INF473G : ETUDE DE LA MÉDIATISATION DES "HATE CRIMES" AUX ETATS-UNIS

Utilisation de graphes sur une base de données d'articles

May 30^{th}, 2023

Auguste CRABEIL, Swann BESSA

ÉCOLE POLYTECHNIQUE

IP PARIS

INTRODUCTION

Véritable fléau des sociétés modernes, le phénomène des "hate crimes" a gagné en ampleur ces dernières années. Aux États-Unis, les tensions ethno-culturelles ont été renforcées par la polarisation de la société, notamment suite aux élections de 2016. Notre étude portera sur deux datasets sur l'année 2017 :

- La première base de données a été trouvée sous la forme d'un CSV fourni par l'organisme de journalisme indépendant ProPublica dans une enquête sur les "hate crimes". ProPublica a réalisé un scraping des articles Google News du 13 février au 28 octobre 2017, en extrayant notamment le titre de l'article, sa date de publication et les mots-clés pour certains. Ce dataset contient 11249 points dont 9150 avec mots-clés. Dans la suite, nous nous baserons uniquement sur les articles contenant des mots-clés.

- La deuxième base de données provient du FBI et contient la liste de tout les "hate crimes" aux États-Unis de 1991 à 2018. Pour chaque "hate crime" sont notamment renseignés la localisation du crime, la date, l'ethnicité de l'assaillant et la communauté de la victime. Ce dataset contient 5698 points sur l'année 2017.

Notre but dans cette étude est d'analyser la médiatisation des différents hate crimes. Ces deux datasets ont pour avantage de permettre la comparaison entre l'impact médiatique et réel des différentes tensions communautaires. Cependant, le lien n'est pas direct entre ces bases de données : la date correspond à la publication de l'article dans le premier, et au jour du crime dans le second. Nous ferons l'approximation de nous restreindre aux hate crimes de 2017 dans la base de données du FBI, en considérant que la plupart des articles du premier dataset portent sur l'année en cours.
Notre étude se déroulera en plusieurs étapes :

1. Extraction des informations clés (ville, état, communauté de la victime) en réalisant de l'Entity Recognition sur les titres d'articles et les mots-clés

2. À partir de l'étape 1, statistiques simples pour analyser la sur/sous-médiatisation des différents hate crimes en comparant les deux datasets

3. Clustering des articles par "Average-Linkage Clustering" et analyse des différents clusters obtenus à partir d'un dendogramme

4. Clustering des articles par Modularity Maximization avec Gephi et analyse des centres (Betweenness Centrality) avec Neo4j

5. Tentative d'identification des articles au "hate crime" précis dans le dataset du FBI.

6. Enrichissement de la base de données des articles par Web Scraping avec BeautifulSoup

7. Réalisation d'un générateur de titres d'articles de "hate crime" à partir d'un graphe dirigé de mots avec NetworkX.

1 STATISTIQUES SIMPLES.

Notre première approche du sujet consiste en une étude des statistiques simples concernant les "hate crimes" et leur médiatisation.

1.1 L'extraction des informations

Fichier utilisé : statistiques_simples.py

Les datapoints du dataset des articles contiennent le titre et les mots-clés de chaque article. À partir de cela, nous souhaitons extraire l'ethnicité de la victime, la ville et l'état du crime.

Titre	Mots-clés	Média
Oildale man sentenced to 15 years for hate crime	car cole dale district hate judge justin latino oildale	The Bakersfield Californian

Figure 1: Représentation d'un point de données initial du dataset des articles

Nous avons commencé par une méthode naïve consistant à classer les mots par ordre d'occurence sur l'ensemble des 11 000 titres et ensuite à sélectionner à la main ceux correspondant à des villes, des états ou des communautés. Le nombre de mots étant trop élevé, nous avons abandonné cette méthode après avoir cependant noté les mots les plus récurrents dans des sets Pythons correspondant respectivement à la communauté de la victime, la ville et l'état (setRace, setCities, et setStates dans le code).
Ensuite nous avons réalisé de l'Entity Recognition avec spaCy afin de décomposer les phrases et catégoriser directement les différents mots. Cette méthode a été plus ou moins fructueuse :

- Concernant les villes et les pays, l'étiquette 'GPE' permet bien d'obtenir à la fois les états et les villes dans les titres. Le problème est que la NER fonctionne sur des bases grammaticales et ne peut pas exploiter les mots-clés, qui n'ont pas de lien grammatical entre eux. (voir Figure 1) Une partie significative de l'information est donc perdue car on n'analyse que le titre, et le nombre de villes/états identifiés était trop faible car ces derniers se trouvaient souvent dans les mots-clés.

