Université de Montréal

Regroupement de textes avec des approches simples et efficaces exploitant la représentation vectorielle contextuelle SBERT

par

Uros Petricevic

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en Informatique

9 décembre 2022

Université de Montréal

Faculté des arts et des sciences

Ce mémoire intitulé

Regroupement de textes avec des approches simples et efficaces exploitant la représentation vectorielle contextuelle SBERT

présenté par

Uros Petricevic

a été évalué par un jury composé des personnes suivantes :

Bang Liu
(président-rapporteur)

Philippe Langlais
(directeur de recherche)

Guy Lapalme
(membre du jury)

Résumé

Le regroupement est une tâche non supervisée consistant à rassembler les éléments semblables sous un même groupe et les éléments différents dans des groupes distincts. Le regroupement de textes est effectué en représentant les textes dans un espace vectoriel et en étudiant leur similarité dans cet espace. Les meilleurs résultats sont obtenus à l'aide de modèles neuronaux qui affinent une représentation vectorielle contextuelle de manière non supervisée. Or, cette technique peuvent nécessiter un temps d'entraînement important et sa performance n'est pas comparée à des techniques plus simples ne nécessitant pas l'entraînement de modèles neuronaux.

Nous proposons, dans ce mémoire, une étude de l'état actuel du domaine. Tout d'abord, nous étudions les meilleures métriques d'évaluation pour le regroupement de textes. Puis, nous évaluons l'état de l'art et portons un regard critique sur leur protocole d'entraînement. Nous proposons également une analyse de certains choix d'implémentation en regroupement de textes, tels que le choix de l'algorithme de regroupement, de la mesure de similarité, de la représentation vectorielle ou de l'affinage non supervisé de la représentation vectorielle. Finalement, nous testons la combinaison de certaines techniques ne nécessitant pas d'entraînement avec la représentation vectorielle contextuelle telles que le prétraitement des données, la réduction de dimensionnalité ou l'inclusion de Tf-idf.

Nos expériences démontrent certaines lacunes dans l'état de l'art quant aux choix des métriques d'évaluation et au protocole d'entraînement. De plus, nous démontrons que l'utilisation de techniques simples permet d'obtenir des résultats meilleurs ou semblables à des méthodes sophistiquées nécessitant l'entraînement de modèles neuronaux. Nos expériences sont évaluées sur huit corpus issus de différents domaines.

Mots-clés: Regroupement de textes, représentation vectorielle contextuelle, réduction de dimensionnalité, apprentissage automatique, traitement automatique du langage naturel, SBERT, Tf-idf, UMAP, TSDEA.

Abstract

Clustering is an unsupervised task of bringing similar elements in the same cluster and different elements in distinct groups. Text clustering is performed by representing texts in a vector space and studying their similarity in this space. The best results are obtained using neural models that fine-tune contextual embeddings in an unsupervised manner. However, these techniques require a significant amount of training time and their performance is not compared to simpler techniques that do not require training of neural models.

In this master's thesis, we propose a study of the current state of the art. First, we study the best evaluation metrics for text clustering. Then, we evaluate the state of the art and take a critical look at their training protocol. We also propose an analysis of some implementation choices in text clustering, such as the choice of clustering algorithm, similarity measure, contextual embeddings or unsupervised fine-tuning of the contextual embeddings. Finally, we test the combination of contextual embeddings with some techniques that don't require training such as data preprocessing, dimensionality reduction or Tf-idf inclusion.

Our experiments demonstrate some shortcomings in the state of the art regarding the choice of evaluation metrics and the training protocol. Furthermore, we demonstrate that the use of simple techniques yields better or similar results to sophisticated methods requiring the training of neural models. Our experiments are evaluated on eight benchmark datasets from different domains.

Keywords: Text clustering, contextual word embedding, dimension reduction, machine learning, natural language processing, SBERT, Tf-idf, UMAP, TSDEA.

Table des matières

Liste des tableaux

Liste des figures

Remerciements

Je voudrais tout d'abord remercier Philippe Langlais de m'avoir donné la liberté de travailler sur un sujet qui m'intéresse et de pouvoir explorer différentes avenues, tout en m'encadrant à travers ce processus. Je voudrais remercier Frank Coggins, pour m'avoir orienté vers le regroupement de texte. Je voudrais également remercier l'entreprise LexRock^{AI}, la Banque Nationale et l'Université de Montréal pour leur soutien financier. Finalement, je remercie mes parents, ma conjointe et mes amis pour leur soutien.

Introduction

Motivation

La polarisation observée actuellement dans notre société inquiète de plus en plus. Nous n'avons qu'à penser à l'assaut du Capitole des États-Unis le 6 janvier 2021[1]. À ce phénomène s'ajoute la propagation des fausses nouvelles sur internet. Les gouvernements et les médias sociaux s'efforcent de trouver des solutions[2,3,4]. Pour faire face à ces problématiques, une analyse des messages véhiculés dans les médias sociaux, journaux et autres plateformes s'impose afin d'en déceler les tendances principales et de mieux comprendre les différents groupes d'individus.

Une des premières étapes pour analyser une grande quantité de données peut être de faire du regroupement de textes, c'est à dire de rassembler les textes similaires. Par exemple, si on prend un sujet comme la vaccination ou la guerre actuelle en Ukraine nous pourrions vouloir regrouper tous les textes qui traitent de ces deux sujets dans des groupes distincts. Par la suite, pour chacun des sujets nous pourrions faire des sous-groupes représentant les thèmes abordés et décider d'actions à prendre comme par exemple les supprimer ou restreindre le partage de données jugées comme fausses ou trompeuses.

Ainsi, le regroupement de textes peut être considéré comme une tâche cruciale dans l'analyse et la résolution de problématiques reliées à l'exploration de données. Organiser ces textes en groupes semblables (souvent par sujet, mais il est possible de le faire aussi sous d'autres angles, par exemple par style d'écriture) devient crucial pour des tâches en traitement de données que ce soit le résumé de textes, l'analyse de patrons fréquents, la recherche ou le filtrage d'informations.

---

[1] https://www.theglobeandmail.com/world/us-politics/article-testimony-by-former-trump-aides-to-be-in-spotlight-as-us-capitol-riot/

[2] https://efus.eu/thematiques/quest-ce-que-la-polarisation-et-comment-y-repondre-a-lechelle-locale/?lang=fr

[3] https://www.theverge.com/2022/6/16/23168987/eu-code-disinformation-online-propaganda-facebook-twitter-tiktok

[4] https://www.facebook.com/formedia/blog/working-to-stop-misinformation-and-false-news

La littérature propose de nombreuses techniques qui exploitent des algorithmes inventés il y a plusieurs décennies, *K-Means* [**MacQueen, 1967**] ou *DBSCAN* [**Ester et al., 1996a**] par exemple. C'est le format des données en entrée qui a changé au fil du temps. Alors que dans les débuts, la représentation des mots était sous forme de sac de mots, [**Willett, 1988**] et [**Aggarwal and Zhai, 2012**], aujourd'hui les meilleures techniques exploitent la représentation vectorielle de mots apprise sur d'immenses quantités de données par des modèles de types *Transformer* [**Vaswani et al., 2017**]. Ces modèles ont permis d'importantes avancées en traitement du langage naturel (TALN) et offrent les meilleurs résultats dans la majorité des tâches. Malgré tout, certaines lacunes ont tout de même été observées. Pour le cas du regroupement de texte, celle qui nous intéresse est la difficulté de ces modèles à avoir une bonne représentation des mots pour des domaines spécifiques, par exemple sur un corpus médical ou informatique. Pour y remédier, certaines techniques d'affinage ont été tentées. De plus, ces modèles représentent les mots dans un nombre élevé de dimensions, ce qui peut rendre le regroupement de texte difficile pour un algorithme comme K-Means.

Les meilleurs résultats sont obtenus à l'aide de modèles neuronaux qui affinent une représentation vectorielle contextuelle de manière non supervisée. Or, cette technique peut nécessiter un temps d'entraînement important et une étude des techniques plus simples semble manquante. Plus précisément, l'état de l'art en représentation vectorielle contextuelle, *SBERT*, est relativement récent [**Reimers and Gurevych, 2019**]. Pour autant que nous le sachions, une étude des meilleures techniques pour mettre à profit cette représentation sans l'affiner à l'aide de modèle neuronaux n'est pas effectuée.

Contributions

Nous proposons, dans ce mémoire, une étude de l'état actuel du domaine. Tout d'abord, nous étudions les meilleures métriques d'évaluation pour le regroupement de textes. Puis, nous évaluons l'état de l'art et portons un regard critique sur leur protocole d'entraînement. Nous proposons également une analyse de certains choix d'implémentation en regroupement de textes, tels que le choix de l'algorithme de regroupement, de la mesure de similarité, de la représentation vectorielle ou de l'affinage non supervisé de la représentation vectorielle. Finalement, nous testons la combinaison de certaines techniques ne nécessitant pas d'entraînement avec la représentation vectorielle contextuelle telles que le prétraitement des données, la réduction de dimensionnalité ou l'inclusion de Tf-idf.

Nos expériences démontrent certaines lacunes dans l'état de l'art quant aux choix des métriques d'évaluation et au protocole d'entraînement. De plus, nous démontrons que l'utilisation de techniques simples permet d'obtenir des résultats meilleurs ou semblables à des méthodes sophistiquées nécessitant l'entraînement de modèles neuronaux. Nos expériences sont évaluées sur huit corpus issus de différents domaines.

Organisation du mémoire

Au chapitre 1, une revue de la littérature est présentée afin de mettre en lumière les différents concepts reliés au regroupement de textes et les tendances actuelles. Au chapitre 2, nous nous intéressons à comprendre les particularités des métriques d'évaluation du regroupement de textes. Au chapitre 3, nous décrivons les données que nous utilisons. Au chapitre 4, nous présentons l'état de l'art en regroupement de textes et tentons de reproduire les résultats. Au chapitre 5, nous présentons les résultats de nos expériences et analysons les meilleurs choix d'implémentation pour effectuer le regroupement de texte.

Chapitre 1

Travaux reliés

Dans ce chapitre, nous introduisons les travaux reliés en présentant les principaux concepts clés du mémoire. Nous commençons tout d'abord par présenter une définition formelle du regroupement. Par la suite, nous décrivons une étape cruciale du regroupement, la représentation vectorielle du texte, avec une attention particulière sur la représentation vectorielle contextuelle utilisant des modèles Transformers. Ensuite, nous explorons l'affinage possible des modèles Transformers et discutons de l'intérêt d'appliquer une réduction de dimensionnalité sur la représentation vectorielle. Puis, nous voyons les algorithmes de regroupement. Finalement, nous décrivons les modèles neuronaux en regroupement de textes.

1.1. Définition formelle de regroupement

Le regroupement (*clustering* en anglais) est une tâche non supervisée de classement d'éléments dans des groupes (*clusters* en anglais) [**Jain et al., 1999**]. C'est un problème de partitionnement qui prend en entrée un ensemble d'éléments finis et qui les divise en groupe de sorte que les éléments semblables sont regroupés sous le même groupe et les éléments différents sont regroupés dans des groupes distincts. Pour ce faire une mesure de similarité doit être établie, souvent c'est en mesurant la distance des éléments dans l'espace latent. Par exemple, en utilisant la distance euclidienne pour la représentation vectorielle du texte. À la figure 1.1 il est possible de voir un exemple de regroupement. Dans les 50 dernières années, de nombreuses techniques de représentation des données et une grande variété d'algorithmes de partitionnement ont été proposées, donnant lieu à une littérature riche.

Il est important de faire la distinction entre le regroupement et la classification. En classification, lors de l'entraînement, les données en entrée sont déjà identifiées à un groupe et le défi est de réussir à grouper de nouvelles données non identifiées. Alors qu'en regroupement les données en entrée ne sont pas identifiées au préalable à des groupes et le défi est de réussir à trouver les groupes dans lesquels les associer. Le regroupement peut être considéré comme

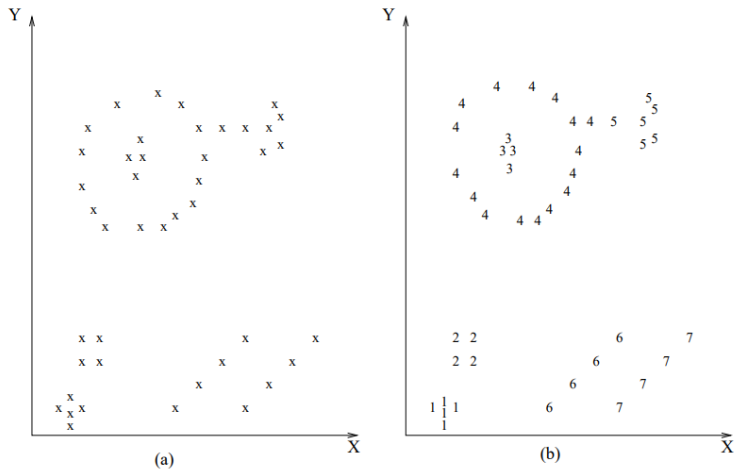


Fig. 1.1. Exemple de regroupement. Les données sont représentées dans un espace à deux dimensions. À la figure 1.1(a) les données en entrée sont représentées. À la figure 1.1(b) le regroupement est effectué, chaque donnée est classée dans un groupe. Figure tirée de [**Jain et al., 1999**].

une tâche plus difficile, car moins d'information sur les données est disponible et l'on doit découvrir les caractéristiques de ces données pour en faire un regroupement significatif.

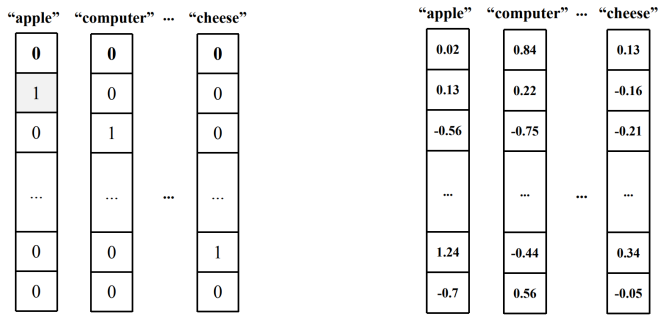
Il est également pertinent de souligner que le regroupement peut être de deux types, soit fort, soit diffus (traduction de *hard* et *fuzzy*). Pour le regroupement fort, chaque élément d'entrée est associé à uniquement un groupe alors que dans le regroupement diffus chaque élément peut être associé à plusieurs groupes avec des degrés différents d'appartenance à chaque groupe. Dans ce mémoire nous nous intéressons à des regroupements forts.

Soulignons aussi une autre distinction importante en regroupement : les algorithmes qui connaissent le nombre de groupes au préalable et ceux dont le nombre de groupes fait partie des éléments à découvrir. Dans des tâches réelles, il est fréquent de ne pas connaître le nombre de groupes, par exemple, le nombre de sujets traités dans des textes extraits de médias sociaux.

Les étapes classiques en regroupement sont les suivantes: représentation adéquate des données à fournir en entrée aux algorithmes de regroupement; pour le texte il s'agit généralement d'une représentation vectorielle. Ensuite, exécution d'un algorithme de regroupement et finalement évaluation du regroupement. Les deux premières étapes sont approfondies dans les sections suivantes du chapitre, alors que l'évaluation sera traitée dans le prochain chapitre.

1.2. Représentation vectorielle du texte

Représenter du texte sous forme d'un vecteur joue un rôle essentiel en traitement du langage naturel. Cette représentation nous permet de faire des opérations sur celle-ci ou



(a) Encodage à chaud .

(b) Plongement de mots.

Fig. 1.2. Représentation vectorielle de mots. Alors qu'à la figure (a) les mots sont représentés par un vecteur de la taille du vocabulaire avec toutes les dimensions égales à 0 sauf pour celle représentant le mot encodé, où la valeur est égale à 1. À la figure (b) les mots sont représentés avec des nombres réels dans chaque dimension et la taille du vecteur est généralement plus petite que la taille du vocabulaire.

d'être utilisée par les algorithmes d'apprentissage machine, ce qui s'avère autrement difficile avec une représentation textuelle. La représentation vectorielle du texte peut se faire au niveau des documents, des phrases, des mots ou même des caractères.

1.2.1. Approches basées sur la fréquence

Les premières techniques de représentation de texte sont basées sur un encodage à chaud (*one-hot encoding* en anglais). Chaque dimension représente un mot du vocabulaire et seulement la dimension du mot encodé possède la valeur à 1, les autres dimensions sont à 0. Une approche plus efficace est TF-IDF [**Sammut and Webb, 2010**] qui assigne une pondération à chaque mot en fonction de sa fréquence dans le document encodé et la fréquence inverse dans la collection (tous les documents). L'intuition est qu'un mot très présent dans le document encodé et peu présent dans les autres documents est représentatif de ce document et donc un poids plus élevé devrait être attribué à ce mot. Le grand avantage de ces techniques se basant principalement sur la fréquence des mots comme caractéristique déterminante est leur simplicité. Elles sont une bonne première méthode à essayer. Cependant, elles comportent de nombreuses limitations [**Naseem et al., 2020**], dont la malédiction de la dimensionnalité et l'incapacité à tenir compte de l'ordre des mots dans le texte.

1.2.2. Approches distributionnelles

L'introduction du plongement de mots (*word embeddings* en anglais) [**Bengio et al., 2003**] et sa démocratisation à l'aide d'un modèle rapide, Word2Vec de Google [**Mikolov et al., 2013**] ont été des importantes avancées en TALN. Cette technique permet de représenter la connaissance de manière diffuse dans toutes les dimensions et dans la plupart des

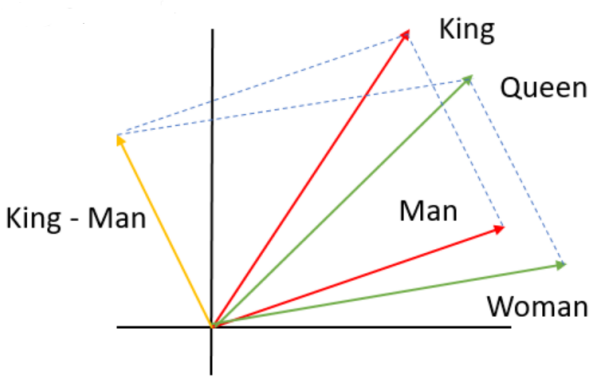


Fig. 1.3. Exemple classique pour illustrer les opérations possibles avec les plongements de mots. Nous pouvons voir qu'en prenant la représentation du mot *King* et en y soustrayant celle de *Man*, puis en additionnant celle de *Woman*, nous devrions obtenir celle du mot *Queen*.

cas ce sont maintenant des nombres réels au lieu de valeurs binaires. La sémantique des mots peut maintenant être apprise et des opérations de similarité peuvent être effectuées. Cette approche s'appuie sur l'hypothèse que des mots similaires apparaissent dans des contextes similaires. L'exemple classique est illustré à la figure 1.3.

Cependant, cette approche a une lacune importante, le contexte d'un mot n'est pas pris en compte. Par exemple, dans les phrases «J'ai appelé mon avocat» et «Je mange un avocat» le mot avocat a la même représentation vectorielle alors qu'il s'agit de deux mots sémantiquement différents.

1.2.3. Approches contextuelles (ou modernes)

En réponse à cette problématique les approches contextuelles ont vu le jour. Les mots n'ont plus une représentation statique, mais plutôt une qui dépend du contexte dans lesquels les mots se trouvent. Les modèles dominants exploitent l'architecture Transformer [**Vaswani et al., 2017**].

Nous allons considérer une particularité de la représentation vectorielle soit celle de la représentation pour les phrases. La technique traditionnelle consiste à prendre la représentation vectorielle de chacun des mots et d'en faire la moyenne ou une moyenne pondérée selon la fréquence pour obtenir la représentation vectorielle de la phrase [**Arora et al., 2017a**].

De plus, nous allons nous concentrer principalement sur le modèle de représentation vectorielle *Sentence-BERT (SBERT)* [**Reimers and Gurevych, 2019**] étant celui avec les résultats à l'état de l'art. Ce modèle sera présenté dans la prochaine section. Depuis 2018 de nombreux modèles exploitant l'architecture de type Transformer ont été proposés, certains étant des variantes de SBERT comme [**Liao, 2021**]. Un autre exemple est *Universal Sentence Encoder (USE)* [**Cer et al., 2018**], mais celui-ci requiert des temps de calcul trop

élevés. Les auteurs de SBERT ont aussi montré que d'autres tentatives de représentations contextuelles offrent en réalité des résultats inférieurs sur la similarité sémantique à des approches distributionnelles comme GloVe [**Pennington et al., 2014**].

1.3. Représentation vectorielle exploitant l'architecture Transformer

Avant de décrire le modèle SBERT nous allons présenter brièvement les modèles sous-jacents, Transformer et BERT, mais pour de plus amples détails voir [**Vaswani et al., 2017**] et [**Devlin et al., 2018**].

