CentraleSupélec

Projet Infonum

Automatisation du test de Bechdel

Auteurs :
Lucie Clemot
Sacha Muller
Guilhem Prince
Mention IA

Encadrants :
Théo Rubenach
Illuin Technology
Frédéric Boulanger
CentraleSupélec

Avril 2023

CentraleSupélec

Table des matières

1 Introduction et reformulation du problème

Combien de films (et de pièces de média plus généralement) que nous consommons contiennent des personnages féminins impactants, et pas secondaires ? Quelle est la véritable place que nous donnons aux protagonistes féminins dans la fiction ? Ces questions d'actualité, qui évoquent un problème sociétal plus profond, possèdent une ébauche de réponse sous le nom du test de Bechdel-Wallace.[1]

Créé presque sur le ton de la blague par la bédéaste Alison Bechdel, ce test fonctionne de la manière suivante : on considère qu'un film le valide s'il remplit les trois critères suivants :

— Il contient au moins deux personnages féminins nommés (niveau 1).

— Ces deux femmes parlent entre elles au moins une fois (niveau 2).

— Cette discussion n'évoque pas d'hommes (niveau 3).

Chaque film possède donc un score entre 0 et 3.

Ce test, à apparence simple, est un outil quantifiable assez efficace pour comparer les films entre eux. L'objectif de notre projet était d'automatiser ce test, de créer un module qui sait identifier quels films le valident ou non de manière rapide et pertinente.

Nous avons fait le choix d'étudier le médium du script. En effet, chaque film repose sur un script, qui est un fichier texte sur lequel nous pourrons mener notre étude. De plus, nous avons créé un module python que nous avons déployé afin de rendre notre outil le plus utile possible. Tout ce projet est disponible en open source sur notre dépot github.

2 Etat de l'art

Il existe quelques projets et initiatives pour automatiser, ou au moins accélérer, le calcul du score de Bechdel.

En 2015, l'article *Key Female Characters in Film Have More to Talk About Besides Men : Automating the Bechdel Test*[2] propose une première approche pour automatiser le test de Bechdel. Pour parser le script, une méthode présentée dans un article de 2014[3] est utilisée. Cette méthode s'appuie sur la structure codifiée des scripts de films et utilise des regex

pour attribuer à chaque ligne du script un label (dialogue, narration, metadata, nom des personnages etc.). Pour les films ne respectant pas la structure attendue, l'article propose d'entraîner un SVM dont les features sont un Bag Of Words des mots contenus sur chaque ligne. Une fois le parsing réalisé, un genre est attribué aux personnages du film. Une première méthode cherche le genre généralement associé au prénom du personnage dans une base de donnée, une deuxième méthode utilise le genre de l'acteur·ice jouant le personnage. C'est en mélangeant les deux méthodes que les meilleurs résultats sont obtenus. Grâce au parsing et aux genre des personnages, l'article construit un graphe de relation entre les personnages qui se parlent pour identifier si deux femmes discutent entre elles. Enfin, pour savoir si le film passe la troisième étape du test, l'article entraîne un SVM sur un certain nombre de features (Bag Of Words des dialogues entre femmes, des propriétés structurelles du graphe de relation, des caractéristiques globales du film etc.), en utilisant les données remplies par les utilisateurs sur le site du test de bechdel [1] comme vérité terrain. Leur méthode obtient au global un F1-Score de $0,79$.

En 2018, Krista Westphal reprend les travaux de *Agarwal et. al* pour sa thèse de master[4], et étend le test aux romans. Surtout, elle donne plus de détail sur sa méthode de parsing : elle utilise un arbre de décision basé sur la présence de mots clés et sur les niveaux d'indentations. Pour les critères 2 et 3 du test, Krista Westphal reprend les mêmes méthodes que dans l'article de 2015. Elle obtient un F1-Score de $0,81$.

Nous nous sommes beaucoup inspirés de ces deux travaux pour implémenter notre parsing du script, et avons nous même implémenté une approche basée sur les prénoms pour le critère deux. Cependant, nous ne trouvions pas l'approche basée sur le genre des acteur·ice·s pertinente car les doubleur·euse·s de dessins animés n'ont pas toujours un genre qui correspond à celui de leur personnage. De plus, nous voulions pouvoir extraire toutes les informations du script directement sans aller chercher d'informations externes. Enfin, pour le troisième critère, la méthode présentée dans l'article nous semblait trop peu explicable pour être utilisée. Nous voulions en effet renvoyer, en plus du score, la liste des scènes qui valident ce score, ce qui est impossible avec leur méthode.