- Concernant les ethnicités, l'étiquette 'NORP' permet d'identifier de nombreux mots traduisant l'appartenance à un groupe ethnique, politique ou religieux. Encore une fois, l'information contenue dans les mots-clés est perdue. Cependant, en sauvegardant les mots identifiés 'NORP' dans setRace (après une sélection à la main car tous ne correspondaient pas à l'ethnicité), nous avons pu obtenir une liste relativement complète des mots se référant aux communautés des victimes. Ces mots (issus des titres) ont ensuite été utilisés pour identifier dans les mots-clés l'éthnicité de la victime, et ont permis d'identifier

dans environ 50% des cas la communauté de la victime. Après vérification, les articles restants ne mentionnaient pour la plupart pas de communauté définie.

Pour contourner la difficulté rencontrée sur les villes et les pays, nous avons finalement abandonné l'Entity Recognition, et plus simplement rempli setCities et setStates avec des fichiers contenant les principales villes américaines et la liste de tout les états. (us-cities.txt et state_names.txt) Nous n'avons pas pu utiliser une liste complète des villes américaines, car il y avait trop de confusions à cause de noms atypiques de villes tels que Nothing en Arizona, Okay en Oklahoma ou encore Accident dans le Maryland.

Un problème demeurait : comment être sûr que la communauté que l'on trouve dans le texte est celle de la victime, et non pas celle de l'assaillant par exemple ? De même, pour les villes et les états : la ville mentionnée pourrait par exemple être la ville d'origine de l'assaillant.
Nous avons d'abord imaginé que la première communauté mentionnée était celle de la victime (exemple : "Black man killed outside of his office"), mais malheureusement c'est aussi souvent l'inverse (exemple : "Angry sikh man throws a glass bottle at gay man").
Nous avons donc opté pour la communauté majoritairement citée dans le titre et les mots-clés. Cette méthode fonctionne bien quand il s'agit de minorités (plus de 90% de réussite sur un échantillon d'une trentaine d'articles), mais très mal pour la communauté blanche aux Etats-Unis. En effet, les articles impliquant la communauté blanche sont partagés entre des crimes anti-blanc et de suprémacistes blancs et nous avons eu beaucoup de mal à les séparer. Nous avons donc décidé, en l'absence de connaissances plus poussées en NLP, de mettre à part cette communauté dans nos recherches.

1.2 Les premières statistiques

Fichiers utilisés : statistiques_simples.py, camembert.py.

Pour réaliser des statistiques simples sur nos deux datasets, nous avons voulu comparer la médiatisation des hate crimes par rapport à leur poids réel. Grâce aux identifications d'ethnicités réalisés en partie 1.1, il était aisé de réaliser un camembert représentant la médiatisation des hate crimes par communauté de la victime. De même, nous avons réalisé un camembert en lisant simplement la colonne "BIAS_DESC" du dataset du FBI, qui correspond à la communauté de la victime. Pour rappel, les articles étudiés ont été publiés entre février et octobre 2017, et les hate crimes étudiés portent sur toute l'année 2017.

La figure 2 (page suivante) compare les deux camemberts. On observe que les "hate crimes" ciblant les communautés juives et LGBT sont assez justement retranscrits dans les médias. En revanche, on observe une nette sous-représentation des "hate crimes" visant les Afro-Américains et une sur-représentation importante des attaques contre les musulmans et les asiatiques. On notera également, pour des minorités plus restreintes telles que les "Native Americans" ou les handicapés, un quasi-silence autour des violences les impactant. Comment expliquer ces écarts entre réalité et médias ?

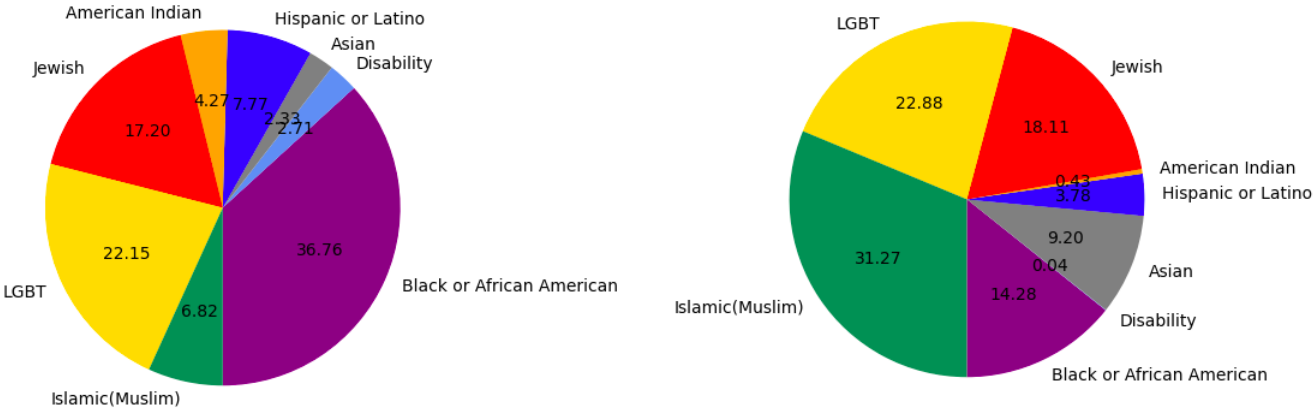

Figure 2: Représentation des différentes ethnicités victimes de "hate crime" dans la réalité (à gauche), et dans les médias (à droite)

