Institut National des Langues et Civilisations Orientales

Département Textes, Informatique, Multilinguisme

Une application d'algorithmes de densité pour la détection de citations en paraphrase

MASTER

TRAITEMENT AUTOMATIQUE DES LANGUES

Parcours :

Ingénierie Multilingue

par

Noélie BOTTERO

Directeur de mémoire :

Pierre Magistry

Encadrants :

Benoît Laurent - Guillaume Lechien

Année universitaire 2021/2022

Résumé

Ce travail porte sur la détection de reprises de citations d'un corpus issu de la presse française, sur le thème de la guerre en Ukraine. Nous utilisons des méthodes de clusterisation par densité afin de repérer les citations identiques et similaires. Nous avons choisi d'utiliser les algorithmes OPTICS et DBSCAN qui permettent de regrouper des formes textuelles similaires sans avoir au préalable de données annotées. Cette application permet également de questionner l'importance du traitement en amont du contenu textuel, de quantifier les performances du modèle et de comparer les résultats des différentes clusterisations. Nous constatons que l'algorithme OPTICS surpasse DBSCAN en termes de scores. La continuité de ce travail serait d'une part, de pouvoir ordonner les citations de manière temporelle, de pouvoir les suivre dans la presse et d'autre part, une aide à l'annotation manuelle.

Mots clés : clustering par densité, citations, presse, dbscan, optics, tfidf, word2vec, doc2vec, scikit-learn

Remerciements

Ce mémoire n'aurait pas été rendu possible sans l'aide et le soutien de certaines personnes que je souhaite aujourd'hui remercier.

Mes remerciements vont d'abord vers mes deux maîtres de stage, Monsieur Benoît Laurent et Monsieur Guillaume Lechien, pour le partage de leurs expériences, leur soutien et leur confiance tout au long du stage. Je remercie l'entreprise Aday de m'avoir permis de réaliser ce stage de fin d'études et de découvrir le monde du traitement automatique des langues en entreprise.

Je souhaite également exprimer toute ma reconnaissance à mon directeur de mémoire, Monsieur Pierre Magistry, pour son suivi, son aide et ses conseils précieux lors de la rédaction de ce mémoire. Je remercie également tous les enseignants du Master qui m'ont permis d'enrichir ma réflexion pendant ces deux années.

Pour finir, je souhaite remercier mes parents pour m'avoir permis de réaliser ces études, pour leur soutien inconditionnel, leur amour et leurs encouragements.

Table des matières

Liste des figures

Liste des tableaux

Première partie

Introduction

0.1 Présentation générale

En grammaire, le propos rapporté est un type de discours visant à énoncer les dires d'une tierce personne, physique ou morale. Il s'agit d'un phénomène courant au sein de la presse. Une nouvelle actualité peut être reprise par différents journaux et c'est également le cas en ce qui concerne les citations. La reprise de citation est encore plus présente aujourd'hui grâce aux ressources disponibles en ligne, aux réseaux sociaux ainsi que par la multitude de journaux existants. Ce travail de recherche porte sur la détection de discours rapportés identiques ou sémantiquement proches dans la presse française. Avant de pouvoir détecter ces citations, nous devons expliquer ce qui est considéré comme un doublon ou non. Dans leurs travaux, [Ferrero and Simac-Lejeune, 2015] distinguent trois niveaux de similarités différents : la copie, la paraphrase et la reformulation. Ils donnent les définitions suivantes :

> "[La copie] consiste à copier mot à mot tout ou partie d'un texte dans un autre. [La paraphrase] aussi appelée reformulation paraphrastique, qui consiste à reprendre une phrase d'un texte pour la détailler ou l'expliciter. Elle conserve donc l'ordre des éléments évoqués, autorisant simplement le changement de vocabulaire, l'ajout, la suppression et la substitution de mots. [La reformulation] autorise elle toutes modifications textuelles à condition que le sens de la phrase soit conservé. Cela donne souvent lieu à un changement d'ordre des concepts [1]."

Nous prendrons ces trois définitions comme références pour caractériser une citation en doublon. Nous explorerons plusieurs méthodes existantes issues du traitement automatique des langues afin de répondre à la problématique principale : quels sont les algorithmes à notre disposition permettant de classer différents contenus textuels en fonction de leur similarité.

Nous devrons également définir la question de la donnée. En d'autres termes il nous faut savoir quel corpus choisir et quels traitements effectuer sur ce dernier pour que les méthodes utilisées soient efficaces. Une fois ces objectifs réalisés nous pourrons analyser sur quels critères - arbitraires ou non - les différentes techniques que nous utiliserons se basent pour obtenir une classification efficace.

De plus, la détection de doublons répond aussi à un besoin industriel. Une des perspectives envisagées serait de pouvoir suivre dans le temps les citations. L'Agence France Presse est en charge de collecter, vérifier et diffuser de nouvelles informations, qui seront ensuite reprises par tous types de médias ou d'entreprises. Il arrive donc fréquemment qu'une source soit reprise par plusieurs journaux. Une des demandes de l'entreprise serait de pouvoir savoir où se retrouve une citation qui a été rapportée, par qui, dans quel journal et si elle a subit une transformation quelconque.

0.2 Plan de lecture

Dans un premier temps, nous effectuerons un état de l'art de ces différentes méthodes existantes. A ce jour, ce travail de recherche n'a pas été réalisé sur les cita-

1. [Ferrero and Simac-Lejeune, 2015], page 228-289

tions mais des travaux similaires existent, notamment sur la classification en clusters de textes courts. Nous présenterons donc les différents algorithmes existants en détection de doublons, nous pouvons notamment citer la classification binaire et la clusterisation. Ensuite, nous traiterons de la question de la vectorisation. Nous verrons les travaux effectués sur l'encodage et le décodage du contenu textuel liés à la classification. Pour conclure l'état de l'art, il sera pertinent de présenter les mesures d'évaluations utilisées pour ce genre de tâche.

Dans un second temps, nous présenterons le corpus utilisé dans ce travail et la motivation de ce choix. Dans un troisième temps, nous rentrerons dans le vif du sujet. Nous appliquerons un algorithme de classification à notre corpus normalisé. Nous nous pencherons plus particulièrement sur l'utilisation d'algorithmes non supervisés, qui permettent de regrouper des séquences similaires sans avoir à annoter un corpus. Une annotation manuelle s'avère être une tâche difficile à mettre en oeuvre quand la taille des données est conséquente. De plus, il s'agirait ici de comparer une citation à toutes les autres du corpus ce qui écarte davantage le choix de faire une campagne d'annotation. Une partie importante de ce travail de recherche est de trouver quelles sont les mesures d'évaluations possibles à utiliser pour évaluer des résultats d'algorithme non supervisée. Pour finir, nous mettrons en avant et analyserons les différents résultats obtenus, suivi d'une discussion sur les futurs perspectives que ce travail peut apporter.

Deuxième partie

État de l’art

CHAPITRE 1

DÉTECTION DE DOUBLONS

Sommaire

La détection de doublons est une tâche largement utilisée dans l'informatique en général. Elle consiste à pouvoir repérer des objets identiques ou similaires. Elle peut être appliquée aux images, à la vidéo ou dans le cas du traitement automatique des langues, à du texte ou de la voix. Nous pouvons citer quelques cas d'utilisation en traitement automatique des langues tel que le résumé de documents [Reztaputra and Khodra, 2017], les systèmes de ranking [Li, 2014], ou encore la détection de plagiat [Sandhya and Chitrakala, 2011].

Pour ce travail de recherche nous souhaitons mettre en lien des citations identiques ou modifiées au fil du temps dans différents journaux. Il existe plusieurs méthodes capables de répondre à cette tâche :

— La classification binaire ;
— Le clustering par partitionnement
— Le clustering par densité
— Le clustering hiérarchique

1.1 Classification binaire

La classification binaire consiste à classer des documents, phrases ou mots sous forme de paires. Les recherches récentes consistent à utiliser des systèmes de réseaux de neurones pour effectuer cette classification. C'est le cas de [Agarwal et al., 2018], qui proposent de classer des tweets similaires par paires grâce à des réseaux de neurones convolutifs accompagnés d'un modèle LSTM (Long Short Term Memory). Selon eux, les réseaux de neurones convolutifs sont plus performants dans ce qui est de repérer des caractéristiques locales (au niveau du mot) tandis que les LSTM sont plus à même d'apprendre sur les dépendances à long terme.

Les corpus utilisés pour cette tâche sont, le plus souvent, annotés manuellement en paires de phrases. [Agarwal et al., 2018] utilisent des corpus déjà existants : le

Twitter Paraphrase SemEval2015 et le Microsoft Paraphrase. Ces deux corpus comportent des paires de phrases qui ont été semi-automatiquement annotées en tant que "paraphrases". Une revue manuelle a ensuite été effectuée sur une partie des corpus. Dans ce même travail de recherche, les paires ont été doublées afin d'obtenir un corpus de taille plus conséquente. Une paire correspond à une phrase 1 et une phrase 2. Elles sont alors inversées, la phrase 1 devient la phrase 2, ce qui donne une nouvelle paire.

Les corpus déjà annotés de cette manière sont très rares, et n'existent quasiment qu'en anglais. Dans certains travaux, comme c'est le cas pour [Mahmoud and Zrigui, 2021], le corpus en langue arabe a été créé dans le cadre de cette classification. A partir de phrases créées ont été générées de nouvelles en changeant certains mots et un score de similarité sémantique leur ont été attribué. Ensuite, une annotation manuelle a été réalisée pour vérifier si les phrases générées étaient correctes.

De nombreux travaux ont été réalisés sur la détection de questions en double, en particulier sur des forums. Par exemple, [Wang et al., 2019] utilisent des modèles neuronaux comme les CNN (Réseaux de neurones convolutifs), RNN (Réseaux de neurones récurrents) et LSTM afin de retrouver des questions en double sur StackOverflow [1]. Ils utilisent notamment le *stemming* pour normaliser le texte. Pour rappel, le *stemming*, ou racinisation consiste à réduire un mot à sa forme neutre qui peut être sa forme dans le dictionnaire. Par exemple, toutes les flexions du verbe "manger" seront alors remplacées par sa forme canonique. Cette technique de normalisation permet, dans le cas de la détection de doublons, d'effectuer un tri automatique en réunissant dans un même groupe du contenu textuel identique. Le lien commun entre ces travaux de recherche est l'utilisation de corpus composés de séquences courtes : phrases, tweets ou titres.

Les méthodes non supervisées sont aujourd'hui les plus utilisées en classification binaire. [Mahmoud and Zrigui, 2021] utilisent un modèle LSTM Bidirectionnel afin de capturer les informations sémantiques d'un mot combiné à son prédécesseur et successeur. Ils montrent également que les réseaux de neurones convolutifs sont performants sur l'apprentissage des contextes de phrases de différentes longueurs par des couches de convolutions et de pooling. Ils sont moins efficaces sur un contexte de plus de trois tokens. Le modèle LSTM non bidirectionnel permet d'effectuer un apprentissage efficace sur des séquences plus longues. Par conséquent, le modèle Bi-LSTM obtient de meilleurs résultats, de par sa capacité à manipuler de longues séquences et de stocker les dépendances sur le long terme, ce qui permet de capturer plus d'information contextuelle que les autres modèles.

1.2 Clusterisation

Il existe beaucoup de travaux sur la détection de doublons, notamment sur l'application d'algorithmes de clusterisation permettant de répondre à cette tâche. La clusterisation permet principalement d'identifier le degré de similarité entre des objets physiques ou non, afin de les regrouper dans des clusters. Pour ce qui est du

1. Lien vers le site de StackOverflow

contenu textuel, le clustering sert à trouver des similitudes sémantiques entre des mots, phrases ou documents.

Le *data clustering* se décline en plusieurs grandes classes d'algorithmes. Il en existe plusieurs types. D'une part, le partitionnement de données dont l'algorithme le plus connu est K-means. D'autre part, les méthodes hiérarchiques où les données sont classées sous forme de dendogrammes, c'est-à-dire en une représentation arborescente. Ensuite, le clustering par densité où les données sont classées en un nombre spécifique de groupes selon un critère de distance entre les points. Il existe également des algorithmes *grid-based* où les observations sont réunies dans un quadrillage.

1.2.1 K-Means et l'elbow method

L'algorithme de clusterisation non supervisé le plus utilisé est appelé "K-Means". Il permet de détecter un nombre défini de clusters, groupes avec des éléments similaires que l'on souhaite obtenir. De manière itérative, l'algorithme positionne les éléments dans un même groupe lorsque la distance entre un point et le centre de son cluster est la plus faible possible [Alian and Awajan, 2020].

[Yuan and Yang, 2019] nous donne une explication sur le fonctionnement de l'algorithme :

> "En utilisant la distance comme mesure et étant donné K, le nombre de clusters dans le set de données, l'algorithme calcule la moyenne de la distance en fonction du point central initial, chaque classe étant décrite par le centroïde."

Donc, l'algorithme K-Means se base principalement sur un nombre de groupes défini au préalable. Cependant, il est parfois difficile de connaître à l'avance ce nombre de clusters. Lorsque que ce nombre n'est pas connu, certains chercheurs utilisent l'*elbow method* qui permet de déterminer à l'avance le nombre optimal de clusters k. Cette mesure se base sur le point central d'un groupe et le carré des distances entre les éléments de chaque groupe [Yuan and Yang, 2019]. En d'autres termes, l'algorithme sera lancé plusieurs fois avec un nombre de k incrémenté de 1 à chaque lancement. La, ou les valeurs de k précédant une stagnation ou une chute des scores d'évaluation indique que l'*elbow* a été atteint [Bholowalia and Kumar, 2014].

[Weißer et al., 2020] présentent une méthode K-means servant à filtrer des sujets de la littérature scientifique du traitement automatique des langues. Ils utilisent en premier lieu l'*elbow method* pour trouver le nombre optimal de clusters à fournir à l'algorithme. L'inertie (SSE, *Sum of Squared Errors*) intervient dans le calcul du coude en tant qu'indicateur de performance. Pour déterminer le nombre de clusters à choisir, plusieurs essais avec différentes valeurs de K sont effectués et le SSE est calculé. Plus le SSE sera minimisé, plus la valeur de K sera un bon candidat comme nombre de cluster optimal. La sortie de ces expériences se présente généralement sous forme de courbe. L'endroit de la courbe qui forme un coude pointe la valeur de k optimale.

Dans les travaux de [Oliveira and Sperandio Nascimento, 2021], l'*elbow method* a été utilisée pour trouver le point où l'inertie [2] commence à diminuer en gardant une valeur de K la plus faible possible. Ainsi, 31 valeurs de K ont été utilisées dans une fourchette de 30 à 61, en sélectionnant le K qui génère le meilleur clustering. Cette méthode a été utilisée afin de détecter le degré de similarité entre documents judiciaires brésiliens.

1.2.2 Clustering

Le clustering basé sur la densité permet de mesurer la similarité entre plusieurs points. Il sert à regrouper des objets qui présentent des caractéristiques semblables et ainsi de les différencier d'autres objets aux propriétés différentes sans avoir une idée du nombre de clusters. [Mohammed et al., 2021] fait état des différentes méthodes de clustering. Il prend également en compte certaines mesures d'évaluation pour quantifier les performances de chacun de ces algorithmes. Les plus utilisées sont DBSCAN [3] et OPTICS [4]. Ces deux algorithmes n'ont pas besoin d'avoir comme paramètre un nombre de clusters défini. DBSCAN a été proposé en 1996 par [Ester et al., 1996] et OPTICS par [Ankerst et al., 1999] en 1999.

Plus d'une vingtaine d'années après leur création, ces deux techniques de clusterisations sont encore largement utilisées de nos jours.

[Jang et al., 2016] utilisent un algorithme de densité pour un système de reconnaissance d'intention de dialogues, couplé à des plongements lexicaux de type Word2Vec. Les dimensions des vecteurs de sorties étant élevées à cause de Word2Vec, une réduction des vecteurs à deux dimensions a été effectuée à l'aide d'un *embedding* nommé T-SNE (T-Distributed Stochastic Neighbor Embedding). L'algorithme de clusterisation choisi est DENCLUE, une amélioration de DBSCAN. [Liao and Cheng, 2016] appliquent l'algorithme DBSCAN avec une vectorisation en Word2Vec afin de regrouper les mots par similarité et analyser les différences sémantiques entre les mots. [Schmitt and Spinosa, 2018] exécutent une analyse de sentiment grâce à des réseaux de neurones convolutifs. Ils utilisent l'algorithme DBSCAN pour regrouper les mots similaires en clusters et supprimer les points de bruit. Ils montrent que les éléments non clusterisés ont un impact important sur les résultats.