1.3.1. Transformer

Un modèle Transformer est un modèle encodeur-décodeur avec deux concepts clés: les couches *Multi-head self-attention* et l'encodage positionnel. En plus du papier de référence, un article[1] intéressant sur internet explique en détail son fonctionnement. *Self-attention* est le processus qui permet au Transformer de comprendre les relations entre les mots. Par exemple, d'identifier à quel groupe nominal un déterminant est associé. Ayant abandonné l'aspect séquentiel des réseaux récurrents de neurones (RNN) [**Sherstinsky, 2020**], qui par leur nature prennent en compte la position des mots dans le texte, un vecteur d'encodage positionnel a été proposé pour compenser. Plus précisément, celui-ci est ajouté à la représentation vectorielle du mot pour injecter de l'information sur la position du mot dans le texte. La position est représentée par un vecteur plus complexe que de seulement donner l'index du mot dans la phrase, les détails de l'implémentation ne sont pas présentés ici. Il devient donc possible de paralléliser l'entraînement et de passer chaque mot simultanément à travers le Transformer.

1.3.2. BERT

BERT (Bidirectional Encoder Representations from Transformers) est un modèle pré-entraîné qui a établi des résultats état de l'art sur de nombreuses tâches en TALN. Il constitue une légère modification d'un modèle Transformer. Un des concepts important du modèle est l'aspect bidirectionnel. Contrairement aux modèles neuronaux précédents qui lisaient le texte de manière séquentielle (de gauche à droite ou de droite à gauche). L'encodeur bidirectionnel de BERT permet d'apprendre le contexte d'un mot en prenant en compte tous les mots voisins (gauche et droite). Il est pré-entraîné en utilisant deux objectifs:

(1) *Masked Languge Model (MLM)*: Un certain nombre de mots du texte en entrée est masqué et le modèle essai de retrouver ces mots.

[1] http://jalammar.github.io/illustrated-transformer/

(2) *Next Sentence Prediction (NSP)*: Deux phrases sont données en entrée et le modèle doit essayer de prédire si l'une précède l'autre.

RoBERTa [**Liu et al., 2019**], une architecture semblable à BERT, réussi à améliorer légèrement les résultats de BERT. D'autres variations populaires de BERT actuellement sont DeBERTa [**He et al., 2020**] et ELECTRA [**Clark et al., 2020**].

1.3.3. SBERT

Un inconvénient important de BERT est qu'il n'y a pas de représentation vectorielle d'une phrase unique qui est calculée, ce qui le rend difficile à utiliser comme modèle pour la représentation vectorielle de phrases. Pour y remédier, plusieurs techniques ont été tentées, par exemple, de passer une phrase unique à travers le réseau et ensuite de faire la moyenne des représentations vectorielles de ses mots pour obtenir un vecteur de taille fixe ou bien d'utiliser la sortie du jeton spécial *CLS*. Pour des exemples, voir [**May et al., 2019, Zhang et al., 2019, Qiao et al., 2019**]. Au lieu d'entraîner un modèle de zéro pour la représentation vectorielle de phrases, les auteurs de SBERT [**Reimers and Gurevych, 2019**] proposent d'affiner le modèle BERT (ou RoBERTa). Ainsi, en moins de 20 minutes, un modèle SBERT peut être affiné sur des texters d'intérêt et offrir des résultats état de l'art en représentation vectorielle de phrases. Pour ce faire, une couche de *pooling* est ajoutée à la sortie. Trois stratégies de *pooling* sont explorées avec les vecteurs de sortie, soit de prendre ceux du jeton CLS, de faire la moyenne de tous les vecteurs ou de prendre le maximum de tous les vecteurs. De plus, pour mettre à jour les poids du modèle, afin de produire une représentation vectorielle incorporant la sémantique et permettant de comparer les phrases avec la similarité cosinus, les auteurs expérimentent avec un *siamese network* (deux instances du même modèles) et *triplet networks* [**Hoffer and Ailon, 2014**] (trois instances du même modèle) où les poids de chacune des instances sont liés. Chacun des modèles reçoit en entrée une phrase différente et l'objectif est d'ajuster les poids du modèle pour que les représentations de phrases similaires soient rapprochées, et celles de phrases non similaires soient éloignées.

1.4. Affinage de modèles Transformer

Les modèles de représentation vectorielle à base de Transformer sont pré-entraînés sur des données générales non spécifiques au domaine dont nous faisons le regroupement. Pour améliorer cette représentation, il est donc important d'affiner ces modèles sur nos données. Comme nous n'avons aucune information sur les données, nous devons utiliser une technique non supervisée. Actuellement, l'approche état de l'art, TSDAE [**Wang et al., 2021a**], repose sur un auto-encodeur. Le texte en entrée est bruité, c'est-à-dire certains mots sont masqués, puis le texte est encodé. Ensuite, le décodeur essaie de retrouver le texte d'origine (sans le

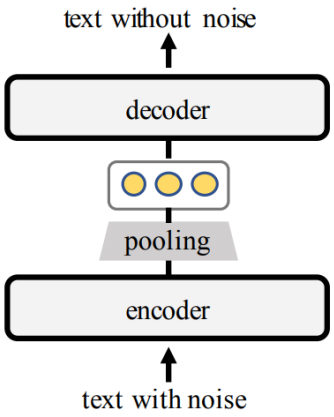


Fig. 1.4. Architecture TSDAE [**Wang et al., 2021a**].

bruit). À l'inférence, seulement l'encodeur est utilisé pour représenter les phrases en vecteurs. Une illustration de l'architecture est présentée à la figure 1.4. D'autres techniques ont été proposées avant TSDEA, principalement basées sur l'apprentissage contrastif (*constrastive learning* en anglais) et utilisant la même phrase ou une modification de la même phrase comme étant la phrase semblable et les autres phrases étant considérées différentes, par exemple SimCSE [**Gao et al., 2021**] et Contrastive Tension [**Carlsson et al., 2021**]. [**Li et al., 2020**] propose comme affinage de rendre la représentation vectorielle plus proche d'une distribution gaussienne.

1.5. Réduction de dimensionnalité

Le fléau de la dimensionnalité ou la malédiction de la dimension (*curse of dimensionnality* en anglais) est un phénomène bien connu en intelligence artificielle [**Bengio et al., 2006**], [**Cabannes et al., 2021**]. Des comportements observés en basse dimension, ne produisent pas les mêmes effets en haute dimension [**Domingos, 2012**]. Par exemple, en regroupement de texte, utiliser la distance euclidienne pour trouver des documents similaires peut bien fonctionner, mais en haute dimension les documents peuvent tous apparaître équidistants les uns des autres et donc il peut être difficile de les différencier. Une solution standard est d'essayer de réduire le nombre de dimensions [**Bouveyron et al., 2007**]. Concrètement, cela consiste à trouver des représentations de dimension inférieure qui conservent le plus possible l'information des données. Plusieurs techniques ont été proposées, celles-ci sont résumées en figure 1.5. La sélection de caractéristiques consiste à diminuer le nombre de dimensions en sélectionnant les dimensions les plus importantes et en éliminant celles qui sont redondantes. L'avantage principal est que l'espace dimensionnel est conservé alors qu'avec l'extraction de caractéristiques un nouvel espace dimensionnel plus petit est créé en appliquant une transformation linéaire ou non linéaire. Dans ce mémoire, nous allons

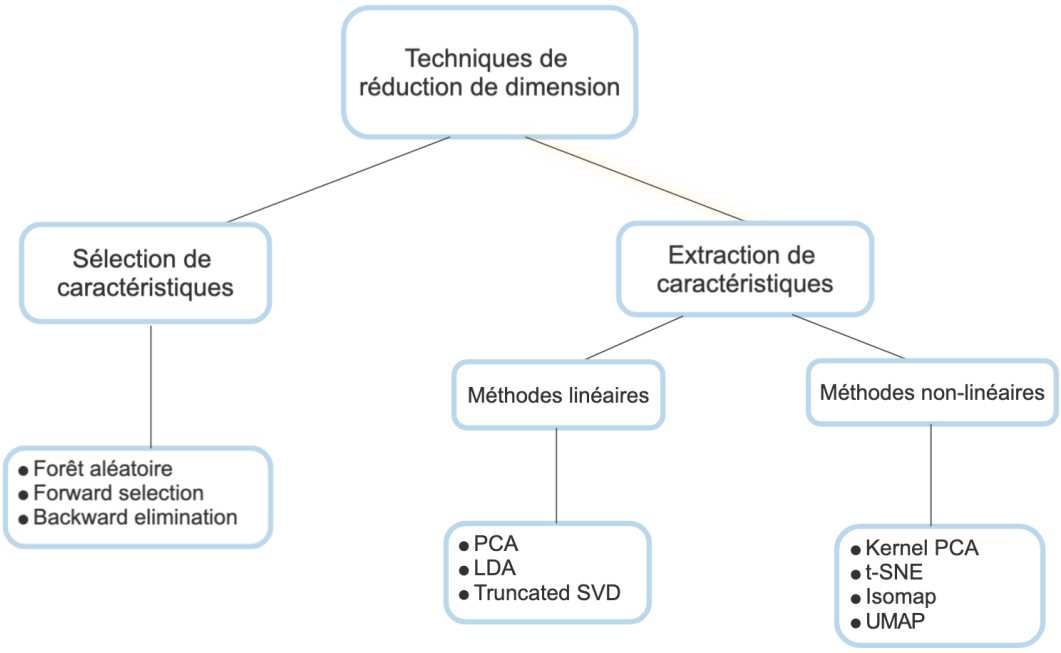


Fig. 1.5. Techniques de réduction de dimensionnalité

nous concentrer sur UMAP [**McInnes et al., 2018**], une des plus récentes techniques de réduction de dimensionnalité, offrant des résultats à l'état de l'art. Très semblable à t-SNE, UMAP conserve une meilleure structure globale des données et a un temps de calcul inférieur. C'est une technique basée sur l'analyse de variété (*Manifold learning* en anglais) utilisant les fonctions propres du Laplacien (*Laplacian eigenmaps* en anglais) et la géométrie riemannienne, pour plus de détails voir l'article de référence ou la documentation[2].

1.6. Algorithmes de regroupement

Les cinquante dernières années ont vu l'essor d'une multitude d'algorithmes de regroupement. De plus, pour plusieurs algorithmes de nombreuses variantes ont été proposées (par exemple, pour une analyse des variantes sur K-Means voir [**Garg and Jain, 2006**] et [**Blömer et al., 2016**]). Les algorithmes de regroupement se divisent principalement en 5 catégories [**Fahad et al., 2014**], soit les méthodes basées sur la hiérarchie, les méthodes basées sur la densité, les méthodes basées sur la grille, les méthodes de partitionnement et les méthodes basées sur des modèles statistiques. Pour une analyse détaillée des types d'algorithmes de regroupement voir [**Xu and Tian, 2015**] et [**Kotsiantis and Pintelas, 2004**]. Nous nous concentrerons sur deux algorithmes importants de regroupement: K-Means, le plus populaire, appartenant à la catégorie des algorithmes de partitionnement et HDBSCAN,

[2] `https://umap-learn.readthedocs.io/`

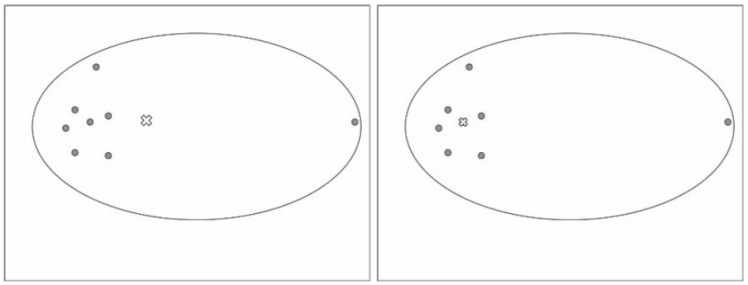


Fig. 1.6. Comparaison du centre de groupe. À la figure (a) K-Means est utilisé, on peut voir que le centre est tiré vers la droite à cause de la donnée aberrante. À la figure (b), grâce à K-Medoids, la donnée aberrante à un impact moindre et le centre du groupe semble plus représentatif.

.

basé sur la hiérarchie. Certaines propriétés intéressantes de HDBSCAN seront discutées par la suite.

1.6.1. K-Means

L'algorithme 1.6.1 présente K-Means. Celui-ci est d'une grande simplicité. L'initialisation est généralement aléatoire, mais certaines méthodes ont été proposées pour permettre une convergence plus rapide en effectuant une meilleure initialisation; voir [**Peña et al., 1999**]. Pour l'assignation des données à un groupe il s'agit de trouver le centre de groupe le plus proche, généralement à l'aide de la distance euclidienne. L'inconvénient de K-Means est qu'il est sensible à des données aberrantes qui peuvent attirer le centre du groupe. Pour y remédier, K-Medoids [**Jin and Han, 2010**] a été proposé, illustré à la figure 1.6. De plus, K-Means prend pour hypothèse une distribution sphérique des données dans l'espace latent.

Algorithme 1.6.1. K-Means

Entrée: Liste de données à regrouper, K : Nombre de groupes

Sortie: Liste de données regroupées en K groupes

Initialisation : Choisir K données comme le centre initial des groupes.

Faire

- Assigner chaque donnée au groupe auquel elle est le plus similaire.
- Recalculer le centre de chaque groupe.

Tant que le critère de convergence n'est pas atteint

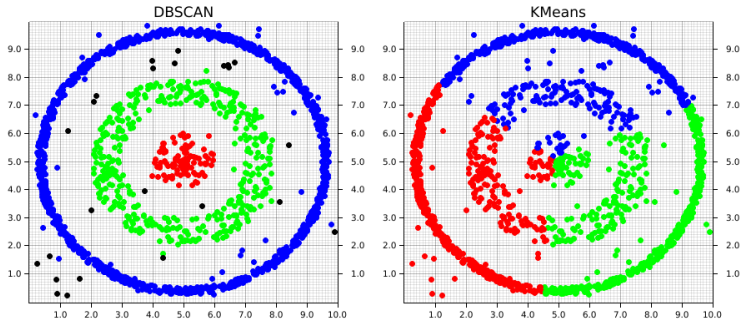


Fig. 1.7. Comparaison entre un regroupement DBSCAN et K-Means. Nous pouvons remarque que K-Means a de la difficulté à faire un regroupement lorsque la forme des groupes n'est pas simple.

1.6.2. HDBSCAN

HDBSCAN [**Campello et al., 2013**] est une extension de DBSCAN [**Ester et al., 1996b**] afin de convertir l'algorithme en un algorithme hiérarchique. Tout d'abord, DBSCAN est un algorithme de densité. L'intérêt de ce type d'algorithme est qu'il peut regrouper des données représentées dans des groupes de formes spéciales et non uniformes. Aussi, il présente une certaine robustesse par rapport aux données aberrantes. Une comparaison entre K-Means et DBSCAN est présentée à la figure 1.7. Une explication détaillée de l'algorithme est présentée sur le site internet implémentant l'algorithme[3], celle-ci peut être résumée en trois étapes:

(1) Estimer les densités en calculant la distance de chacun des points avec ses plus proches voisins.
(2) Choisir les régions de haute densité.
(3) Combiner les points pour former un arbre. Chaque région de haute densité est un sous-arbre et peut être considérée comme un groupe.

Cet algorithme est intéressant, car il ne nécessite pas de connaissance au préalable sur les données, le nombre de groupes n'a pas à être fourni. Celui-ci est trouvé en fonction du nombre de régions à haute densité. De plus, en étant hiérarchique, il est possible de choisir à quel niveau couper l'arbre pour avoir un regroupement à différents niveaux de détails (regroupement très général en peu de groupes ou très précis avec un nombre de groupes élevé). Enfin, il permet de détecter les points aberrants, ceux-ci étant ceux qui se trouvent à l'extérieur des régions de haute densité.

[3] `https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html`

1.7. Modèles neuronaux de regroupement de textes

Le modèle état de l'art actuellement en regroupement de textes avec le nombre de groupes connu au préalable est SCCL [**Zhang et al., 2021**], celui-ci sera décrit au chapitre 4 lorsque nous tentons de reproduire ses performances. Auparavant, d'autres approches ont été tentées. Par exemple, [**Xu et al., 2017b**] propose tout d'abord d'utiliser SIF, un modèle de représentation vectorielle des phrases utilisant *FastText*, *PCA* et une moyenne pondérée de la représentation de chaque mot [**Arora et al., 2017b**]. Ensuite, un auto-encodeur est entraîné sur cette représentation vectorielle. Puis, les poids de l'encodeur sont affinés en utilisant le centre de chacun des regroupements. Pour sa part, [**Rakib et al., 2020a**], propose un regroupement itératif. Tout d'abord, un premier regroupement est effectué. Puis, une détection des données aberrantes est appliquée et ces données sont retirées afin d'entraîner un classificateur sur les données restantes. Enfin, en utilisant le modèle entraîné, les données aberrantes sont classifiées. Ce processus est répété de manière itérative jusqu'à ce qu'un critère d'arrêt soit atteint.

Pour ce qui est du regroupement avec le nombre de groupes non connu au préalable, l'état de l'art est DeepDPM [**Ronen et al., 2022**]. Cette méthode s'appuie sur les mélanges de processus de Dirichlet (DPM) (*Mixtures of Dirichlet processes* en anglais) [**Antoniak, 1974**]. Auparavant, [**Wang et al., 2021b**] a proposé aussi d'utiliser un modèle DPM et [**Chen, 2015**] d'utiliser un auto-encodeur, mais leurs modèles ont de la difficulté à gérer de grands corpus.

Chapitre 2

Méthodes d'évaluation d'un regroupement

Évaluer un regroupement s'avère plus complexe que l'évaluation en apprentissage supervisé, où pour la classification par exemple, il suffit de calculer la précision ou le rappel. Il s'agit d'une tâche difficile et les meilleures métriques à utiliser restent inconnues [**Aggarwal and Reddy, 2013**]. Comment évaluer un regroupement? Qu'est-ce qui distingue un bon regroupement d'un mauvais? Quelles sont les particularités lorsque le nombre de groupes n'est pas connu au préalable? Ce sont toutes des questions qui restent à éclaircir. Autant que nous le sachions, il n'y a pas de recherche récente qui clarifie ces questions. Il existe deux catégories de métriques d'évaluation d'un regroupement; l'évaluation externe ou interne. La principale différence réside dans la présence ou non d'informations externes pour aider à l'évaluation comme par exemple, un regroupement de référence pour chacune des données. Une analyse d'une dizaine de métriques pour chacune des catégories est présentée dans [**Aggarwal and Reddy, 2013**]. Dans ce mémoire nous nous concentrons sur les méthodes d'évaluations externes. Les plus populaires sont la précision (ACC), l'information mutuelle normalisée (NMI) et l'indice de Rand ajusté (ARI). Dans la littérature, ces métriques semblent principalement être choisies à des fins de comparaison avec les recherches précédentes. Dans un premier temps, nous décrivons ces trois métriques. Suivi d'une analyse de ces métriques avec une attention particulière sur l'effet de l'uniformité de K-Means. Finalement, nous faisons une proposition de métriques à privilégier.

2.1. Principales métriques d'évaluation externes

Comme mentionné précédemment, les principales métriques d'évaluation externes en regroupement de texte sont la précision (ACC), l'information mutuelle normalisée (NMI) et l'indice de Rand ajusté (ARI). Avant de décrire ces métriques, définissons tout d'abord :

k : nombre de groupes présents dans le corpus.

k' : nombre de groupes dans lesquels le regroupement est effectué. Si k est fourni au préalable à l'algorithme de regroupement, alors $k' = k$.

C : $\{c_i\}_{i\in[1,n]} \in [1,k]$, l'assignation de référence de chaque donnée du corpus à un groupe. C'est-à-dire, l'information externe que nous possédons sur le corpus quant à l'appartenance de chaque donnée à un groupe connu au préalable.

C' : $\{c'_i\}_{i\in[1,n]} \in [1,k']$, le résultat de l'algorithme de regroupement, l'assignation de chaque donnée du corpus à un groupe.

G : $\{G_i\}_{i\in[1,k]}$, sous-ensembles de données du corpus qui sont associées au groupe i par le regroupement de référence.

G' : $\{G'_j\}_{j\in[1,k']}$, sous-ensembles de données du corpus qui sont associées au groupe j obtenu par l'algorithme de regroupement.

n : taille du corpus.

$n_{i\cdot}$: $|G_i|$, taille du sous-ensemble G_i; le nombre de données du corpus qui sont associées au groupe i par le regroupement de référence.

$n_{\cdot j}$: $|G'_j|$, taille du sous-ensemble G'_j; le nombre de données du corpus qui sont associées au groupe j obtenu par l'algorithme de regroupement.

n_{ij} : $|G_i \cap G'_j|$, la taille de l'intersection de G_i et G'_j; le nombre de données appartenant au groupe i parmi le regroupement de référence et appartenant au groupe j parmi le regroupement obtenu par l'algorithme de regroupement.

Pour établir la qualité d'un regroupement, les métriques tentent principalement de comparer C et C'.