- La surreprésentation des crimes anti-musulmans n'est en réalité pas choquante, car l'élection de Donald Trump (janvier 2017) a provoqué une hausse de 91% des crimes anti-musulmans dans la première période de 2017 (source : Al Jazeera). Une surmédiatisation paraît donc cohérente.
- La sous-représentation des hate crimes anti-Afro Américains est certainement liée au fait que ces violences ont lieu depuis des décennies. On pourrait diagnostiquer une banalisation de telles violences dans la société américaine, à telle point qu'elles en deviennent sous-médiatisées. Le même raisonnement peut s'appliquer aux "Native Americans".

Nous avons présenté ici quelques raisons logiques pour expliquer les différences de représentations des communautés. Dans la suite, nous envisageons d'approfondir notre analyse en utilisant des outils informatiques, en particulier des graphes, afin d'obtenir des informations supplémentaires grâce à une étude globale de la base de données.

2 CLUSTERING DES ARTICLES.

Algorithme utilisé : clustering_articles.py

Afin de comprendre la médiatisation différenciée des crimes contre certaines communautés, nous avons étudié la fréquence de certains types d'articles. Pour cela, nous avons comparé les termes utilisés dans différents articles, tant au niveau des titres que des mots-clés (notre analyse s'est donc limitée aux articles pour lesquels nous disposions des mots-clés correspondants). Nous avons ainsi cherché à identifier les thèmes et les sujets récurrents, pouvant expliquer pourquoi les crimes contre certaines ethnies sont davantage mis en avant. Il est important de préciser que cette méthode se base uniquement sur les termes utilisés, sans prendre en compte d'autres éléments tels que la gravité des crimes.

Pour effectuer cela, nous avons utilisé la méthode TF-IDF (Term Frequency-Inverse Document Frequency). Cette approche nous permet d'identifier les regroupements d'articles autour de certains termes clés. Bien que TF-IDF prenne en compte la fréquence d'apparition des termes dans l'ensemble des textes, nous avons néanmoins dû exclure les mots "hate" et "crime(s)" de l'analyse, car ils faussaient les résultats. En effet, tous les articles traitent du thème des "hate crimes", de sorte que ces termes n'apportaient aucune information significative et leur présence dans deux articles mêmes très différents avait un impact non négligeable.

Ainsi, nous avons "vectorizé" tous les articles (les résultats étaient les meilleurs lorsque la "vectorization" était faite sur la concaténation des titres et des mots clés). En utilisant la similarité cosinus, nous avons construit une matrice de similarité entre les articles. Ce processus implique le calcul du produit scalaire entre les vecteurs représentant deux articles, suivi de la normalisation de ces valeurs. Nous obtenons alors une mesure de similarité qui reflète la proximité en termes de contenu et de distribution des termes entre les articles.

Les calculs étant longs, nous nous basons dans la suite de cette partie uniquement sur 2000 articles.

2.1 Clustering par Average-linkage

La première idée que nous avons eu est de considérer que deux articles sont liés si ils sont suffisamment similaires et d'en déduire des clusters. Cela reviens à faire du Single-linkage. Pour avoir des meilleurs résultats nous avons plutôt effectué de l'Average-linkage pour être moins sensible aux valeurs aberrantes et éviter les effets de chaîne. L'Average-linkage est une méthode de liaison qui calcule la distance moyenne entre les points de deux groupes pour évaluer leur similarité et les fusionner progressivement.

Pour visualiser les résultats de la méthode d'Average-linkage, nous avons procédé à la création d'un dendrogramme où la hauteur des branches représente la similarité cosinus. Cependant, en raison du grand nombre de points, les résultats sont peu lisibles directement sur le dendrogramme, nous avons donc opté pour l'affichage direct des clusters. Pour obtenir ces clusters, nous avons choisi un nombre de clusters k, approximativement égal au nombre d'ethnies, ce qui nous permet de déterminer une hauteur pour séparer les articles en groupes distincts.