En 2021, Data For Good a lancé l'initative BechdelAI [5]. Celle ci étudie le test de Bechdel et les films qui le passent ou non sous différents aspects, en étudiant d'une part les inégalités de représentation féminine dans les films,

mais également en développant des outils open source d'analyse de contenu et et de quantification sur des questions de représentation féminine, de male ou female gaze... Les rendus et avancées de cette initiative dépassent le scope de notre étude, mais apportent des éléments de contexte intéressants sur l'aspect visuel de la représentation féminine. Iels proposent notamment un outil de computer vision qui permet de visionner uniquement les scènes d'un film ou plusieurs femmes sont présentes. Il s'agit donc d'un élément intéressant pour remettre en question notre choix de médium du script : le fichier vidéo aurait été un choix pertinent également. Nous restons cependant convaincus qu'il est plus simple de télécharger le script d'un film que son contenu en fichier vidéo.

3 Choix d'implémentation

Une fois le choix du médium fait, nous avons réfléchi à la meilleure méthode pour résoudre notre problème.

3.1 Création de la base de données

Afin de pouvoir entrainer nos différents modèles et de tester notre module, il était nécessaire de créer une base de données de scripts ainsi que leur score de Bechdel. Pour ce faire, nous avons scrappé la base de scripts IMSDb, et l'avons intersectée avec la liste de films sur le site officiel du Test de Bechdel[1]. Ainsi, nous avons obtenu une base de 700 scripts, avec des parsings et scores variables. Les scripts étudiés sont en anglais.

3.2 Structure utilisée

Nous avons approché notre structure de données de la manière suivante :

— Nous avons créé une classe Script, qui est initialisée à l'aide d'un fichier texte représentant un script.

— Tout objet de la classe Script est composé d'objets Scènes, qui elles sont composées d'objets Dialogues.

— Un script contient également une liste d'objets Personnages, qui eux ont (entre autres) un genre.

— Enfin, un script possède un ensemble d'objets Passages Narratifs .

De plus, une fois cette structure peuplée, il était question d'implémenter le calcul du score de Bechdel. Les trois critères reposent en fait sur trois aspects clés :

— Notre capacité à identifier les personnages, les dialogues et les passages narratifs ;

— Notre capacité à prédire le genre d'un personnage ;

— Et notre capacité à juger du contenu d'un dialogue.

En effet, une fois ces différents points implémentés, il suffit de lier les deux premiers pour le critère 1, de se servir de notre capacité de parsing pour le critère 2, et de se servir du dernier point pour le critère 3.

4 Implémentation des critères

Afin de peupler la structure d'un script et de calculer le score qui lui est associé, il a fallu se concentrer sur les trois critères sus-cités.

4.1 Parsing des scripts

Avant de pouvoir conclure sur la présence de deux personnages féminins nommées ou non dans le film, il faut pouvoir identifier les dialogues et les personnages à partir du script.

En nous inspirant des travaux de Krista Westphal [4], nous avons construit notre propre arbre de décision pour parser les scripts.

En effet, les scripts de films sont écrits de manière très codifiée : différents niveaux d'indentations marquent la différence entre les passages narratifs, les dialogues et les didascalies. Le début d'une nouvelle scène est indiquée par une ligne écrite en majuscule décrivant le lieu ou la scène se joue.

```
S       INT. THE PLACE WHERE THE SCENE HAPPENS
N       This paragraph usually describes what is happening
N       in the scene : who is doing what, how does the camera
N       moves, how is the surroundings etc.
E
C                   CHARACTER NAME
M               (something the character
M               does while speaking)
D           The text the character says goes below
D           their name.
```

Notre objectif est d'assigner à chaque ligne du script un label parmi les labels suivants :

— Empty line (E) : Ligne est vide

— Scenes boundary (S) : Première ligne d'une nouvelle scène

— Scene description (N) : Narration

— Character (C) : Nom d'un personnage s'apprêtant à parler

— Dialogue (D) : Dialogue

— Metadata (M) : Précisions sur ce que fait le personnage en parlant

— Unknown (?) : Ligne inclassable (numéros de pages indiqués sur certains scripts etc.)