2.1.1. Précision (ACC)

La précision (*accuracy* en anglais) est définie comme l'erreur de classification. Pour calculer cette erreur nous devons tout d'abord faire un alignement entre C et C'. En effet, comme nous n'avons pas d'associations entre les groupes de références [1,..,k] et les groupes du regroupement [1,..,k'], nous devons trouver l'alignement le plus probable de [1,..,k] à [1,...,k'], c'est à dire, celui qui minimise l'erreur de classification. Cette opération est effectuée généralement à l'aide de l'algorithme hongrois [**Steiglitz, 1982**]. Ensuite, il est possible de calculer la précision en divisant le nombre d'éléments bien classifiés sur le nombre total d'éléments. Plus formellement :

$$ACC = \frac{\sum_{i=1}^{n} \delta(c_i, map(c'_i))}{n}, \tag{2.1.1}$$

où:

$\delta(c,c')$: est une fonction qui retourne 1 si c et c' sont égaux et 0 sinon.

map() : est une fonction d'alignement qui associe à chaque valeur de $c'_x \in [1,k']$ une valeur $\in [1,k]$.

Il est à noter que l'algorithme hongrois résout un problème d'affectation balancé, c'est-à-dire, si l'on représente le problème à l'aide d'un graphe biparti, les deux sous-ensembles ont

la même taille. Ainsi, dans le cas où k n'est pas connu au préalable et que donc, k n'est pas nécessairement égal à k', il est important de s'assurer d'utiliser un algorithme d'alignement non balancé.

2.1.2. L'information mutuelle normalisée (NMI)

L'information mutuelle normalisée (*Normalized Mutual Information* en anglais) mesure l'information partagée entre C et C':

$$NMI(C,C') = \frac{MI(C,C')}{\sqrt{H(C)H(C')}}, \tag{2.1.2}$$

où *MI* est l'information mutuelle et *H* l'entropie. Le dénominateur permet de normaliser l'information mutuelle et ainsi la fixer dans un intervalle [0,1]. Lorsque les données sont groupées parfaitement, le NMI a une valeur de 1 et, lorsque C et C' sont indépendants, une valeur de 0. En pratique, le NMI de l'équation 2.1.2 est approximée avec l'équation suivante [**Chen et al., 2011, Wu and Schölkopf, 2006, Strehl and Ghosh, 2002**]:

$$NMI = \frac{\sum_{i=1}^{k}\sum_{j=1}^{k'} n_{i,j} log(\frac{n \cdot n_{i,j}}{n_{i\cdot} \cdot n_{\cdot j}})}{\sqrt{(\sum_{i=1}^{k} n_{i\cdot} log\frac{n_{i\cdot}}{n})(\sum_{j=1}^{k'} n_{i,j} n_{\cdot j} log\frac{n_{\cdot j}}{n})}}, \tag{2.1.3}$$

2.1.3. L'indice de Rand ajusté (ARI)

Un peu moins populaire que les deux dernières métriques, mais tout de même utilisé dans quelques recherches, telles que [**Li et al., 2018**] et [**Guan et al., 2020**], l'ARI (*Rand index adjusted for chance* en anglais) [**Hubert and Arabie, 1985**] est l'indice de Rand (RI) [**Rand, 1971**] ajusté pour la chance. L'ajustement pour la chance permet de donner à un regroupement aléatoire un score de 0, tandis qu'un regroupement parfait obtient un score de 1. Le RI est une métrique qui compare les paires de données afin de déterminer le taux d'accord entre deux partitions (C et C' dans notre cas). Pour chaque paire de données du corpus, on regarde si les deux éléments sont assignés à un même groupe ou non dans C et également dans C'. Contrairement à la précision, l'indice de Rand n'a pas besoin d'alignement. Il correspond à:

$$RI = \frac{a+b}{N}, \tag{2.1.4}$$

où:

- a : Le nombre de paires d'éléments du corpus qui appartiennent conjointement à même groupe dans C et C' respectivement.
- b : Le nombre de paires d'éléments du corpus qui appartiennent conjointement à des groupes différents dans C et C' respectivement.
- N : n(n-1)/2, Le nombre total de paires possibles, où n est le nombre de données.

	G_1	G_2	...	G_k	Somme
G'_1	n_{11}	n_{12}	...	n_{1k}	$n_{1.}$
G'_2	n_{21}	n_{22}	...	n_{2k}	$n_{2.}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$G'_{k'}$	$n_{k'1}$	$n_{k'2}$	...	$n_{k'k}$	$n_{k'.}$
Somme	$n_{.1}$	$n_{.2}$	...	$n_{.k}$	n

Tableau 2.1. Table de contingence pour comparer deux partitions.

Ensuite, afin d'ajuster pour la chance, il suffit de soustraire l'espérance d'un RI aléatoire :

$$ARI = \frac{RI - E[RI]}{max(RI) - E[RI]} \tag{2.1.5}$$

Pour calculer l'ARI une matrice de contingence, telle que présentée dans le tableau 2.1 est utilisée. En assumant une distribution généralisée hypergéométrique comme modèle aléatoire, il est possible de démontrer [**Hubert and Arabie, 1985**] que $a + b$ de l'équation 2.1.4 est équivalent à $\sum_{i,j} \binom{n_{ij}}{2}$ et que:

$$E\left[\sum_{i,j}\binom{n_{ij}}{2}\right] = \left[\sum_{i}\binom{n_{i.}}{2}\sum_{j}\binom{n_{.j}}{2}\right]/\binom{n}{2} \tag{2.1.6}$$

Pour ainsi obtenir l'équation finale, après simplifications algébriques :

$$ARI = \frac{\sum_{i,j}\binom{n_{ij}}{2} - \left[\sum_{i}\binom{n_{i.}}{2}\sum_{j}\binom{n_{.j}}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_{i}\binom{n_{i.}}{2} + \sum_{j}\binom{n_{.j}}{2}\right] - \left[\sum_{i}\binom{n_{i.}}{2}\sum_{j}\binom{n_{.j}}{2}\right]/\binom{n}{2}} \tag{2.1.7}$$

2.2. L'uniformité de K-Means

	G_1	G_2	G_3	G_4	G_5
G'_1	10	0	0	0	0
G'_2	10	0	0	0	0
G'_3	10	0	0	0	0
G'_4	0	0	0	10	0
G'_5	0	2	6	0	2

Regroupement 1

	G_1	G_2	G_3	G_4	G_5
G'_1	27	0	0	2	0
G'_2	0	2	0	0	0
G'_3	0	0	6	0	0
G'_4	3	0	0	8	0
G'_5	0	0	0	0	2

Regroupement 2

Tableau 2.2. Tableaux de contingence de deux regroupements. Corpus de 50 documents groupés en 5 groupes. La taille des groupes est 30, 2, 6, 10 et 2. CV = 1.166, ce qui implique un corpus déséquilibré. Exemples tirés de [**Aggarwal and Reddy, 2013**].

Un des phénomènes bien connus de K-Means est qu'il tend à produire des groupes de tailles uniformes [**Xiong et al., 2009**]. Pour comprendre ce phénomène, il est possible

Groupement	ACC	NMI	ARI
1	48	62	33
2	90	70	75

Tableau 2.3. Résultat des regroupements présentés au tableau 2.2

d'utiliser le coefficient de variation (CV), une statistique qui mesure le degré de dispersion d'une distribution aléatoire. CV est défini comme le ratio entre l'écart-type et la moyenne. Contrairement à l'écart-type, c'est un nombre sans unité, ce qui en fait une métrique intéressante pour comparer deux distributions avec des moyennes différentes. En général, plus la valeur de CV est grande, plus les données sont dispersées. L'exemple au tableau 2.2 illustre l'uniformité de K-Means. Dans le regroupement 1, cinq groupes de tailles égales sont présents, CV = 0. Par contre, dans le regroupement 2, les cinq groupes ont des tailles différentes avec un CV = 1.125, ce qui est beaucoup plus proche de la vraie distribution des données. En effet, le premier regroupement a mis les trois plus petits groupes dans le même groupe et nous pouvons ainsi observer le phénomène d'uniformité de K-Means, alors que dans le deuxième, il est facile d'associer chacun des groupes aux groupes de référence. Ainsi, il est possible de conclure que le regroupement 1 est de moindre qualité que le regroupement 2. En outre, [**Wu et al., 2007**] a démontré empiriquement avec un intervalle de confiance de 95% que K-Means tend à produire un regroupement avec un CV borné par [0.09, 0.85]. Ainsi, pour des corpus avec une valeur de CV supérieure à 0.85, l'effet d'uniformité risque d'être présent. Appliquer K-Means sur des corpus déséquilibrés risque donc de ne pas produire un regroupement efficace. Compte tenu de ce phénomène, il semble essentiel qu'une bonne métrique d'évaluation soit sensible à l'uniformité du regroupement afin que les résultats reflètent la qualité réelle du regroupement. Au tableau 2.3, on peut voir le résultat des différentes métriques pour chacun des deux regroupements du tableau 2.2. On remarque que toutes les métriques semblent tenir compte de l'uniformité, avec un score plus élevé pour le deuxième regroupement, représentant mieux la distribution d'origine. Par contre, la différence est moins marquée pour le NMI où il y a seulement 8 points de différence entre les deux regroupements alors que les autres métriques sont autour du double pour le deuxième regroupement.

2.3. Recommandation de métriques à utiliser

Dans les travaux récents en regroupement de texte [**Zhang et al., 2021, Rakib et al., 2020b, Xu et al., 2017b, Hadifar et al., 2019**] les auteurs justifient l'usage de certaines métriques d'évaluation en citant d'anciens travaux [**Cai et al., 2005, Xu et al., 2003, Huang et al., 2014, Chen et al., 2011, Strehl and Ghosh, 2002, Wu and Schölkopf, 2006**], et ne font pas leur propre analyse. En examinant ces travaux cités, ces

derniers n'ont guère fait plus d'élaboration à ce sujet. Par ailleurs, la plupart des recherches utilisent l'algorithme K-Means. Ces éléments nous amènent à vouloir faire une investigation plus poussée de ces métriques et valider qu'elles sont appropriées pour K-Means. Avec notre exemple du tableau 2.2 les métriques semblent bien refléter l'uniformité de K-Means. Par contre, les auteurs de [**Aggarwal and Reddy, 2013**] effectuent une expérience plus poussée. En effet, ils simulent plusieurs corpus synthétiques issus d'un mélange de deux lois gaussiennes avec différents niveaux de déséquilibre. Aussi, ils échantillonnent un corpus de nouvelles de journaux avec différents niveaux de déséquilibre. Plus le niveau de déséquilibre augmente, plus la différence du coefficient de variation (DCV) entre le regroupement de référence et le regroupement K-Means augmente, et ce, en raison de l'uniformité de K-Means. Le résultat attendu est alors qu'une bonne métrique sera corrélée avec le DCV. C'est d'ailleurs ce que les auteurs observent pour le NMI et l'ARI, mais ce n'est pas le cas pour l'ACC. Même en essayant de normaliser l'ACC, celle-ci n'est pas corrélée avec le DCV. Nous pouvons en conclure que l'ACC est un mauvais choix de métrique d'évaluation de regroupement puisqu'elle pourrait ne pas refléter une différence de distribution pour des corpus déséquilibrés. De plus, les auteurs affirment que le NMI et l'ARI respectent d'importantes propriétés mathématiques. Ces deux métriques sont donc à privilégier, alors que l'ACC devrait être abandonnée.

En conclusion, après avoir décrit les principales métriques d'évaluation, nous avons souligné l'importance de prendre en compte l'uniformité de K-Means et avons redécouvert les travaux oubliés de [**Aggarwal and Reddy, 2013**], qui n'étaient pas pris en comptes dans le choix des métriques d'évaluation par les travaux récents. Nous recommandons d'utiliser le NMI et l'ARI comme métriques d'évaluation.

Chapitre 3

Données utilisées

Nous décrivons dans ce chapitre les corpus utilisés. Nous effectuons nos expériences sur huit corpus employés dans les recherches récentes. Le tableau 3.1 présente les détails de chacun des corpus. Nous observons une bonne variété du type de données, soit des données issues de nouvelles, de recherches Google, des données plus techniques telles que dans le domaine informatique ou médical et des données de médias sociaux. En regardant la longueur moyenne des textes, nous remarquons que ce sont des textes très courts. Une explication possible de l'intérêt d'utiliser des textes courts est qu'intuitivement ne n'avons pas besoin d'un texte complet pour déterminer son sujet ou pour le grouper avec des textes semblables. Généralement, le titre ou l'introduction suffisent pour des textes tels que les nouvelles, les articles scientifiques ou les publications sur les médias sociaux. Il serait cependant intéressant de comparer la performance des techniques présentées dans ce mémoire avec des textes plus longs. Les corpus peuvent être téléchargés depuis le répertoire GitHub[1] de [**Rakib et al., 2020b**]. Cependant, pour avoir AgNews sans prétraitement, nous l'avons téléchargé depuis le site de PyTorch[2].

[1] `https://github.com/rashadulrakib/short-text-clustering-enhancement/tree/master/data`
[2] `https://pytorch.org/text/stable/datasets.html#ag-news`

Corpus	Vocabulaire	Textes		Groupes		
		Total	Longueur moyenne	K	CV	Plus grand/ plus petit
AgNews	21 000	8 000	23	4	0,00	1
StackOverflow	15 000	20 000	8	20	0,00	1
Biomedical	19 000	20 000	12	20	0,00	1
SearchSnippets	31 000	12 340	18	8	0,44	7
GoogleNewsTS	20 000	11 109	28	152	1,04	143
GoogleNewsS	18 000	11 109	22	152	1,04	143
GoogleNewsT	8 000	11 109	6	152	1,04	143
Tweets	5 000	2 472	8	89	1,57	249

Tableau 3.1. Description des corpus. Vocabulaire: nombre de mots uniques par corpus. Total: nombre de données par corpus. Longueur moyenne: le nombre moyen de mots par texte. K: nombre de groupes. CV: coefficient de variation. Plus grand/plus petit: ratio de la taille du grand groupe sur le plus petit (un ratio de 1 signifie que tous les groupes ont la même taille).

3.1. AgNews

Label	data
4	Toshiba unleashes slew of notebooks The wireless-enabled systems target all levels of businesses; at least one model is designed to handle intense graphic presentations. \
4	Cingular completes AT T Wireless acquisition Following approval from two U.S. government agencies, Cingul…
1	Hurricane Jeanne Takes Aim at Florida WEST PALM BEACH, Fla. - Hurricane Jeanne trekked westward Friday on…
3	Air NZ fears for future after Qantas bid fails AIR New Zealand today claimed it faced an uncertain future after the countrys High Court threw out plans for a tie-up with Qantas, its Australian-based rival.
1	Darfur Rebels Say Peace Talks on Last Legs ABUJA (Reuters) - Darfur rebels said on Thursday that peace talks with the Sudan government had reached a stalemate and were on the verge of collapse.
1	Putin plan for political overhaul easily passes its first vote MOSCOW Russia #39;s lower house of Parliam…
4	CA updates Unicenter offerings Computer Associates Monday announced the general availability of three Unicenter performance management products for mainframe data management.
1	Pak says India paranoid on F-16 Islamabad: Pakistan on Monday accused India of being quot;paranoid quot;…
3	Nintendo, Sony devices to target holiday shoppers NEW YORK -- Nintendo Co. and Sony Corp. made separate p…
2	Benson Leads Texas to 44-14 Win Over Baylor AUSTIN, Texas (Sports Network) - Cedric Benson ran for 188 …

Fig. 3.1. Échantillon des données présentes dans le corpus AgNews.

AgNews est un corpus de nouvelles publiées sur le site internet de 2 000 sources de nouvelles [3]. Il contient plus d'un million de nouvelles dont près de la moitié est classifiée. [**Zhang et al., 2015**] y sélectionne le titre et la description de 30 000 nouvelles distribuées également parmi les quatre catégories les plus populaires (Monde, Sports, Affaires, Science/Technologie). Ensuite, en suivant l'approche employée par [**Zhang et al., 2021**] et [**Rakib et al., 2020b**], nous sélectionnons aléatoirement 8 000 nouvelles parmi les 30 000 nouvelles (2 000 nouvelles par catégorie). Un échantillon des données est présenté à la figure 3.1.

[3] `http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html`

SearchSnippets - Sujet	Nb. de questions	Nb. de données
Affaire	70	1500
Informatique	70	1500
Culture, arts et divertissement	105	2210
Éducation, science	128	2660
Ingénierie	16	370
Santé	54	1180
Politique, société	70	1500
Sports	66	1420
Total	579	12340

Tableau 3.2. Sujets représentant chacun des groupes de **SearchSnippets**. Pour chaque sujet, un certain nombre de questions est rentré dans le moteur de recherche et 20 ou 30 extraits de résultats sont sélectionnés par question pour former le corpus.

3.2. SearchSnippets

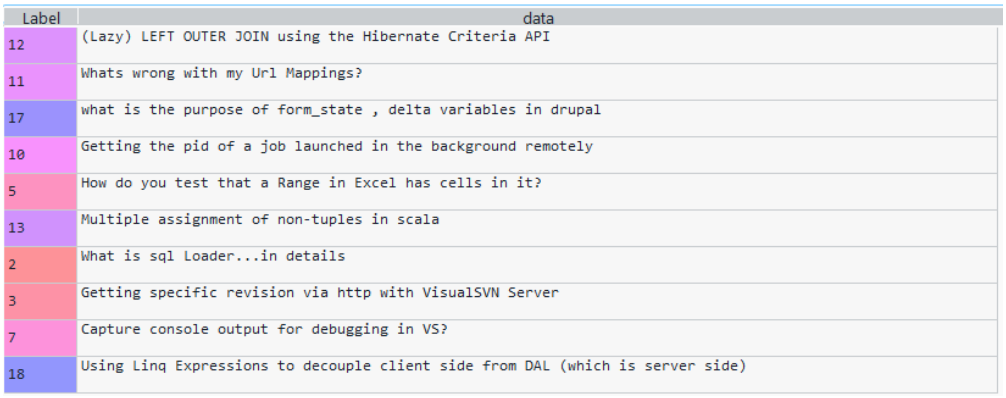
Label	data
1	inflationdata inflation inflation rate currentinflation inflation inflation updated monthly inflation web…
8	chicagobears home page chicago bears team audio video clips team news depth charts transactions statistic…
8	healthclubdirectory health club directory health club fitness center health club fitness center hour gym …
6	bls gov oco ocos pharmacists pharmacists oversee pharmacy students serving interns pharmacists pursuing n…
7	bls gov oco cgs federal government excluding postal service federal government duties defending united fo…
7	wikipedia wiki communist party communist party wikipedia encyclopedia modern usage communist party politi…
4	plato stanford edu entries descartes epistemology descartes epistemology stanford encyclopedia philosophy…
8	nfl gamecenter gamebook nfl atl sea nfl com gamebook game weather sunny played outdoor turf fieldturf tem…
4	amazon information theory robert ash amazon com information theory books ash amazon com information theor…
4	physics udel edu bnikolic teaching phys lectures introduction solid physics solid physics crystalline sol…

Fig. 3.2. Échantillon des données présentes dans le corpus SearchSnippets.

SearchSnippets [**Phan et al., 2008**] est constitué de 12 340 extraits de résultats de recherches Google distribués parmi 8 groupes. Pour chaque recherche un certain nombre d'extraits de résultats est sélectionné, le tableau 3.2 fourni les détails. Un échantillon des données est présenté à la figure 3.2.

3.3. StackOverflow

StackOverflow est une sélection faite par [**Xu et al., 2017a**] parmi 20 mots-clés de 20 000 titres de questions sur StackOverflow publiées entre le 31 juillet 2012 et le 14 août 2012. Le tableau 3.3 illustre les différents mots-clés. Un échantillon des données est présenté à la figure 3.3.

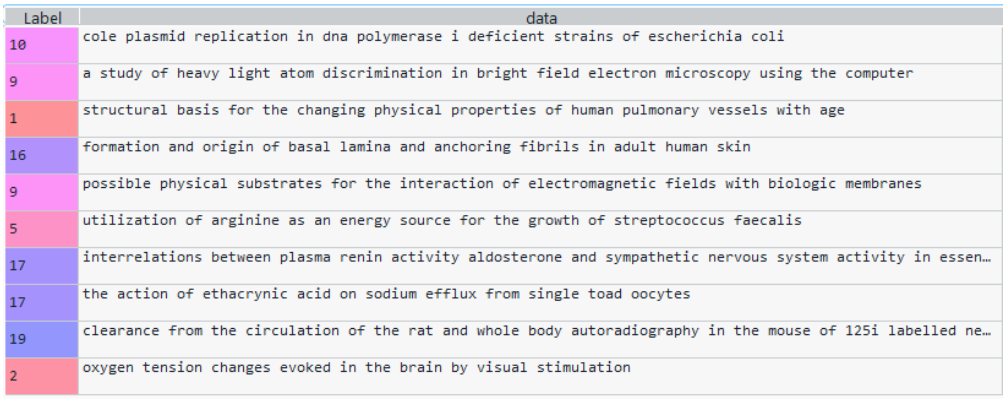


Label	data
12	(Lazy) LEFT OUTER JOIN using the Hibernate Criteria API
11	Whats wrong with my Url Mappings?
17	what is the purpose of form_state , delta variables in drupal
10	Getting the pid of a job launched in the background remotely
5	How do you test that a Range in Excel has cells in it?
13	Multiple assignment of non-tuples in scala
2	What is sql Loader...in details
3	Getting specific revision via http with VisualSVN Server
7	Capture console output for debugging in VS?
18	Using Linq Expressions to decouple client side from DAL (which is server side)

Fig. 3.3. Échantillon des données présentes dans le corpus StackOverflow.