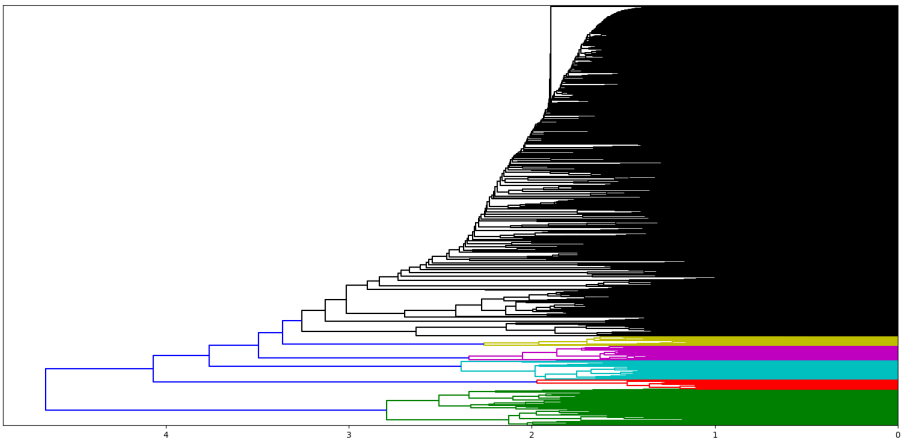

Figure 3: Dendogramme obtenu par Average-linkage

On peut observer sur le Dendogramme que beaucoup d'articles ont une similarité très proche, ce qui fait que la séparation entre les articles est souvent très fine (elle l'est d'ailleurs encore plus avec Single-linkage). Cela provoque des problèmes de clustering (pour k = 10, on obtient un cluster avec 1342 articles soit plus de la moitié). Malgré tout on peut observer plusieurs phénomènes intéressants :

-certains clusters précis indiquent un évènement particulier traité par de nombreux articles. Avec par exemple, une figure contenant les titres d'un cluster pour k = 10.

```
killing me inside how lebron confronts racism could be another crowning moment
lebron james responds to racist vandalism of his la home
against lebron shows racism very much alive in today s america
how emmett till s murder affected muhammad ali
on eve of finals lebron james has somber response to apparent
lebron james brentwood home vandalized
lebron james is the target of a possible
lebron james discusses racism in america after being target of alleged
is alive every single day lebron james says after racist graffiti incident
how lebron james investigation highlights broader concern
lebron james responds to la home being vandalized with racial slur
lebron james says being black in america is tough following his la home being
vandalized
after being targeted in a lebron james gives america a wake up call on racism
what happened to lebron james is sadly one of many disturbing incidents lately
racists attack smithsonian lebron james home
targeting lebron james reminds us that is alive and well in the good old usa
lebron james being black in america is tough
```

Figure 4: Exemple d'un cluster dans lequel tous les articles traitent du sujet d'un acte de vandalisme contre le joueur de basket Lebron James.

-certains clusters moins précis indiquent une communauté en particulier.

Ces résultats sont intéressants : nous arrivons, simplement par les termes utilisés, à retrouver des articles qui traitent du même sujet malgré la grande variété des titres.

Mais cette approche simpliste ne prend pas en compte la complexité d'un graphe de documents et se limite à une mesure de similarité entre deux articles. Lorsqu'il y a de nombreux articles avec une distance très proche, cela entraîne la création de clusters déséquilibrés, rendant l'analyse difficile. Il est donc nécessaire d'adopter des méthodes plus avancées qui considèrent la structure globale du graphe .

2.2 Clustering par Modularity maximization

Pour faire cette analyse plus poussé de nos données, on utilise la similarité cosinus. Nous considérons un graphe où les articles sont les points et la similarité entre les aricles correspond aux poids des arêtes. Nous ne gardons que les arêtes dont le poids est suffisamment important ($similarité \geq 0.1$ pour ne conserver qu'environ 1% des arêtes). Cela nous donne un graphe assez lourd mais prenant en compte le maximum d'informations tout en étant analysable dans les temps dont nous disposions.

Nous pouvons alors appliquer des méthodes de clustering prenant en compte la structure globale du graphe. Nous avons choisi un clustering par Modularity maximization pour plusieurs raisons :

- Détection de structures communautaires : Contrairement à certaines méthodes de clustering qui cherchent à minimiser une fonction de coût globale, la maximisation de la modularité se concentre spécifiquement sur la détection de structures communautaires intrinsèques dans un réseau ou un graphe.

- Flexibilité dans le nombre de clusters : Contrairement aux méthodes de clustering qui nécessitent une spécification préalable du nombre de clusters souhaité, la maximisation de la modularité ne requiert pas cette information. Ici, nous ne savons pas exactement quel nombre de clusters est le plus intelligent.

- Interprétabilité des résultats : Les clusters obtenus par la maximisation de la modularité sont basés sur la structure du graphe et les interactions entre les nœuds. Cela permet une interprétation plus intuitive et une meilleure compréhension des résultats de clustering. Les clusters peuvent représenter des articles qui ont des liens significatifs entre eux.

Pour effectuer les clusterings par Modularity maximization, nous avons utilisé Gephi à partir des fichiers csv obtenus grâce au script python, ce qui nous permettait de directement visualiser les clusters. Voici le résultat visuel que nous avons obtenu :

Figure 5: Clustering par Modularity maximization.