1.2.3 OPTICS

Le deuxième algorithme qui nous intéresse est OPTICS. Il possède le même fonctionnement que DBSCAN, il est en quelque sorte une extension de celui-ci mais il calcule simultanément les différentes distances, c'est-à-dire qu'il partitionne sur des rayons de clusters de tailles différentes. Le paramètre epsilon correspond donc à la valeur du rayon autour d'un objet. Il crée des rayons de tailles différentes à l'instar de DBSCAN où la distance epsilon reste fixe. Si un objet se trouve dans ce rayon, cela signifie qu'il fait partie du noyau de densité défini par le paramètre epsilon. Si, dans cette zone, le nombre minimal de points (*min_samples*) est atteint, alors cette zone deviendra un cluster (Figure 1.1).

2. La somme du carré de la distance euclidienne de chaque point à son centroïde
3. Density-Based Spatial Clustering of Applications with Noise
4. Ordering Points to Identify the Clustering Structure

DBSCAN parvient à sa limite lorsqu'il existe un trop grand nombre de densités différentes. Au final, il essaie de conserver les clusters de plus haute densité, ce qui correspond au plus petit epsilon. OPTICS joue un rôle important sur des représentations d'éléments de la langue qui auront des densités variables.

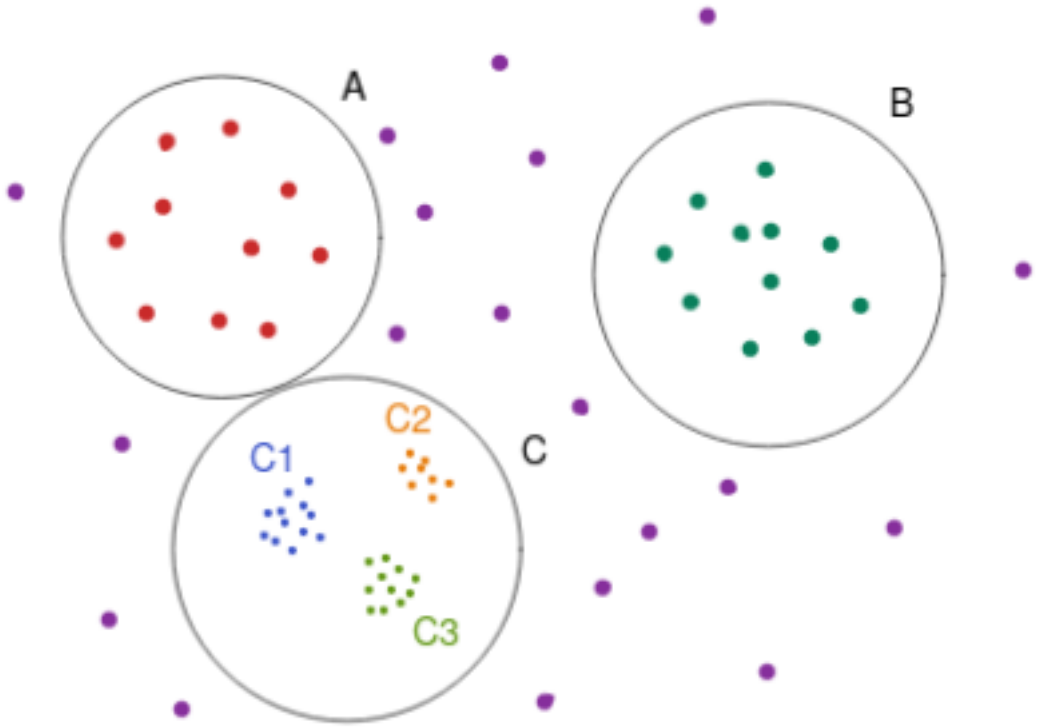

FIGURE 1.1 – Schéma de clusterisation de différentes densités, inspiré de [Ankerst et al., 1999]

Si nous gardons une seule valeur d'epsilon (d'écartement maximum entre les points d'un cluster), il sera impossible de distinguer C1, C2 et C3. Nous aurons juste A, B et C. OPTICS permet de partitionner un espace de données qui a des densités variables.

L'algorithme OPTICS semble être moins utilisé que DBSCAN. Néanmoins, il existe des travaux de recherche sur l'explication de l'algorithme et ses améliorations possibles. [Schubert and Gertz, 2018] cherchent à améliorer la structure des clusters extraits depuis l'algorithme OPTICS. [Bansal et al., 2022] tentent de créer automatiquement des playlists de chaînes Youtube grâce à des algorithmes de clusterisation. Ils comparent différentes méthodes dont K-Means++, DBSCAN et OPTICS. [Vijayan and P, 2021] utilisent le clustering pour créer des vocabulaires d'images, aussi appelés sacs de mots visuels (Bag of Visual Words). Cette méthode consiste à extraire les caractéristiques importantes d'une image. Si l'on prend un portrait, les points clés seront les yeux, le nez ou encore la bouche. Plusieurs algorithmes de clustering sont testés : K-means, Mini-Batch K-Means, DBSCAN et OPTICS. La méthode la plus adaptée à cette tâche est OPTICS, qui obtient un score de précision de 79.01%.

Pour citer une autre recherche, [Farnaghi et al., 2020] tentent de récupérer des éléments spatio-temporels dans des tweets afin de repérer automatiquement des catastrophes [5]. Une autre méthode de classification existante est le grid-based clustering. Cet algorithme divise les données en un nombre fini de cellules pour former un quadrillage, où les clusters sont déterminés à partir de cette grille.

5. Leur cas d'étude porte sur l'ouragan Florence, survenu en 2018.

CHAPITRE 2

VECTORISATION DE TEXTES

Sommaire

En traitement automatique des langues la représentation vectorielle du sens permet de quantifier ou comparer la similarité entre le sens de différents mots.

2.1 TF-IDF

Le TF-IDF (term frequency–inverse document frequency) sert à refléter l'importance d'un mot dans un corpus ou une collection de documents. La valeur TF-IDF d'un mot augmente proportionnellement au nombre d'occurrences de ce mot dans un document, et cette augmentation est compensée par le nombre de documents du corpus contenant ce mot. Cette propriété permet de mieux concilier le fait que certains mots sont généralement plus susceptibles d'apparaître. Aujourd'hui, le TF-IDF compte parmi les méthodes de pondération des mots les plus utilisées.

Cet acronyme signifie "Term Frequency and Inverse Document Frequency". Sa formule est la suivante :

$$idf_i = log \frac{|D|}{|d \in D : t \in d|} \tag{2.1}$$

$$tfidf_{t,f} = tf_{t,d} \cdot idf_{t,D} \tag{2.2}$$

tf est la fréquence du terme t dans un document d. D est le nombre total de documents dans le corpus et d indique le nombre de documents qui contiennent le terme t.

La pondération TF-IDF permet de déterminer quelles proportions certains mots d'un document peuvent avoir par rapport au reste du corpus. Cette pondération calcule le nombre de fois où le mot-clé apparaît dans un document, multiplié par le nombre de fois où ce mot apparaît et par sa fréquence dans un ensemble de documents. Cependant, avec cette méthode, nous perdons le contexte des mots [Mohammed et al., 2021]. Pour notre tâche il est important de connaître les différents contextes d'un mot afin de pouvoir différencier les citations. Certaines citations peuvent avoir les mêmes mots-clés mais ne pas être pour autant de la paraphrase.

Dans les travaux de [Weißer et al., 2020], le TF-IDF est utilisé afin de regrouper des articles scientifiques par thématiques :

> "Le résultat de la pipeline est un ensemble de vecteurs, chacun représentant un titre dans le corpus [...], la longueur du vecteur étant la quantité de mots dans l'ensemble du corpus. Cette matrice terme-document avec des décimales TF-IDF est adaptée au traitement et à l'analyse mathématique [1]".

Ils utilisent le TF-IDF pour vectoriser les titres, mots clés et résumés d'articles mais ils précisent que les plongements lexicaux sont plus adaptés sur des textes longs. La vectorisation grâce au TF-IDF renvoie une bonne prise en compte des termes spécifiques mais elle est fortement dépendante des termes présents.

2.2 Plongements lexicaux

Un plongement lexical (*word embedding*) est un terme qui désigne des méthodes de modélisation de contenu textuel dans le traitement automatique des langues, où les mots sont positionnés dans un espace vectoriel. La vectorisation s'effectue à partir d'un espace comportant autant de dimensions que de mots, vers un espace vectoriel continu de dimensions plus faibles. La plupart des nouvelles techniques de plongements sont basées sur des architectures de réseaux neuronaux plutôt que sur les modèles traditionnels de type n-gram ou sur l'apprentissage non supervisé. [Mikolov et al., 2013] ont mis au point des représentations sous forme de plongements lexicaux, il s'agit de Word2Vec [2] et Doc2Vec [3].

Avec la méthode en plongement de mots de Word2Vec, la vectorisation est effectuée en déterminant avec quel mot, le mot cible apparaît le plus souvent. La proximité sémantique entre les tokens est également calculée. A l'instar du TF-IDF, Word2Vec se base sur une méthode non supervisée pour créer une vectorisation des mots. Un réseau de neurones est entraîné sur des données non étiquetées afin de créer le modèle qui générera les vecteurs de mots. Word2Vec se décompose en deux formes différentes. La première, appelée Continous-Bag-OF-Words (CBOW), consiste à prédire un mot selon le contexte, c'est-à-dire les mots voisins. La seconde, appelée Skip-Gram, utilise le mot courant pour prédire les mots du contexte environnant. Plus le mot courant est proche, plus le poids du mot contextuel est important.

La représentation Doc2Vec consiste en une extension de Word2Vec. Il possède le même fonctionnement que Word2Vec mais une matrice supplémentaire est générée, représentant la vectorisation du paragraphe [Mikolov et al., 2013]. La vectorisation en Doc2Vec permet de mapper chaque paragraphe d'un document en un unique vecteur en utilisant son identifiant et la représentation vectorielle de ses mots [Magalhães et al., 2020].

Il a été démontré que l'utilisation de plongements de mots améliorent les performances des tâches en traitement automatique des langues. Selon

1. [Weißer et al., 2020], page 3
2. Word to Vector
3. Document to Vector

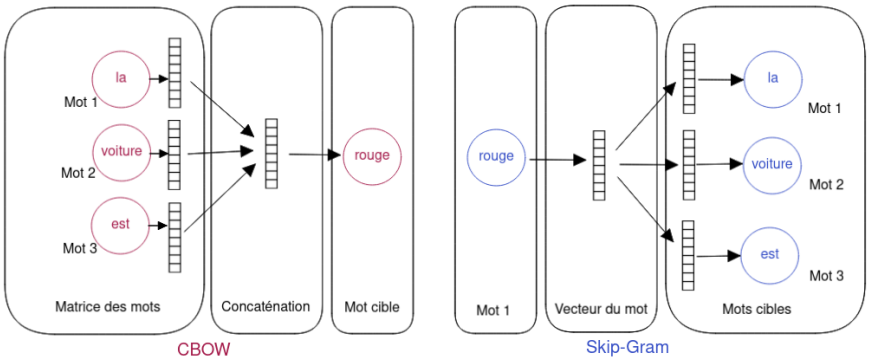

FIGURE 2.1 – Représentation des modèles CBOW et Skip-Gram

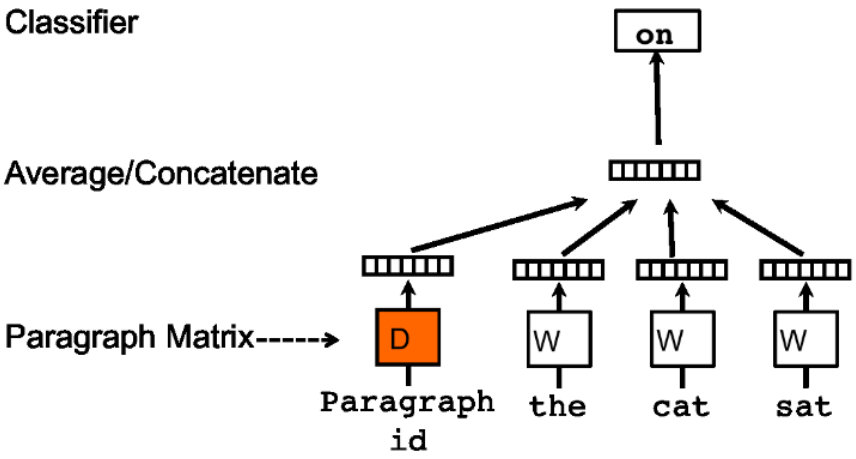

FIGURE 2.2 – Représentation de l'implémentation de Doc2Vec [Le and Mikolov, 2014]

[Mohammed et al., 2021], ils permettent de représenter les vecteurs de dimensions plus faibles et ainsi d'améliorer la similarité des co-occurrences dans ceux-ci :

> "L'utilisation de données à grandes dimensions ou de grande taille comme paramètres pour les algorithmes de clustering entraînera une faible précision et des performances instables. Par conséquent, la meilleure solution à ce problème est d'utiliser des embeddings de mot pour représenter les données du corpus dans un espace vectoriel de faible dimension et pour améliorer la similarité des co-occurrences mots à mots des vecteurs d'une manière très efficace [4]".

[Magalhães et al., 2020] préfèrent utiliser des embeddings au TF-IDF pour analyser différentes mesures de distances à travers des techniques de convolution dans des graphes. [Weißer et al., 2020] indiquent également que les plongements de mots sont plus performants sur des textes longs.

4. [Mohamed and Oussalah, 2020], page 553

CHAPITRE 3

MESURES D'ÉVALUATION

Sommaire

La validation et l'évaluation des résultats issus d'une clusterisation, et qui plus est, d'une méthode non supervisée est primordiale. La validation de la clusterisation constitue un sujet épineux dans la littérature scientifique. Selon si les données données à l'algorithme sont étiquetées ou non, les mesures utilisées peuvent être différentes. Il existe deux méthodes principales pour évaluer une clusterisation basée sur la densité : le score de silhouette et l'indice de Davies-Bouldin.

3.1 Score de silhouette

Cette mesure a été mise au point par [Rousseeuw, 1987] en 1987. Le score de silhouette est une mesure d'évaluation utilisée pour mesurer la qualité d'une clusterisation. Pour rentrer plus dans les détails, cette mesure prend en compte deux critères : l'homogénéité des clusters et la séparation entre les points. Pour un point i, le score de silhouette $s(i)$ permet d'évaluer si l'élément appartient bien au cluster désigné et à quel point il est proche du centre de celui-ci. Cette mesure renvoie un score entre -1 et 1, 1 signifiant que les clusters trouvés sont bien délimités. Une valeur proche de 0 montrent une bonne clusterisation mais certains points ne font pas partie du bon groupe. Pour calculer un score de silhouette, deux paramètres sont nécessaires. Le premier est le résultat de la clusterisation et le second, les différentes "proximités" entre les points qui représentent la distance entre eux.

$$s(i) = \frac{b(i) - a(i)}{max(a(i)), b(i))} \tag{3.1}$$

> "Soit $a(i)$ la non-similarité moyenne d'un objet i par rapport à tous les autres objets de son propre cluster A. A représente le deuxième meilleur choix pour i, ainsi que $b(i)$ comme la non-similarité moyenne de i par rapport aux objets du cluster le plus proche de A [1]."

1. [Weißer et al., 2020], page 4

3.2 Indice de Davies-Bouldin

Cette mesure, introduite par [Davies and Bouldin, 1979], calcule la similarité moyenne de chaque cluster avec le cluster le plus proche. Elle possède plusieurs propriétés :

- La similarité n'est jamais négative
- Les similarités entre deux clusters sont symétriques
- Si la similarité entre deux clusters est égale à 0 cela signifie que la dispersion est nulle ; il s'agit donc de clusters identiques. Si la distance entre les clusters augmente alors que leur dispersion reste stable, alors la similarité baisse. Si la distance entre clusters reste constante alors que la dispersion augmente alors la similarité augmente également.

[Mary, 2012] utilise cette mesure d'évaluation afin de mesurer la qualité de la clusterisation d'algorithmes et notamment de DBSCAN. Sa formule est la suivante :

$$DBI = \frac{1}{n}\sum_{i=1}^{n} max_{i \neq j}\left(\frac{S_n(Q_i) + S_n(Q_j)}{S(Q_{i'}Q_i)}\right) \tag{3.2}$$

n correspond au nombre de clusters, s_n, la distance moyenne de tous les éléments d'un cluster comparé au centre de leur cluster et $S(Q_i, Q_j)$, la distance entre les centroïdes des clusters [Mary, 2012]. Cette mesure nous intéresse car elle demande le moins possible d'intervention humaine dans le choix des paramètres pour que la mesure soit la moins supervisée possible par rapport à nos algorithmes, eux mêmes non supervisés.