Si tous les scripts étaient parfaitement rédigés, il suffirait d'identifier les quatre niveaux d'indentation présentés dans l'exemple. Malheureusement, la plupart des scripts n'ont pas une indentation parfaite. En explorant manuellement plusieurs scripts, on remarque que souvent, il existe plusieurs niveaux d'indentations associés à un même label, plutôt que plusieurs labels au sein d'un même niveau d'indentation. Pour cette raison, on décide de concaténer toutes les lignes ayant un même niveau d'indentation puis d'identifier le label du texte résultant de cette concaténation en analysant ses caractéristiques. On utilise la présence de mots ou de caractères spéciaux importants, le taux de capitalisation des lignes et la longueur moyenne d'une ligne pour identifier le label d'un niveau d'indentation à l'aide de règles de décision identifiées manuellement. Une version simplifiée de l'arbre utilisé est visible Figure 1.

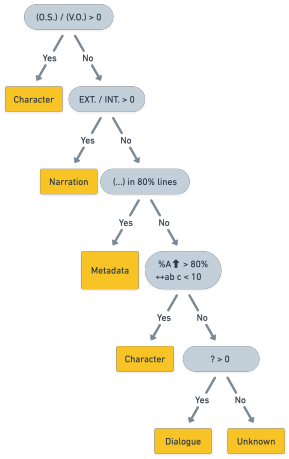

FIGURE 1 – Arbre de décision simplifié utilisé pour attribuer un label aux lignes partageant un même niveau d'indentation. Les deux premiers noeuds vérifient la présence de mots clés typique des dialogues ou de la narration, le troisième la présence de parenthèses, le quatrième que les lignes comportent peu de caractères et sont écrites en majuscule, enfin le dernier noeud porte sur la présence de points d'interrogations.

Sans vérité terrain à laquelle comparer nos labels, nous avons estimé qu'un script devait contenir au moins un dialogue, une narration, moins de mille personnages et plus d'une scène pour qu'on considère son parsing comme satisfaisant. Avec ce critère en tête, notre arbre de décision parvient à parser correctement les deux tiers des scripts de notre base de donnée.

Pour parser le tiers restant, nous avons entrainé un modèle de classification sur les lignes des scripts correctement parsés. Notre modèle consiste en un modèle Bert pré-entraîné dont les poids ont été gelés, sur lequel nous avons branché quelques couches entièrement connectées que nous avons entraînées sur notre dataset. Nous avons obtenu une accuracy de 85%, ce que nous estimons acceptable. Ce modèle nous a permis d'obtenir un parsing satisfaisant pour tous les films de notre base de données.

4.2 Prédiction du genre des personnages

Il est délicat et complexe de supposer le genre d'une personne dans un contexte habituel. Cependant, nous avons itéré sur plusieurs méthodes afin d'être les plus précis possible, malgré des potentielles erreurs.

Dans un premier temps, nous nous sommes basés sur les prénoms des personnages. En utilisant une classification Naive Bayes sur une table de prénoms associés à un genre (obtenue sur data.gouv.fr), basée sur des features de texte (comme le compte des différentes lettres, le suffixe, le préfixe), nous obtenions un classifieur avec une précision de 85%. Cependant, les limites de cette classification sont assez claires : on ne peut pas deviner le genre d'une personne à partir de son prénom uniquement.

Ceci nous a donc mené à la deuxième approche : l'exploitation des passages narratifs. En effet, les pronoms utilisés pour décrire un personnage dans la narration d'un film sont une source assez fiable pour genrer ce personnage. Nous avons implémenté dans un premier temps une méthode naïve, qui étant donnée un personnage, va identifier tous les passages narratifs dans lesquels son nom est cité. Ensuite, nous comptons tous les pronoms genrés présents dans ces paragraphes. Nous prédirons donc le genre associé aux pronoms les plus fréquents. Dans un second temps, nous avons exploité le module Neural Coref de SpaCy, un module de coréférence neuronale qui associe nom et appellations utilisées pour désigner le nom. Cette méthode bien plus sophistiquée est aussi la plus performante. Dans les cas où la coréférence n'est pas concluante pour un personnage, nous utilisons la méthode naïve pour prédire un genre.

Il convient de noter que le module neuralcoref[6] n'est compatible qu'avec une version ancienne de SpaCy qui nous a contraint à se ramener à une version python 3.7.