On obtient 37 clusters dont 17 qui n'ont qu'un article (en général des titres très particuliers comme "Marshmallow attack : Hate crime or effective therapy technique ? Popdust"). Mais les 20 autres clusters ont tous des tailles raisonnables, ce qui nous permets d'analyser plus précisément chacun des clusters contrairement à l'énorme cluster obtenu par Average-linkage. Nous obtenons d'ailleurs une modularité de 0.764, ce qui est un indicateurs d'un bon regroupement par clusters.

Pour comprendre le sujet principal de chacun de ces clusters nous les avons importé sur Neo4j. Et nous avons pu observer une très bonne répartition par type de "hate crime". Par exemple, voici les titres d'articles d'un cluster qui regroupe uniquement des articles traitant du sujet de la transidentité :

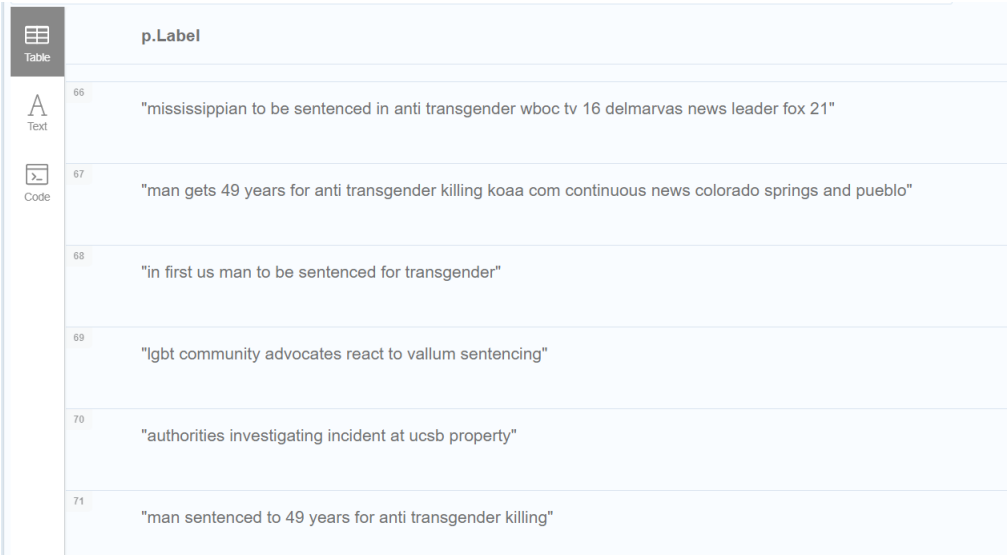

Figure 6: L'un des clusters regroupant uniquement des articles traitant du sujet de la transidentité.

Nous sommes fiers de ce résultat, car nous avons réussi à regrouper les articles non seulement en fonction des mots utilisés, mais également en fonction de thèmes spécifiques, même lorsque certains titres ne partagent aucun mot en commun. Ce résultat aurait été impossible à obtenir uniquement avec une approche de liaison simple, car il aurait fallu prendre en compte l'ensemble du graphe pour établir des liens entre des articles traitant du même sujet sans utiliser de termes similaires.

Maintenant que l'on sait que les articles sont regroupés en grands thèmes (malgré plusieurs erreurs de rangement, et des clusters qui sont parfois difficiles à catégoriser), nous pouvons revenir à notre question de la différence de représentation des communautés en étudiant les thèmes de chacun de ces clusters. Pour cela, nous avons cherché les noeuds de plus grande "Betweenness centrality" dans chacun des clusters (on extrait un sous-graphe contenant uniquement les noeuds du cluster). Ces noeuds sont centraux dans chacun des clusters car ils font le lien entre les différents fait divers traités par les articles du cluster et ainsi, ils nous permettent de connaître les thèmes des clusters.

Pour effectuer cette recherche, nous avons utilisé le module GDS de Neo4j. Pour le cluster concernant la transidentité (cluster numéro 16), nous utilisons les commandes suivantes :

```
CALL gds.betweenness.stream('Graph_16')
YIELD nodeId, score
RETURN gds.util.asNode(nodeId).Id AS Id, score
ORDER BY score DESC
```

Figure 7: Commandes permettant d'obtenir le noeud de plus grande "Betweenness centrality"

Nous avons ainsi des clusters et des thèmes pour les clusters, nous allons ainsi pouvoir essayer d'apporter de nouveaux arguments pour expliquer les résultats obtenus dans la partie 1.

2.3 Analyses des résultats obtenus

Voici les résultats obtenus :

- Un dendogramme permettant de regrouper des articles traitant du même fait divers.
- Des clusters regroupant les articles par grands thèmes avec le thèmes de ces clusters obtenus en regardant la "Betweenness centrality".