StackOverflow
svn, oracle, bash, apache, excel, matlab, cocoa, visual-studio, osx, wordpress, spring, hibernate, scala, sharepoint, ajax, drupal, qt, haskell, linq, magento
Biomedical
aging, chemistry, cats, erythrocytes, glucose, potassium, lung, lymphocytes, spleen, mutation, skin, norepinephrine, insulin, prognosis, risk, myocardium, sodium, mathematics, swine, temperature

Tableau 3.3. Mots-clés représentant chacun des groupes de **StackOverflow** et sujets représentant chacun des groupes de **Biomedical**.



Label	data
10	cole plasmid replication in dna polymerase i deficient strains of escherichia coli
9	a study of heavy light atom discrimination in bright field electron microscopy using the computer
1	structural basis for the changing physical properties of human pulmonary vessels with age
16	formation and origin of basal lamina and anchoring fibrils in adult human skin
9	possible physical substrates for the interaction of electromagnetic fields with biologic membranes
5	utilization of arginine as an energy source for the growth of streptococcus faecalis
17	interrelations between plasma renin activity aldosterone and sympathetic nervous system activity in essen…
17	the action of ethacrynic acid on sodium efflux from single toad oocytes
19	clearance from the circulation of the rat and whole body autoradiography in the mouse of 125i labelled ne…
2	oxygen tension changes evoked in the brain by visual stimulation

Fig. 3.4. Échantillon des données présentes dans le corpus Biomedical.

3.4. Biomedical

Biomedical est un corpus publié sur BioASQ [4] où 20 000 titres de publications scientifiques ont été sélectionnés de manière aléatoire par [**Xu et al., 2017a**] parmis les 20 groupes présentés au tableau 3.3. Un échantillon des données est présenté à la figure 3.4.

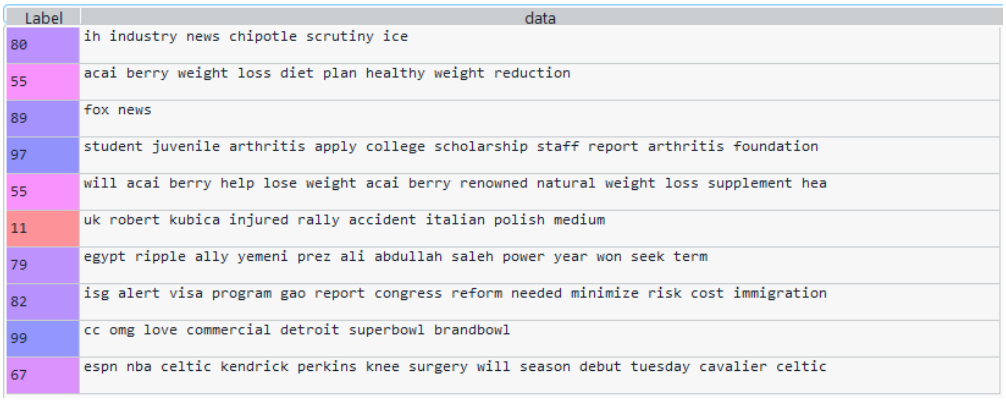
[4] `http://participants-area.bioasq.org/`

3.5. Tweets

Tweets est composé de 2572 tweets regroupés manuellement en 89 catégories en 2011 et 2012 à *Text REtrieval Conference*[5]. Le corpus est publié par [**Yin and Wang, 2016**]. Le tableau 3.4 montre un extraits des sujets formant les groupes. Un échantillon des données est présenté à la figure 3.5.

Tweets - Sujet	
river boat cruises	berries and weight loss
British Government cuts	Bedbug epidemic
Chicago blizzard	FDA approval of drugs
2022 FIFA soccer	BBC World Service staff cuts
Pakistan diplomat arrest murder	NIST computer security

Tableau 3.4. Échantillon des sujets de 10 groupes parmi les 89 sujets du corpus **Tweets**.



Label	data
80	ih industry news chipotle scrutiny ice
55	acai berry weight loss diet plan healthy weight reduction
89	fox news
97	student juvenile arthritis apply college scholarship staff report arthritis foundation
55	will acai berry help lose weight acai berry renowned natural weight loss supplement hea
11	uk robert kubica injured rally accident italian polish medium
79	egypt ripple ally yemeni prez ali abdullah saleh power year won seek term
82	isg alert visa program gao report congress reform needed minimize risk cost immigration
99	cc omg love commercial detroit superbowl brandbowl
67	espn nba celtic kendrick perkins knee surgery will season debut tuesday cavalier celtic

Fig. 3.5. Échantillon des données présentes dans le corpus Tweets.

3.6. Les corpus GoogleNews

GoogleNews est un corpus de titres et descriptions de 11 109 nouvelles regroupées en 152 événements extraits des nouvelles Google[6]. Ce corpus est ensuite divisé en trois sous-corpus: **T** (titre), **S** (description) et **TS** (titre et description). Le corpus est extrait automatiquement par [**Yin and Wang, 2014**], puis analysé manuellement pour valider la qualité des regroupements. Un échantillon des données est présenté aux figures 3.6, 3.7 et 3.8.

[5] `https://trec.nist.gov/data/microblog.html`
[6] `https://news.google.com/`

Label	data
101	storm upending holiday travel york cbsnewyork ap wall storm upending holiday travel plan american road sk…
31	coalition compromise long time coalition agreement includes bit party involved wanted call agreement admi…
66	bryant ink year extension lakers los angeles kobe bryant signed year contract extension los angeles laker…
46	pope francis decries trickle economics pope francis prays vigil saint peter square vatican september upi …
4	macy thanksgiving day parade coming side controversy cnn year magic kick thursday morning macy thanksgivi…
111	bizarre fire ant raft survive constant flood behaves solid liquid red raft fire ant describes unusual phy…
39	couple jailed life slitting throat teenage daughter servant rajesh nupur talwar accused killing teenager …
126	stock future edge higher york stock future edged higher extending nasdaq composite time year blue chip st…
44	google glass half empty half full greg swan google glass view headshot thursday october swan variety adop…
98	glaxosmithkline plc adr gsk glaxosmithkline growth glaxosmithkline gsk strong vaccine portfolio includes …

Fig. 3.6. Échantillon des données présentes dans le corpus GoogleNewsTS.

Label	data
7	finnish tech company nokia nyse nok aim apple nasdaq aapl ipad air commercial touted advantage released l…
33	summer france warned ally al qaeda root northern mali establish base jihadist extremism north africa warn…
4	woman wait flight phoenix sky harbor airport phoenix tuesday morning nov winter storm moved east coast na…
49	dekalb fleck recruiting jordan lynch northern illinois assistant lynch high school coach sold blue collar…
20	comet ison minute exposure nasa marshall space flight center msfc nov time picture comet ison mile earth …
1	duke cambridge prince pop tuesday november sang stage taylor swift jon bon jovi special charity event ken…
86	sliding shirtless motor oil bad guy fire hose rolling car windshield dude head count jason statham kickin…
58	updated wednesday november video article browser support javascript flash player currently disabled brows…
13	rebecca brown confessed online shopping addict year wellsville mo sitting front computer thanksgiving nig…
101	winter storm warning canceled morning snow time time today limited lake snow today tonight temperature mi…

Fig. 3.7. Échantillon des données présentes dans le corpus GoogleNewsS.

Label	data
76	xbox user banned swearing
131	winged robot fly jellyfish
73	love hormone work magic
68	blood improves redeem oldboy
65	kanye west kim kardashian beautiful woman time
65	khloe kardashian baby north kanye west
103	russia police moscow detain suspected member islamic extremist
83	euro jump asia ecb easing speculation
58	lostprophets singer ian watkins pleads gui mu iple child sex crime
141	keeping thanksgiving

Fig. 3.8. Échantillon des données présentes dans le corpus GoogleNewsT.

Chapitre 4

Reproduction de l'état de l'art

Dans ce chapitre nous tentons de reproduire les résultats du modèle état de l'art. Nous décrivons tout d'abord le modèle. Puis présentons nos résultats et les analysons. Les auteurs fournissent tous les détails d'implémentation en annexe dans [**Zhang et al., 2021**] et le code sur GitHub[1].

4.1. Description du modèle

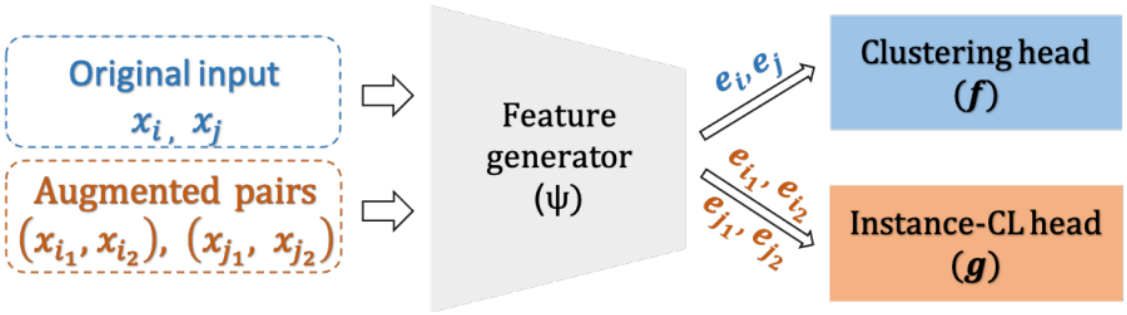


Fig. 4.1. Description de la configuration du modèle SCCL. Figure tirée de [**Zhang et al., 2021**].

SCCL est un modèle d'apprentissage contrastif (*Constrastive learning* en anglais). Pour effectuer l'apprentissage constrastif, des données similaires sont créées en substituant certains mots à leurs synonymes [**Ma, 2019**]. Concrètement, certains mots sont substitués dans le texte d'origine pour créer une nouvelle phrase semblable à celle d'origine et c'est cette nouvelle phrase qui est utilisée comme un exemple positif, le reste des phrases étant des exemples négatifs. L'objectif est de rapprocher la représentation vectorielle des phrases positives et d'éloigner celles étant négatives. Ainsi, la représentation vectorielle sera affinée au corpus à regrouper. Une illustration du modèle est présentée à la figure 4.1.

[1] `https://github.com/amazon-research/sccl`

Plus précisément, l'entrée du modèle est un corpus et deux copies augmentées du corpus. Chaque copie du corpus est une variation de la donnée originale par substitution de certains mots avec des mots similaires. Ces données sont fournies ensuite à un réseau de neurones (ψ). Celui-ci, projette les données dans un espace vectoriel avec comme point d'appui la représentation vectorielle SBERT, qui sera affinée au fil des itérations. Il s'agit d'un modèle conjoint. La première tête du modèle *(f)* effectue un regroupement sur le corpus en utilisant entre autres comme point d'appui la distribution t de Student pour calculer la probabilité d'assigner une donnée à un certain groupe et la divergence K-L pour rapprocher le regroupement d'une distribution cible. La deuxième tête du modèle effectue un apprentissage contrastif, elle utilise les deux copies augmentées du corpus. Pour chaque donnée de la première copie, la donnée correspondante dans la deuxième copie est utilisée comme exemple positif, alors que le reste des données est considéré comme exemples négatifs. L'objectif est de distinguer les exemples positifs des exemples négatifs. Une perte conjointe est définie afin de permettre d'affiner la représentation vectorielle du réseau de neurones ψ. Les auteurs rapportent les résultats obtenus en appliquant K-Means sur la représentation vectorielle affinée.

4.2. Résultats

	AgNews		**SearchSnippets**		**StackOverflow**		**Biomedical**	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
Référence	88,2	68,2	85,2	71,1	75,5	74,5	46,2	41,5
Reproduction	87,5	67,4	84,5	69,7	75,0	73,5	43,8	39,6
Différence	-0,7	-0,8	-0,7	-1.4	-0,5	-1,0	-2,4	-1,9

	GoogleNewsTS		**GoogleNewsT**		**GoogleNewsS**		**Tweets**	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
Référence	89,8	94,9	75,8	88,3	83,1	90,4	78,2	89,2
Reproduction	82,0	93,4	70,1	86,6	74,3	88,6	64,8	85,8
Différence	-7,2	-1,5	-5,7	-1,7	-8,8	-1,8	-13,4	-3,4

Tableau 4.1. Reproduction des résultats de SCCL [**Zhang et al., 2021**]. Moyenne de cinq exécutions.

Malgré nos efforts pour reproduire à l'identique les résultats décrits par [**Zhang et al., 2021**] à l'aide du code fourni dans leur GitHub, nous avons obtenu des résultats inférieurs sur tous les corpus. Au tableau 4.1, nous pouvons y voir l'écart de performance sur chacun des corpus. En moyenne nous sommes 4,9 points en dessous pour la précision avec des résultats variant de -0,5 à -13,4. Pour le NMI nous sommes en moyenne à 1,7 point en dessous avec des résultats variant de -0,8 à -3,4. Alors que l'écart n'est pas plus marqué sur certains corpus pour la métrique NMI, nous remarquons que pour l'ACC cet écart est plus prononcé sur les corpus déséquilibrés. En effet, les corpus GoogleNews et Tweets sont en moyenne 8,8

points inférieurs à l'état de l'art pour l'ACC alors que le reste des corpus cet écart est de 1,1 points. Tel que discuté à la section 2.2 l'effet d'uniformité de K-Means est plus marqué sur les corpus déséquilibrés et a un impact sur la métrique ACC. Ainsi, il est possible que ce phénomène induise plus de variabilité dans les résultats.

4.3. Analyse

Nous élaborons sur les détails techniques du modèle et de notre difficulté à reproduire l'état de l'art.

4.3.1. Hyperparamètres

Nous avons utilisé les mêmes hyperparamètres que les auteurs, ceux-ci sont égaux pour tous les corpus sauf α, le degré de liberté de la distribution t de Student qui est à 10 pour le corpus Biomedical et à 1 pour les autres corpus. Aucune explication n'est fournie par les auteurs sur ce changement de valeur.

4.3.2. Division du corpus

Tel que mentionné par l'auteur principal sur leur GitHub et rapporté à la figure 4.2, il n'y a pas de division des corpus en données d'entraînement et en données de test/validation, car c'est un apprentissage non supervisé.

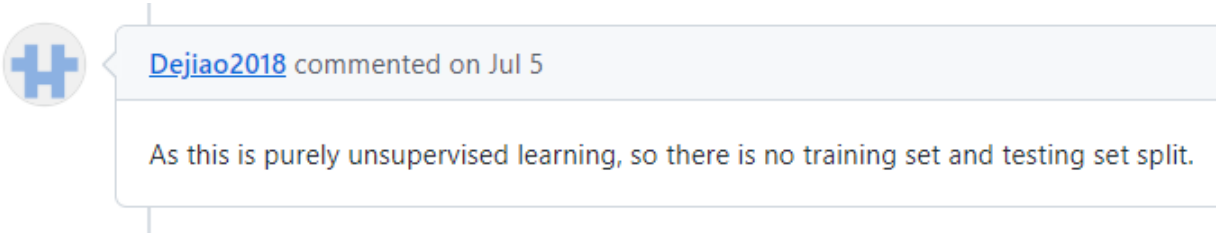


Fig. 4.2. Réponse de l'auteur principal quant à la séparation du corpus en données d'entraînement et données de test. Source: `https://github.com/amazon-science/sccl/issues/21`.

4.3.3. Critère d'arrêt

```
## STOPPING CRITERION (due to some license issue, we still need some time to release the data)
# you need to implement your own stopping criterion, the one we typically use is
# diff (cluster_assignment_at_previous_step - cluster_assignment_at_previous_step) / all_data_samples <= criterion
```

Fig. 4.3. Description sur le critère d'arrêt trouvé dans la version initiale du code pour reproduire les résultats de [**Zhang et al., 2021**].

Aucune description des conditions d'arrêt de l'entraînement du modèle n'est proposée dans [**Zhang et al., 2021**]. En regardant dans le code pour trouver une réponse, la version

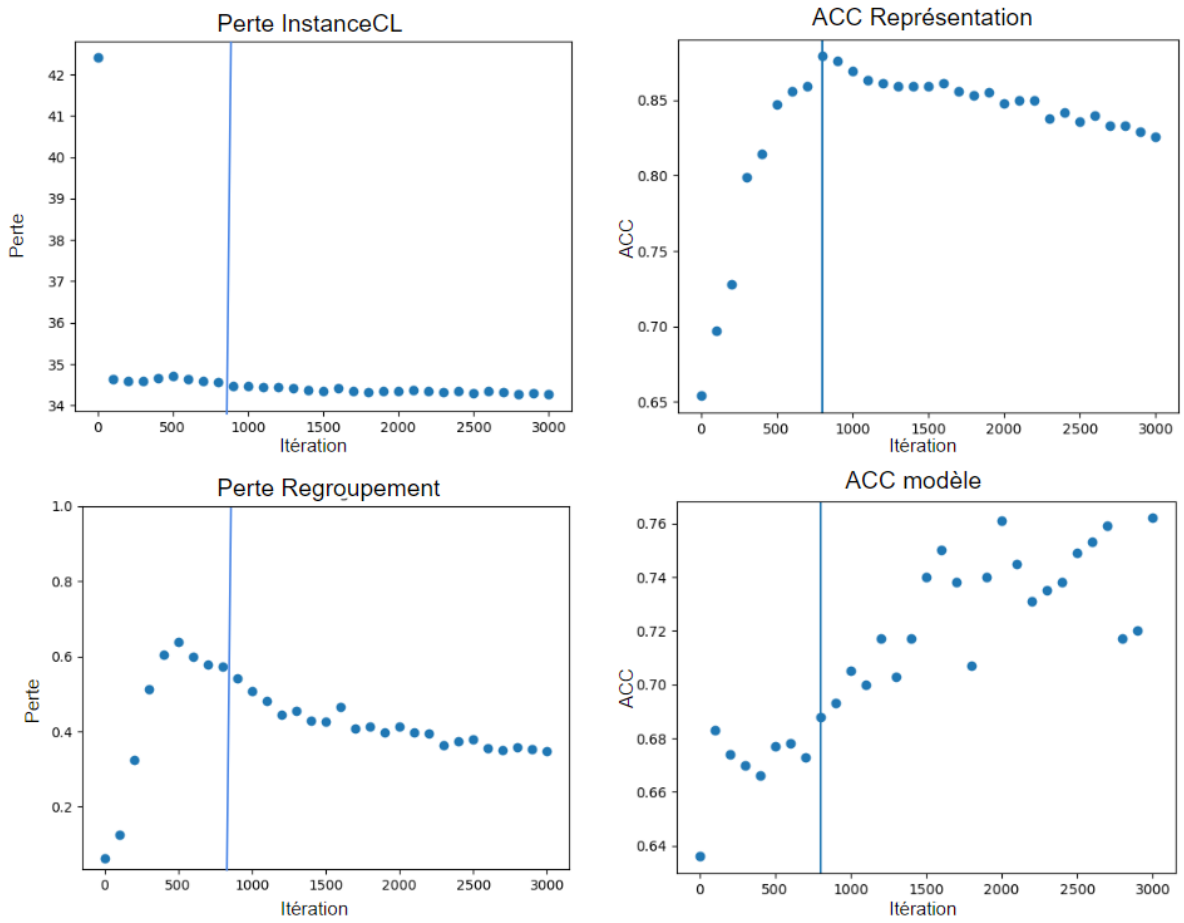


Fig. 4.4. Exemple d'entraînement sur le corpus GoogleNews TS pour la reproduction de SCCL. Évolution de la perte de chacune des deux têtes du modèle. Précision du modèle; la précision de la tête du modèle effectuant le regroupement. Précision représentation; précision en appliquant K-Means sur la représentation vectorielle affinée. La ligne bleue représente le moment où le modèle offre les meilleurs résultats sur la représentation vectorielle affinée.

initiale y contient une description sur le sujet, mais fut enlevée dans la version suivante. Cette description est rapportée à la fig. 4.3. Nous avons contacté les auteurs qui nous ont répondu que c'est lorsque la perte arrête de diminuer ou que les prédictions du modèle restent approximativement les mêmes. Or, comme on peut le voir au tableau 4.4 les deux pertes continuent toujours de diminuer et la précision du modèle d'augmenter. La précision sur la représentation vectorielle affinée utilisée normalement une fois l'entraînement terminé plafonne à la ligne bleue, moment où on se rapproche le plus des résultats à reproduire. Par contre, à ce moment ni les pertes ni la précision du modèle n'a atteint le critère d'arrêt proposé par les auteurs.

De plus, n'ayant pas réussi à trouver comment reproduire les résultats, pour nous y rapprocher le plus possible nous n'avons pas utilisé de critère d'arrêt, malgré que ce n'est pas un protocole à préconiser. Nous avons entraîné un modèle pour chacun des corpus

sur le nombre maximal d'itérations utilisé par les auteurs (3000). Ensuite, nous avons pris les résultats de l'itération du modèle offrant la meilleure précision sur la représentation vectorielle affinée. En d'autres mots, nous avons pris les meilleurs résultats possibles.