4.3 Topic modeling sur le contenu des dialogues

Etant donné le contenu d'un dialogue, ou d'un texte en général, comment identifier qu'il évoque un homme ou non ? Nous avons choisi de suivre une démarche assez simple, mais néanmoins très efficace. En effet, nous avons considéré que si un texte ne contient aucun pronom masculin, ou appellation masculine classique (boy, man...), ou prénom d'un personnage masculin, alors il n'évoque pas d'homme. Cette méthode, peu coûteuse en calcul, est assez performante dans les faits.

Elle présente néanmoins quelques défauts, qui en pratique ne nous sont pas apparus lors des tests, mais pourraient tout à fait être présent. En effet, cette approche ne fonctionne pas pour les phrases citant des personnages masculins célèbres : deux femmes parlant de Barack Obama ou Albert Ein-

stein, sans prononcer les mots "il, lui, son, homme ..." seront considérées comme passant le test, alors qu'elles parlent évidemment d'un homme. Ce problème peut être partiellement réglé en ajoutant des prénoms masculins courants à notre liste de mots interdits, mais nous pourrons toujours trouver des noms moins communs (Mahatma Gandhi, par exemple). Autre défaut, deux femmes parlant d'un.e boulanger·ère (a baker), sans préciser son genre, alors que l'on sait d'après les scènes précédentes que c'est un homme, passent aussi le test. Ici, il faudrait réfléchir à comment intégrer la coréférence dans la narration pour ce critère.

La piste d'un modèle de classification sur les phrases de dialogues n'a pas été explorée, par manque de dataset de phrases labellisées en fonction de la mention d'un homme pour l'entraînement.

4.4 Dureté des règles

L'énoncé des critères du test de Bechdel est légèrement ambigu. En effet, les femmes doivent-elles être seules dans la discussion ? Est-ce que la totalité de la conversation doit valider le troisième critère, ou est-ce qu'un seul échange aller-retour suffit ?

Afin d'inclure cette réflexion dans notre travail, nous avons implémenté deux règles : une sur la présence de femmes uniquement dans le dialogue, et une sur la totalité de la discussion ou non pour le critère 3. En utilisant ces critères comme paramètres, nous laissons à l'utilisateur le choix de définir le test comme il le souhaite.

Cependant, nous considérons que si un homme est présent dans la conversation, alors le paramètre sur le contenu de la conversation ne peut pas être strict (aucune phrase dans la conversation n'évoque un homme). Il existe donc seulement 3 combinaisons de règles possibles, soit trois scores par film.

4.5 Rajout de l'intervention utilisateur·ice

Nous avons assez vite identifié que la faiblesse majeure de notre projet était aussi la plus simple à corriger : la prédiction de genre des personnages. Ainsi, nous opérons de la manière suivante : une fois un premier score calculé, nous proposons à l'utilisateur·ice de parcourir les personnages nommés et leurs genres. Si l'utilisateur·ice considère qu'il y a une erreur, iel peut recalculer le score en corrigeant les personnages de son choix, et ce sans avoir

à re-parser le script.

5 Aménagement de l'outil

5.1 Structure du répertoire

Une fois que notre outil a atteint un niveau satisfaisant, en terme de capacité à identifier si un film valide le test ou non, il s'agissait de l'aménager pour le rendre accessible, en le mettant en ligne.
Nous avons opéré de manière assez classique, en distinguant le Front-end du Back-end, et en construisant une API pour faire appel à notre back (notre objet script) depuis notre site internet.

5.1.1 Paramètrage de l'API

Nous avons choisi, suite à un échange avec notre encadrant, d'utiliser Fast-API pour créer notre API.
Les quelques méthodes nécessaires pour faire fonctionner notre outil sont telles que suit :

— Une première méthode POST, capable de recevoir un fichier texte et de le transformer en objet script, et d'ajouter dans la base de données les informations utiles (la liste de personnages et leurs genres, le score calculé...)

— Une méthode GET pour récupérer, étant donné un identifiant unique d'un film, son score et les éléments pertinents (les scènes qui valident ce score...)

— Une méthode POST capable de recalculer le score étant donnés un script et une correction des genres des personnages par un·e utilisateur·ice

— Une méthode GET rendant le contenu d'une scène étant donné un film et un identifiant de scène, notamment les lignes passant ou non le test.

5.1.2 Implémentation du site internet

Dans un premier temps, nous avions codé un site sur streamlit. Assez rapidement cependant, nous avons choisi de nous engager sur une piste plus ambitieuse mais aussi plus intéressante, en codant le site en React, en utilisant l'outil Create React App.