Troisième partie

Corpus

CHAPITRE 4

LA QUESTION DES DONNÉES

Sommaire

4.1 Introduction

Dans ce chapitre, nous allons expliciter le processus derrière le choix des données pour ce travail de recherche. Nous commencerons par présenter le corpus utilisé pour ce projet et comment il a été constitué.

Ensuite, nous expliquerons les différents traitements réalisés sur ce corpus de l'extraction des citations à sa transformation en vecteur.

Le premier questionnement qui se pose quand il s'agit de créer une tâche en traitement automatique des langues, est le corpus à utiliser. Dans notre cas, nous avons besoin de constituer un corpus dans lequel des doublons ou des séquences similaires seront présentes.

4.2 KiDiKoi

Au cours de cette deuxième année de Master, nous avons travaillé en collaboration avec l'entreprise Aday sur un algorithme permettant de détecter du discours rapporté dans des articles de presse. Le discours rapporté, ou citation est un énoncé produit par une personne physique ou morale. Notre ligne directrice était de récupérer ces énoncés. Nous avions besoin d'un marqueur afin de pouvoir délimiter ces citations donc nous avons fait le choix de prendre en compte uniquement les passages entre guillemets.

En plus des citations, nous extrayions également le locuteur de la citation, le relateur et le vecteur. Le relateur correspond au passage introduisant le discours rapporté. Le vecteur est un moyen utilisé par le locuteur pour transmettre un discours. Un second algorithme a été réalisé dans le but de relier ces trois éléments à leur citation correspondante. Les deux tableaux ci-dessous font état des performances des deux modèles.

Types d'entités	**Précision**	**Rappel**	**F-mesure**
Total	0,862	0,760	0,808
Citation	0,918	0,922	0,920
Source	0,837	0,632	0,720
Relateur	0,825	0,728	0,773
Vecteur	0,643	0,400	0,493

TABLE 4.1 – Scores du modèle NER de KiDiKoi

Type de relation	**Précision**	**Rappel**	**F-Mesure**
Citation_Source	0.643	0.828	0.724
Citation_Relateur	0.582	0.863	0.695
Citation_Vecteur	0.625	0.092	0.161

TABLE 4.2 – Scores du modèle REL de KiDiKoi

En travaillant sur ce projet, nous nous sommes rendus compte que certaines citations pouvaient être reprises d'un article à l'autre, quelquefois en restant identiques et d'autres fois en étant modifiées :

Citation	Journal & Date de parution
la chine est disposée à travailler avec la russie pour assumer sa responsabilité de grande puissance, jouer un rôle de premier plan et injecter de la stabilité et de l'énergie positive dans un monde parcouru par le chaos	Le Figaro 15/09/2022
la chine est disposée à travailler avec la russie pour assumer leur responsabilité de grandes puissances, jouer un rôle de premier plan et injecter de la stabilité et de l'énergie positive dans un monde parcouru par le chaos	Le Figaro 15/06/2022
des mesures supplémentaires pour renforcer la sécurité et la stabilité de notre continent	Le Figaro 22/06/2022
mesures additionnelles pour renforcer la sécurité	Le Parisien 19/05/2022
si l'ukraine tombe, alors les pays baltes seront les prochains	Le Figaro 13/06/2022
si l'ukraine tombe, alors les pays baltes seront les prochains	Les Echos 14/06/2022
si ça continue comme ça, cela aura des conséquences sérieuses (...) pour le secteur alimentaire dans l'ensemble, la hausse de l'inflation sera inévitable	Le Figaro 10/03/2022
si ça continue comme ça, cela aura des conséquences sérieuses (. . .) pour le secteur alimentaire dans l' ensemble	Le Parisien 10/03/2022
a quoi bon un tel monde, si la russie n'en fait plus partie ?	Le Monde 27/09/2022
a quoi bon un monde sans la russie ?	Le Monde 05/05/2022

TABLE 4.3 – Exemples de citations copiées ou modifiées

CHAPITRE 5

CONSTITUTION DU CORPUS

Il n'existe pas, dans la littérature scientifique, de corpus en français comportant uniquement du discours rapporté. L'entreprise Aday, faisant principalement de la veille média, a accès à la quasi-totalité de la presse française. Le corpus a donc pu être récupéré grâce à un programme mis à disposition par l'entreprise. Il s'agit de plusieurs scripts permettant de récupérer des articles de presse en français depuis leur base de données. Nous avons choisi d'extraire les fichiers au format XML car ils devront être en premier lieu passés au modèle de détection d'entités de KiDiKoi, qui prend en entrée du format XML.

Dans le cadre de ce mémoire, nous nous intéresserons uniquement à l'extraction de *spans* de type citations. Les scores du modèle sur les relations, c'est-à-dire les liens entre la citation et ses dépendances ne sont pas assez concluants pour que le modèle soit utilisé dans ce travail de recherche. L'utiliser pourrait fausser l'évaluation et l'interprétation des résultats.

Nous avons fait le choix initial de sélectionner des articles parmi les dix journaux les plus connus. Cela permettra d'obtenir le plus d'articles possibles et de maximiser les chances de voir apparaître des citations rapportées. Notre corpus a pour thématique la guerre en Ukraine. Afin d'obtenir des documents traitant uniquement de la guerre en Ukraine, nous avons délimité la recherche d'une part au niveau des dates : du 25 février 2022, date de début de la guerre, jusqu'au 1er octobre 2022. D'autre part, nous avons sélectionné des mots clés devant être présents dans les articles pour délimiter le sujet : [Ukraine, Russie, Poutine, Zelensky].

Journal	**Nombre d'articles**	**Nombre de citations**	**Code journal**
Le Figaro	2978	9168	FIGR
Le Parisien	1503	4515	PAFR
Le Monde	1212	4309	MOFR
Ouest-France	862	551	OUFR
Les Échos	534	890	ECHO
Libération	327	863	LIBE
Aujourd'hui en France	301	598	PAAF
Humanités	240	409	HUMW
L'Équipe	46	63	EQIP
Total	8003	21366	

TABLE 5.1 – Corpus sur le thème de la guerre en Ukraine

Une fois les articles récupérés, ils sont traités par le modèle de reconnaissance d'entités nommées de KiDiKoi afin de récupérer les citations. Bien que le label "citation" du modèle a un score de F-mesure de 0.92, certaines citations provenant d'un même article sont détectées deux fois. Pour cette raison et pour ne pas fausser en amont la clusterisation, nous sommes passés par un pré-traitement des doublons pour ne garder, dans un même fichier, qu'une seule fois une même citation.


```
{"results": [{
  "citation":
    {"text": "« allait continuer son agression jusqu'à ce que nous capitulions, ce qui n'arrivera jamais »", "start": 68, "end": 160},
  "source":
    {"text": "Dmytro Kuleba", "start": 6, "end": 19}},
  {
  "citation":
    {"text": "« management externe »", "start": 137, "end": 159},
  "source":
    {"text": "Vladimir Poutine", "start": 0, "end": 16},
  "relateur":
    {"text": "a estimé que", "start": 17, "end": 29}}
]}

```


FIGURE 5.1 – Exemple de sortie du modèle NER de KiDiKoi

CHAPITRE 6

NORMALISATION DES DONNÉES

Pour que plusieurs citations identiques ou similaires puissent être classées dans un même groupe, une étape de normalisation est nécessaire. Après de premiers essais de clusterisation, nous avons pu constater les variations qui peuvent exister entre deux séquences similaires mais qui ne sont pas complètement identiques. Ci-dessous se trouve une liste des différentes normalisations effectuées sur le corpus. Grâce à ces différents tests, nous avons ensuite tokénisé les phrases grâce au module NLTK[1] et avons pu déterminer quels étaient les traitements pouvant potentiellement améliorer le clustering, et ceux qui étaient à mettre de côté.

1. Mise en minuscule
2. Suppression des guillemets
3. Suppression des espaces en double
4. Suppression des citations de longueur cinq ou moins
5. Suppression des mots grammaticaux
6. Racinisation des termes

Nous avons dans un premier temps décidé de passer tout le corpus en minuscule. La suppression de la casse permettra d'éviter que les noms propres et début de phrases soient orthographiés de manières différentes pour mieux capter les similitudes. Ces cas restent néanmoins rares. Nous pouvons citer l'exemple suivant :

> "Le gouvernement a décidé d'informer l'**Otan** de la volonté de la Suède de devenir membre de l'Alliance"
>
> "Nous n'allons pas fermer les yeux sur l'opération militaire turque en Syrie parce qu'Ankara a mis son veto à l'adhésion à l'**OTAN** de la Suède et de la Finlande"

Ensuite, nous avons décidé de supprimer les guillemets. Comme nous travaillons sur des citations, elles possèdent supposément toutes un guillemet de début et un guillemet de fin. Nous ne souhaitons pas que l'algorithme de clusterisation utilise ce critère pour déterminer des clusters. Garder les guillemets pourraient donc fausser les résultats. Nous avons également fait le choix de ne pas conserver les citations d'une longueur de moins de cinq tokens. Certaines phrases courtes peuvent être identiques sans pour autant traiter du même sujet. C'est le cas pour les séquences suivantes :

> "**J'aime** bien"
> "**J'aime**"

1. Lien vers le site de NLTK

Nous déciderons de ne pas supprimer les mots grammaticaux. En jetant un oeil aux citations, certaines variations peuvent se reposer uniquement sur la suppression ou l'ajout d'un seul mot, souvent grammatical. Pour cette raison, il est important de garder les stopwords. Comme nous l'avons vu précédemment, dans certaines recherches il est question de raciniser les phrases. C'est le cas de [Oliveira and Sperandio Nascimento, 2021]. Dans leurs travaux, la racinisation est utile car ils traitent des documents entiers et donc plus de contenu textuel. Dans notre corpus, il peut y avoir des séquences en paraphrase où seule la conjugaison d'un verbe, ou l'accord d'un mot peut varier. Nous voulons prendre en compte le plus d'information possible, raciniser certains mots pourrait donc faire perdre de l'information nécessaire à la clusterisation. Les citations les plus courtes pourraient se retrouver identiques après une racinisation alors qu'elles ne le seraient pas à l'origine.

Citation	**Journal**
À l'heure actuelle, aucune contamination n'a été relevée à la station et le niveau de radioactivité est normal	Le Figaro
À l'heure actuelle, aucune contamination n'a été relevée à la station et le niveau de radioactivité est normal	Le Parisien
l'ouzbékistan reconnaissait l'indépendance , la souveraineté et l'intégrité territoriale de l'ukraine	Le Figaro
un rôle dans l'atteinte ou la menace de l'intégrité territoriale , de la souveraineté et de l'indépendance de l'ukraine	Le Figaro
l'afrique est l'otage de ceux qui ont commencé la guerre contre notre etat	Le Monde
l'afrique est l'otage de ceux qui ont commencé la guerre contre notre etat	Le Figaro
l'afrique est l'otage de ceux qui ont commencé la guerre contre notre état	Le Monde
l'otage de ceux qui ont commencé la guerre contre notre etat	L'Equipe
envoyer un message sans équivoque et retentissant au monde entier : les etats-unis sont aux côtés de l'ukraine	Le Monde
notre délégation s'est rendue à kiev pour envoyer un message sans équivoque et retentissant au monde entier : les etats-unis sont aux côtés de l'ukraine	Aujourd'hui en France

TABLE 6.1 – Exemples de citations en doublons

Quatrième partie

Expérimentations

CHAPITRE 7

MÉTHODES

Sommaire

7.1 Introduction

Dans cette partie, nous traiterons des algorithmes que nous utiliserons pour effectuer une clusterisation sur nos données. Nous avons expliqué dans la partie précédente de quelle manière notre corpus a été constitué et normalisé. La donnée textuelle doit être encodée, c'est-à-dire transformée en objet calculable par l'algorithme de clusterisation. Nous présenterons donc les méthodes de vectorisation que nous devrons appliquer pour notre tâche. Nous présenterons les méthodes choisies ainsi que leurs implémentations respectives. Pour terminer, nous évaluerons ces clusterisations grâce aux différentes mesures que nous avons précédemment présentées. Nous analyserons également les résultats grâce à ces scores mais également en parcourant les citations clusterisées. Pour réaliser ces expérimentations nous utiliserons les implémentations de DBSCAN et OPTICS issues de la bibliothèque *scikit-learn* [1]

7.2 Implémentation

7.2.1 Vectorisation

Au vu de la littérature scientifique, nous avons pu faire ressurgir trois vectorisations principales utilisées pour la détection de paraphrases : Il s'agit de la vectorisation TF-IDF et les plongements de mots comme Word2Vec et Doc2Vec. Une fois le corpus normalisé, nous sommes en possession d'une liste comprenant toutes nos citations au format textuel. Cette liste doit être transformée en un vecteur numérique afin de pouvoir être interprétée par nos algorithmes de clustering.

La vectorisation en TF-IDF a été réalisée à l'aide de la méthode TfidfVectorizer [2] de scikit-learn. Elle permet de convertir du contenu textuel en une matrice TF-IDF.

1. Scikit-learn
2. Algorithme de vectorisation en TF-IDF implémenté par scikit-learn.

Cette méthode requiert un paramètre obligatoire : l'objet que nous souhaitons vectoriser. Cela peut être une liste de fichiers, un objet pouvant être ouvert en lecture sous forme de bytes ou une séquence d'éléments au format texte brut. Nous choisirons la troisième option. Le tableau 7.1 montre le résultat d'une vectorisation en TF-IDF pour la citation suivante :

> "concentrer le gros des efforts sur l'objectif principal : la libération du Donbass."
> "le contrôle total du Donbass et du sud de l'Ukraine"

Token	**Poids TF-IDF**
ukraine	0.176
de	0.176
sud	0.176
et	0.176
total	0.176
contrôle	0.176
donbass	0.353
du	0.53
libération	0.176
la	0.176
principal	0.176
objectif	0.176
sur	0.176
efforts	0.176
des	0.176
gros	0.176
le	0.353
concentrer	0.176

TABLE 7.1 – Exemple d'un vecteur TF-IDF sur deux citations du corpus

Pour ce qui est de Word2Vec et Doc2Vec, nous avons utilisé ces deux vectorisations grâce à la librairie Python Gensim [3]. Afin d'entraîner notre modèle, nous en avons choisi un en français, issu de FastText [4]. L'entraînement du modèle Word2Vec, en plus du contenu tokénisé, prend comme paramètre la taille du vecteur qui correspond au nombre de dimensions. Pour ce qui est de Doc2Vec, nous avons fait le choix de l'entraîner directement sur nos données afin de comparer ses résultats avec les embeddings pré-entraînés.

L'algorithme d'entraînement produit des vecteurs avec beaucoup de dimensions. La méthode T-SNE (T-Distributed Stochastic Neighbor Embedding), créée par [Maaten and Hinton, 2008], permet de réduire les dimensions d'une vectorisation. Dans des travaux similaires, comme par exemple [Jang et al., 2016] une réduction est faite en amont de DBSCAN pour créer un système de reconnaissance d'intention de dialogues. Le T-SNE est aussi utilisé par [Patidar et al., 2016] dans le but de détecter des activités dans des mails grâce à des méthodes comme DBSCAN.

3. Site web de Gensim
4. Lien vers les vecteurs pré-entraînés

Une réduction de la dimension des vecteurs semble donc nécessaire à l'application d'un algorithme de clusterisation. Pour cette tâche, nous utiliserons l'implémentation du T-SNE par scikit-learn [5].

7.2.2 Clustering

Après avoir établi les solutions existantes afin de clusteriser un corpus, nous pouvons déjà entrevoir quelles méthodes nous pouvons sélectionner. Celles qui seraient les plus à même de répondre efficacement à notre tâche sont les méthodes de clusterisations basées sur la densité, principalement car elles ne requièrent pas de données annotées et car elles se sont montrées efficaces sur des larges set de données. Les deux plus utilisées sont DBSCAN et OPTICS. Nous allons appliquer ces deux méthodes avec des vectorisations et normalisations différentes ainsi qu'avec une variation des hyperparamètres que nous allons détailler.

DBSCAN

Le premier algorithme, DBSCAN, se compose de deux paramètres principaux. Le premier, que nous appellerons *min_samples*, permet de spécifier le nombre d'occurrences minimum pour un seul cluster. Par exemple, s'il est paramétré à 3, alors tous les clusters à deux éléments seront pris en compte comme du bruit. Le second paramètre à sélectionner est la distance maximale que doivent avoir deux points pour être considérés comme voisins et donc membres d'un même groupe. La distance epsilon correspond à une valeur entre 0 et 1, où 1 correspond à deux éléments identiques.