A partir de ces résultats, nous avons pu déduire quelques remarques pouvant expliquer, autrement que par des explications socio-politiques, les résultats obtenus partie 1.

Premièrement, nous avons vu que nous pouvions regrouper les articles par fait divers. Nous nous sommes alors rendus comptes que certains événements, comme le cambriolage du joueur de basket Lebron James, étaient relaté par une centaine d'article sur les 2000 étudiés dans cette partie. Ainsi, un événement unique peut changer la médiatisation des "hate crimes" de toute sa communauté. Nous pouvons alors nous demander si il n'y a pas une surreprésentation de certains "hate crimes".

De plus, nous pouvons remarquer que certains clusters n'utilisent que très peu de fois les termes désignant les communautés visées par le crime. Voici, par exemple, une partie d'un des cluster parlant des crimes anti-Afro Américains qui traite du meurtre d'un étudiant dans l'état du Maryland (Richard Collins III) :

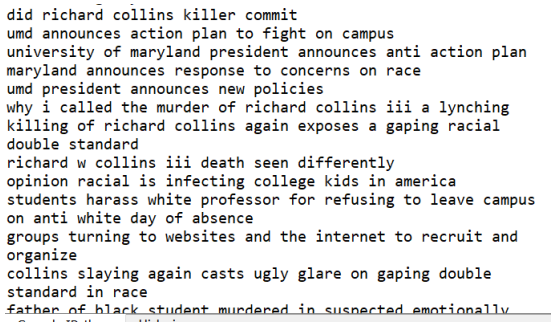

Figure 8: Articles traitant du meurte de Richard Collins sans désigner sa communauté.

Nous constatons qu'il y a très peu de termes faisant référence à l'appartenance à une communauté spécifique, même si le crime a été identifié comme étant motivé par la haine raciale.

Cette observation nous amène à envisager une extension de notre étude, que nous n'avons pas fait ici, qui se penche sur la manière dont les communautés sont mentionnées dans ce type d'articles. Cela soulève plusieurs questions : est-ce que certaines communautés sont implicites plutôt que directement mentionnées par les journalistes ? Et si oui, pourquoi ? Est-ce que cela explique les différences observées dans la partie 1, ou est-ce que cela les accentue davantage ?

Nous avons constaté que certains crimes étaient largement médiatisés et nous avons cherché à établir des correspondances entre ces crimes spécifiques et ceux répertoriés dans le dataset du FBI. En utilisant des techniques d'extraction d'informations, nous avons pu récupérer des données à partir d'articles pertinents. Notre approche consistait à croiser ces informations provenant de différents articles traitant du même crime afin de le retrouver dans le dataset du FBI. Cependant, malgré nos efforts, nos tentatives informatiques n'ont pas abouti. Même manuellement pour des crimes connus que nous avons identifiés dans les articles et pour lesquels nous avions toutes les informations nécessaires, nous n'avons pas pu les retrouver dans le dataset du FBI.

Cela nous amène à considérer deux possibilités : soit le dataset du FBI est incomplet et ne contient pas ces crimes spécifiques, soit ces crimes médiatisés sont rangés différemment dans les catégories du FBI.

Dans le cas d'un dataset du FBI complet, nous aurions adopté la démarche suivante :

- Clusteriser les articles se référant au même hate crime
- Grâce aux clusters, croiser les informations provenant des différents articles pour avoir le plus d'informations possibles sur chaque hate crime
- À partir de ces informations, retrouver le hate crime correspondant dans la base de données du FBI

3
WEB SCRAPING ET OUTIL DE GÉNÉRATION DE TITRES

Dans cette partie, nous allons étudier la mise en place d'un outil de génération aléatoire de titres d'articles à partir de notre Dataset des articles. Dans un premier temps, nous verrons comment cet outil a été mis en place. Dans un second temps, nous présenterons une tentative d'enrichissement de la base de données par Web Scraping sur Google.

3.1 Génération automatique de titres à partir d'un graphe de mots

Fichier utilisé : generateur_de_titres.py

Dans cette sous-partie, nous nous focaliserons sur le dataset correspondant aux articles de presse. Notre but était d'analyser la structure grammaticale des titres d'articles pour exploiter tout le potentiel de ce dataset dans notre projet.

Plus précisément, nous avons voulu créer un générateur randomisé de titres d'articles. Pour cela, nous nous sommes basés sur un graphe dirigé de mots. Nous précisons que les titres d'articles générés , de par la nature aléatoire de notre générateur, ne reflètent en rien la réalité. Notre unique but était de créer un outil efficace permettant de simuler des vrais titres d'articles. Le graphe de mots utilisé contient comme noeuds :

- Des noeuds "START" et "END", correspondant au début et à la fin du titre d'article
- Des noeuds correspondant aux mots des articles, tels que "shooting", "lesbian", "Kansas",...