Ne nous y connaissant pas vraiment, une montée en compétence a été nécessaire pour créer notre site internet. Au début, à force de tutoriels et de stack overflow, nous avons réussi à implémenter les premières fonctionnalités de base, telles que l'upload d'un script sur le site, l'affichage d'un score ou encore les menus déroulants pour changer le genre des personnages. Ensuite, nous nous sommes attaqués à des tâches plus complexes, comme la communication avec l'api ou la gestion du chargement (ajout d'un spinner) et des potentielles erreurs. Au fur et à mesure, coder en React n'était plus vraiment le réactif limitant et nous voulions rajouter toujours plus de fonctionnalités. Le temps pressait, alors nous avons décidé de mettre en pause le développement sur le front-end pour se concentrer sur le déploiement du site et les autres parties du projet qui devaient être finalisées.

5.2 Déploiement

Nous avons choisi de mettre le côté front et back sur une machine virtuelle. Actuellement, nous louons une machine virtuelle gratuite chez Google, et elle se trouve dans l'Oregon aux Etats Unis. Ses performances sont cependant assez faibles, elle ne contient que 1GB de RAM.

Nous avons utilisé le logiciel *nginx* en tant que reverse proxy (proxy inverse), c'est à dire en tant que serveur qui agit en intermédiaire, qui reçoit les requêtes et les redirige vers le front-end et le back-end successivement.

Nous avons également dû améliorer le fonctionnement de l'API pour qu'elle puisse gérer plusieurs utilisateur·ice·s en même temps, problème que nous n'avions pas remarqué en testant en local. En effet, notre première version de l'API attendait de recevoir les résultats du calculs pour renvoyer le score lorsque nous lui envoyions un script. Le calcul prend quelques dizaines de seconde et pendant ce laps de temps l'API était complètement paralysée, ce qui empêchait les autres utilisateur·ice·s d'explorer les résultats qu'iels venaient de recevoir. En effet, pour afficher une scène passant le test le front envoie une requête à l'API, et cette requête simple, ne requérant aucun calcul et simplement de regarder dans la base de donnée, était bloquée jusqu'à ce que les calculs plus conséquents soient finis, ce qui rendait le site inutilisable.

Nous avons donc mis en place un système de polling : lorsque l'API reçoit un script elle lance les calculs lourds en arrière plan et renvoie au front un numéro d'identification, puis le front demande régulièrement à l'API si les résultats sont prêts avec son ID, jusqu'à ce que les calculs soient terminés. Avec cette méthode, le site peut techniquement répondre à la demande de plusieurs utilisateur·ice·s, cependant, en pratique, à cause des faibles perfor-

mances de notre machine virtuelle, les temps de calculs augmentent significativement lorsque plusieurs requêtes sont envoyés, pouvant atteindre jusqu'à une heure de calcul pour seulement trois utilisateur·ice·s simultané·e·s. En pratique, le site n'est donc pas utilisable à l'échelle que nous espérions.

6 Performances du module

Afin de quantifier la performance de notre outil, nous utilisons notre base de donnée de 700 scripts déjà labellisés. Il faut cependant noter que ces labels sont choisis à la main par des utilisateurs·ices externes, qui ne sont d'ailleurs pas toujours d'accord entre eux. Avec ces labels biaisés donc, nous itérons à travers toute la base de donnée et construisons un ensemble de vrais positifs (VP), faux positifs (FP), vrais négatifs (VN) et faux négatifs (FN). On définit ici un positif comme un script qui obtient un score de 3, et un négatif comme un script qui obtient un score inférieur à 3 (qui ne valide donc pas le test de Bechdel).

Notre objectif final était de maximiser la qualité de l'outil de ce que nous proposons à l'utilisateur·ice. Etant donnée sa capacité à corriger un mauvais genrage, ou à utiliser son esprit critique pour identifier si le contenu d'une scène est satisfaisant ou non, nous préférons offrir à l'utilisateur·ice les éléments de contexte nécessaires (et souvent suffisants) pour valider ou non le score prédit.

Il s'agit également de calculer ces performances pour tous les couples de règles possibles, c'est à dire de la version du test la plus stricte à la plus souple. Nous obtenons alors les résultats dans le tableau suivant, sur un total de 694 scripts labelisés.