4.3.4. Conclusion

Nous concluons que la seule façon de se rapprocher des résultats c'est d'utiliser la précision sur la représentation affinée et de sélectionner la meilleure itération. Par contre, utiliser ce type de critère d'arrêt n'est pas une bonne approche à notre avis et ne nous a pas permis d'obtenir les mêmes résultats que l'état de l'art.

Chapitre 5

Expériences et résultats

Nous étudions différentes idées présentées dans le chapitre 1 afin de valider leur pertinence dans notre contexte. Plus précisément, nous validons le choix de l'algorithme de regroupement entre K-Means et K-Medoids. Puis, examinons la meilleure mesure de similarité pour notre représentation vectorielle entre la similarité cosinus et la distance euclidienne. Par la suite, nous comparons trois représentations vectorielles SBERT. Nous testons une combinaison de deux approches avec différentes représentations SBERT : l'affinage de la représentation vectorielle et la réduction de dimensionnalité. Finalement, nous nous intéressons à savoir si le prétraitement des données et si l'inclusion de la notion de la fréquence des mots aident au regroupement. Nous comparons nos résultats avec deux méthodes plus traditionnelles (Tf-idf et Word2Vec), puis, avec un modèle ne connaissant pas le nombre de groupes au préalable (HDSCAN). Nous comparons nos expériences également au modèle état de l'art (SCCL). Tel que mentionné au chapitre 2, les métriques d'évaluation à privilégier sont le NMI et l'ARI, nous utilisons donc ces métriques. Malgré notre incapacité de reproduire les résultats état de l'art du modèle SCCL, nous utilisons les résultats rapportés par les auteurs pour comparer nos expériences. Nous concluons ce chapitre avec un regard global sur l'état actuel du regroupement de texte et en faisons nos recommandations.

5.1. Validation de la mesure de similarité cosinus et de la distance euclidienne

Il est mentionné dans [**Sohangir and Wang, 2017**] que la similarité cosinus semble mieux performer que la distance euclidienne. Par contre, plusieurs implémentations d'algorithme de regroupement utilisent la distance euclidienne. Nous voulons valider la pertinence de la distance cosinus sur la représentation vectorielle de *SBERT*.

Comme nous pouvons le voir dans le tableau 5.1, les résultats sont légèrement meilleurs pour la similarité cosinus. En effet, sans la réduction de dimensionnalité la similarité cosinus

est meilleure dans 6/8 cas pour le NMI et 7/8 cas pour l'ARI. Dans le cas où la réduction de dimensionnalité est appliquée, la similarité cosinus pour le NMI performe mieux dans 7/16 cas, a la même performance dans 4/16 cas et performe moins bien de seulement 0,1 point dans 4/16 cas. Pour l'ARI avec la réduction de dimensionnalité, la distance cosinus performe mieux dans 9/16 cas, a la même performance dans 2/16 cas. Ainsi, nous pouvons en conclure que la similarité cosinus semble plus efficace que la distance euclidienne. De plus, nous remarquons que pour le modèle MPNet il n'y a aucune différence, probablement parce que le modèle normalise les vecteurs avant de les retourner. Également, nous observons qu'avec ou sans la réduction de dimensionnalité la similarité cosinus offre de meilleurs résultats, ainsi cet aspect ne serait pas un facteur déterminant pour le choix entre ces deux métriques. Par les définitions mêmes de ces deux métriques, étant donné que la similarité cosinus offre de meilleurs résultats, nous pouvons soupçonner que la magnitude des dimensions des vecteurs serait un facteur moins important que leur direction. Cette hypothèse est en accord avec le principe qu'en haute dimensionnalité les vecteurs sont quasi équidistants [**Hall et al., 2005**]. Il serait intéressant de faire une analyse plus globale en comparant avec d'autres distances comme la distance Hellinger (norme L1). Pour la suite des expériences, nous utilisons la similarité cosinus.

5.2. K-Means et K-Medoids

Nous testons l'efficacité de K-Medoids qui se veut comme une solution à certaines lacunes de K-Means. Nous utilisons la configuration par défault de scikit-learn-extra[1]. Par défaut c'est un algorithme alternatif [**Park and Jun, 2009**] qui est utilisé à celui proposé initialement pour K-Medoids (PAM) [**Kaufman and Rousseeuw, 1990**]. PAM est sensé produire de meilleurs résultats, mais son temps d'exécution est trop long. Les détails des deux algorithmes peuvent être trouvés sur le site de scikit-learn-extra[2].

Au tableau 5.2 et à la figure 5.1 nous remarquons tout d'abord que K-Medoids a une performance nettement inférieure à K-Means lorsque la réduction de dimensionnalité n'est pas appliquée. En effet, pour tous les corpus et tous les modèles SBERT, K-Means performe mieux. Nous remarquons par contre que la différence est moins prononcée sur les corpus fortement déséquilibrés (GoogleNews et Tweets). En moyenne sur les corpus excluant GoogleNews et Tweets, le NMI est inférieur de 16,8 points et le ARI de 15,2 points. Alors que sur les corpus déséquilibrés le NMI est inférieur de 4,0 points et le ARI de 3,2 points. Cette observation nous porte à croire que l'effet d'uniformité serait moins présent chez K-Medoids que K-Means. Au mieux de notre connaissance, il n'y a pas d'article qui compare l'effet d'uniformité de K-Means et K-Medoids. D'autre part en comparant avec les résultats où

[1] `https://scikit-learn-extra.readthedocs.io/en/stable/generated/sklearn_extra.cluster.KMedoids.html`

[2] `https://scikit-learn-extra.readthedocs.io/en/stable/modules/cluster.html`

	AgNews		**SearchSnippets**		**StackOverflow**		**Biomedical**	
	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI
DistilBERT-E	33,1	34,5	**32,3**	**27,4**	**51,6**	34,1	33,6	19,9
DistilBERT-C	**34,0**	**36,2**	31,3	26,5	51,0	**34,7**	**33,9**	**20,3**
Différence	0,9	1,7	-1.0	-0,9	-0,6	0,6	0,3	0,4
DistilBERT-UMAP-E	49,4	52,4	**46,3**	43,0	**55,1**	**43,6**	**33,8**	**21,6**
DistilBERT-UMAP-C	**49,7**	52,4	46,2	**43,3**	54,7	41,8	33,7	21,1
Différence	0,3	0,0	-0,1	0,3	-0,4	-1,8	-0,1	-0,5
MPNet-E	60,2	62,4	58,0	50,7	70,6	57,1	41,6	27,6
MPNet-C	60,2	62,4	58,0	50,7	70,6	57,1	41,6	27,6
Différence	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
MPNet-UMAP-E	54,6	45,1	61,4	55,8	70,1	62,1	38,1	24,8
MPNet-UMAP-C	**56,8**	**47,8**	**62,0**	**57,1**	70,1	62,1	**38,4**	**25,2**
Différence	2,2	2,7	0,6	1,3	0,0	0,0	0,3	0,4

	GoogleNewsTS		**GoogleNewsT**		**GoogleNewsS**		**Tweets**	
	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI
DistilBERT-E	86,8	59,6	78,1	46,9	81,0	52,2	78,9	40,3
DistilBERT-C	**87,5**	**61,9**	**79,0**	**48,9**	**81,3**	**52,9**	**79,8**	**43,2**
Différence	0,7	2,5	0,9	2,0	0,3	0,7	0,9	2,9
DistilBERT-UMAP-E	**90,8**	**71,4**	84,4	60,0	**86,1**	**65,3**	84,1	47,8
DistilBERT-UMAP-C	90,7	71,2	84,4	**60,3**	86,0	64,8	84,1	**48,4**
Différence	-0,1	-0,2	0,0	0,3	-0,1	-0,5	0,0	0,6
MPNet-E	91,6	69,3	86,6	57,9	88,0	62,4	87,3	52,6
MPNet-C	91,6	69,3	86,6	57,9	88,0	62,4	87,3	52,6
Différence	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
MPNet-UMAP-E	93,9	78,3	**90,1**	**69,3**	90,9	72,3	89,0	52,0
MPNet-UMAP-C	**94,0**	**79,1**	89,9	68,8	**91,1**	**73,7**	89,0	**52,1**
Différence	0,1	0,8	-0,2	-0,5	0,2	1,4	0,0	0,1

Tableau 5.1. Comparaison entre la similarité cosinus (C) et la distance euclidienne (E) comme mesure de similarité pour différents modèles de représentation vectorielle SBERT avec ou sans réduction de dimensionnalité (UMAP). Chaque résultat est une moyenne de cinq exécutions.

la réduction de dimensionnalité est appliquée nous remarquons des gains importants pour K-Medoids, alors que ces gains sont moins marqués pour K-Means. À la lumière de ces résultats il est possible que K-Medoids performe moins bien en haute dimension, et qu'une fois la dimension réduite sa performance rejoigne celle de K-Means. K-Medoids réussi même

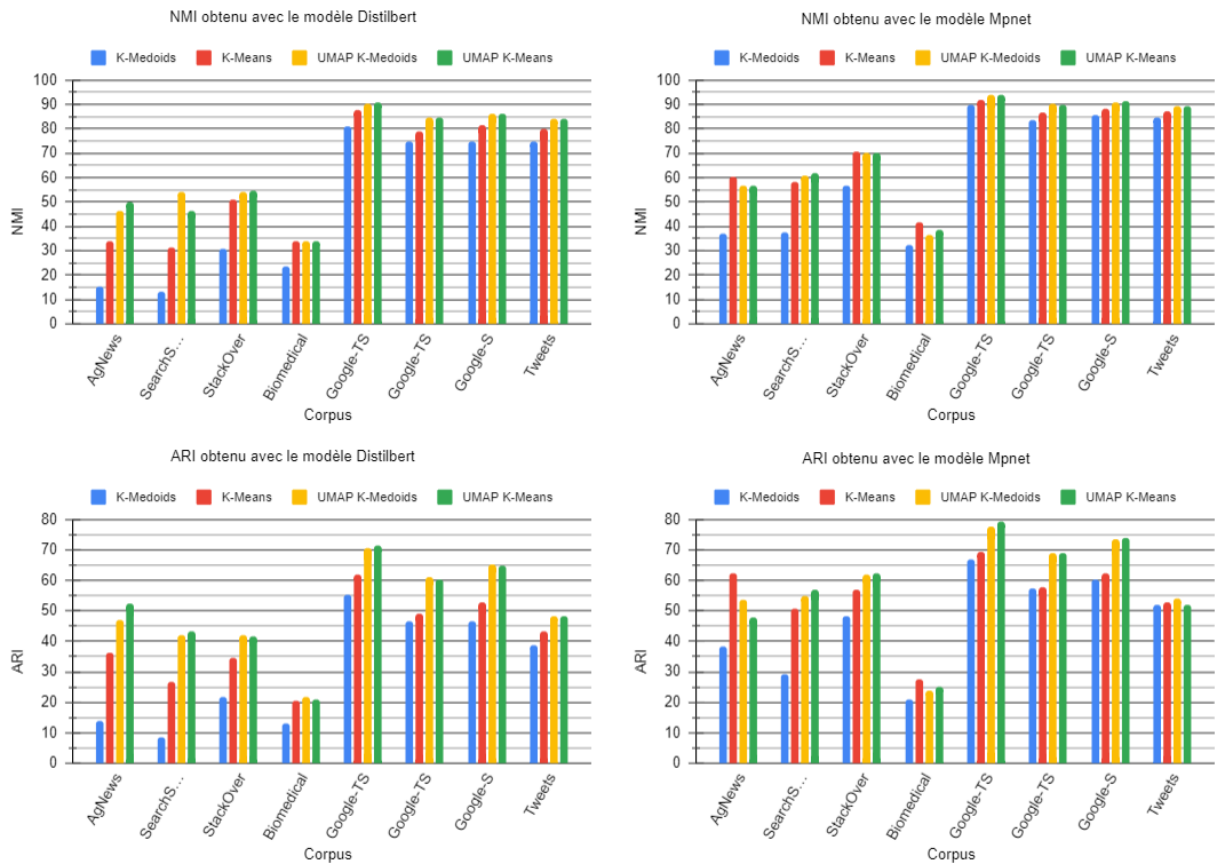


Fig. 5.1. Comparaison entre l'algorithme de regroupement K-Means et K-Medoids sur deux modèles SBERT avec et sans réduction de dimensions.

à avoir un meilleur score pour le NMI dans 5/16 situations et pour l'ARI dans 6/16 situations. Le graphique à la figure 5.1 permet de mieux visualiser ces différences. Finalement, il est possible que K-Medoids n'a pas pu être mis complètement en valeur, car une de ses forces est d'être plus robuste aux données aberrantes, comme nos corpus sont soigneusement préparés, il risque d'y avoir très peu de ce type de données. De plus, l'algorithme PAM, tel que précisé à la section 5.2, n'a pas été testé afin de valider le plein potentiel de K-Medoids dû à son temps d'exécution trop long. En effet, la complexité de l'algorithme PAM pour le regroupement de n éléments en k groupes est de $O(k(n-k)^2)$, alors que pour K-Means celle-ci est linéaire $O(n)$. Il serait intéressant de tester certaines solutions proposées pour réduire la complexité de PAM, tel que [**Park and Jun, 2009**]. Pour la suite des expériences nous considérons seulement K-Means.

	AgNews		**SearchSnippets**		**StackOverflow**		**Biomedical**	
	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI
DistilBERT K-Medoids	15,1	13,8	13,1	8,6	31,0	21,8	23,3	13,1
DistilBERT K-Means	**34,0**	**36,2**	**31,3**	**26,5**	**51,0**	**34,7**	**33,9**	**20,3**
Différence	-18,9	-22,4	-18,2	-17,9	-20,0	-12,9	-10,6	-7,2
DistilBERT-UMAP K-Medoids	46,2	46,8	**54,2**	42,2	54,2	**42,2**	**34,1**	**21,8**
DistilBERT-UMAP K-Means	**49,7**	**52,4**	46,2	**43,3**	**54,7**	41,8	33,7	21,1
Différence	-3,5	-5,6	8,0	-1,1	-0,5	0,4	0,4	0,7
MPNet K-Medoids	37,2	38,3	37,7	29,3	56,8	48,4	32,4	20,7
MPNet K-Means	**60,2**	**62,4**	**58,0**	**50,7**	**70,6**	**57,1**	**41,6**	**27,6**
Différence	-23,0	-24,1	-20,3	-21,4	-13,8	-8,7	-9,2	-6,9
MPNet-UMAP K-Medoids	56,4	**53,4**	60,8	54,8	70,1	61,8	36,5	23,7
MPNet-UMAP K-Means	**56,8**	47,8	**62,0**	**57,1**	70,1	**62,1**	**38,4**	**25,2**
Différence	-0.4	5,6	-1,2	-2,3	0,0	-0,3	-1,9	-1,5

	GoogleNewsTS		**GoogleNewsT**		**GoogleNewsS**		**Tweets**	
	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI
DistilBERT K-Medoids	81,0	55,4	74,6	46,7	74,7	46,4	74,9	38,6
DistilBERT K-Means	**87,5**	**61,9**	**79,0**	**48,9**	**81,3**	**52,9**	**79,8**	**43,2**
Différence	-6,5	-6,5	-4,4	-2,2	-6,6	-6,5	-4,9	-4,6
DistilBERT-UMAP K-Medoids	90,5	70,5	**84,5**	**61,1**	86,0	**65,3**	**84,3**	48,4
DistilBERT-UMAP K-Means	**90,7**	**71,2**	84,4	60,3	86,0	64,8	84,1	48,4
Différence	-0,2	-0,7	0,1	0,8	0,0	0,5	0,2	0,0
MPNet K-Medoids	89,9	67,0	83,8	57,3	85,7	60,3	84,4	51,9
MPNet K-Means	**91,6**	**69,3**	**86,6**	**57,9**	**88,0**	**62,4**	**87,3**	**52,6**
Différence	-1,7	-2,3	-2,8	-0,6	-2,3	-2,1	-2,9	-0,7
MPNet-UMAP K-Medoids	93,7	77,8	**90,1**	68,8	91,0	73,3	**89,1**	**54,0**
MPNet-UMAP K-Means	**94,0**	**79,1**	89,9	68,8	**91,1**	**73,7**	89,0	52,1
Différence	-0,3	-1,3	0,2	0,0	-0,1	-0,4	0,1	1,9

Tableau 5.2. Comparaison entre l'algorithme de regroupement K-Means et K-Medoids sur deux modèles SBERT avec et sans réduction de dimensionnalité.

5.3. Modèles SBERT

Comme représentation vectorielle contextuelle de phrases nous avons choisi celle de SBERT. Plus précisément nous avons sélectionné trois modèles. DistilBERT (distilbert-base-nli-stsb-mean-tokens) pour être comparable avec [**Zhang et al., 2021**], malgré que ce

modèle a été déprécié pour sa mauvaise qualité[3]. Le modèle état de l'art MPNet (all-mpnet-base-v2), T5 est sensé être encore meilleur, mais c'est un modèle près de 20 fois plus gros que MPNet, avec un temps d'encodage du texte nettement plus lent pour des résultats à peine meilleurs selon la documentation de SBERT. Puis, nous testons un modèle RoBERTa (all-roberta-large-v1), un des meilleurs modèles permettant aussi l'affinage. Nous utilisons la table de performance fourni par SBERT[4] pour sélectionner nos modèles, celle-ci se trouve également en Annexe A. Nous pouvons y voir les modèles disponibles au moment de faire nos expériences ainsi que leur performance et leur taille.

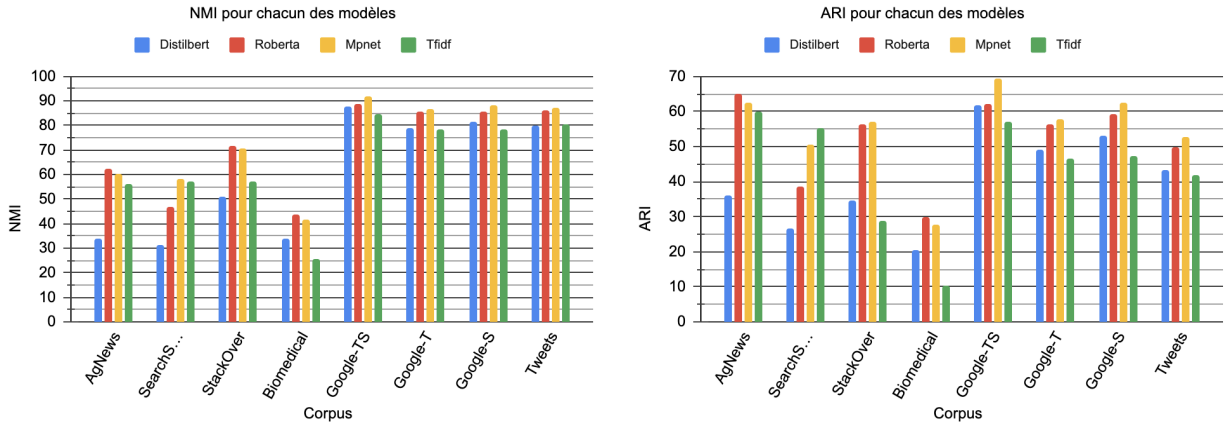


Fig. 5.2. Performance des modèles SBERT sélectionnés et comparaison avec un modèle de référence (Tf-idf). Moyenne sur 5 exécutions.

Pour les trois modèles sélectionnés, il est possible de voir leur performance sur le regroupement au tableau 5.3 et à la figure 5.2. Tout d'abord, notre modèle de référence, Tf-idf, qui conserve les 3500 mots les plus représentatifs à des performances proches et à l'occasion meilleures que les modèles SBERT. Cela peut s'expliquer par l'importance de repérer les mots représentatifs de chaque regroupement plutôt que d'en comprendre le sens, avantage principal d'un modèle Tf-idf. Nous discutons de cet aspect plus loin dans ce chapitre. Ensuite, tel que mentionné dans la documentation de SBERT, nous remarquons que DistilBERT performe nettement moins bien que les deux autres modèles. Pour ce qui est de RoBERTa et MPNet, ceux-ci ont des performances semblables avec MPNet démontrant des performances légèrement meilleures. RoBERTa performe mieux sur les corpus AgNews, StackOverflow et Biomedical avec la métrique NMI et seulement sur AgNews et Biomedical avec la métrique ARI. Étonnamment, comme nous pouvons constater au tableau A.1 en Annexe A, le modèle MPNet est plus petit que RoBERTa. MPNet produit des vecteurs de 728 dimensions alors que RoBERTa de 1024 dimensions. Une hypothèse possible est que la

[3] `https://huggingface.co/sentence-transformers/distilbert-base-nli-stsb-mean-tokens`
[4] `https://www.sbert.net/docs/pretrained_models.html`

différence de performance soit seulement due à la difficulté de K-Means à faire du regroupement en haute dimension. Également, MPNet selon ses auteurs [**Song et al., 2020**] est une amélioration sur les modèles précédents comme RoBERTa. Les deux modèles sont affinés sensiblement sur les mêmes données, avec plus d'un milliard de paires d'entraînements [5] [6]. Même pour l'entraînement du modèle de base, les auteurs de Mpnet ont utilisé les mêmes données que RoBERTa. Donc, cette différence semble plutôt s'expliquer par l'amélioration de l'architecture du modèle que par le choix des données.

