALN*.
— **NER** : Named Entity Recognition, la reconnaissance, grâce à la sémantique d'un texte, des entités nommées présentes dans celui-ci. Ex : localisations, personnes, métier, etc.
— **NED** : Named-Entity Disambiguation, sous domaine du NER consistant à lever les ambiguïtés sur la nature d'une entité nommée.
— **Fine-tuning** : À partir d'un modèle déjà pré-entrainé sur une grande quantité de données, nous ré-entrainons ce même modèle sur nos données personnelles. L'intérêt demeure que ces données personnelles peuvent être de plus petites tailles qu'un entraînement *"from scratch"*.
— **Web Scraping** : Récupération automatique de données directement depuis les pages Web de sites internet.

9 Bibliographie

— Site Web GenderedNews
— Repo Gitlab
— GenderedNews Papier arxiv
— Apache Superset
— MongoDB documentation
— SQLite documentation
— Twitter API
— Named Entity Recognition
— SpaCy pipeline
— SpaCy pipeline - training
— Article sur la customisation d'une pipeline spacy
— Article sur la customisation d'une pipeline spacy - 2