OPTICS

Comme pour DBSCAN, OPTICS prend comme paramètre un nombre minimum de points pour avoir un cluster valide. Mais, contrairement à DBSCAN, pour utiliser OPTICS nous n'avons pas à spécifier de valeur pour epsilon car nous souhaitons que l'algorithme puisse découvrir des clusters parmi des densités différentes. Lorsque ce paramètre n'est pas spécifié, le temps de calcul est plus long que DBSCAN mais ne dépasse pas les cinq minutes. En plus de ces deux algorithmes, nous devrons également appliquer les différentes vectorisations sur notre corpus.

7.2.3 Optimisation des hyperparamètres

Faire varier les hyperparamètres peut être une tâche fastidieuse. En plus de cela, nous devons également prendre en compte les différentes vectorisations ainsi que les deux algorithmes. Cela donne entre 200 et 300 lancements différents. Nous avons choisi d'utiliser Optuna [6], qui est un framework servant à optimiser les hyperparamètres. Avec cet outil nous pouvons obtenir la meilleure combinaison pour nos expériences. Naïvement, nous avons lancé une session d'Optuna avec les paramètres suivants :

- Algorithme : DBSCAN ou OPTICS
- Vectorisation : TF-IDF, Word2Vec ou Doc2Vec
- Distance epsilon [0.1, 0.9]. Nous avons décidé de faire varier la distance, uniquement dans le cadre de DBSCAN, avec une incrémentation de 0.1.

5. Documentation des différentes techniques de réduction par scikit-learn
6. Lien vers le site web d'optuna

— min_samples : [2, 9]

Pour ce premier lancement, l'optimisation par Optuna s'arrête quand le score de silhouette n'est plus maximisé.

Étant donné la quantité importante de lancements à effectuer, nous avons pris quelques décisions pour réduire le temps d'exécution d'Optuna, tout en conservant les hyperparamètres pertinents. Tout d'abord, nous avons séparé ces expérimentations en six catégories différentes selon l'algorithme et la vectorisation choisie. Ensuite, nous avions en premier lieu lancé cet algorithme avec des min_samples de valeurs allant de 2 à 9. Nous nous sommes rendus compte qu'un cluster valide doit être au minimum constitué d'une paire identique. Mettre une valeur de 3 nous ferais perdre les clusters valides à deux éléments. Nous laisserons ce paramètre à 2 pour la suite des expériences. De plus, nous avons introduit plus tôt l'indice de Davies-Bouldin. Nous lancerons Optuna en essayant de maximiser le score de silhouette et à minimiser l'indice de Davies-Bouldin.

CHAPITRE 8

RÉSULTATS

Sommaire

8.1 Premières expériences

Comme nous l'avons expliqué précédemment, nous avons utilisé le score de silhouette et l'indice de Davies-Bouldin pour évaluer les différentes clusterisations. Pour rappel, nous avons choisi ces deux mesures car elles sont utilisées pour la validation de clusterisation par densité mais aussi car ce sont les seules qui ne nécessite pas d'avoir un corpus annoté. Elles reposent uniquement sur les résultats de l'algorithme non supervisé.

En annexe se trouvent les tableaux A.2-3-4-5-6 et 8 présentant les résultats de nos expériences lancées avec Optuna. Les figures A.1-2-3-4-5-6 sont les représentations visuelles des meilleurs modèles.

En plus de la silhouette, nous considérons que les meilleurs modèles sont ceux qui possèdent le plus de clusters avec le moins de points de bruits possibles. Dans ce premier lancement d'Optuna, nous avons décidé de garder les points de bruit pour le calcul des scores afin de voir si ces derniers ont un impact sur les scores.

Eps	Clusters	Bruit	Silhouette	Davies-Bouldin	Vectorisation
0.6	3	11	0.254	0.505	d2v
0.1	1207	18829	0.999	0.001	tfidf
0.1	620	20090	0.991	0.695	w2v

TABLE 8.1 – Résultats de l'algorithme DBSCAN avec des différentes vectorisations (Avec le bruit)

Vectorisation	Clusters	Bruit	Silhouette	Min samples
w2v	6460	2617	0.608	2
tfidf	6302	3335	0.624	2
d2v	3098	5548	0.172	3

TABLE 8.2 – Résultats de l'algorithme OPTICS avec des différentes vectorisations (Avec le bruit)

Pour DBSCAN (Table 8.1), le nombre de clusters varie de 1 à 1827. Les scores de silhouette sont tous dans le négatif, sauf quand il y a une vectorisation grâce à Doc2Vec. Dans ce cas-ci, la moyenne des scores se situe aux alentours de 0.15. Cependant, les meilleurs scores de silhouette de DBSCAN-Doc2Vec ne sont pas représentatifs car le nombre de clusters oscille entre 1 et 3. De plus, les points de bruits sont soit très nombreux (environ 97% du corpus) soit inexistants. Nous savons qu'il existe dans notre corpus des citations uniques et donc qu'il y a tout de même des points de bruit. Nous pouvons en conclure que DBSCAN-Doc2Vec, malgré le score de silhouette le plus élevé, ne parvient pas à donner une clusterisation efficace.

Ensuite, les résultats des scores de TF-IDF et Word2Vec sont plutôt semblables. Les scores de silhouette sont presque tous dans le négatif, quelque soit le nombre de clusters. Le meilleur score de silhouette de DBSCAN-Word2Vec est de 0.091 mais comme pour Doc2Vec, 97% du corpus est caractérisé en bruit. Par contre, pour ce qui est de DBSCAN-TFIDF, tous les scores de silhouette se retrouvent dans le négatif. De plus, faire varier la distance epsilon n'apporte pas de changement majeur dans les résultats.

Nous pouvons voir dans les résultats que l'algorithme OPTICS surpasse DBSCAN (Table 8.2). Le nombre de clusters varie de 591 à 6464. Les trois meilleurs scores par vectorisation pour OPTICS se trouvent dans le tableau 8.1. OPTICS-TFIDF possède le meilleur score de silhouette, mais il n'est pas le modèle ayant le plus de clusters et le moins de bruit. En revanche, la vectorisation en Doc2Vec a la même silhouette que dans le cas de DBSCAN. Malgré cela, il parvient à clusteriser plus de citations, avec moins de bruit.

De par cette première expérimentation, nous pouvons déduire que la variation de la distance epsilon n'apporte pas de changements drastiques dans les résultats. Ce phénomène est appuyé par le fait que OPTICS surpasse de loin DBSCAN. Nous voulons savoir si le bruit a un impact sur les scores, nous allons donc relancer les meilleurs modèles de OPTICS et de DBSCAN avec des paramètres différents que nous allons détailler.

8.2 Suite des expérimentations

Comme nous l'avons expliqué précédemment, nous avons décidé de prendre en compte au minimum deux membres par clusters, c'est pour cette raison que le paramètres min_samples sera toujours à 2 désormais. Ensuite, nous avons supprimé les points de bruit du calcul de silhouette pour voir s'ils peuvent impacter les résultats. Enfin, nous avons ajouté le calcul de l'indice de Davies-Bouldin pour compléter

l'évaluation. Les meilleurs modèles seront ceux où le score de silhouette sera maximisé et où l'indice Davies-Bouldin sera minimisé. Nous avons décidé pour DBSCAN de choisir un intervalle de distance epsilon entre 0.1 et 0.9, avec une incrémentation de 0.1.

DBSCAN					
Eps	**Clusters**	**Bruit**	**Silhouette**	**Indice Davies-Bouldin**	**Vectorisation**
0.6	3	11	0.254	0.505	Doc2Vec
0.1	1207	18829	0.999	0.001	TF-IDF
0.1	620	20090	0.991	0.695	Word2Vec

TABLE 8.3 – Résultats des meilleurs modèles de l'algorithme DBSCAN avec des différentes vectorisations (Sans le bruit)

Une fois de plus, nous pouvons voir pour DBSCAN que les scores de la vectorisation en Doc2Vec est bien en deça de ceux des autres vectorisation. De plus, la distance epsilon est différente puisqu'elle est à 0.6. En revanche, les scores d'évaluation de DBSCAN-TFIDF sont très élevés. Les scores de silhouette et de Davies-Bouldin atteignent respectivement leur maximum et minimum. Néanmoins, encore beaucoup d'éléments clusterisés comme bruit persistent, ce qui représente environ 90% du corpus total.

OPTICS				
Clusters	**Bruit**	**Silhouette**	**Indice Davies-Bouldin**	**Vectorisation**
6512	3209	0.631	0.418	TF-IDF
6680	2390	0.625	0.394	Doc2Vec
6967	1804	0.663	0.367	Word2Vec

TABLE 8.4 – Résultats de l'algorithme OPTICS avec des différentes vectorisations (Sans le bruit)

Dans le cas de OPTICS, les scores de silhouette restent similaires mis à part pour la vectorisation avec Doc2Vec où ce score augmente légèrement. Cela est dû au fait que le nombre de clusters à été multiplié par deux en changeant le nombre d'éléments minimums dans un cluster de 3 à 2. Les scores de OPTICS-TFIDF et OPTICS-Word2Vec sont presque semblables à ceux obtenus précédemment, car le bruit reste faible (14% du corpus)

Nous pouvons en conclure que le bruit a un impact sur les données clusterisées. Lorsqu'un modèle a beaucoup de bruit, si ce dernier n'est pas compté, alors les scores d'évaluation s'améliorent. Ensuite, nous avons vu que l'algorithme OPTICS est plus performant que DBSCAN si l'on se fit, en plus des scores, à la quantité de clusters et de bruit. Une des raisons serait que OPTICS peut former des clusters parmi différentes densités. DBSCAN se montre très dépendant de la distance epsilon, qui se répercute sur la qualité du clustering, ce qui l'empêche de classifier des données ayant de trop larges différences de densités.

8.3 Analyse sémantique des clusters

Après cette première approche statistique, il est maintenant temps d'analyser directement les citations qui ont été clusterisées. Pour cet analyse nous choisirons les meilleur modèle pour chaque vectorisation :

— OPTICS-Doc2Vec
— OPTICS-Word2Vec
— DBSCAN-TFIDF

OPTICS-Doc2Vec
Exemple n°1
- brutale, injustifiée et par ailleurs cynique
- je n'exclus rien ni aucune initiative politique.
- injustifiable, inutile, inapplicable et nuisible
Exemple n°2
- le nucléaire ne doit pas être un objet de la guerre
- une preuve d'intelligence politique
- je me retirerais de la ville
- l'horreur de la guerre mondiale
- le nucléaire ne doit pas être un objet de la guerre
Exemple n°3
- nous, on nous demande de négocier avec nos agresseurs, alors que quand la russie a agressé l'ukraine,
- ce sont toutes les nations qui ont demandé que cette agression s'arrête
- nous allons donner à l'ukraine les armes pour se battre et se défendre tout au long des jours difficiles qui sont devant elle
Exemple n°4
- la russie n'a pas réellement réussi à trouver d'autres acheteurs pour compenser la baisse de la demande de l'ue, bien que l'interdiction ait été connue depuis des mois
- de toute évidence, quelqu'un en europe fait un mauvais calcul : il est essentiel de repenser la stratégie pour sauver les emplois et les entreprises en italie
- de toute évidence, quelqu'un en europe fait un mauvais calcul : il est essentiel de repenser la stratégie pour - sauver les emplois et les entreprises en italie
Exemple n°5
- ça accentue un sentiment d'impunité
- la patrie, mes amis, ce n'est lécher le cul du président, et encore moins l'embrasser en permanence
- l'intéressé conteste :
- l'utilisation d'armes nucléaires n'est pas une peur sans fondement
- réseau d'incubateurs d'entrepreneurs d'entreprises agricoles innovantes
- l'armée russe dit

TABLE 8.5 – Quelques clusters issus du modèle OPTICS-Doc2Vec

Les résultats de la clusterisation de chaque modèle ont été enregistrés dans un tableur comportant la citation et le cluster auquel elle appartient. Pour les trois meilleurs modèles, nous avons choisi des exemples que nous pensons être représentatifs de chacune des clusterisations.

8.3.1 OPTICS-Doc2Vec

La table n°8.5 fait état de quelques exemples issues de la clusterisation OPTICS-Doc2Vec. Malgré les scores semblables aux autres modèles avec OPTICS, la clusterisation n'est pas la même. Dans tous les exemples, nous allons avoir une première paire de citations identiques et des citations en trop. Dans certains clusters, nous discernons un schéma récurrent. Ils peuvent être basés sur un token en particulier : "injustifié" et "injustifiable" dans l'exemple n°1 ou un caractère suivi d'une apostrophe dans l'exemple n°5. Certains mots identiques peuvent se retrouver dans plusieurs citations, mais le contexte est différent. De plus, le modèle se base fréquemment sur des entités nommées comme dans l'exemple n°4, où plusieurs noms de pays sont répétés. Nous en déduisons que la vectorisation avec Doc2Vec ne permet pas de détecter correctement de la paraphrase. La vectorisation semble donner davantage de poids à certains tokens. Notre corpus traitant d'une même thématique, certains mots reviennent très souvent. Il s'agit peut être d'une des raisons qui rend complexe la clusterisation de ce modèle.

8.3.2 OPTICS-Word2Vec

Ce modèle possède des résultats de clusterisation similaires au modèle vectorisé grâce à Doc2Vec. Davantage de poids est donné à certains tokens tel que "(...)" dans l'exemple n°1 ou les guillemets et apostrophes dans l'exemple n°3. L'algorithme se base souvent sur la ponctuation pour former des groupes, ce qui repose la question de la normalisation. Nous aurions pu retirer ce genre de ponctuation afin que ces deux modèles réalisent une clusterisation uniquement sur des mots. De plus, nous pouvons nous demander si les passages entre parenthèses font partie intégrante d'une citation ou non.

8.3.3 DBSCAN-TFIDF

En revanche, les résultats du modèle DBSCAN-TFIDF sont très homogènes. Les citations ont le plus souvent été classées en paires identiques ou avec très peu de variations. Dans l'exemple n°3, il y a un déterminant supplémentaire. Dans l'exemple n°1, des crochets ont été rajoutés. Ce modèle présente une très bonne clusterisation mais les paires varient seulement d'un ou deux tokens. Il a de très bons scores d'évaluation mais beaucoup de bruit. Nous pouvons alors nous demander si des citations qui devraient être dans ces clusters n'ont pas été détectées car elles présentent trop de variations.

Les modèles OPTICS-Doc2Vec et OPTICS-Word2Vec semblent être plutôt efficaces pour classer les citations par thématique. Le corpus traitant du même thème peut aussi jouer sur la détection. Les calculs de ces embeddings se basent sur une sémantique trop générale ce qui les rendraient trop tolérants à la variation. Au contraire, le modèle DBSCAN-TFIDF est efficace au premier abord et plus proche du contenu

OPTICS-Word2Vec
Exemple n°1
- la ligne de 750 kilovolts (kv) réparée alimente la plus grande centrale nucléaire d'europe (...) avec l'électricité nécessaire pour assurer le refroidissement des réacteurs et d'autres fonctions de sécurité
- la ligne de 750 kilovolts réparée alimente la plus grande centrale nucléaire d'europe (. . .) avec l'électricité nécessaire pour assurer le refroidissement des réacteurs et d'autres fonctions de sécurité
- pour se représenter l'horreur de l'assaut (. . .) , il faut en avoir été le témoin , raconte le colonel de l'armée russe lev engelhardt .
Exemple n°2
- densifier sa présence militaire sur la frontière russo-finlandaise
- a informé l'équipage du croiseur que les officiers , les aspirants et les marins continueraient de servir dans la marine
- a informé l'équipage du croiseur que les officiers , les aspirants et les marins continueraient de servir dans la marine
Exemple n°3
- "nous n'avions pas le choix" : "filtrage" et le crime de guerre de transfert forcé de civils ukrainiens vers la russie
- on a créé une histoire autour de nos chaussures, c'est cela qui fait la différence
- l'"esprit de la guerre d'hiver", c'est l'idée que, même en position d'infériorité, nous nous battrons et que chacun fera des sacrifices
- c'est l'idée que, même en position d'infériorité et alors que l'épreuve s'annonce difficile, nous nous battrons et que chacun fera des sacrifices
Exemple n°4
- sécurité et la stabilité du cyberespace
- est un objectif réaliste et réalisable

TABLE 8.6 – Extrait de la clusterisation de OPTICS-Word2Vec

textuel mais les paires de citations trouvées sont le plus souvent identiques. Ces résultats sont surprenants car nous nous attendions à ce que l'utilisation d'embeddings pré-entraînés surpasse la vectorisation en TF-IDF qui est moins complexe, comme nous l'avons vu dans différents travaux de recherche.