5.4. Affinage de la représentation vectorielle contextuelle SBERT

Comme nous avons certains corpus avec des données très spécifiques, propres à un domaine en particulier, nous appliquons la technique TSDEA avec les paramètres par défaut[7] sur le modèle RoBERTa de SBERT pour valider si la représentation vectorielle s'améliore. Il est à noter que certaines architectures des modèles SBERT n'autorisent pas[8] l'application de la technique TSDEA. Il n'existe pas de liste des modèles SBERT compatibles avec TSDEA, nous avons procédé par essai-erreur, en commençant par les meilleurs modèles SBERT pour trouver le premier modèle compatible.

Les résultats de nos expériences se trouvent au tableau 5.3. Malgré l'intérêt de TSDEA pour l'affinage des modèles de manière non supervisée et contrairement aux résultats de l'article de référence [**Wang et al., 2021a**], nous n'observons pas une amélioration importante, même que la performance se détériore sur les corpus AgNews, StackOverflow, Biomedical. De plus, les corpus où les meilleurs gains sont observés impliquent des modèles performant moins bien initialement. Par exemple, sur SearchSnippets, RoBERTa a un NMI de 46,6 et un ARI de 38,5 alors que MPNet est à un NMI de 58,0 et 50,7. Une fois TSDEA appliqué RoBERTa se rapproche de MPNet avec un NMI de 57,3 et un ARI de 53,1. Nous suspectons que si nous avions testé la technique sur DistilBERT nous aurions vu une amélioration intéressante, mais sans battre les meilleurs modèles (RoBERTa et MPNet). Cette observation nous porte à croire que les meilleurs modèles n'ont plus la capacité d'apprendre davantage alors que les modèles de base oui. De plus, ce que nous remarquons c'est que dans l'article de référence les auteurs utilisent des modèles de base, moins performants, et observent des gains importants sur ceux-ci. Par contre, ces modèles affinés ne sont pas comparés aux meilleurs modèles, seulement aux modèles de base. Nous concluons que l'affinage des modèles SBERT

[5] https://huggingface.co/sentence-transformers/all-roberta-large-v1

[6] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

[7] https://www.sbert.net/examples/unsupervised_learning/TSDAE/README.html

[8] https://github.com/UKPLab/sentence-transformers/blob/6510b431428274885d13a4874c862f535227bd6b/sentence_transformers/losses/DenoisingAutoEncoderLoss.py

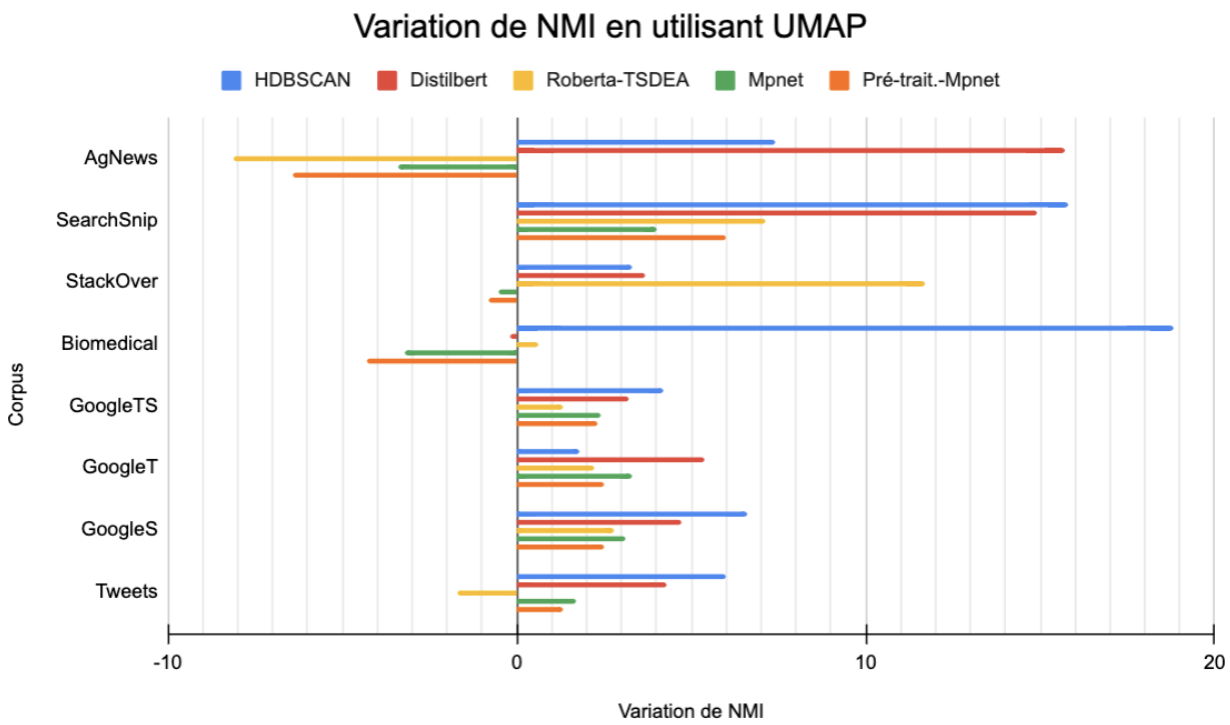
n'est pas nécessaire pour le regroupement de textes, car les meilleurs modèles performent mieux généralement.

5.5. Réduction de dimensionnalité

Motivé par la difficulté de K-Means à faire du regroupement en haute dimension nous appliquons la technique UMAP de réduction de dimensions. Les paramètres utilisés sont un nombre de voisins à 15 (*n_neighbors=15*), une réduction à 32 dimensions (*n_components=32*) et la similarité cosinus (*metric='cosine'*).

Le tableau 5.3 présente la variation de performance en appliquant UMAP. Nous observons le même phénomène qu'avec l'affinage des modèles. Les modèles performant moins bien initialement bénéficient du meilleur gain de performance. En effet, DistilBERT et HDBSCAN ont une amélioration de NMI et ARI sur tous les corpus et celle-ci est plus prononcée en général. En moyenne sur les huit corpus, HDBSCAN a un gain de performance en NMI de 8,0 et en ARI de 6,8 et DistilBERT en NMI de 6,5 et en ARI de 9,1. Alors que pour les modèles performant mieux initialement ce gain est pour RoBERTa-TSDAE en NMI de 2,0 et en ARI de -0,7, pour MPNet en NMI de 0,9 et en ARI de 2,3 et pour Prétrait.-MPNet en NMI de 0,4 et en ARI de 2,4. Il est intéressant d'observer que malgré que HDBSCAN utilise aussi un modèle de représentation vectorielle MPNet celui-ci offre des meilleurs gains sur HDBSCAN que sur K-Means. En consultant, l'article de référence d'UMAP [**McInnes et al., 2018**], nous en concluons que cette observation s'explique par le fait qu'UMAP ne produit pas nécessairement des groupements sphériques, aspect pouvant rendre le regroupement avec K-Means plus difficile. D'autre part, sur les meilleurs modèles nous remarquons qu'appliquer UMAP offre des résultats inférieurs sur les corpus AgNews, Tweets, StackOverflow et Biomedical. À notre avis, deux phénomènes permettent de l'expliquer. Premièrement, la taille du corpus: AgNews et Tweets sont les plus petits corpus et UMAP construit un graphe avec les plus proches voisins, or avec un plus petit nombre de données, une même donnée peut être plus proche voisin de plusieurs regroupements. Cet aspect est également soulevé sur gitlab[9]. Nous avons conservé le paramètre du nombre de plus proches voisins fixe pour tous les corpus, soit 15. Il serait intéressant de le faire varier en fonction de la taille du corpus. Ensuite, StackOverflow et Biomedical sont des corpus avec des données très techniques et par conséquent la représentation vectorielle risque d'être de moindre qualité. Ainsi, sur ce genre de corpus, il est possible que l'analyse de variété (*Manifold* en anglais) soit plus difficile, une des théories sur laquelle s'appuie UMAP. En d'autres mots, ces données ne possèdent pas une représentation de plus petite dimension équivalente, ou d'une manière plus illustrée: représenter un cube en deux dimensions est une tâche facile, par contre pour une structure déformée cette tâche s'avère plus complexe.

[9] `https://github.com/satijalab/seurat/issues/3077`



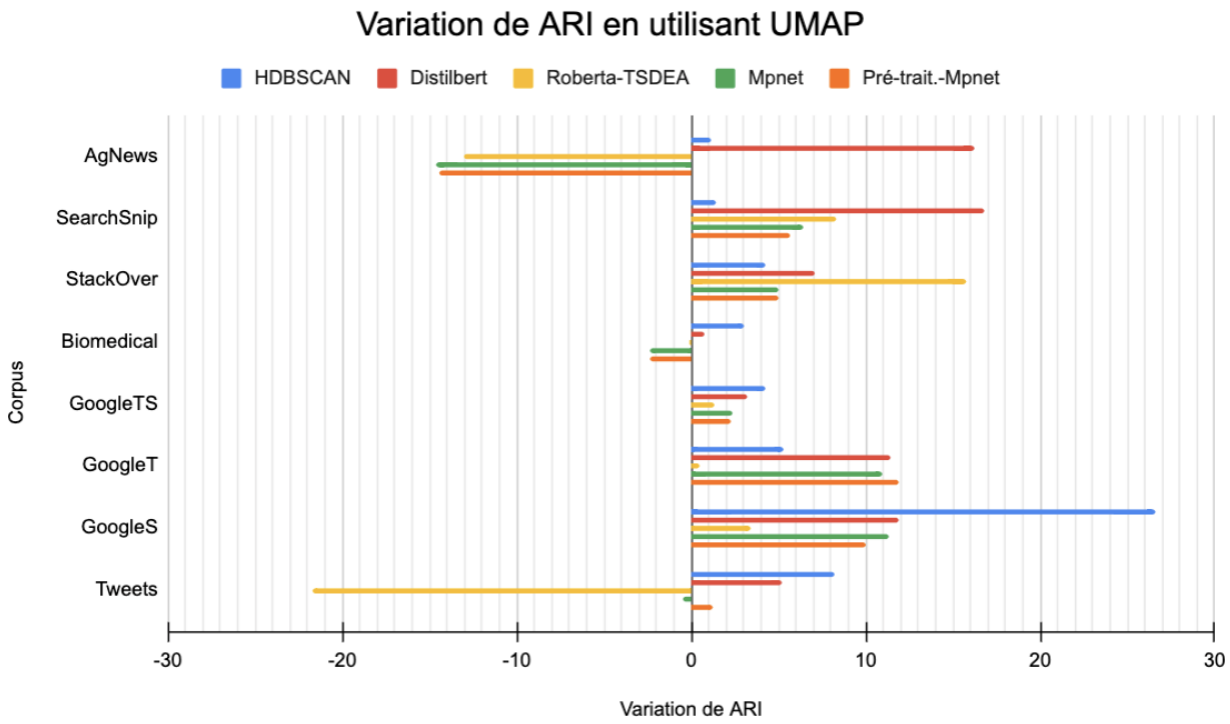


Fig. 5.3. Variation de NMI et ARI en appliquant la technique de réduction de dimensionnalité UMAP sur cinq différents modèles et testé sur les huit corpus.

Nous concluons que cette technique de réduction de dimensionnalité est intéressante pour des corpus n'étant pas trop petits (AgNews et Tweets ne devraient pas être considérés) et

n'ayant pas un haut niveau de technicité (StackOverflow et Biomedical ne devraient pas être considérés). La définition exacte de la quantité de données suffisante pour faire du regroupement avec UMAP reste cependant à définir.

5.6. Prétraitement des données

Nous étudions si porter une attention particulière au prétraitement des données améliore les résultats. Avant l'arrivée de la représentation vectorielle contextuelle des données, le prétraitement des données était une étape importante pour accomplir une tâche en TALN. Elle a quelque peu été abandonnée sous l'argument que maintenant les modèles de types *Transformers* sont entraînés sur des données brutes ou qu'un prétraitement suffisant (dont l'application de *WordPiece* [**Wu et al., 2016**]) est déjà inclus dans les librairies qui utilisent ces modèles.

Le tableau 5.4 illustre les prétraitements effectués sur les corpus. Les techniques standard sont appliquées. La mise en minuscule est appliquée si elle n'a pas déjà été appliquée avant le téléchargement du corpus. Ensuite, la suppression des mots vides est appliquée. Pour certains corpus ayant un thème particulier, des mots vides spécifiques à ce corpus sont supprimés en regardant les mots plus fréquents du corpus. Par exemple, pour AgNews, un corpus de nouvelles, nous pouvons supprimer les jours de la semaine. Ensuite, sur les corpus non techniques nous avons appliqué la lemmatisation. En analysant manuellement des échantillons de chacun des corpus, nous avons remarqué que certains corpus contenaient des mots courts sans signification, c'est pourquoi nous avons appliqué la suppression de mots avec 2 caractères et moins. Puis, en regardant les mots les moins fréquents de chacun des corpus, nous avons supprimé les mots avec une fréquence inférieure à 1,2 ou 3 selon les corpus. Certains corpus contenaient des caractéristiques propres qui nous ont permis de supprimer davantage de contenu, comme la présence d'URL, d'identificateurs HTML ou bien la répétition de mots ou groupes de mots. En annexe C, il est possible de voir pour chacun des corpus des exemples de données avant et après le prétraitement et permettent d'illustrer l'intérêt que nous voyons dans le prétraitement de ces données.

5.6.1. Résultats

Malgré la perte d'intérêt du prétraitement des données depuis l'arrivée des modèles *Transformers*, nous remarquons que celle-ci a une incidence sur les résultats. Il est possible de voir les résultats de nos expériences au tableau 5.3 et aux graphiques de la figure 5.4. Tout d'abord, il est important de souligner que nous n'avons pas pu tester complètement l'impact du prétraitement des données sur tous les corpus, car les corpus GoogleNews, Tweets, SearchSnippets et Biomedical sont déjà pré-traités lorsque nous les téléchargeons. Nous

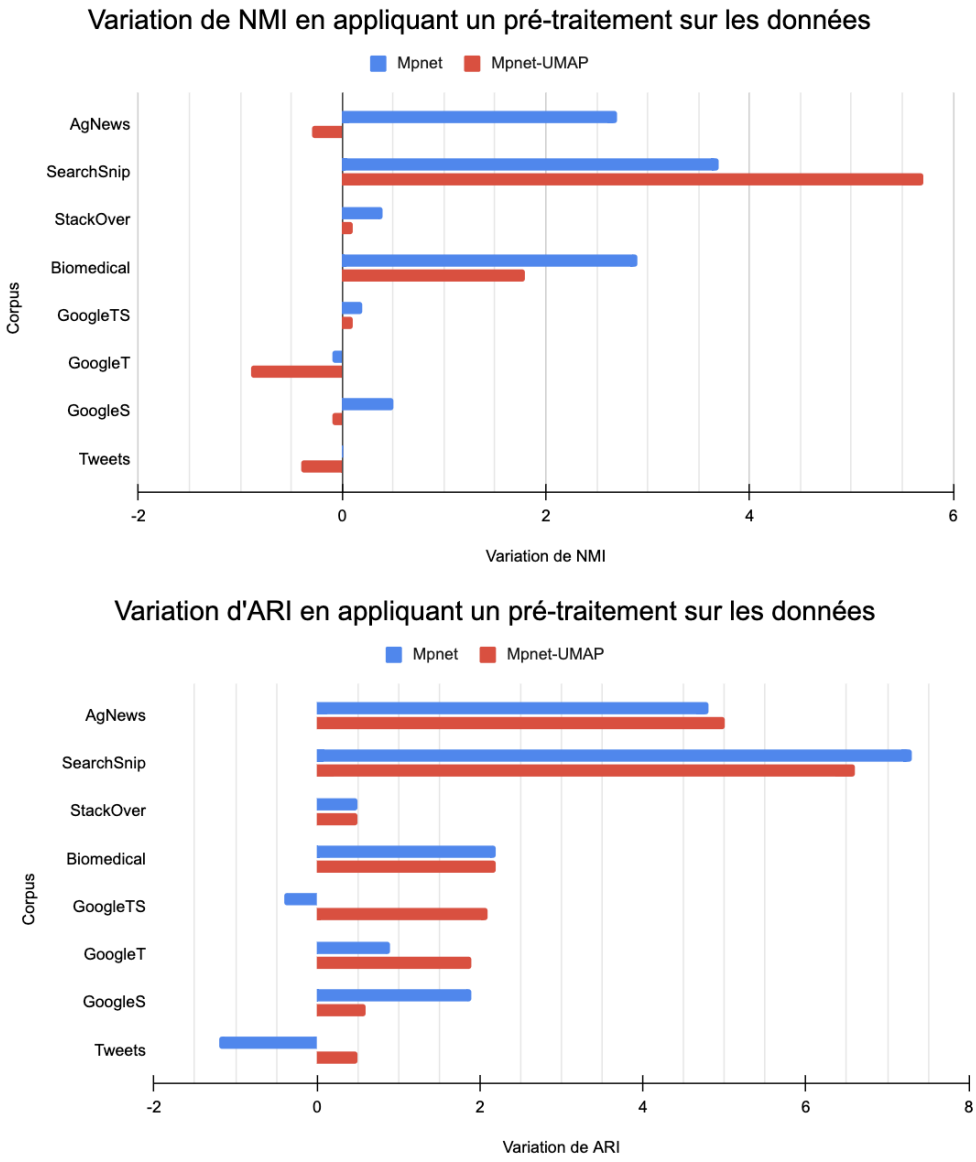


Fig. 5.4. Variation de NMI et d'ARI en appliquant un prétraitement des données

avons tout de même trouvé des prétraitements supplémentaires pertinents. Les corpus GoogleNewsT et Tweets ont un petit vocabulaire et une moyenne de mots par phrases inférieure aux autres corpus, comme on peut le constater au tableau 3.1, cet aspect rend le prétraitement plus difficile, car il empêche des techniques plus agressives comme la suppression de mots sous un certain seuil de fréquence. Nous remarquons au tableau 5.3 que le prétraitement nous permet d'atteindre les meilleurs résultats parmi nos expériences effectuées sur les corpus AgNews, SearchSnippets, StackOverflow et Biomedical et que de façon générale on peut en conclure en regardant les graphiques de la figure 5.4 que le prétraitement améliore les résultats. En conclusion, le prétraitement des données demeure pertinent même avec les modèles récents.

5.7. Inclure la fréquence des mots

Nous pensons que certains groupements peuvent facilement être faits en portant attention aux mots importants. En effet, en travaillant sur le prétraitement des données, nous avons remarqué que dans plusieurs corpus certains groupes sont facilement identifiables en repérant des mots seulement présents dans ces groupes. Par exemple, dans le corpus Biomedical, être en mesure de voir que le mot *cat* est présent dans seulement un groupe permet de regrouper facilement les données appartement à ce groupe, de même que dans le corpus StackOverflow repérer les données contenant le mot *linq* permet de regrouper les données appartenant au groupe traitant de *linq*. Plusieurs autres exemples sont montrés en annexe B. La technique permettant d'inclure cette information est Tf-idf et nous testons si une combinaison de Tf-idf et SBERT permet un meilleur regroupement. Pour ce faire nous concaténons les vecteurs de Tf-idf et SBERT. [**Bianchi et al., 2021**] propose une technique semblable en détection de sujets. Afin d'éviter d'avoir un vecteur de trop grande dimension nous appliquons Tf-idf une fois les données prétraitées et la taille du vocabulaire réduit pour conserver seulement les 3 500 mots les plus fréquents.

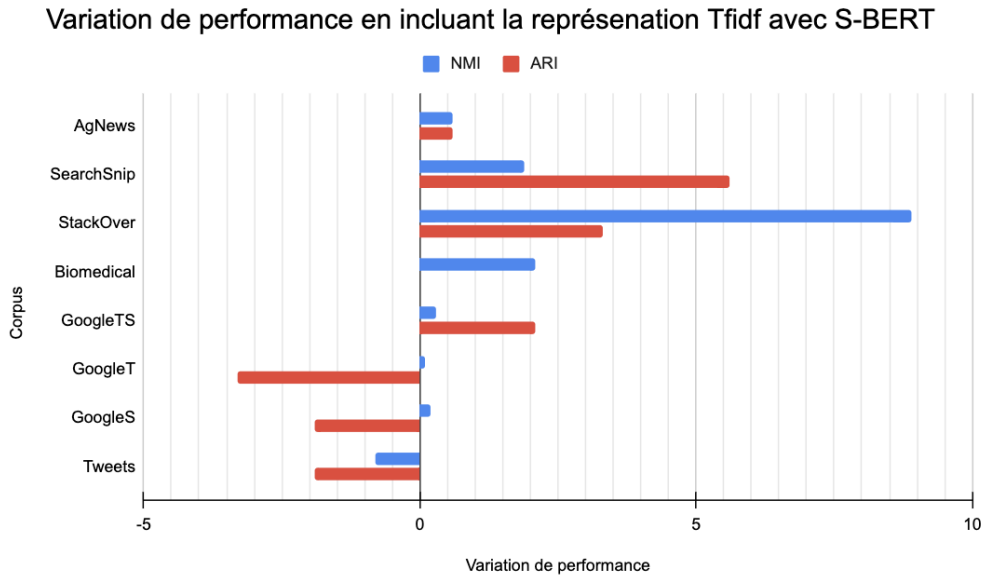


Fig. 5.5. Variation de NMI et d'ARI en concaténant les dimensions de Tf-idf à ceux de MPNet. Variation mesurée sur le modèle MPNet avec données pré-traitées.