	Règles strictes (True, True)	**Règles intermédiaires (True, False)**	**Règles souples (False, False)**
Rappel définition	Les deux femmes doivent être seules et ne pas évoquer un homme de la conversation.	Les deux femmes doivent être seules et doivent échanger deux répliques sans évoquer un homme.	Les deux femmes doivent échanger deux répliques sans évoquer un homme.
Accuracy	62%	65%	69%
Précision	80%	81%	72%
Rappel	37%	45%	68%
F1-score	51%	58%	70%
VP	136	164	248
FP	35	38	95
VN	292	289	232
FN	231	203	119

7 Déroulement et avancement du projet

Dans l'ensemble, nous sommes satisfaits de l'état final de notre projet, et considérons avoir atteint nos objectifs. Nous sommes légèrement frustrés de notre incapacité de déployer complètement le site internet, il s'agissait là d'un manque de temps. Cependant, nous considérons que notre contribution open source permettra au projet d'être facilement repris par quelqu'un qui aura du temps à lui consacrer.

Les membres du groupe sommes très fiers de ce que nous avons produit, et c'est avec regret que nous ne prolongeons pas cette expérience en continuant de développer notre outil. Nos projets personnels et professionnels nous en empêchent cependant.

	Critère de validation	**Etat d'avancement**
Structure d'objet script	Fournir un score quel que soit le script en entrée	Terminé
Modèles sophistiqués	Parsing systématiquement performant et genrage plus précis	Terminé
Création du Back-end	Création d'une API et ses méthodes	Terminé
Création du Front-end	Développement en React et ajout intervention utilisateur	Terminé
Déploiement du site	Site accessible en ligne et capable de répondre à la demande de plusieurs utilisateurs simultanément	En cours
Contribution Open Source	Mise à disposition d'un dépôt git construit, propre, et utilisable tel quel	Terminé

D'un point de vue de mise en oeuvre du projet, nous avons procédé de manière itérative, et avons toujours travaillé en présentiel en équipe. Après une phase de reformulation du problème, du choix de l'utilisation du script, nous avons procédé à la fabrication du MVP : la structure d'objet Script, utilisable via un terminal. Une fois cet outil satisfaisant, nous nous sommes tournés vers le déploiement. Ainsi, nous avons eu l'occasion de développer de nouvelles capacités de développement, plutôt axées front-end.

Les trois membres du groupe ont chacun apporté leur domaine d'expertise leurs apprentissages issus d'expériences passées dans le projet. Nous avons toujours été ouverts aux idées les uns des autres, et chacun·e a pu choisir quels éléments de la liste de tâches iel souhaitait réaliser.

8 Voies d'amélioration

Plusieurs fonctionnalités et améliorations étaient envisagées, mais n'ont pas pu être réalisées par manque de temps, et parfois de moyens. Parmi elles, on retrouve des éléments liés à l'accessibilité du site, ainsi que des éléments liés aux propriétés du site.

— Nous souhaitions obtenir un nom de domaine plus impactant ainsi qu'une machine virtuelle plus performante (et donc payante). En effet, notre site actuel ne peut répondre qu'à un·e seul·e utilisateur·ice à la fois. Il serait également possible d'améliorer notre gestion des requêtes par l'API, comme mettre en place une file d'attente pour éviter de lancer les calculs en arrière plan en simultané et ralentir significativement la machine. Utiliser un framework comme Celery afin de répondre intelligemment et efficacement à une demande de plusieurs utilisateurs·ices en simultané serait une bonne piste.

— Nous souhaitions également rendre notre base de données de scripts disponible sur le site, afin qu'un·e utilisateur·e n'ait pas l'obligation de télécharger un script en ligne pour tester l'outil. Nous n'avons pas eu le temps de mettre cela en place, mais nous pensons que notre machine virtuelle actuelle dispose d'assez de stockage pour conserver ces scripts.

— Nous aurions aimé communiquer davantage sur ce projet et cet outil, qui selon nous pourraient en intéresser beaucoup.

9 Considérations éthiques et environnementales

9.1 Aspect éthique et sociétal du projet

Nous sommes conscients des ramifications éthiques de notre projet. D'une part, notre outil permet de faciliter la conversation à propos de la place que prennent les personnages féminins dans les films. En ce sens, nous avons un impact positif et adjuvant sur l'avancement d'une problématique sociétale : la représentation féminine. D'autre part cependant, force est de constater que notre prédiction de genre est parfois inexacte, ce qui peut être problématique. Nous avons rajouté, pour pallier à ce problème, l'intervention utilisateur·ice citée précédemment. Nous sommes désormais plus à l'aise avec notre module.