8.3.4 Analyse d'une citation

Pour appuyer notre propos, nous avons décidé de choisir parmi le corpus, une citation reprise dans plusieurs journaux et de montrer comment elle a été clusterisé par les différents modèles. Ce cluster annoté manuellement est composé de cinq citations. Les trois premières et deux dernières sont identiques et entre ces deux sous-groupes, les propositions ont été inversées, et le verbe "faire" est remplacé par "aller".

Cluster d'origine

- j'étais prêt à le faire, mais apparemment, et je dois en prendre acte, ce n'était pas souhaité à kiev

DBSCAN-TFIDF

Exemple n°1
ne pas se considérer comme un impérialiste
ne pas [se] considérer comme un impérialiste

Exemple n°2
lors de leur audition en tant que témoins, ils ont expliqué avoir découvert le drapeau après avoir escaladé le sommet et l'avoir filmé sur un téléphone
lors de leur audition en tant que témoins, ils ont expliqué avoir découvert le drapeau après avoir escaladé le sommet et l'avoir filmé sur un téléphone

Exemple n°3
plus grande catastrophe géopolitique du xxe siècle
la plus grande catastrophe géopolitique du xxe siècle

- j'étais prêt à le faire mais, apparemment, et je dois en prendre acte, ce n'était pas souhaité à kiev
- j'étais prêt à le faire mais apparemment, et je dois en prendre acte, ce n'était pas souhaité à kiev
- j'étais prêt à y aller, mais apparemment ce n'était pas souhaité à kiev et j'en prends acte
- j'étais prêt à y aller mais apparemment ce n'était pas souhaité à kiev et j'en prends acte

DBSCAN-TFIDF

- j'étais prêt à le faire mais, apparemment, et je dois en prendre acte, ce n'était pas souhaité à kiev
- j'étais prêt à le faire, mais apparemment, et je dois en prendre acte, ce n'était pas souhaité à kiev
- j'étais prêt à le faire mais apparemment, et je dois en prendre acte, ce n'était pas souhaité à kiev
- j'étais prêt à y aller, mais apparemment ce n'était pas souhaité à kiev et j'en prends acte
- j'étais prêt à y aller mais apparemment ce n'était pas souhaité à kiev et j'en prends acte

OPTICS-Word2Vec

- j'étais prêt à le faire mais, apparemment, et je dois en prendre acte, ce n'était pas souhaité à kiev
- j'étais prêt à le faire, mais apparemment, et je dois en prendre acte, ce n'était pas souhaité à kiev
- je n'ai jamais arrêté de me battre depuis ce temps-là .
- si j'étais plus jeune, je serai prêt à me battre pour la transcarpatie, mais pas pour la partie est de l'ukraine, ce n'est pas chez moi
- j'étais prêt à le faire mais apparemment, et je dois en prendre acte, ce n'était pas souhaité à kiev
- j'étais prêt à y aller, mais apparemment ce n'était pas souhaité à kiev et j'en prends acte

- j'étais prêt à y aller mais apparemment ce n'était pas souhaité à kiev et j'en prends acte

OPTICS-Doc2Vec

- c'est un honneur d'être ici et de pouvoir transmettre l'héritage de mantas
- j'étais prêt à y aller, mais apparemment ce n'était pas souhaité à kiev et j'en prends acte
- j'étais prêt à y aller mais apparemment ce n'était pas souhaité à kiev et j'en prends acte
- j'étais prêt à le faire mais, apparemment, et je dois en prendre acte, ce n'était pas souhaité à kiev
- aucun pays ne mérite d'être envahi
- mais surtout, le message, c'est l'importance de l'amour de sa patrie et la beauté d'être prêt à donner sa vie pour elle
- j'étais prêt à le faire, mais apparemment, et je dois en prendre acte, ce n'était pas souhaité à kiev
- soutenir le régime néonazi de kiev
- ce n'était pas souhaité à kiev
- soutenir le régime néonazi de kiev
- cette guerre ne s'achèvera pas facilement ou rapidement
- montrer, et particulièrement aux états-unis, qu'il existe une véritable 'amitié ' sino-russe – terme consacré depuis plusieurs mois – et que leur coopération militaire est tout à fait opérante
- j'étais prêt à le faire mais apparemment, et je dois en prendre acte, ce n'était pas souhaité à kiev

Le modèle avec la vectorisation en TF-IDF parvient à réunir ces cinq citations dans un même groupe. Les deux autres modèles détectent aussi ces citations mais ne les réunissent pas toutes dans le même cluster. En plus de cela, certaines citations considérées comme du bruit se sont glissées à l'intérieur. Pour OPTICS-Doc2Vec, même les citations identiques n'ont pas été clusterisées dans un même groupe et beaucoup de bruit persiste.

8.4 Conclusion

A la suite de ces expérimentations et analyses, nous rencontrons plusieurs interrogations. Nous avons montré que nos modèles Word2Vec et Doc2Vec se focalisent essentiellement sur les ponctuations, il serait alors judicieux de les supprimer pour constater une différence dans les résultats. En analysant les citations dans leurs clusters respectifs, le modèle DBSCAN-TFIDF semble être le plus performant pour notre tâche. L'utilisation des scores de silhouette et l'indice de Davies-Bouldin seuls ne sont pas suffisant pour valider une clusterisation. Une évaluation humaine est fortement nécessaire quand les données ne sont pas annotées au préalable pour savoir si la clusterisation est correcte. De plus, nous avons montré que la vectorisation joue un rôle essentiel pour la détection de doublons.

Cinquième partie

Discussions

CHAPITRE 9

COMPLÉMENTS SUR LES MESURES D'ÉVALUATION

Sommaire

9.1 Annotation manuelle d'un sous-corpus

En plus des mesures d'évaluation non supervisées, nous souhaitons mesurer la précision, le rappel et la f-mesure pour avoir une idée supplémentaire des performances de nos algorithmes. Pour cela, en nous aidant des résultats des meilleurs modèles, nous avons annoté une partie du corpus en clusters, avec des éléments considérés comme du bruit. Nous avons un total de 720 citations regroupées 312 clusters et 46 citations annotées en tant que bruit. Un extrait du corpus est disponible en annexe (Tableau A.7).

9.2 Évaluation

Pour évaluer ces modèles, nous avons choisi les mesures les plus utilisées en traitement automatique des langues. Il s'agit de la précision, du rappel et de la F-Mesure. Nous souhaitons savoir si les mesures d'évaluation que nous avons utilisé plus tôt et celles-ci sont corrélées. Nous savons désormais que le bruit peut avoir un impact sur les scores, c'est pourquoi nous appliquerons ces mesures sur les trois meilleurs modèles de OPTICS et DBSCAN. Ce calcul a été réalisé grâce à la librairie scikit-learn [1].

La métrique utilisée est nommée *pair confusion matrix*. Il s'agit d'une matrice de 2 sur 2 entre deux clusters créée en considérant toutes les paires possibles au sein de cette matrice. Grâce aux labels de la référence et de la prédiction, ces paires sont classées en tant que vrais positifs, faux positifs, vrais négatifs et faux négatifs.

1. Matrice de confusion par paires

	Précision	**Rappel**	**F-Mesure**
Word2Vec	0.348	0.255	0.294
Doc2Vec	0.051	0.164	0.078
TF-IDF	0.913	0.315	0.468

TABLE 9.1 – Résultats de l'évaluation des trois modèles basés sur OPTICS

	Précision	**Rappel**	**F-Mesure**
Word2Vec	0.03	0.906	0.058
Doc2Vec	0.005	0.964	0.011
TF-IDF	0.019	0.901	0.037

TABLE 9.2 – Résultats de l'évaluation des trois modèles basés sur DBSCAN

Nous constatons que le modèle OPTICS-TFIDF obtient les meilleurs scores. Pour rappel, la précision mesure la quantité d'éléments correctement classifiés par un algorithme, rapporté au nombre d'éléments total trouvé par l'algorithme.

$$\text{Précision} = \frac{\text{vrais positifs}}{\text{vrais positifs} + \text{faux positifs}} \tag{9.1}$$

Le rappel, comme la précision mesure la quantité d'éléments correctement classifiés par l'algorithme mais rapporté au nombre d'éléments présents dans les données de référence.

$$\text{Rappel} = \frac{\text{vrais positifs}}{\text{vrais positifs} + \text{faux négatifs}} \tag{9.2}$$

La F-mesure correspond à la moyenne pondérée de la précision et du rappel.

$$\text{F-mesure} = \frac{(1+\beta^2) \cdot \text{Précision} \cdot \text{Rappel}}{\beta^2 \cdot \text{Précision} + \text{Rappel}} \tag{9.3}$$

Une précision élevée et un rappel faible signifie que les paires trouvées sont peu nombreuses, mais ont été correctement classées. Nous utilisons une matrice qui combine toutes les citations sous forme de paires, il n'est donc pas étonnant que le rappel soit faible. De plus, nous pouvons voir dans les matrices de confusion (disponibles en annexe) que les scores sont noyés par des paires négatives.

	Silhouette	**Davies-Bouldin**
DBSCAN		
Word2Vec	0.991	0.009
Doc2Vec	0.995	0.005
TF-IDF	0.996	0.007
OPTICS		
Word2Vec	0.745	0.311
Doc2Vec	0.517	0.49
TF-IDF	0.844	0.212

TABLE 9.3 – Scores de silhouette et de Davies-Bouldin du sous-corpus

Si nous comparons ces scores avec ceux non supervisés, nous pouvons constater que les meilleurs modèles ne sont pas les mêmes. Dans le cas d'OPTICS, Word2Vec est la configuration avec les meilleurs scores de silhouette et de Davies-Bouldin. Ici, le modèle qui obtient la meilleure F-mesure est OPTICS-TFIDF. Pour DBSCAN en revanche, nous retrouvons un contraste entre un très bon score de silhouette mais une F-Mesure très basse. De plus, les scores de F-mesure obtenus par OPTICS sont meilleurs que ceux de DBSCAN. Bien que les scores de silhouette et de Davies-Bouldin ont retourné de bons scores dans l'ensemble, nous pouvons voir qu'ils ne sont pas en adéquation avec la f-mesure.

CHAPITRE 10

PERSPECTIVES D'AMÉLIORATIONS

Sommaire

10.1 Normalisation des citations

Suite à nos expériences, nous pouvons envisager des améliorations de nos modèles. Tout d'abord pour ce qui est de la normalisation, nous devrions faire un tri plus poussé des tokens et caractères à conserver afin que les embeddings pré-entraînés donnent moins de poids aux ponctuations. Nous pensons spécialement aux passages entre parenthèses dont la présence dans une citation peut être questionnable. Ensuite, nous pourrions tester ces modèles sur un corpus ne résultant pas d'une même thématique. Cependant, nous pensons que nous trouverons moins de citations en paraphrase et beaucoup plus de bruit dans ce cas-ci car les thématiques peuvent être très diverses.

10.2 Améliorations de l'encodage des données

Une autre solution serait de trouver d'autres manières d'encoder nos citations. GloVe est un algorithme de représentation de vecteur sémantique, créé de manière non supervisée grâce à des réseaux neuronaux. [Tu et al., 2018] réalisent une segmentation de contenu textuel scolaire. Ils utilisent GloVe pour représenter les textes en une représentation unidimensionnelle. Grâce au Latent Dirichlet Allocation (LDA), technique similaire au T-SNE, les vecteurs de GloVe ont ensuite été concaténés afin d'effectuer une clusterisation des différents éléments par similarité et ainsi de les représenter par thématique. [Pennington et al., 2014] précisent que cet embedding pré-entraîné est plus performant que Word2Vec :

> "Nous notons que les performances de Word2Vec diminuent si le nombre d'échantillons négatifs augmente au delà de 10. Cela est probablement dû au fait que la méthode d'échantillonnage négatif ne s'approche pas bien

de la distribution de probabilité cible. Pour le même corpus, le même vocabulaire, la même taille de fenêtre et le même temps d'entraînement, GloVe surpasse systématiquement Word2Vec. Il obtient de meilleurs résultats plus rapidement, et obtient également les meilleurs résultats indépendamment de la vitesse [1]."

10.3 Améliorations existantes des algorithmes de clusterisation

10.3.1 DP-DBSCAN

[Almamory and Kamil, 2019] parviennent à créer une version améliorée de DBSCAN qui pourrait classer des éléments en cluster mais également selon leurs densités différentes. Un des inconvénients de DBSCAN est qu'il ne regroupe par efficacement des donnés ayant des densités variables. L'algorithme qu'ils proposent, nommé DP-DBSCAN, permet de séparer une première fois le corpus en deux selon leurs variations de densités.

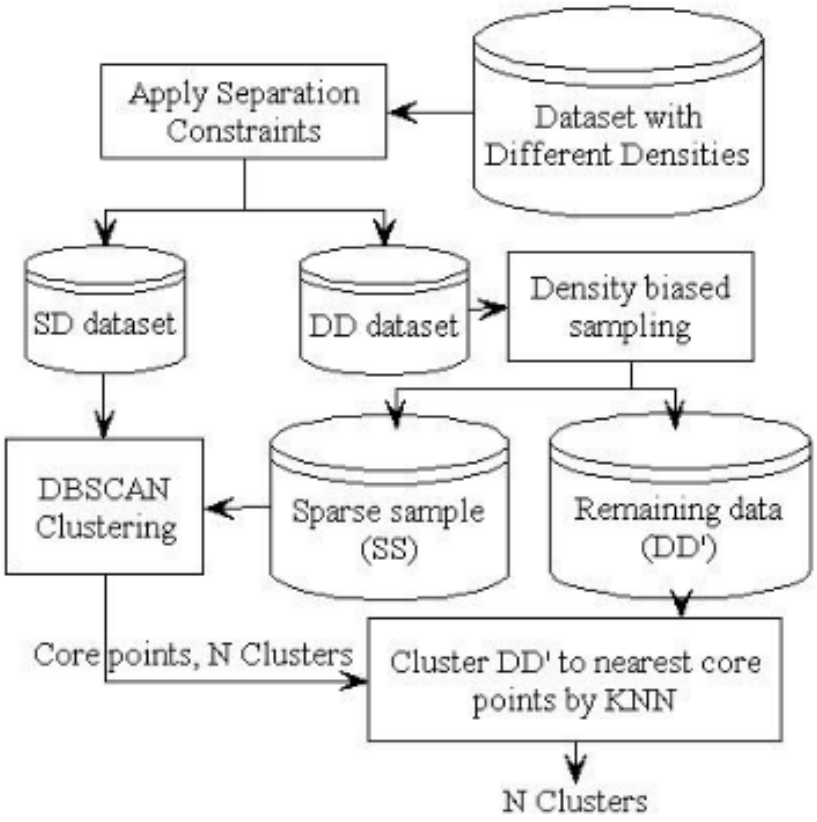

FIGURE 10.1 – Diagramme du système proposé par [Almamory and Kamil, 2019]

10.3.2 OPTICS k-XI

[Charlon, 2019] mettent au point une variante de l'algorithme OPTICS, nommé k-Xi, qui est un mélange entre OPTICS et K-Means :

1. [Pennington et al., 2014], page 1541

> "Pour chaque plus grande différence successive, OPTICS k-Xi tentera de former un cluster de tous les points adjacents avec une distance plus petite, en distinguant les zones de densités différentes de la même manière que OPTICS Xi. Si le cluster nouvellement formé contient moins d'observations que le paramètre pts [min_samples], ou s'il réduit la taille d'un cluster précédemment formé en dessous du paramètre pts, le nouveau cluster est écarté et la plus grande différence suivante est considérée. [2]."

10.4 Ajout de métadonnées relatives aux citations

10.4.1 Locuteur et coréférence

Une autre perspective d'amélioration serait d'ajouter des métadonnées relatives aux citations qui seraient utilisées pendant la clusterisation. Le modèle d'extraction de citations KiDiKoi détecte également le locuteur. Cependant, afin de pouvoir ajouter le locuteur, nous devrions améliorer ce modèle ainsi que celui qui lie la citation à sa source. Certaines citations sont détectées sans source, d'autres sont liées à de multiples sources. Pour cette raison, il serait envisageable de réaliser une résolution de coréférence lorsqu'il y a plusieurs sources, afin d'écarter les pronoms personnels et ne garder que les entités nommées. Techniquement, cela pourrait se traduire en liant un pronom personnel à son locuteur par le biais d'une base de connaissance. Grâce à cela, nous pourrions vérifier si plusieurs citations détectées comme paraphrase ont le même locuteur ou non, car nous pensons qu'il est possible qu'une citation puisse être rapportée par plus d'une personne.