Bien que cette technique implique que K-Means se retrouve à devoir faire un regroupement sur une représentation vectorielle de plus de 4 000 dimensions, nous remarquons que pour les corpus ayant des termes spécifiques pour chaque groupe (StackOverflow et Biomedical) nous obtenons de meilleurs résultats que l'état de l'art (tableau 5.3). De manière plus générale, nous observons au tableau 5.5 qu'inclure Tf-idf permet d'améliorer les résultats sur

tous les corpus à l'exception de Tweets pour le NMI et pour l'ARI sur tous les résultats à l'exception de GoogleNewsS, GoogleNewsT et Tweets. La première intuition pour expliquer les résultats sur le corpus Tweets est qu'il ne contient pas de mots permettant de facilement identifier un groupe. Or, en analysant manuellement le corpus, nous remarquons que le principal groupe traite du Super Bowl et que repérer ces mots permet de facilement regrouper une grande quantité d'exemples. Alors, il est possible que le haut niveau de déséquilibre de Tweets empêche la mise en valeur de la fréquence des mots, phénomène également observé par [**Jiang et al., 2021**]. Les corpus GoogleNews sont également des corpus déséquilibrés, par contre, une explication possible sur le gain de performance observé pour GoogleNewsTS est que la longueur moyenne du texte plus élevé (voir tableau 3.1) permette de compenser le déséquilibre du corpus. En conclusion, il semble bénéfique d'appliquer cette technique sur les corpus n'ayant pas un haut niveau de déséquilibre.

5.8. Analyse des corpus utilisés

Tout d'abord, nous remarquons des résultats plus élevés sur les corpus GoogleNewsTS, GoogleNewsT, GoogleNewsS et Tweets par rapport aux autres corpus pour la métrique NMI alors que pour la métrique ARI l'écart est moindre. Cette différence s'explique par la tendance de NMI à surestimer le score des corpus déséquilibrés [**de Souto et al., 2012**].

D'autre part, en analysant manuellement le corpus AgNews, nous remarquons que deux des quatre groupes se confondent sémantiquement. En effet, le groupe *Sci/Tech* et le groupe *Business* sont difficiles à distinguer, par exemple les entreprises en technologie cotées à la bourse font souvent parler d'elles lors de la publication de leurs résultats financiers, nouvelles considérées dans le groupe *Business*, mais ces articles contiennent de nombreux termes reliés au groupe *Sci/Tech*.

Ensuite, les modèles de références (Tf-idf, Word2Vec et HDBSCAN) ont des performances semblables aux meilleurs modèles sur les corpus Biomedical, Google et Tweets. Il pourrait s'agir d'un indice que ces corpus ne sont pas assez difficiles.

5.9. Analyse globale

Pour nos expériences, nous nous sommes comparés à l'état de l'art (SCCL). Dans un premier temps, nous avons tenté de reproduire les résultats et comme mentionné précédemment nous avons rencontré certaines difficultés. Dans, [**Lin, 2019**] et [**Sculley et al., 2018**], les auteurs décrivent la pression actuelle dans le domaine de l'intelligence artificielle à publier des articles acceptés dans de grandes conférences et de la comparaison avec des modèles de références ayant une faible performance. Est-ce le cas pour [**Zhang et al., 2021**] avec leur modèle SCCL? Nous laissons cette question ouverte. Nous avons ensuite commencé nos expériences en validant l'intérêt d'utiliser K-Means au lieu de K-Medoids. Puis, nous avons

confirmé l'utilisation de la similarité cosinus par rapport à la distance euclidienne comme mesure de similarité entre les représentations vectorielles. Par la suite, nous avons observé que le modèle MPNet de SBERT semble offrir la meilleure représentation vectorielle pour le regroupement. Nous avons ensuite testé deux approches pour améliorer la représentation vectorielle SBERT, son affinage (TSDEA) et la réduction de dimensionnalité (UMAP). Pour TSDEA, nous n'avons pas obtenu des résultats intéressants en le comparant avec des modèles ayant une bonne performance initiale. En analysant l'article de référence [**Wang et al., 2021a**], nous remarquons que cette technique fut comparée avec des modèles de référence faibles. En effet, il n'y a pas de comparaison avec les meilleurs modèles de SBERT tels que RoBERTa ou MPNet. Pour ce qui est de UMAP, nous observons des résultats intéressants permettant d'atteindre de meilleurs résultats en regroupement. Nous observons également des gains de performance intéressant pour le prétraitement des données et l'ajout de la représentation vectorielle Tf-idf à la représentation vectorielle SBERT. À la lumière de ces trois expériences concluantes, nous établissons l'approche à préconiser pour le regroupement des données. Tout d'abord, le prétraitement des données permet un gain de performance avec ou sans UMAP ou Tf-idf. Ensuite, lorsque le corpus est déséquilibré, UMAP devrait être priorisé et pour un corpus balancé la concaténation de SBERT avec Tf-idf devrait être utilisée. En effet, on remarque que sur les corpus fortement déséquilibrés (GoogleNewsTS, GoogleNewsS, GoogleNewsT et Tweets), UMAP offre de meilleurs résultats alors que pour les corpus équilibrés (AgNews, StackOverflow, Biomedical, SearchSnippets) SBERT avec Tf-idf performe mieux. Finalement, grâce à ces deux approches, nous avons réussi à obtenir des résultats supérieurs à l'état de l'art sur quatre des huit corpus (StackOverflow, Biomedical, GoogleNewsT et GoogleNewsS) et des résultats à moins de 5 points NMI de l'état de l'art sur les autres corpus. De plus, nos approches ne requièrent l'entraînement d'aucun modèle, mais plutôt la combinaison de techniques existantes avec la plus récente représentation vectorielle contextuelle (SBERT).

	AgNews		SearchSnippets		StackOverflow		Biomedical	
	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI
Tf-idf	56,2	60,1	57,2	55,1	57,4	28,9	25,5	10,3
Prétrait.-Tf-idf	58,1	63,5	61,3	59,6	66,1	25,7	30,5	8,2
Word2Vec	7,7	4,2	18,6	3,5	60,0	30,4	27,0	9,8
HDBSCAN	20,5	0,6	21,5	0,1	4,1	0,8	19,0	0,4
HDBSCAN-UMAP	27,9	1,7	37,3	1,5	44,4	5,0	37,8	3,4
DistilBERT	34,0	36,2	31,3	26,5	51,0	34,7	33,9	20,3
DistilBERT-UMAP	49,7	52,4	46,2	43,3	54,7	41,8	33,7	21,1
RoBERTa	62,5	65,0	46,6	38,5	71,9	56,4	43,5	**29,8**
RoBERTa-TSDEA	61,8	64,7	57,3	53,1	61,3	51,1	37,7	26,6
RoBERTa-TSDEA-UMAP	53,7	51,7	64,4	61,4	73,0	66,9	38,3	26,5
MPNet	60,2	62,4	58,0	50,7	70,6	57,1	41,6	27,6
MPNet-UMAP	56,8	47,8	62,0	57,1	70,1	62,1	38,4	25,2
Prétrait.-MPNet	62,9	67,2	61,7	58,0	71,0	57,6	44,5	**29,8**
Prétrait.-MPNet-UMAP	56,5	52,8	67,7	**63,7**	70,2	**62,6**	40,2	27,4
Prétrait.-MPNet-Tf-idf	63,5	**67,8**	63,6	63,6	**78,9**	60,9	**46,6**	**29,8**
SCCL	**68,2**		**71,1**		74,5		41,5	

	GoogleNewsTS		GoogleNewsT		GoogleNewsS		Tweets	
	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI
Tf-idf	84,5	56,9	78,6	46,7	78,3	47,4	80,3	41,9
Prétrait.-Tf-idf	86,6	56,1	80,2	28,3	83,7	41,0	80,1	39,4
Word2Vec	89,1	57,0	80,3	27,6	84,3	45,5	78,6	38,3
HDBSCAN	81,1	18,7	75,1	10,0	76,4	11,0	79,5	41,5
HDBSCAN-UMAP	85,3	44,2	76,9	15,3	83,0	37,6	85,5	49,7
DistilBERT	87,5	61,9	79,0	48,9	81,3	52,9	79,8	43,2
DistilBERT-UMAP	90,7	71,2	84,4	60,3	86,0	64,8	84,1	48,4
RoBERTa	88,7	62,1	85,6	56,3	85,8	59,1	86,3	49,9
RoBERTa, TSDEA	92,2	71,5	85,7	64,4	85,5	62,5	86,1	67,4
RoBERTa-TSDEA-UMAP	93,5	75,5	87,9	64,9	88,3	65,9	84,4	45,7
MPNet	91,6	69,3	86,6	57,9	88,0	62,4	87,3	52,6
MPNet-UMAP	94,0	79,1	**89,9**	68,8	**91,1**	73,7	89,0	52,1
Prétrait.-MPNet	91,8	68.9	86,5	58,8	88,5	64,3	87,3	51,4
Prétrait.-MPNet-UMAP	94,1	**81,2**	89,0	**70,7**	91,0	**74,3**	88,6	**52,6**
Prétrait.-MPNet-Tf-idf	92,1	71,0	86,6	55,5	88,7	62,4	86,5	49,5
SCCL	**94,9**		88,3		90,4		**89,2**	

Tableau 5.3. Regroupement de textes à l'aide d'un algorithme sans connaissance du nombre de groupe (HDBSCAN) et à l'aide de k-Means sur trois modèles SBERT (RoBERTa, DistilBERT et MPNet). Application d'une technique de réduction de dimension (UMAP) et d'affinage (TSDEA). prétraitement (Prétrait.) des données sur le modèle HDBSCAN et MPNet. Code de couleur pour comparaison avec l'état de l'art, vert: résultats meilleurs, jaune: résultats inférieurs de moins de 5 points NMI, rouge: résultats inférieurs de plus de 5 points NMI.

	AgNews	Biomedical	SearchSnippets	StackOverflow	Tweets	GoogleNews (T,S,TS)
Mise en minuscule	x		x	x		
Suppression de mots vides	x	x	x	x	x	x
Suppression de mots vides spécifiques au corpus	x				x	x
Lemmatisation	x		x		x	x
Suppression de mots courts ($\leq$ 2 caractères)	x	x	x			
Suppression de mots avec fréquence ≤ 3	x					
Suppression de mots avec fréquence ≤ 2		x	x			
Suppression de mots avec fréquence ≤ 1					x	x
Suppression de urls (href)	x					
Suppression de html (&lt, &gt)	x					
Suppression de répétition de mots ou groupes de mots			x			

Tableau 5.4. Prétraitements supplémentaires effectués sur chacun des corpus

Conclusion

Nous sommes à l'ère où la quantité d'information présente sur internet augmente sans cesse. Au moment notre société fait face à de nombreux enjeux face à ce phénomène tels que la modération de contenu ou l'exploitation de ces données à différentes fins. Le regroupement de textes est généralement une des premières étapes en exploration de données afin de comprendre les principaux thèmes abordés dans une vaste collection de données. Les recherches récentes proposent d'entraîner les modèles en utilisant la représentation vectorielle contextuelle, mais une étude des techniques simples combinées à la représentation vectorielle contextuelle n'est pas effectuée.

Dans ce mémoire, nous avons proposé de faire le regroupement du texte avec la représentation vectorielle SBERT combinée à des techniques ne nécessitant pas d'entraînement, comme le prétraitement des données, la réduction de dimensionnalité UMAP ou l'inclusion de Tf-idf. Pour ce faire, nous avons tout d'abord étudié les métriques d'évaluation et proposé d'utiliser le NMI et l'ARI. Ensuite, nous avons tenté de reproduire les résultats du modèle état de l'art (SCCL), mais ce fut sans succès. Par la suite, nous avons analysé différents choix d'implémentations pour effectuer le regroupement. Nous avons validé le meilleur choix de mesure de similarité entre la similarité cosinus et la distance euclidienne avec des résultats légèrement meilleurs pour la similarité cosinus. Puis, nous avons validé le choix de l'algorithme de regroupement K-Means par rapport à K-Medoids. Ensuite, nous avons comparé trois modèles de représentation vectorielle SBERT. Nous avons évalué l'intérêt d'effectuer l'affinage de la représentation avec la technique TSDEA et avons rejeté cette idée. Puis, nous avons confirmé la pertinence de faire du prétraitement de données avec les modèles récents. Nous avons également effectué une analyse des corpus utilisés. Finalement, nous avons étudié la réduction de dimensionnalité UMAP et l'inclusion de Tf-idf à la représentation vectorielle SBERT.

Nos expériences ont montré qu'il est possible d'obtenir des résultats état de l'art sur la moitié des corpus et sur l'autre moitié nos résultats sont à moins de 5 points NMI de l'état de l'art. Nous proposons comme approche d'appliquer un prétraitement aux données avant de les représenter sous forme vectorielle. Lorsque les corpus sont déséquilibrés d'utiliser UMAP et pour les corpus balancés la concaténation de SBERT avec Tf-idf.

Quelques limites peuvent être soulignées dans notre approche. Tout d'abord, nos expériences ont été effectuées sur des textes courts avec une longueur moyenne maximale de 28 mots. L'efficacité de la représentation vectorielle SBERT et de notre approche sur des textes plus longs reste à valider. Ensuite, nous nous sommes concentrés sur le regroupement lorsque le nombre de groupes est connu au préalable. C'est d'ailleurs le cas de la plupart des travaux récents. Or, cette situation ne représente probablement pas des tâches réelles où ce nombre de groupes risque d'être inconnu. Comme mentionné par certains chercheurs dont Yoshua Bengio[10], il est temps de repenser la recherche rapide en intelligence artificielle afin de non seulement faire avancer la science, mais aussi résoudre des problèmes réels de notre société.

Travaux futurs

Un corpus supplémentaire, *20 Newsgroup*[11], pourrait être ajouté à la liste des corpus. Celui-ci fut populaire dans les recherches en regroupement de textes au début des années 2000 [**Banerjee et al., 2005, Dhillon et al., 2002, Slonim and Tishby, 2000**], mais fut ensuite oublié par les recherches plus récentes, rendant ainsi difficile la comparaison des nouvelles méthodes avec les anciennes.

Nous avons effectué une analyse entre les algorithmes de regroupement K-Means et K-Medoids, cette analyse pourrait être élargie à différentes variantes de ces algorithmes et à d'autres algorithmes de regroupement.

Nous avons comparé la similarité cosinus et la distance euclidienne, il serait intéressant d'élargir cette analyse à d'autres mesures de similarité. Aucune étude à notre connaissance n'existe sur les meilleures mesures de similarité à utiliser avec les représentations vectorielles telles que SBERT.

L'algorithme Tf-idf performe moins bien sur des corpus déséquilibrés, il serait intéressant d'étudier comment remédier à cette lacune et tester certaines solutions proposées telles que [**Jiang et al., 2021**]. De plus, nous avons utilisé une approche naïve pour combiner SBERT avec Tf-idf, d'autres approches pourraient être étudiées, peu de recherches existent sur ce sujet.

Il serait intéressant d'étudier l'influence du nombre de plus proches voisins à considérer pour la méthode de réduction de dimensionnalité UMAP. En effet, celui-ci fut fixe pour tous nos corpus, mais il devrait probablement être ajusté en fonction de la taille du corpus.

La littérature propose de nombreuses techniques de regroupement de textes, mais la plupart prennent pour hypothèse que le nombre de groupes est connu au préalable et ce fut

[10] `https://yoshuabengio.org/2020/02/26/time-to-rethink-the-publication-process-in-machine-learning/`

[11] `https://www.kaggle.com/datasets/crawford/20-newsgroups`

le cas dans nos expériences aussi. Certaines méthodes existent pour adapter ces techniques au cas où le nombre de groupes est inconnu, mais celles-ci sont coûteuses en temps de calcul comme le soulève [**Ronen et al., 2022**]. Or, dans la pratique le nombre de groupes n'est généralement pas connu et fait parti de la problématique. Nous devrions également chercher à trouver combien de groupes distincts il y a un dans un corpus au lieu de le donner en entrée aux algorithmes de regroupement. Les recherches futurs devraient prendre cette orientation.

Finalement, en tant que communauté scientifique, nous pouvons continuer à travailler sur de meilleures représentations vectorielles avec des modèles toujours plus grands. Par contre, peut-être que nous devrions repenser complètement la façon d'effectuer les tâches et que la prochaine frontière réside dans des modèles capables de raisonner [**LeCun, 2022, Bakhtin et al., 2022**].

Références bibliographiques

[Aggarwal and Reddy, 2013] Aggarwal, C. and Reddy, C. (2013). DATA CLUSTERING Algorithms and Applications.

[Aggarwal and Zhai, 2012] Aggarwal, C. C. and Zhai, C. (2012). A Survey of Text Clustering Algorithms, pages 77–128. Springer US, Boston, MA.

[Antoniak, 1974] Antoniak, C. E. (1974). Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems. The Annals of Statistics, 2(6):1152 – 1174.

[Arora et al., 2017a] Arora, S., Liang, Y., and Ma, T. (2017a). A simple but tough-to-beat baseline for sentence embeddings. In ICLR.

[Arora et al., 2017b] Arora, S., Liang, Y., and Ma, T. (2017b). A simple but tough-to-beat baseline for sentence embeddings. In ICLR.

[Bakhtin et al., 2022] Bakhtin, A., Brown, N., Dinan, E., Farina, G., Flaherty, C., Fried, D., Goff, A., Gray, J., Hu, H., Jacob, A. P., Komeili, M., Konath, K., Kwon, M., Lerer, A., Lewis, M., Miller, A. H., Mitts, S., Renduchintala, A., Roller, S., Rowe, D., Shi, W., Spisak, J., Wei, A., Wu, D., Zhang, H., and Zijlstra, M. (2022). Human-level play in the game of diplomacy by combining language models with strategic reasoning. Science.

[Banerjee et al., 2005] Banerjee, A., Krumpelman, C., Ghosh, J., Basu, S., and Mooney, R. J. (2005). Model-based overlapping clustering. In Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, KDD '05, page 532–537, New York, NY, USA. Association for Computing Machinery.

[Bengio et al., 2006] Bengio, Y., Delalleau, O., and Roux, N. (2006). Label propagation and quadratic criterion. Semi-Supervised Learning.

[Bengio et al., 2003] Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. J. Mach. Learn. Res., 3(null):1137–1155.

[Bianchi et al., 2021] Bianchi, F., Terragni, S., and Hovy, D. (2021). Pre-training is a hot topic: Contextualized document embeddings improve topic coherence. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 759–766, Online. Association for Computational Linguistics.

[Blömer et al., 2016] Blömer, J., Lammersen, C., Schmidt, M., and Sohler, C. (2016). Theoretical Analysis of the k-Means Algorithm – A Survey, pages 81–116. Springer International Publishing, Cham.

[Bouveyron et al., 2007] Bouveyron, C., Girard, S., and Schmid, C. (2007). High-dimensional data clustering. Computational Statistics Data Analysis, 52(1):502–519.

[Cabannes et al., 2021] Cabannes, V., Pillaud-Vivien, L., Bach, F., and Rudi, A. (2021). Overcoming the curse of dimensionality with laplacian regularization in semi-supervised learning. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, Advances in Neural Information Processing Systems, volume 34, pages 30439–30451. Curran Associates, Inc.

[Cai et al., 2005] Cai, D., He, X., and Han, J. (2005). Document clustering using locality preserving indexing. IEEE Transactions on Knowledge and Data Engineering, 17(12):1624–1637.

[Campello et al., 2013] Campello, R. J. G. B., Moulavi, D., and Sander, J. (2013). Density-based clustering based on hierarchical density estimates. In Pei, J., Tseng, V. S., Cao, L., Motoda, H., and Xu, G., editors, Advances in Knowledge Discovery and Data Mining, pages 160–172, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Carlsson et al., 2021] Carlsson, F., Gyllensten, A. C., Gogoulou, E., Hellqvist, E. Y., and Sahlgren, M. (2021). Semantic re-tuning with contrastive tension. In International Conference on Learning Representations.

[Cer et al., 2018] Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Sung, Y.-H., Strope, B., and Kurzweil, R. (2018). Universal sentence encoder.

[Chen, 2015] Chen, G. (2015). Deep learning with nonparametric clustering.