Les arêtes (pondérées) du graphe sont construites de la façon suivante :

1. On itère sur l'ensemble des titres du dataset (11000 titres)
2. À chaque fois que 2 mots se suivent dans la phrase : si aucune arête entre les noeuds correspondants n'existe dans le graphe, on crée une arête de poids 1. Si une arête existe déjà, on augmente son poids de 1.
3. Concernant les noeuds "START" et "END", on réalise le procédé de l'étape 2 entre le noeud "START" et le noeud correspondant au premier mot de la phrase. On fait la même chose entre le noeud correspondant au dernier mot de la phrase et le mot "END"

Dans le cas où l'on a 2 phrases "Je mange du chocolat" et "Il mange du riz", le graphe a l'allure suivante :

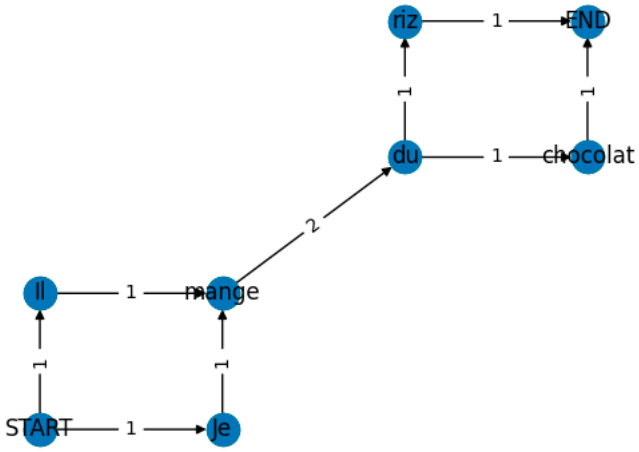

Figure 9: Graphe de mots obtenu dans un cas simple

Le principe de l'algorithme est alors le suivant :

- On commence au noeud "START"
- À chaque étape, on choisit aléatoirement une des arêtes partant du noeud actuel, en pondérant la probabilité de chaque arête par son poids. L'arête choisie mène au prochain noeud.
- La succession des noeuds nous permet alors de construire la phrase, jusqu'au moment où on atteint le noeud "END".

Ainsi dans ce cas, on obtiendra avec probabilité 1/4 chacune des phrases : "Je mange du chocolat", "Il mange du chocolat", "Je mange du riz", "Il mange du riz".

On construit ainsi un graphe à partir des 11000 titres d'articles dont on dispose, et on réalise alors le processus randomisé précédent pour générer des titres aléatoirement.
On sélectionne les titres avec des longueurs d'environ une ligne, car on constate que sinon les résultats sont trop longs et alors trop incohérents, ou trop courts.
Notre première approche a fourni des résultats mitigés (figure 10). En effet, le fait qu'on ne lie un mot qu'au mot d'après fournit des phrases localement cohérentes mais globalement incohérentes.

```
Portland man alleged to have been filed for the 2018 legislation
Editorial: Be 'Hate Crime Against Muslims Angered After Early Doubt
Noose in black college student nothing new garden in memory of hate crime
Man Shot Outside Target Jewish Family Receives Posthumous Degree
Georgia Pair, Who Disfigured Gay Man at Missoula & Weather
```

Figure 10: Génération de titres aléatoires avec notre premier modèle

Nous avons donc décidé de réaliser 2 changements :

- Créer un nouveau graphe G2 sur le même principe que le graphe précédent, sauf qu'il connecte les mots qui ont un mot entre eux dans le titre. Dans le cas de la phrase "Je mange du chocolat", on aurait les arêtes dirigées ("START","mange"),("Je","du"),("mange","chocolat"), ("du","END").
- Dans la décision randomisée du prochain mot, on prend alors en compte à la fois le mot précédent et le mot actuel pour la décision du mot suivant, et non plus seulement le mot actuel. [1]
- Fusionner les mots récurrents tels que "to", "I", "this" avec le mot consécutif. Ainsi dans une phrase telle que "I love this cat", on créérait seulement deux noeuds : "I love" et "this cat".

Ces changements ont permis d'atténuer les défauts précédemment évoqués, bien que le problème de cohérence globale demeure. Nous avons donc décidé de créer un troisième graphe G3 liant les mots qui ont 2 mots entre eux.[2]
Les résultats sont alors assez satisfaisants, avec des titres d'articles qui semblent réels, à quelques erreurs de syntaxe près. (voir figure 11) On précise que les titres présentés sont donnés après élimination des titres déjà existants dans le dataset, et des titres trop courts. Le problème de cet algorithme est en effet qu'il génère 80% de déchets environ, dont 5% de titres déjà existants et 75% de titres trop courts (exemples : "Gay men", "New York Hate Crime", "Minnesota Mosque"). Nous avons donc décidé d'éliminer la possibilité de choisir "END" pour des titres trop courts (quand il existe d'autres solutions), ce qui a réduit le nombre de déchets à 20% environ.