Ensuite, nous avons pu rajouter le nom du fichier qui correspond à l'article et sa date de parution. Grâce à cela nous pourrions lier les citations à leurs journaux respectifs pour découvrir si certains ont plus souvent recours au discours rapporté, mais aussi déterminer l'ordre chronologique d'une citation grâce à la date de parution d'un article.

2. [Charlon, 2019], page 5

CHAPITRE 11

CAS D'APPLICATION POSSIBLES

Sommaire

11.0.1 Déduplication de corpus

[Lee et al., 2022] utilisent des méthodes nommées respectivement ExactSubstr et NearDup afin de dédupliquer des corpus de grandes tailles. La méthode ExactSubstr permet d'extraire des éléments identiques grâce à un tableau de suffixes des mots. NearDup quant à elle, consiste à estimer la similarité des n-grams entre toutes les paires d'exemples grâce au calcul de l'indice de Jaccard. Selon eux, 1% des données peuvent représenter de la paraphrase entre l'entraînement et le test. Ils montrent que cette déduplication pourrait permettre d'améliorer l'entraînement des modèles en termes de temps et de coût en supprimant les données présentent en double. Les méthodes que nous avons explorées pourraient elles aussi être appliquées à la déduplication de corpus.

11.0.2 Topic modeling et annotation semi-automatique

Les embeddings pré-entraînés comme Word2Vec et Doc2vec ne parviennent pas à détecter correctement de la paraphrase que ce soit par DBSCAN ou OPTICS. Cependant, ils semblent être performants pour classer des séquences par thématiques, ils seraient donc adaptés pour des tâches comme la détection de thématique ou le topic modeling [Murshed et al., 2022]. L'annotation manuelle de corpus est une tâche très lourde en termes de temps et de coût. Dans le cadre d'une détection de doublons, confronter une citation à toutes les autres est très fastidieux. La clusterisation avec une vectorisation en TF-IDF renvoie de bons résultats, le plus souvent avec des paires identiques ou présentant très peu de variations, c'est pourquoi il pourrait être pertinent de l'utiliser en tant qu'outil d'aide à l'annotation. Cette aide permettrait d'accélérer ce travail et de créér un corpus entièrement étiqueté.

Conclusion générale

Dans ce mémoire de recherche, nous avons appliqué des algorithmes clusterisation par densité pour de la détection de citations en paraphrase. Nous avons fait état des différents travaux concernant ce sujet de recherche, ainsi que les différentes architectures à notre disposition. Nous avons fait le choix d'utiliser principalement la librairie python de scikit-learn par facilité d'implémentation.

Au cours des différentes expériences, nous avons constaté que chaque bloc de l'architecture a un impact sur l'autre. La normalisation joue un rôle important puisqu'elle est le point d'entrée de notre modèle. Il a fallu trouver le juste équilibre entre garder la citation au format le plus brut possible qui soit aisément détectable tout en l'uniformisant au maximum. Le choix de la vectorisation était également crucial dans notre analyse, et résulte en des clusterisations très différentes en termes de qualité et de quantité des clusters. Le TF-IDF a tendance à détecter des copies exactes de citations, alors que le Doc2Vec rend les clusters trop denses et donc moins appréhendables par OPTICS et DBSCAN. L'algorithme de clusterisation a aussi été un critère décisif. Parmi K-Means, DBSCAN et OPTICS, nous avons préféré opter pour des méthodes basées sur la densité qui se montrent plus efficaces sur des larges set de données et lorsque le nombre de cluster n'est pas connu à l'avance [Mary, 2012].

Ce travail de recherche répond également à un besoin industriel de l'entreprise Aday qui serait de pouvoir suivre les citations dans la presse. Pour cela, il est important de pouvoir détecter la déformation d'une citation à travers le temps ou à travers un autre journal.

BIBLIOGRAPHIE

[Agarwal et al., 2018] Agarwal, B., Ramampiaro, H., Langseth, H., and Ruocco, M. (2018). A deep network model for paraphrase detection in short text messages. *Information Processing & Management*, 54(6) :922–937. – Cité page 15.

[Alian and Awajan, 2020] Alian, M. and Awajan, A. (2020). Factors Affecting Sentence Similarity and Paraphrasing Identification. *International Journal of Speech Technology*. – Cité page 17.

[Almamory and Kamil, 2019] Almamory, S. and Kamil, I. (2019). A new density based sampling to enhance dbscan clustering algorithm. *Malaysian Journal of Computer Science*, 32 :315–327. – Cité pages 6 et 58.

[Ankerst et al., 1999] Ankerst, M., Breunig, M. M., Kriegel, H.-P., and Sander, J. (1999). Optics : ordering points to identify the clustering structure. *ACM SIGMOD Record*, 28(2) :49–60. – Cité pages 6, 18 et 19.

[Bansal et al., 2022] Bansal, I., Gupta, K., Aakriti, Sehgal, B., and Sharma, V. (2022). A social network approach for automated generation of youtube playlists. In *2022 IEEE International Conference on Data Science and Information System (ICDSIS)*, page 1–6. – Cité page 19.

[Bholowalia and Kumar, 2014] Bholowalia, P. and Kumar, A. (2014). Ebk-means : A clustering technique based on elbow method and k-means in wsn. *International Journal of Computer Applications*, 105(9) :17–24. – Cité page 17.

[Charlon, 2019] Charlon, T. (2019). opticskxi : Optics k-xi density-based clustering. – Cité pages 58 et 59.

[Davies and Bouldin, 1979] Davies, D. and Bouldin, D. (1979). A cluster separation measure. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-1 :224–227. – Cité page 26.

[Ester et al., 1996] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231, Portland, Oregon. AAAI Press. – Cité page 18.

[Farnaghi et al., 2020] Farnaghi, M., Ghaemi, Z., and Mansourian, A. (2020). Dynamic spatio-temporal tweet mining for event detection : A case study of hurricane florence. *International Journal of Disaster Risk Science*, 11(3) :378–393. – Cité page 20.

[Ferrero and Simac-Lejeune, 2015] Ferrero, J. and Simac-Lejeune, A. (2015). Détection automatique de reformulations - Correspondance de concepts appliquée à la détection du plagiat - Editions RNTI. *EGC 2015*, (28) :287–298. – Cité page 11.

[Jang et al., 2016] Jang, J., Lee, Y., Lee, S., Shin, D., Kim, D., and Rim, H. (2016). A novel density-based clustering method using word embedding features for dialogue intention recognition. *Cluster Computing*, 19(4) :2315–2326. – Cité pages 18 et 40.

[Le and Mikolov, 2014] Le, Q. V. and Mikolov, T. (2014). Distributed representations of sentences and documents. (arXiv :1405.4053). arXiv :1405.4053 [cs]. – Cité pages 6 et 23.

[Lee et al., 2022] Lee, K., Ippolito, D., Nystrom, A., Zhang, C., Eck, D., Callison-Burch, C., and Carlini, N. (2022). Deduplicating Training Data Makes Language Models Better. arXiv :2107.06499 [cs]. – Cité page 61.

[Li, 2014] Li, H. (2014). Learning to rank for information retrieval and natural language processing, second edition. *Synthesis Lectures on Human Language Technologies*, 7(3) :1–121. – Cité page 15.

[Liao and Cheng, 2016] Liao, X. and Cheng, G. (2016). Analysing the semantic change based on word embedding. In Lin, C.-Y., Xue, N., Zhao, D., Huang, X., and Feng, Y., editors, *Natural Language Understanding and Intelligent Applications*, Lecture Notes in Computer Science, page 213–223, Cham. Springer International Publishing. – Cité page 18.

[Maaten and Hinton, 2008] Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86) :2579–2605. – Cité page 40.

[Magalhães et al., 2020] Magalhães, D., Pozo, A., and Santana, R. (2020). *An empirical comparison of distance/similarity measures for Natural Language Processing*. – Cité pages 22 et 23.

[Mahmoud and Zrigui, 2021] Mahmoud, A. and Zrigui, M. (2021). BLSTM-API : Bi-LSTM Recurrent Neural Network-Based Approach for Arabic Paraphrase Identification. *Arabian Journal for Science and Engineering*, 46. – Cité page 16.

[Mary, 2012] Mary, S. (2012). A density based dynamic data clustering algorithm based on incremental dataset. *Journal of Computer Science*, 8 :656–664. – Cité pages 26 et 63.

[Mikolov et al., 2013] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. (arXiv :1310.4546). arXiv :1310.4546 [cs, stat]. – Cité page 22.

[Mohamed and Oussalah, 2020] Mohamed, M. and Oussalah, M. (2020). A hybrid approach for paraphrase identification based on knowledge-enriched semantic heuristics. *Language Resources and Evaluation*, 54(2) :457–485. – Cité page 23.

[Mohammed et al., 2021] Mohammed, S., Jacksi, K., and Zeebaree, S. (2021). A state-of-the-art survey on semantic similarity for document clustering using GloVe and density-based algorithms. *Indonesian Journal of Electrical Engineering and Computer Science*, 22 :552–562. – Cité pages 18, 21 et 23.

[Murshed et al., 2022] Murshed, B. A. H., Mallappa, S., Abawajy, J., Saif, M. A. N., Al-ariki, H. D. E., and Abdulwahab, H. M. (2022). Short text topic modelling approaches in the context of big data : taxonomy, survey, and analysis. *Artificial Intelligence Review*. – Cité page 61.

[Oliveira and Sperandio Nascimento, 2021] Oliveira, R. and Sperandio Nascimento, E. G. (2021). Clustering by Similarity of Brazilian Legal Documents Using Natural Language Processing Approaches. – Cité pages 18 et 36.

[Patidar et al., 2016] Patidar, M., Rohatgi, S., Chaudhary, A., Singh, M. P., Agarwal, P., and Shroff, G. (2016). Activity detection from email meta-data clustering. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, page 568–575. – Cité page 40.

[Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. (2014). Glove : Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, page 1532–1543, Doha, Qatar. Association for Computational Linguistics. – Cité pages 57 et 58.

[Reztaputra and Khodra, 2017] Reztaputra, R. and Khodra, M. L. (2017). Sentence structure-based summarization for Indonesian news articles. In *2017 International Conference on Advanced Informatics, Concepts, Theory, and Applications (ICAICTA)*, pages 1–6. – Cité page 15.

[Rousseeuw, 1987] Rousseeuw, P. J. (1987). Silhouettes : A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20 :53–65. – Cité page 25.

[Sandhya and Chitrakala, 2011] Sandhya, S. and Chitrakala, S. (2011). Plagiarism Detection of Paraphrases in Text Documents with Document Retrieval. In Wyld, D. C., Wozniak, M., Chaki, N., Meghanathan, N., and Nagamalai, D., editors, *Advances in Computing and Information Technology*, Communications in Computer and Information Science, pages 330–338, Berlin, Heidelberg. Springer. – Cité page 15.

[Schmitt and Spinosa, 2018] Schmitt, M. F. L. and Spinosa, E. J. (2018). Outlier detection on semantic space for sentiment analysis with convolutional neural networks. In *2018 International Joint Conference on Neural Networks (IJCNN)*, page 1–8. – Cité page 18.

[Schubert and Gertz, 2018] Schubert, E. and Gertz, M. (2018). Improving the cluster structure extracted from optics plots. – Cité page 19.

[Tu et al., 2018] Tu, Y., Xiong, Y., Chen, W., and Brinton, C. (2018). A domain-independent text segmentation method for educational course content. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, page 320–327. – Cité page 57.

[Vijayan and P, 2021] Vijayan, V. and P, P. K. (2021). Analysis of various clustering algorithms to enhance bag-of-visual-words for drowsiness prediction. In *2021 International Conference on Artificial Intelligence and Machine Vision (AIMV)*, page 1–6. – Cité page 19.

[Wang et al., 2019] Wang, W. Y., Singh, S., and Li, J. (2019). Deep Adversarial Learning for NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Tutorials*, pages 1–5, Minneapolis, Minnesota. Association for Computational Linguistics. – Cité page 16.

[Weißer et al., 2020] Weißer, T., Saßmannshausen, T., Ohrndorf, D., Burggräf, P., and Wagner, J. (2020). A clustering approach for topic filtering within systematic literature reviews. *MethodsX*, 7 :100831. – Cité pages 17, 22, 23 et 25.

[Yuan and Yang, 2019] Yuan, C. and Yang, H. (2019). Research on k-value selection method of k-means clustering algorithm. *J*, 2(22) :226–235. – Cité page 17.

ANNEXE

E Nombre d'éléments minimal pour former un cluster
D Valeur de la distance maximale entre deux points
C Nombre de clusters identifiés
B Points de bruit identifiés
S Score de silhouette
V Vectorisation choisie
A Algorithme choisi