Dans une démarche d'inclusivité, notre code peut genrer un personnage comme Femme, Homme, ou personne Non-binaire. Se pose alors la question de préciser les règles du test de Bechdel : est-ce qu'une conversation entre deux minorités de genre qui n'évoque pas d'homme permet à un film d'atteindre le score de 3 ? Ces considérations, peu complexes à mettre en place, dépassent le scope de notre étude, et nous ne souhaitions pas s'éloigner autant du test initial. Cependant, nous pensons qu'avec la croissance de représentation d'individus non-binaires dans les médias, il serait pertinent de mettre le test à jour et d'implémenter ces modifications.

9.2 Impact environnemental du projet

Par soucis de transparence, et dans une démarche ingénieur responsable, nous avons tenté de quantifier l'impact environnemental de notre projet. Il nous semble que l'étape la plus émissive de la création du site est l'entrainement du modèle de Deep Learning utilisé pour le parsing de script. D'autre part, il nous semblait pertinent de quantifier l'impact environnemental d'une inférence de notre module, c'est à dire d'un appel à l'API, c'est à dire d'une utilisation du site.
L'ensemble des estimations d'impact ont été réalisés à l'aide du module codecarbon[7].

— En ce qui concerne l'entrainement du modèle de parsing, nous le réalisons en local, donc en France métropolitaine. L'entrainement prend environ 2 heures. Le mix électrique utilisé pour calculer l'impact provient de la documentation de codecarbon.[8] Cette étape n'est réalisée qu'une fois.

— L'inférence ne prend que quelques secondes, mais cela varie en fonction du script. L'inférence est réalisée sur notre machine virtuelle qui se situe dans l'Oregon, aux Etats-Unis. Le mix électrique de l'état de l'Oregon provient de la documentation officielle de l'état.[9] Cette étape est réalisée à chaque fois qu'un·e utilisateur·ice cherche à utilsier le site.

Voici l'impact calculé de ces deux étapes.

	Entrainement Modèle	**Inférence**
Mix électrique intensité carbone	55 gCO2e/kWh	128,94 gCO2e/MJ
Energie consommée (kWh)	0,107	0,0019
Emissions totales (gCO2e)	5,9	0,9

Si le site était amené à subir une forte demande, ses émissions pourraient devenir inquiétantes. Cependant, il nous semble qu'en l'état, ces chiffres d'émissions sont tout à fait raisonnables.

10 Conclusion et remerciements

Après de nombreux mois de travail sur ce projet d'automatisation du test de Bechdel, nous sommes dans l'ensemble très satisfaits de nos résultats. Dans le contexte de notre première contribution open source pour certains, nous restons cependant sur notre faim : nous aurions aimé aller plus loin, et réaliser les éléments cités dans la partie "Voies d'amélioration". C'est néanmoins avec fierté que nous finissons ce projet.

Nous souhaitons remercier CentraleSupélec et le département InfoNum pour l'accompagnement, ainsi que Théo Rubenach, notre encadrant côté Illuin Technology, pour son aide et ses conseils tout au long de l'année.

11 Bibliographie

Références

[1] Bechdel. Bechdel test official website, 2023. bechdeltest.com.

[2] Apoorv Agarwal, Jiehan Zheng, Shruti Kamath, Sriramkumar Balasubramanian, and Shirin Ann Dey. Key female characters in film have more to talk about besides men : Automating the Bechdel test. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 830–840, Denver, Colorado, May–June 2015. Association for Computational Linguistics.

[3] Apoorv Agarwal, Sriramkumar Balasubramanian, Jiehan Zheng, and Sarthak Dash. Parsing screenplays for extracting social networks from movies. In *Proceedings of the 3rd Workshop on Computational Linguistics for Literature (CLFL)*. Association for Computational Linguistics, 2014.

[4] Krista Westphal. Using natural language processing to automate the bechdel test. 2018.

[5] DataForGood. Initiative bechdelai, 2021. dataforgood.fr.

[6] huggingface. Neuralcoref documentation, 2021. github.com.

[7] codecarbon. Codecarbon, 2020. codecarbon.io.

[8] codecarbon. Codecarbon eu carbon intensity electricity table, 2020. github.com.

[9] State of Oregon Department of Environmental Quality. Electricity carbon intensity values for 2022, 2022. oregon.gov.