[Chen et al., 2011] Chen, W.-Y., Song, Y., Bai, H., Lin, C.-J., and Chang, E. Y. (2011). Parallel spectral clustering in distributed systems. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(3):568–586.

[Clark et al., 2020] Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators.

[de Souto et al., 2012] de Souto, M. C., Coelho, A. L., Faceli, K., Sakata, T. C., Bonadia, V., and Costa, I. G. (2012). A comparison of external clustering evaluation indices in the context of imbalanced data sets. In 2012 Brazilian Symposium on Neural Networks, pages 49–54.

[Devlin et al., 2018] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding.

[Dhillon et al., 2002] Dhillon, I. S., Mallela, S., and Kumar, R. (2002). Enhanced word clustering for hierarchical text classification. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '02, page 191–200, New York, NY, USA. Association for Computing Machinery.

[Domingos, 2012] Domingos, P. (2012). A few useful things to know about machine learning. Commun. ACM, 55(10):78–87.

[Ester et al., 1996a] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996a). A density-based algorithm for discovering clusters in large spatial databases with noise. In KDD.

[Ester et al., 1996b] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996b). 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. AAAI Press.

[Fahad et al., 2014] Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A. Y., Foufou, S., and Bouras, A. (2014). A survey of clustering algorithms for big data: Taxonomy and empirical analysis. IEEE Transactions on Emerging Topics in Computing, 2:267–279.

[Gao et al., 2021] Gao, T., Yao, X., and Chen, D. (2021). Simcse: Simple contrastive learning of sentence embeddings.

[Garg and Jain, 2006] Garg, S. and Jain, R. (2006). Variations of k-mean algorithm: A study for high-dimensional large data sets. Information Technology Journal.

[Guan et al., 2020] Guan, R., Zhang, H., Liang, Y., Giunchiglia, F., Huang, L., and Feng, X. (2020). Deep feature-based text clustering and its explanation. IEEE Transactions on Knowledge and Data Engineering, PP:1–1.

[Hadifar et al., 2019] Hadifar, A., Sterckx, L., Demeester, T., and Develder, C. (2019). A self-training approach for short text clustering. In Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019), pages 194–199, Florence, Italy. Association for Computational Linguistics.

[Hall et al., 2005] Hall, P., Marron, J. S., and Neeman, A. (2005). Geometric representation of high dimension, low sample size data. Journal of the Royal Statistical Society. Series B (Statistical Methodology), 67(3):427–444.

[He et al., 2020] He, P., Liu, X., Gao, J., and Chen, W. (2020). Deberta: Decoding-enhanced bert with disentangled attention.

[Hoffer and Ailon, 2014] Hoffer, E. and Ailon, N. (2014). Deep metric learning using triplet network.

[Huang et al., 2014] Huang, P., Huang, Y., Wang, W., and Wang, L. (2014). Deep embedding network for clustering. 2014 22nd International Conference on Pattern Recognition, pages 1532–1537.

[Hubert and Arabie, 1985] Hubert, L. and Arabie, P. (1985). Comparing partitions. Journal of Classification, 2(1):193–218.

[Jain et al., 1999] Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: A review. ACM Comput. Surv., 31(3):264–323.

[Jiang et al., 2021] Jiang, Z.-Y., Gao, B., He, Y., Han, Y., Doyle, P., and Zhu, Q. (2021). Text classification using novel term weighting scheme-based improved tf-idf for internet media reports. Mathematical Problems in Engineering, 2021:1–30.

[Jin and Han, 2010] Jin, X. and Han, J. (2010). K-Medoids Clustering, pages 564–565. Springer US, Boston, MA.

[Kaufman and Rousseeuw, 1990] Kaufman, L. and Rousseeuw, P. (1990). Finding Groups in Data: An Introduction To Cluster Analysis.

[Kotsiantis and Pintelas, 2004] Kotsiantis, S. and Pintelas, P. (2004). Recent advances in clustering: A brief survey. WSEAS Transactions on Information Science and Applications, 1:73–81.

[LeCun, 2022] LeCun, Y. (2022). A path towards autonomous machine intelligence. preprint posted on openreview.

[Li et al., 2020] Li, B., Zhou, H., He, J., Wang, M., Yang, Y., and Li, L. (2020). On the sentence embeddings from pre-trained language models. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 9119–9130, Online. Association for Computational Linguistics.

[Li et al., 2018] Li, C., Guo, J., Lu, Y., Wu, J., Zhang, Y., Xia, Z., Wang, T., Yu, D., Chen, X., and Liu, P. (2018). Lda meets word2vec: A novel model for academic abstract clustering. pages 1699–1706.

[Liao, 2021] Liao, D. (2021). Sentence embeddings using supervised contrastive learning.

[Lin, 2019] Lin, J. (2019). The neural hype and comparisons against weak baselines. SIGIR Forum, 52(2):40–51.

[Liu et al., 2019] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach.

[Ma, 2019] Ma, E. (2019). Nlp augmentation. https://github.com/makcedward/nlpaug.

[MacQueen, 1967] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations.

[May et al., 2019] May, C., Wang, A., Bordia, S., Bowman, S. R., and Rudinger, R. (2019). On measuring social biases in sentence encoders.

[McInnes et al., 2018] McInnes, L., Healy, J., and Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction.

[Mikolov et al., 2013] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space.

[Naseem et al., 2020] Naseem, U., Razzak, I., Khan, S. K., and Prasad, M. (2020). A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models.

[Park and Jun, 2009] Park, H.-S. and Jun, C.-H. (2009). A simple and fast algorithm for k-medoids clustering. Expert Systems with Applications, 36(2, Part 2):3336–3341.

[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), pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

[Peña et al., 1999] Peña, J., Lozano, J., and Larrañaga, P. (1999). An empirical comparison of four initialization methods for the k-means algorithm. Pattern Recognition Letters, 20(10):1027–1040.

[Phan et al., 2008] Phan, X.-H., Nguyen, L., and Horiguchi, S. (2008). Learning to classify short and sparse text web with hidden topics from large-scale data collections. pages 91–100.

[Qiao et al., 2019] Qiao, Y., Xiong, C., Liu, Z., and Liu, Z. (2019). Understanding the behaviors of bert in ranking.

[Rakib et al., 2020a] Rakib, M. R. H., Zeh, N., Jankowska, M., and Milios, E. (2020a). Enhancement of short text clustering by iterative classification.

[Rakib et al., 2020b] Rakib, M. R. H., Zeh, N., Jankowska, M., and Milios, E. E. (2020b). Enhancement of short text clustering by iterative classification. CoRR, abs/2001.11631.

[Rand, 1971] Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association, 66:846–850.

[Reimers and Gurevych, 2019] Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks.

[Ronen et al., 2022] Ronen, M., Finder, S. E., and Freifeld, O. (2022). Deepdpm: Deep clustering with an unknown number of clusters.

[Sammut and Webb, 2010] Sammut, C. and Webb, G. I., editors (2010). TF–IDF, pages 986–987. Springer US, Boston, MA.

[Sculley et al., 2018] Sculley, D., Snoek, J., Wiltschko, A., and Rahimi, A. (2018). Winner's curse? on pace, progress, and empirical rigor.

[Sherstinsky, 2020] Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D: Nonlinear Phenomena, 404:132306.

[Slonim and Tishby, 2000] Slonim, N. and Tishby, N. (2000). Document clustering using word clusters via the information bottleneck method. In Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '00, page 208–215, New York, NY, USA. Association for Computing Machinery.

[Sohangir and Wang, 2017] Sohangir, S. and Wang, D. (2017). Improved sqrt-cosine similarity measurement. Journal of Big Data, 4:25.

[Song et al., 2020] Song, K., Tan, X., Qin, T., Lu, J., and Liu, T.-Y. (2020). Mpnet: Masked and permuted pre-training for language understanding.

[Steiglitz, 1982] Steiglitz, K. (1982). Combinatorial Optimization: Algorithms and Complexity, volume 32.

[Strehl and Ghosh, 2002] Strehl, A. and Ghosh, J. (2002). Cluster ensembles - a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research, 3:583–617.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need.

[Wang et al., 2021a] Wang, K., Reimers, N., and Gurevych, I. (2021a). Tsdae: Using transformer-based sequential denoising auto-encoder for unsupervised sentence embedding learning.

[Wang et al., 2021b] Wang, Z., Ni, Y., Jing, B., Wang, D., Zhang, H., and Xing, E. (2021b). Dnb: A joint learning framework for deep bayesian nonparametric clustering. IEEE Transactions on Neural Networks and Learning Systems, pages 1–11.

[Willett, 1988] Willett, P. (1988). Recent trends in hierarchic document clustering: A critical review. Information Processing Management, 24(5):577–597.

[Wu et al., 2007] Wu, J., Xiong, H., Chen, J., and Zhou, W. (2007). A generalization of proximity functions for k-means. In Seventh IEEE International Conference on Data Mining (ICDM 2007), pages 361–370.

[Wu and Schölkopf, 2006] Wu, M. and Schölkopf, B. (2006). A local learning approach for clustering. In Schölkopf, B., Platt, J., and Hoffman, T., editors, Advances in Neural Information Processing Systems, volume 19. MIT Press.

[Wu et al., 2016] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, , Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation.

[Xiong et al., 2009] Xiong, H., Wu, J., and Chen, J. (2009). K-means clustering versus validation measures: A data-distribution perspective. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 39(2):318–331.

[Xu and Tian, 2015] Xu, D. and Tian, Y. (2015). A comprehensive survey of clustering algorithms. Annals of Data Science, 2.

[Xu et al., 2017a] Xu, J., Xu, B., Wang, P., Zheng, S., Tian, G., and Zhao, J. (2017a). Self-taught convolutional neural networks for short text clustering. CoRR, abs/1701.00185.

[Xu et al., 2017b] Xu, J., Xu, B., Wang, P., Zheng, S., Tian, G., Zhao, J., and Xu, B. (2017b). Self-taught convolutional neural networks for short text clustering. Neural Networks, 88:22–31.

[Xu et al., 2003] Xu, W., Liu, X., and Gong, Y. (2003). Document clustering based on non-negative matrix factorization. In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, SIGIR '03, page 267–273, New York, NY, USA. Association for Computing Machinery.

[Yin and Wang, 2014] Yin, J. and Wang, J. (2014). A dirichlet multinomial mixture model-based approach for short text clustering. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, page 233–242, New York, NY, USA. Association for Computing Machinery.

[Yin and Wang, 2016] Yin, J. and Wang, J. (2016). A model-based approach for text clustering with outlier detection. In 2016 IEEE 32nd International Conference on Data Engineering (ICDE), pages 625–636.

[Zhang et al., 2021] Zhang, D., Nan, F., Wei, X., Li, S., Zhu, H., McKeown, K., Nallapati, R., Arnold, A., and Xiang, B. (2021). Supporting clustering with contrastive learning.

[Zhang et al., 2019] Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2019). Bertscore: Evaluating text generation with bert.

[Zhang et al., 2015] Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification.

Annexe A

Modèles SBERT

Model Name	Performance Sentence Embeddings (14 Datasets)	Performance Semantic Search (6 Datasets)	Avg. Performance	Speed	Model Size
all-mpnet-base-v2	69.57	57.02	63.30	2800	420 MB
gtr-t5-xxl	70.73	55.76	63.25	50	9230 MB
gtr-t5-xl	69.88	55.88	62.88	230	2370 MB
sentence-t5-xxl	70.88	54.40	62.64	50	9230 MB
gtr-t5-large	69.90	54.85	62.38	800	640 MB
all-mpnet-base-v1	69.98	54.69	62.34	2800	420 MB
multi-qa-mpnet-base-dot-v1	66.76	57.60	62.18	2800	420 MB
multi-qa-mpnet-base-cos-v1	66.29	57.46	61.88	2800	420 MB
all-roberta-large-v1	70.23	53.05	61.64	800	1360 MB
sentence-t5-xl	69.23	51.19	60.21	230	2370 MB
all-distilroberta-v1	68.73	50.94	59.84	4000	290 MB
all-MiniLM-L12-v1	68.83	50.78	59.80	7500	120 MB
all-MiniLM-L12-v2	68.70	50.82	59.76	7500	120 MB
multi-qa-distilbert-dot-v1	66.67	52.51	59.59	4000	250 MB

Fig. A.1. Principaux modèles SBERT disponibles pour la représentation vectorielle du texte.

Annexe B

Exemples sur le corpus GoogleNewsTS de groupes définis par la présence de certains mots

Fig. B.1. Groupe défini par la présence de *grand theft auto*.

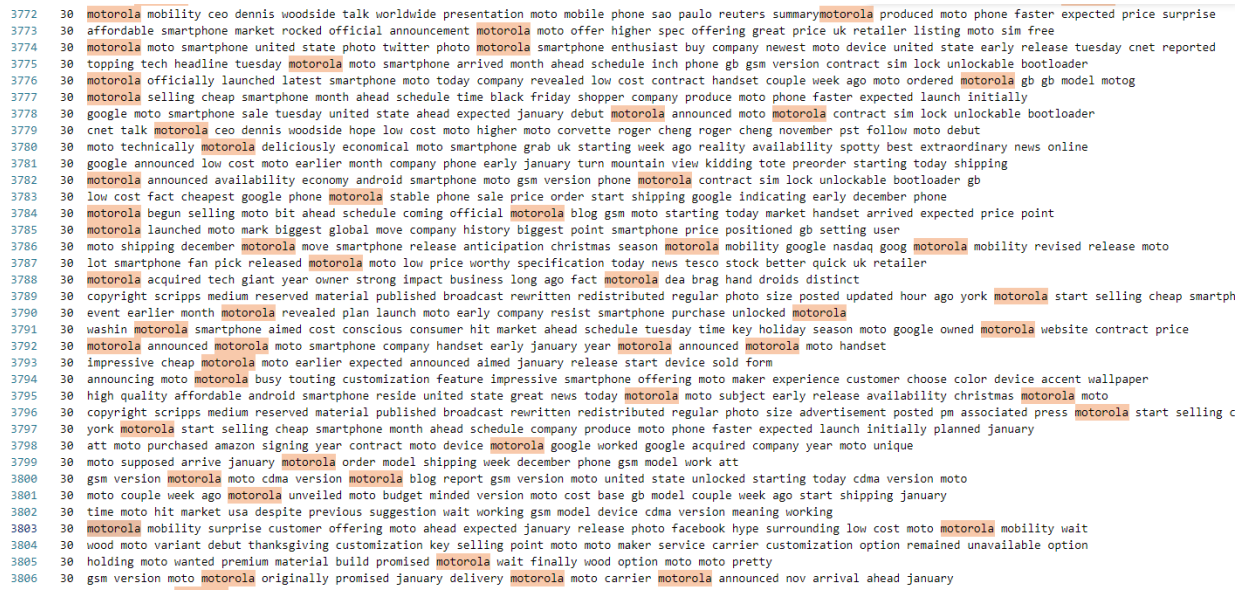


Fig. B.2. Groupe défini par la présence de *motorola*.

Fig. B.3. Groupe défini par la présence de *taylor swift*.

Annexe C

Exemples de données prétraitées pour chacun des corpus

```
INITIAL: Brazilian Soldier Wounded in Haiti Unrest &lt;p&gt;&lt;/p&gt;&lt;p&gt; By Joseph Guyler Delva&lt;/p&gt;&lt;p&gt;
GONAIVES, Haiti (Reuters) - A Brazilian soldier with theU.N. peacekeeping force in Haiti was shot and wounded
Saturdaywhen peacekeepers and local police faced gunfire in a crackdownon armed gangs in the Haitian capital.&lt;/p&gt;
PRÉTRAITÉ: brazilian soldier wounded haiti unrest brazilian soldier peacekeeping force haiti shot wounded
peacekeeper local police faced armed gang haitian capital

INITIAL: Revlon 3rd-Quarter Loss Widens  CHICAGO (Reuters) - Cosmetics maker Revlon Inc. &lt;A
HREF="http://www.investor.reuters.com/FullQuote.aspx?ticker=REV.N target=/stocks/quickinfo/fullquote"&gt;REV.N&lt;/A&gt;
on  Wednesday said its third-quarter net loss widened on  refinancing costs and declining sales.
PRÉTRAITÉ: loss widens cosmetic maker inc net loss widened cost declining sale

INITIAL: Oracle Q1 net up 16 Net income in the fiscal first-quarter rose to \$509 million,
from \$440 million a year earlier, while revenue rose seven percent to \$2.
PRÉTRAITÉ: oracle net net income rose million million revenue rose seven percent
```

Fig. C.1. Exemples de données prétraitées sur AgNews.

```
INITIAL:   median age at onset of asthma and allergic rhinitis in tecumseh michigan
PRÉTRAITÉ: age onset asthma allergic

INITIAL:  somatotopic localization in cat motor cortex
PRÉTRAITÉ:localization cat motor cortex

INITIAL: pharmacokinetic study of maleate acid of 2 n n dimethylaminoethanol 14c1 cyclohexylpropionate
cyprodenate and of n n dimethylaminoethanol 14c1 in animals
PRÉTRAITÉ: study acid animals
```

Fig. C.2. Exemples de données prétraitées sur Biomedical.

```
INTIAL:    xbox owner rage hdmi snafu judders brit euro telly
PRÉTRAITÉ: xbox owner rage euro

INTIAL:    stem cell transplant repair damaged gut mouse model inflammatory bowel
PRÉTRAITÉ: stem cell gut model inflammatory

INTIAL: kilo mach sonic boom probed fireball ember ad supernova
PRÉTRAITÉ: mach boom ad supernova
```

Fig. C.3. Exemples de données prétraitées sur GoogleNewsT.

```
INTIAL:   paleontologist working alberta unearthed complete fossilized skeleton tiny rhinoceros dinosaur
guy year died pristine condition providing scientist clue
PRÉTRAITÉ:working alberta unearthed complete skeleton tiny dinosaur guy died condition providing scientist

INTIAL: toddler year foot meter long wandered river alberta canada drowned year ago beast preserved
skin left impression nearby rock fossil
PRÉTRAITÉ: toddler foot meter long river alberta canada ago beast skin left impression nearby rock fossil

INTIAL:lindsey vonn partially tore acl day ago chance competing sochi olympics day doubt lot left achieve season day day time
PRÉTRAITÉ:lindsey vonn partially tore acl ago chance competing sochi olympics doubt lot left achieve season
```

Fig. C.4. Exemples de données prétraitées sur GoogleNewsS.

```
INTIAL:    kanye west super sized rant madison square garden going year single interview kanye west medium blitz week interesting
thing heard west interview bret easton elli zane lowe jimmy kimmel sway
PRÉTRAITÉ: kanye west super sized rant madison square garden going single interview kanye west medium blitz week interesting
thing heard west interview jimmy kimmel

INTIAL:   scammer aim black friday cyber monday ti season cyberscams stacking unprecedented plunder cybergrinches loading
post facebook scammer aim black friday cyber monday wkyc incorrect
PRÉTRAITÉ:aim black cyber season stacking unprecedented loading post facebook aim black cyber incorrect

INTIAL:    wintry blast hit west killed storm head east samantha hernandez scrape ice window kenneth field spray concoction vinegar
water soften ice saturday odessa texas edyta blaszczyk ap albuquerque stormy weather west
PRÉTRAITÉ: wintry blast hit west killed storm head east samantha ice window field spray water ice saturday texas stormy weather west
```

Fig. C.5. Exemples de données prétraitées sur GoogleNewsTS.

```
INTIAL: wikipedia wiki labor party israel labor party israel wikipedia encyclopedia israeli labor party hebrew
מפלגת העבודה הישראלית[U+200E] mifleget haavoda hayisraelit israel avoda hebrew עבודה[U+200E] center-left
PRÉTRAITÉ:labor party israel center left

INTIAL:    bkgm rgb rgb cgi backgammon fun frustration danish championship semifinal match match score peter friis jensen lars trabolt
crawford game rolls
PRÉTRAITÉ: championship semifinal match score peter game roll

INTIAL:    mapsofworld fifa world cup world cup final germany world cup soccer world cup fifa world match germany fifa world cup
berlin olympic stadium berlin olympic stadium reconstructed
PRÉTRAITÉ: fifa world final germany world soccer world fifa world match germany fifa world berlin olympic stadium
```

Fig. C.6. Exemples de données prétraitées sur SearchSnippets.

```
INTIAL: Magento : Call to a member function count() on a non-object
PRÉTRAITÉ: magento call member function count object

INTIAL: Want to add custom option from the frontend of magento
PRÉTRAITÉ: want custom option frontend magento

INTIAL: Import & modify data in Matlab
PRÉTRAITÉ: import modify data matlab
```

Fig. C.7. Exemples de données prétraitées sur StackOverflow.

```
INTIAL:    christina aguilera national anthem politicsthe votechristina aguilera national anthem flubbed itth
PRÉTRAITÉ: christina aguilera national anthem aguilera national anthem flubbed

INTIAL: tx fishing news coolfishing day guna fishing
PRÉTRAITÉ: fishing news day fishing

INTIAL: antonin scalia charm tea party caucus simmi aujla politico
PRÉTRAITÉ: antonin scalia tea party caucus politico
```

Fig. C.8. Exemples de données prétraitées sur Tweets.