TABLE A.1 – Légende des tableaux de résultats A.2-6

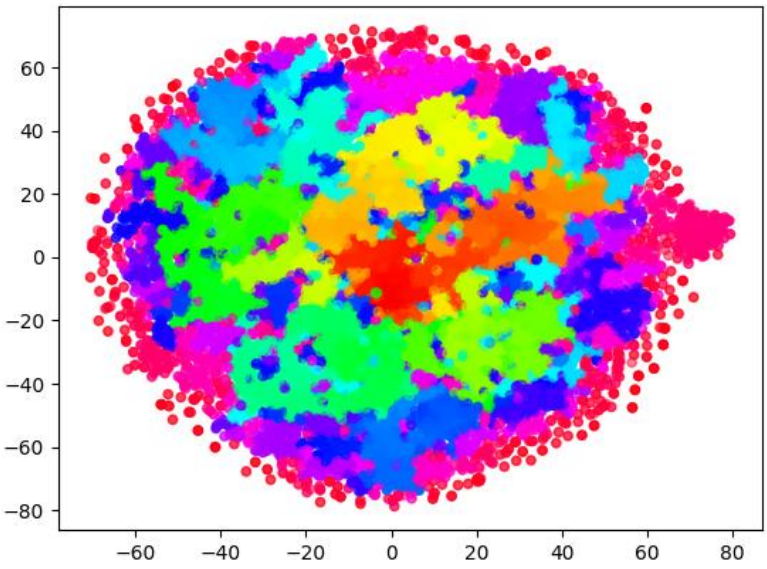

FIGURE A.1 – Résultat de la clusterisation : OPTICS-Word2Vec

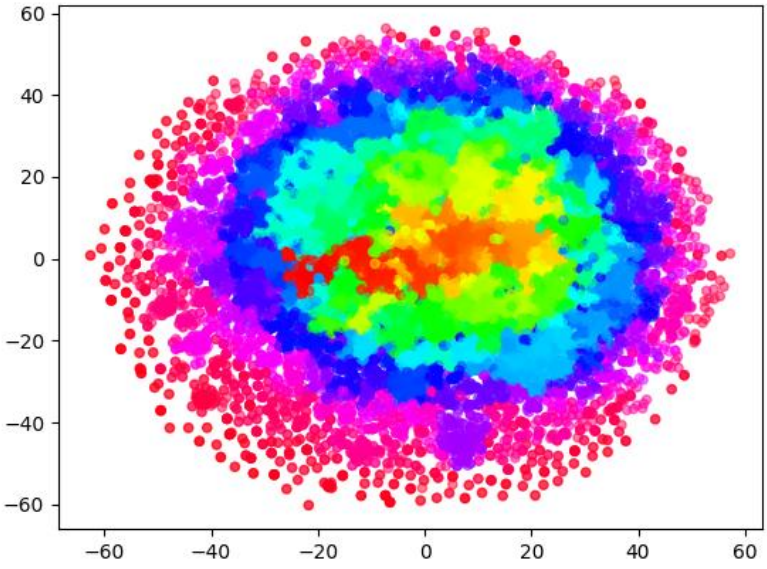

FIGURE A.2 – Résultat de la clusterisation du modèle OPTICS-TFIDF

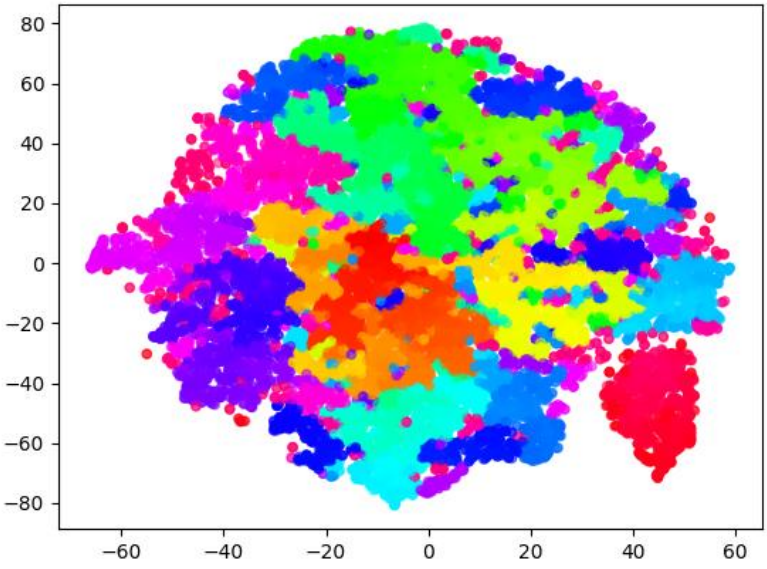

FIGURE A.3 – Résultat de la clusterisation du modèle OPTICS-Doc2Vec

E	D	C	B	S	V	A
2	auto	6464	2696	0.609	w2v	optics
2	auto	6460	2617	0.608	w2v	optics
2	auto	6410	2712	0.401	w2v	optics
2	auto	6302	3335	0.624	tfidf	optics
2	auto	6300	3389	0.622	tfidf	optics
2	auto	6290	3361	0.63	tfidf	optics
2	auto	6286	3437	0.625	tfidf	optics
2	auto	6270	3356	0.623	tfidf	optics
2	auto	6265	3403	0.361	tfidf	optics
2	auto	6227	3384	0.629	tfidf	optics
2	auto	6205	3404	0.361	tfidf	optics
2	auto	6033	3708	0.273	d2v	optics
3	auto	3685	4022	0.307	w2v	optics
3	auto	3473	4665	0.260	w2v	optics
3	auto	3454	4838	0.243	w2v	optics
3	auto	3439	4681	0.619	w2v	optics
3	auto	3418	4681	0.61	w2v	optics
3	auto	3417	4784	0.611	w2v	optics
3	auto	3408	4702	0.607	w2v	optics
3	auto	3373	4959	0.608	w2v	optics
3	auto	3181	5529	0.2	tfidf	optics
3	auto	3098	5548	0.172	d2v	optics
4	auto	2274	6297	0.127	w2v	optics
4	auto	2125	5981	0.155	d2v	optics
4	auto	2113	6513	0.129	tfidf	optics
2	0.9	1827	16308	-0.576	tfidf	dbscan
5	auto	1668	6086	0.152	d2v	optics
5	auto	1663	7340	0.046	w2v	optics
2	0.8	1603	17153	-0.627	tfidf	dbscan
2	0.7	1459	17627	-0.658	tfidf	dbscan
6	auto	1398	6127	0.15	d2v	optics
2	0.6	1338	17995	-0.69	tfidf	dbscan
6	auto	1277	8355	-0.032	w2v	optics
5	auto	1272	9759	-0.126	tfidf	optics
2	0.5	1222	18308	-0.717	tfidf	dbscan
7	auto	1182	6340	0.135	d2v	optics
2	0.4	1130	16293	-0.734	w2v	dbscan
6	auto	1115	8572	-0.018	tfidf	optics
5	auto	1101	11737	0.591	tfidf	optics
2	0.4	1101	18600	-0.742	tfidf	dbscan
7	auto	1036	8901	-0.070	w2v	optics
8	auto	1020	6500	0.125	d2v	optics
2	0.3	991	18863	-0.7615296	tfidf	dbscan
9	auto	932	6149	0.14475657	d2v	optics
2	0.2	908	19058	-0.77720505	tfidf	dbscan
7	auto	907	8997	-0.048878662	tfidf	optics
2	0.5	873	10096	-0.77915627	w2v	dbscan
8	auto	861	9351	-0.10729026	w2v	optics
2	0.1	847	19189	-0.78792834	tfidf	dbscan

TABLE A.2 – Résultats de toutes les expériences lancées 1/5

E	D	C	B	S	V	A
2	0.1	847	19189	-0.7887543	tfidf	dbscan
10	auto	822	6496	0.12267699	d2v	optics
8	auto	795	9186	-0.055169877	tfidf	optics
9	auto	701	10059	-0.15719073	w2v	optics
10	auto	611	10525	-0.1866659	w2v	optics
9	auto	602	10827	-0.1800752	tfidf	optics
10	auto	591	9921	-0.11237885	tfidf	optics
2	0.3	583	19517	-0.8345885	w2v	dbscan
2	0.6	500	5069	-0.8269125	w2v	dbscan
3	0.9	432	19098	-0.77946997	tfidf	dbscan
2	0.2	339	20186	-0.8766286	w2v	dbscan
3	0.4	326	17901	-0.8218948	w2v	dbscan
3	0.8	313	19733	-0.810671	tfidf	dbscan
2	0.7	257	2419	-0.83974814	w2v	dbscan
3	0.7	254	20037	-0.8138957	tfidf	dbscan
3	0.5	251	11340	-0.835435	w2v	dbscan
4	0.9	189	19868	-0.8120051	tfidf	dbscan
3	0.6	183	20305	-0.8279246	tfidf	dbscan
2	0.1	144	20610	-0.88538337	w2v	dbscan
3	0.5	138	20476	-0.81472224	tfidf	dbscan
4	0.4	137	18796	-0.8171525	w2v	dbscan
3	0.6	133	5803	-0.83466536	w2v	dbscan
2	0.8	126	1186	-0.8273046	w2v	dbscan
4	0.8	111	20356	-0.8112499	tfidf	dbscan
3	0.3	104	20475	-0.8185014	w2v	dbscan
4	0.5	102	12314	-0.8260468	w2v	dbscan
3	0.4	98	20606	-0.819447	tfidf	dbscan
5	0.9	81	20337	-0.7935088	tfidf	dbscan
5	0.4	74	19289	-0.8079729	w2v	dbscan
4	0.7	67	20606	-0.7635591	tfidf	dbscan
2	0.9	64	566	-0.7716047	w2v	dbscan
3	0.3	64	20717	-0.77387804	tfidf	dbscan
3	0.7	58	2817	-0.7773548	w2v	dbscan
5	0.5	56	12957	-0.8012017	w2v	dbscan
4	0.6	54	6403	-0.80015683	w2v	dbscan
6	0.5	47	13435	-0.77846575	w2v	dbscan
5	0.8	46	20627	-0.70734787	tfidf	dbscan
3	0.2	41	20792	-0.7729648	tfidf	dbscan
6	0.9	40	20568	-0.7579641	tfidf	dbscan
4	0.6	39	20739	-0.7347389	tfidf	dbscan
6	0.4	38	19669	-0.7444654	w2v	dbscan
7	0.4	36	19879	-0.685168	w2v	dbscan
4	0.3	36	20703	-0.70053357	w2v	dbscan
3	0.2	36	20792	-0.7699843	w2v	dbscan
5	0.6	35	6841	-0.76432925	w2v	dbscan
7	0.5	33	13910	-0.7571528	w2v	dbscan
3	0.1	32	20819	-0.76233935	tfidf	dbscan

TABLE A.3 – Résultats de toutes les expériences lancées 2/5

E	D	C	B	S	V	A
8	0.4	30	20053	-0.65763795	w2v	dbscan
2	0.01	29	20859	-0.8133148	w2v	dbscan
3	0.8	26	1386	-0.70617557	w2v	dbscan
5	0.7	25	20780	-0.59359974	tfidf	dbscan
4	0.5	22	20824	-0.64717287	tfidf	dbscan
7	0.9	21	20694	-0.6646557	tfidf	dbscan
4	0.7	20	3096	-0.69349736	w2v	dbscan
6	0.6	20	7205	-0.6880265	w2v	dbscan
8	0.5	19	14365	-0.6997651	w2v	dbscan
6	0.8	19	20770	-0.4975716	tfidf	dbscan
5	0.3	19	20783	-0.6419527	w2v	dbscan
3	0.9	18	658	-0.6874891	w2v	dbscan
8	0.6	17	7765	-0.66016793	w2v	dbscan
9	0.4	17	20242	-0.6384796	w2v	dbscan
7	0.6	16	7492	-0.680086	w2v	dbscan
9	0.5	16	14694	-0.6636102	w2v	dbscan
4	0.4	15	20856	-0.6712601	tfidf	dbscan
5	0.6	12	20853	-0.49880826	tfidf	dbscan
5	0.7	11	3325	-0.5274365	w2v	dbscan
9	0.6	11	8061	-0.62159365	w2v	dbscan
8	0.9	11	20774	-0.5547488	tfidf	dbscan
7	0.8	11	20822	-0.49262944	tfidf	dbscan
3	0.1	11	20876	-0.6812413	w2v	dbscan
6	0.3	10	20834	-0.5462142	w2v	dbscan
4	0.8	9	1527	-0.50886995	w2v	dbscan
6	0.7	8	3503	-0.35959673	w2v	dbscan
4	0.3	8	20885	-0.50169104	tfidf	dbscan
6	0.7	7	20871	-0.48070845	tfidf	dbscan
5	0.8	6	1629	-0.4436519	w2v	dbscan
9	0.9	6	20816	-0.43974745	tfidf	dbscan
7	0.3	6	20859	-0.22028695	w2v	dbscan
4	0.2	6	20884	-0.515984	w2v	dbscan
5	0.5	5	20892	-0.33879346	tfidf	dbscan
4	0.9	4	736	-0.40218836	w2v	dbscan
6	0.9	4	826	-0.24909617	w2v	dbscan
6	0.8	4	1732	-0.3470958	w2v	dbscan
7	0.7	4	3661	-0.35801318	w2v	dbscan
8	0.8	4	20872	-0.35554835	tfidf	dbscan
2	0.1	3	7	-0.10714677	d2v	dbscan
3	0.1	3	7	0.1625821	d2v	dbscan
4	0.1	3	11	0.15585358	d2v	dbscan
5	0.1	3	11	0.17639792	d2v	dbscan

TABLE A.4 – Résultats de toutes les expériences lancées 3/5

E	D	C	B	S	V	A
5	0.9	3	781	-0.14212511	w2v	dbscan
8	0.3	3	20888	0.086056426	w2v	dbscan
7	0.7	3	20895	-0.41364613	tfidf	dbscan
5	0.2	3	20896	-0.22229503	w2v	dbscan
6	0.6	3	20899	-0.111291185	tfidf	dbscan
4	0.1	3	20900	-0.4596487	w2v	dbscan
2	0.4	2	0	-0.051543407	d2v	dbscan
2	0.5	2	0	0.030510955	d2v	dbscan
2	0.6	2	0	-0.09544389	d2v	dbscan
2	0.7	2	0	0.17701256	d2v	dbscan
2	0.8	2	0	0.1610149	d2v	dbscan
2	0.3	2	2	0.1596997	d2v	dbscan
9	0.2	2	10	0.16187511	d2v	dbscan
6	0.1	2	18	0.14938907	d2v	dbscan
8	0.8	2	1890	-0.17830884	w2v	dbscan
8	0.7	2	3802	-0.08754132	w2v	dbscan
9	0.7	2	3965	-0.08813693	w2v	dbscan
9	0.3	2	20897	0.091186404	w2v	dbscan
6	0.2	2	20901	-0.18876828	w2v	dbscan
5	0.1	2	20904	-0.34142962	w2v	dbscan
6	0.1	2	20904	-0.19486824	w2v	dbscan
4	0.1	2	20909	-0.39447564	tfidf	dbscan
4	0.2	2	20909	-0.37567976	tfidf	dbscan
4	0.9	1	1	0.1624735	d2v	dbscan
5	0.9	1	1	0.08313447	d2v	dbscan
6	0.9	1	1	0.16859485	d2v	dbscan
7	0.9	1	1	0.023673747	d2v	dbscan
8	0.9	1	1	0.15751103	d2v	dbscan
3	0.4	1	2	0.17154387	d2v	dbscan
3	0.5	1	2	0.16133566	d2v	dbscan
4	0.5	1	2	0.17821267	d2v	dbscan
5	0.5	1	2	0.16332367	d2v	dbscan
3	0.6	1	2	-0.044003148	d2v	dbscan
4	0.6	1	2	-0.052173536	d2v	dbscan
5	0.6	1	2	0.15921676	d2v	dbscan
6	0.6	1	2	0.13724943	d2v	dbscan
7	0.6	1	2	0.15680724	d2v	dbscan
3	0.7	1	2	0.1684625	d2v	dbscan
4	0.7	1	2	-0.1168296	d2v	dbscan
5	0.7	1	2	0.19273971	d2v	dbscan
6	0.7	1	2	0.14755985	d2v	dbscan
7	0.7	1	2	0.17127243	d2v	dbscan
8	0.7	1	2	0.1717244	d2v	dbscan
9	0.7	1	2	0.16433059	d2v	dbscan
3	0.8	1	2	0.17903732	d2v	dbscan
4	0.8	1	2	0.13241796	d2v	dbscan
5	0.8	1	2	0.1646727	d2v	dbscan
6	0.8	1	2	0.15000828	d2v	dbscan
7	0.8	1	2	0.15210448	d2v	dbscan

TABLE A.5 – Résultats de toutes les expériences lancées 4/5

E	D	C	B	S	V	A
8	0.8	1	2	0.13389465	d2v	dbscan
9	0.8	1	2	0.17555673	d2v	dbscan
9	0.9	1	2	-0.13639946	d2v	dbscan
4	0.4	1	3	0.14837842	d2v	dbscan
5	0.4	1	3	0.13321234	d2v	dbscan
6	0.4	1	3	0.17328303	d2v	dbscan
7	0.4	1	3	0.1879847	d2v	dbscan
6	0.5	1	3	0.17766866	d2v	dbscan
7	0.5	1	3	0.1581189	d2v	dbscan
8	0.5	1	3	0.21051641	d2v	dbscan
9	0.5	1	3	-0.05642069	d2v	dbscan
8	0.6	1	3	-0.1648226	d2v	dbscan
9	0.6	1	3	0.16435628	d2v	dbscan
2	0.2	1	4	0.12298624	d2v	dbscan
3	0.2	1	4	0.13954726	d2v	dbscan
4	0.2	1	4	0.11147678	d2v	dbscan
5	0.2	1	4	0.033948615	d2v	dbscan
6	0.2	1	4	0.17009825	d2v	dbscan
3	0.3	1	4	0.14787464	d2v	dbscan
4	0.3	1	4	0.15581027	d2v	dbscan
5	0.3	1	4	0.13252729	d2v	dbscan
6	0.3	1	4	0.16536397	d2v	dbscan
7	0.3	1	4	0.18195572	d2v	dbscan
8	0.3	1	4	0.14692584	d2v	dbscan
9	0.3	1	4	0.15108125	d2v	dbscan
8	0.4	1	4	0.1734531	d2v	dbscan
9	0.4	1	4	-0.022837218	d2v	dbscan
7	0.2	1	6	0.18238677	d2v	dbscan
8	0.2	1	9	0.18762529	d2v	dbscan
7	0.1	1	24	0.070339225	d2v	dbscan
8	0.1	1	24	0.16210562	d2v	dbscan
9	0.1	1	24	0.14673074	d2v	dbscan
7	0.9	1	876	0.06525951	w2v	dbscan
8	0.9	1	926	0.05431273	w2v	dbscan
9	0.9	1	952	0.066	w2v	dbscan
7	0.8	1	1811	0.057	w2v	dbscan
9	0.8	1	1980	0.058	w2v	dbscan
9	0.8	1	20896	0.136	tfidf	dbscan
7	0.2	1	20907	0.18	w2v	dbscan
8	0.2	1	20907	0.179	w2v	dbscan
9	0.2	1	20907	0.183	w2v	dbscan
8	0.7	1	20909	-0.096	tfidf	dbscan
7	0.1	1	20910	0.178	w2v	dbscan
5	0.4	1	20912	0.173	tfidf	dbscan

TABLE A.6 – Résultats de toutes les expériences lancées 5/5

ID	Citation	N° de cluster
647	banque des copains de poutine	278
648	banque des copains de vladimir poutine	278
649	la banque des copains de poutine	278
650	oui , les fantômes de l ' esprit de revanche , les violations flagrantes de la souveraineté des etats , l ' intolérable mépris des peuples , la volonté impérialiste ressurgissent du passé pour s ' imposer dans le quotidien de notre europe , de nos voisins , de nos amis	279
651	oui , les fantômes de l ' esprit de revanche , les violations flagrantes de la souveraineté des états , l ' intolérable mépris des peuples , la volonté impérialiste ressurgissent du passé pour s ' imposer dans le quotidien de notre europe , de nos voisins , de nos amis	279
652	oui , les fantômes de l'esprit de revanche , les violations flagrantes de la souveraineté des états , l'intolérable mépris des peuples , la volonté impérialiste ressurgissent du passé pour s'imposer dans le quotidien de notre europe , de nos voisins , de nos amis	279
653	œuvrer pour notre souveraineté énergétique , pour accompagner les français , nos entreprises , dans le contexte de cette guerre	280
654	œuvrer pour notre souveraineté énergétique , pour accompagner les français , nos entreprises dans le contexte de cette guerre	280
655	on a besoin de tous nos compatriotes pour qu ' ils comprennent qu ' acheter français , consommer français , aimer notre agriculture , c ' est la clé pour garder cette souveraineté et ne pas se réveiller demain avec la gueule de bois	281
656	on a besoin de tous nos compatriotes pour qu'ils comprennent que acheter français , consommer français , aimer notre agriculture , c'est la clé pour garder cette souveraineté et ne pas se réveiller demain avec la gueule de bois	281
657	la partie russe (avait) confirmé être prête à fournir toute l ' assistance nécessaire aux inspecteurs	282
658	la partie russe a confirmé être prête à fournir toute l'assistance nécessaire aux inspecteurs	282