[1] La pondération en probabilité en fonction des arêtes du premier graphe G et du deuxième graphe G2 est la suivante pour un potentiel mot suivant Ms, en supposant que le mot actuel est Ma et le mot précédent Mp :

$$G[Mp][Ms]^2 + G2[Ma][Ms]^2 \tag{1}$$

(Cette formule a été trouvée par tâtonnements successifs,G[i][j] correspond au poids de l'arête de i à j)

[2] La formule devient alors :

$$G[Mp][Ms]^2 + G2[Ma][Ms]^2 + G3[Mpp][Ms]^2 \tag{2}$$

```
Ky. governor pushes 'Charlottesville Provisions' to make violent protests a hate
crime
Man gets 49 years for anti-transgender hate crime
PayPal escalates tech industry's war on racist graffiti
Berkeley Investigates This As Possible Hate Crimes
Police investigate hate crime for allegedly stabbing
Scottish YouTuber Faces a Year in Prison For Hate Crime
Fatal Stabbing of Black Man charged with hate crime
The Latest: Man gets 49 years for anti-transgender hate crime
Man who accidentally shot dead in US, Shooter yells 'get out of my country'
```

Figure 11: Génération de titres aléatoires avec le modèle utilisant trois graphes G, G2 et G3 (Ces titres n'étaient pas déjà existants dans le dataset)

3.2 Web Scraping des résultats de recherche Google

Fichiers utilisés : scraping_google.py, scraping_google_news.csv

Pour améliorer la qualité des résultats et réduire le nombre de déchets, nous avons voulu augmenter le volume de notre dataset en réalisant nous-mêmes le Web Scraping des articles de Google News sur les "hate crimes". Le dataset initial est issu d'un projet de l'organisme de journalisme indépendant ProPublica dont le but est de documenter les "hate crimes", nous n'avions donc pas eu à réaliser l'extraction de données.

Pour la réaliser nous avons utilisé les bibliothèques requests et BeautifulSoup. En recréant le code de l'URL, nous avons navigué sur les différentes pages de Google puis extrait le code source avec requests. Puis nous avons sélectionné les titres d'articles grâce à leur classe HTML et inscrit ces titres dans un fichier csv (scraping_google_news.csv).
Nous avons rencontré plusieurs difficultés lors de cette opération :

- Nous étions au débût bloqués sur la page de cookies de Google. Nous avons dû envoyer le cookie "YES+026" avec requests pour pouvoir en sortir.
- Après avoir extrait trop de données de Google trop rapidement, notre adresse IP a été suspendue pour activité suspecte. Nous avons donc dû espacer les requêtes avec des temps randomisés de l'ordre de la minute pour y échapper.
- Enfin, Google limite le nombre de résultats de recherche à environ 500, ce qui nous a empêché d'en obtenir plus.

Nous avons finalement obtenu un dataset d'environ 500 points (scraping_google_news.csv), étant limités par le nombre de résultats que Google propose. Nous n'avons donc pas pu significativement augmenter notre dataset (contenant initialement 11000 points) et avoir une influence détectable sur le générateur de titres. Une approche pour obtenir plus de points pourrait consister à formuler différentes requêtes autour des hate crimes, comme "hate crime in new york" ou "hate crime in los angeles". À raison de 500 points par requête, on pourrait assez sensiblement augmenter la taille de notre dataset et donc certainement la qualité de notre générateur de titres.

CONCLUSION

En conclusion, notre projet a porté sur l'analyse de données liées aux "hate crimes" à partir d'un dataset sur des articles de presse et un autre du FBI contenant des données officielles. À partir de statistiques simples, nous avons observé des différences dans la représentation des communautés par les médias, ce qui soulève des questions quant à la manière dont les "hate crimes" sont rapportés.

Une analyse des articles sous formes de graphes nous a permis de pousser notre analyse plus loin. A l'aide d'outils de clustering (Agglomerative Clustering avec TF-IDF puis Modularity Maximization), nous avons réalisé des catégorisations à différentes granularités des articles. Cela nous a permis de fournir des explications aux différences de représentation observées en première partie.

Enfin, pour réaliser une exploitation plus poussée de notre dataset, nous avons fait le choix de réaliser un générateur aléatoire de titres d'articles. Plusieurs itérations nous ont permis d'aboutir à un résultat assez satisfaisant. Le Web Scraping, dont le but était d'enrichir la base de données, n'a pas totalement abouti mais les causes ont été identifiées et un dataset plus important pourrait être extrait facilement à partir de notre étude.

En ouverture, nous pourrions mettre en oeuvre des moyens de Natural Language Processing pour déterminer la communauté de la victime dans des articles où cette dernière n'est pas mentionnée explicitement, en la comparant aux articles où elle est mentionnée.