TABLE A.7 – Extrait du corpus annoté

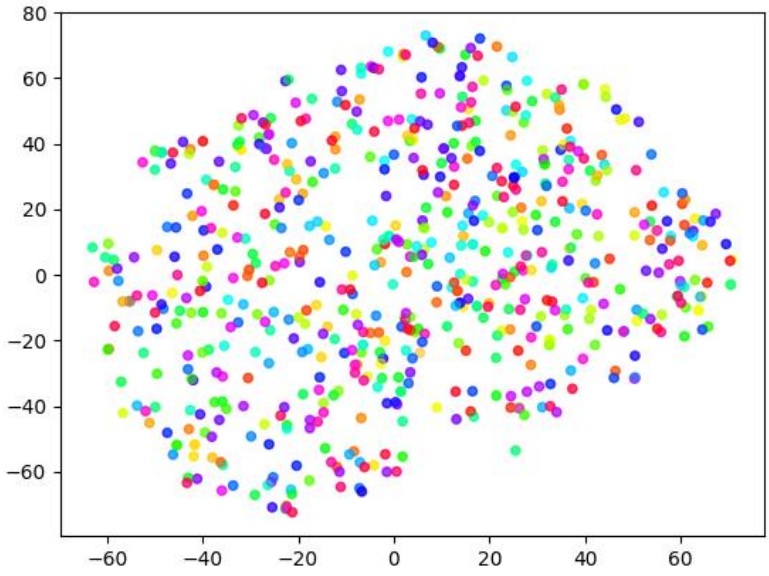

FIGURE A.4 – Résultat de la clusterisation du modèle DBSCAN-Word2Vec

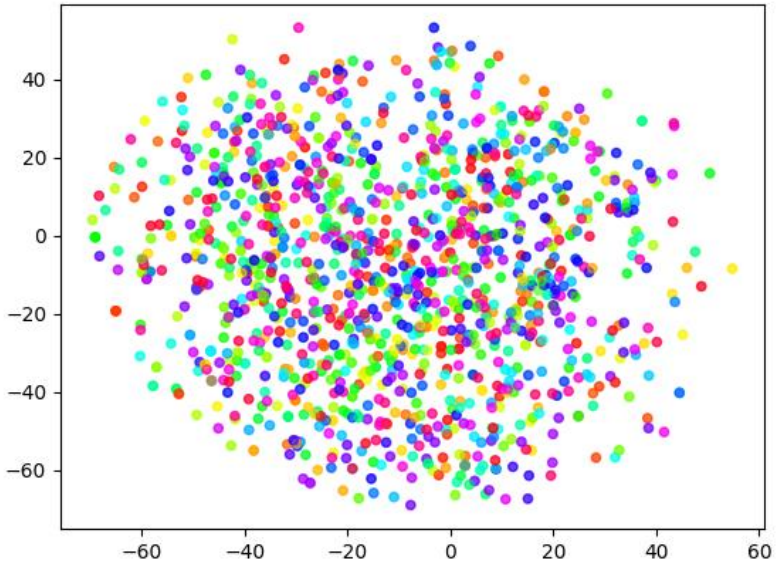

FIGURE A.5 – Résultat de la clusterisation du modèle DBSCAN-TFIDF

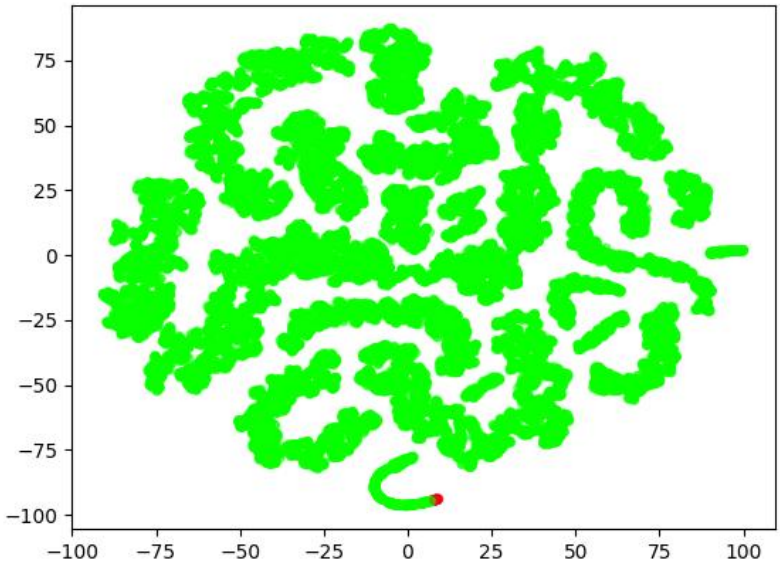

FIGURE A.6 – Résultat de la clusterisation du modèle DBSCAN-Doc2Vec

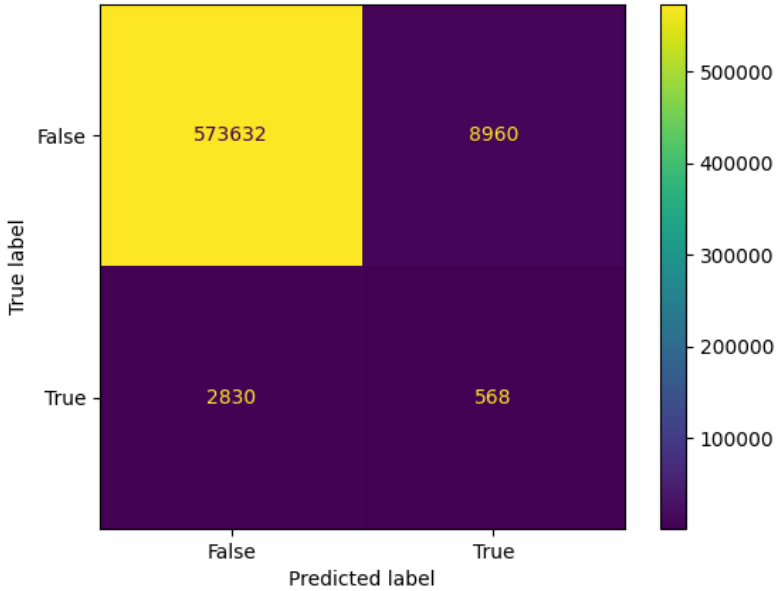

FIGURE A.7 – Matrice de confusion (Corpus annoté - OPTICS-Doc2Vec)

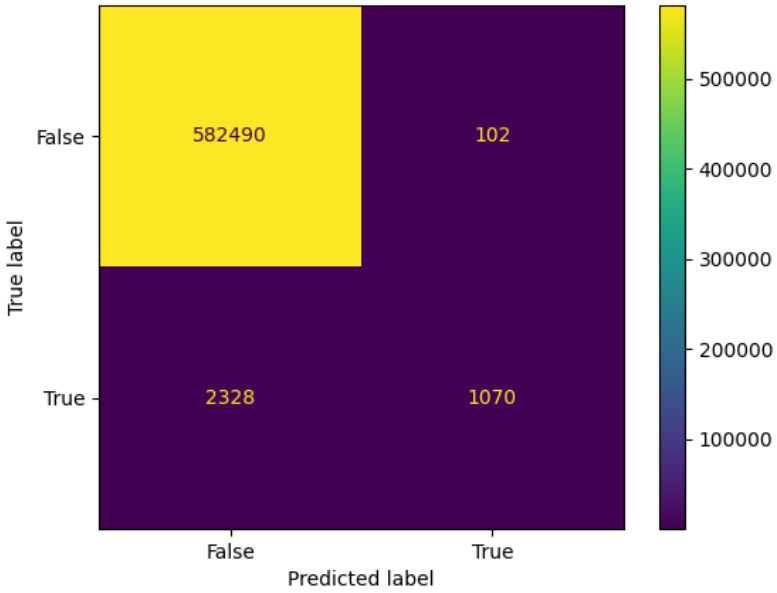

FIGURE A.8 – Matrice de confusion (Corpus annoté - OPTICS-TFIDF)

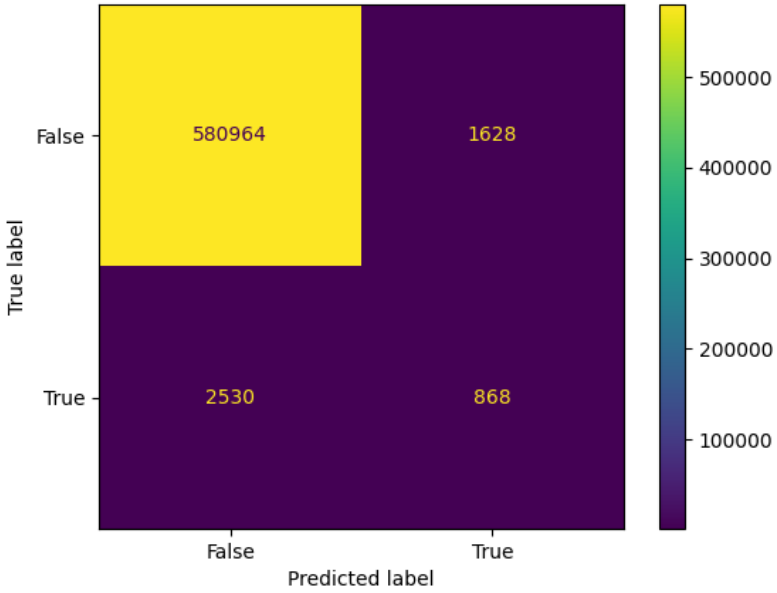

FIGURE A.9 – Matrice de confusion (Corpus annoté - OPTICS-Word2Vec)

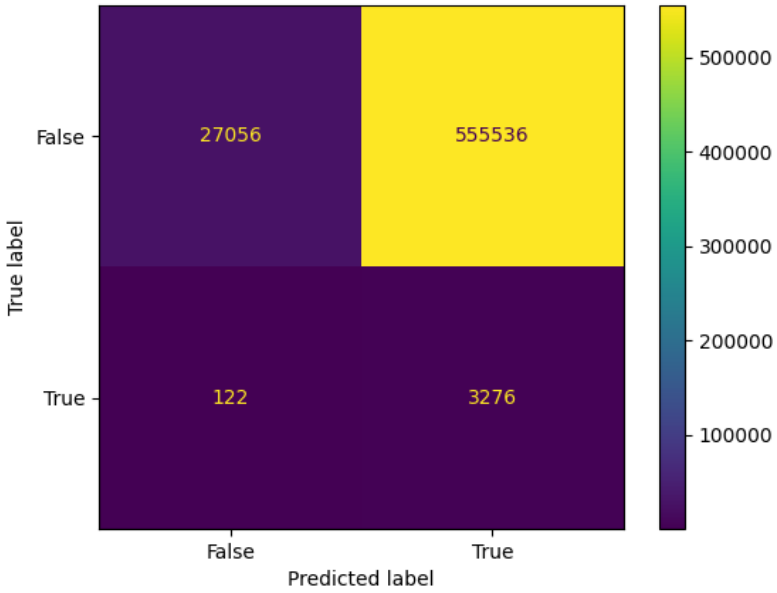

FIGURE A.10 – Matrice de confusion (Corpus annoté - DBSCAN-Doc2Vec)

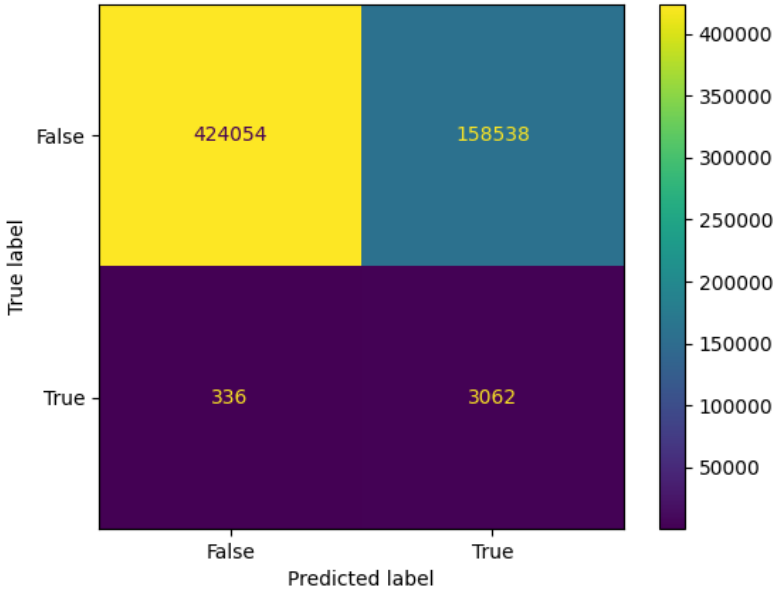

FIGURE A.11 – Matrice de confusion (Corpus annoté - DBSCAN-TFIDF)

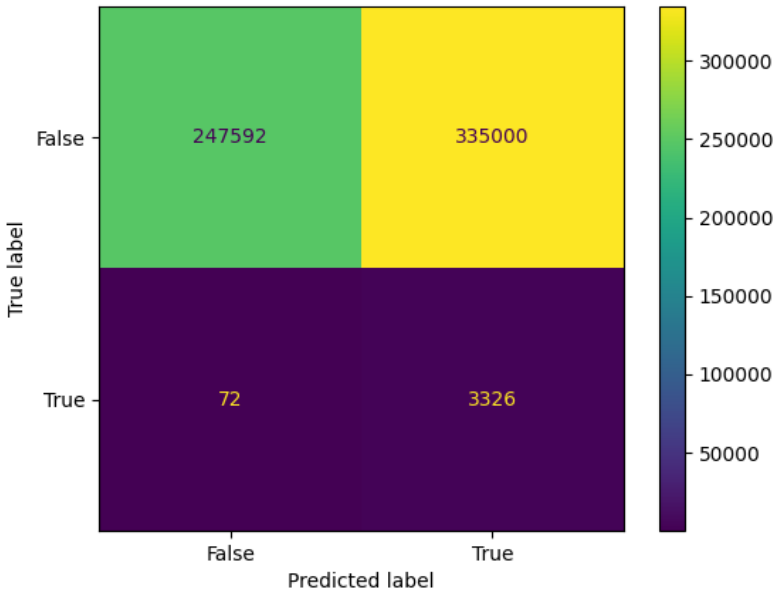

FIGURE A.12 – Matrice de confusion (Corpus annoté - DBSCAN-Word2Vec)

Eps	Clusters	Bruit	Silhouette	Davies-Bouldin	Vectorisation
0.1	275	2784	-0.754	0.839	d2v
0.2	31	278	-0.184	0.697	d2v
0.3	8	86	0.11	0.759	d2v
0.4	6	40	0.14	0.531	d2v
0.5	4	20	0.245	0.521	d2v
0.6	3	11	0.254	0.505	d2v
0.7	2	7	0.251	0.516	d2v
0.8	2	7	0.252	0.515	d2v
0.9	3	4	0.15	0.533	d2v
0.1	1207	18829	0.999	0.001	tfidf
0.2	1267	18699	0.997	0.004	tfidf
0.3	1342	18512	0.996	0.006	tfidf
0.4	1444	18257	0.988	0.018	tfidf
0.5	1554	17977	0.979	0.032	tfidf
0.6	1654	17681	0.969	0.052	tfidf
0.7	1764	17324	0.957	0.075	tfidf
0.8	1897	16857	0.914	1.367	tfidf
0.9	2099	16039	0.853	2.274	tfidf
0.1	620	20090	0.991	0.695	w2v
0.2	1061	18734	0.633	2.637	w2v
0.3	805	7649	-0.683	1.862	w2v
0.4	167	1731	-0.83	1.156	w2v
0.5	34	435	-0.695	0.953	w2v
0.6	9	157	-0.486	1.514	w2v
0.7	7	71	-0.437	1.278	w2v
0.8	6	44	-0.455	1.645	w2v
0.9	2	24	-0.085	1.055	w2v

TABLE A.8 – Résultats de l'algoritme DBSCAN selon des distances différentes