

1 Introduction

Le clustering consiste en un apprentissage non supervisé visant à caractériser la distribution des données et les relations entre les différentes données. Effectivement, nous n'avons aucune connaissance sur les données a priori de notre analyse. Le clustering est la méthode la plus populaire de l'apprentissage non supervisée et est souvent catégorisé comme une classification automatique des données sur lesquelles on l'applique. Son objectif principal est de classer nos données en plusieurs sous-ensembles d'objets. Nous allons commencer par présenter notre dataset, nos méthodes de prétraitement, notre méthodologie, notre analyse des données et une discussion concernant les avantages et désavantages de nos modèles ainsi que des améliorations qui peuvent être apportées à nos modèles.

En ce qui concerne notre méthodologie, notre méthode naïve consiste à utiliser l'algorithme de K-means avec un nombre de clusters aléatoire de 100 et en lui faisant passer une matrice TF-IDF ne prenant en compte que les unigrammes. Nous avons amélioré notre premier modèle en trouvant le nombre de clusters grâce à la méthode du coude avec une représentation graphique de l'inertie. De plus, nous utilisons une matrice TF-IDF contenant les bigrammes et trigrammes à dimension réduite en utilisant LSA. Après avoir obtenu les résultats, nous avons décidé d'utiliser une méthode plus complexe qui consiste à utiliser LDA. Celle-ci permet de réduire la dimension de notre problème tout en effectuant une analyse plus efficace et réaliste que K-means. En expliquant nos différentes manières d'analyser nos résultats, nous expliquerons en quoi notre modèle LDA est meilleur que les autres.

2 Présentation du dataset

Nous avons choisi d'extraire 500 000 tweets pour ne pas causer une surcharge de mémoire. De plus, on remarque que ces tweets sont assez bien répartis dans le temps. Nous pouvons donc avoir une bonne représentation de l'ensemble de données.

En ce qui concerne les features choisis, nous avons décidé d'en prendre plus que ce dont nous avons besoin. Cela nous permettrait d'avoir plus de choix et plus de flexibilité dans notre méthodologie et dans notre analyse. Nous avons décidé de garder les attributs text, [user, location], [user, screen_name], [user, name], created_at, lang, retweet_count, favorite_count et [user, description]. En regardant dans notre fichier csv hydraté, nous avons pu constater que nous avons uniquement 484 884 tweets. Cela s'explique par le fait que plusieurs comptes ont été suspendus ou supprimés et des tweets supprimés.

Pour notre étude, nous allons uniquement utiliser les features text ,created_at et lang.

3 Preprocessing

3.1 Retrait des tweets non-anglais

Au départ, nous retirons les tweets qui ne sont pas anglais, puisque les étapes de pré-traitement diffèrent d'un langage à un autre. Nous trouvons que nous avons assez de tweets en anglais afin d'appliquer un modèle de clustering.

3.2 Pipeline

En ce qui concerne les méthodes de prétraitement de données, nous avons un pipeline composé de plusieurs étapes: tokenization, retrait des stopwords, stemming, retrait de la ponctuation, des liens http, des mots RT, des #, des @et des emojis.

La tokenization est très utile puisque cette étape nous permettra de séparer les différents tweets en tokens. Nous avons décidé d'utiliser le TweetTokenizer de la librairie NLTK. Ce tokenizer est spécialisé pour les données de Twitter. Il possède des paramètres afin de retirer des répétitions de lettres dans les mots.

Le retrait des stopwords (mots vides) est utile, car il retire plusieurs mots insignifiants du vocabulaire. Effectivement, ces mots sont peu significatifs puisqu'ils sont peu susceptibles de changer et comprennent majoritairement des mots de classes fermées. Ils n'ont pas d'impact sur le contexte d'un tweet. Nous utilisons la liste prédéfinie des stopwords anglais de NLTK.

Nous avons choisi de stemmatiser nos tweets puisque ce qui nous importe dans cette analyse sont les concepts. Cela fait en sorte que les mots tels que « work » et « working » représentent le même concept. On se retrouve alors avec les racines des mots ce qui réduit la taille de notre vocabulaire. Ces dernières sont assez pour indiquer le contexte d'un tweet.

Nous avons aussi choisi de retirer toutes les ponctuations, les liens http, les mots RT, les hashtags #, les mentions @et les emojis puisqu'après la tokenisation, tous ces éléments représentent des tokens d'un seul caractère. Cela augmente la dimensionnalité sans aider à prédire le contexte d'un tweet. En ce qui concerne les emojis, nous importons la librairie emoji et vérifions qu'aucun caractère de notre corpus ne contient un caractère avec un Unicode d'emojis.

4 Méthodologie

4.1 Vectorisation de nos données en utilisant TF-IDF

Nous avons décidé de représenter nos tweets avec la matrice TF-IDF. Nous utilisons le vectorizer de sklearn `TfidfVectorizer` en appelant la méthode `fit_transform` et en lui passant notre corpus prétraité. Ce vectorizer normalise le nombre de tokens en réduisant l'importance (IDF) des mots les plus fréquents dans les tweets et augmente celle des mots les moins fréquents. Nous précisons un vocabulaire d'une taille de 8000 mots contenant les mots ayant les fréquences les plus élevées.

4.2 Algorithme K-means

Le bon nombre de clusters est celui que te permet de raconter une histoire sur tes données. Alors, en se basant sur notre jugement nous avons choisi 100 comme nombre de clusters. Nous avons utilisé un modèle K-Means de la librairie sklearn, avec `n_clusters` à 100. Le paramètre `random_state` est mis à 42 afin d'initialiser les centroïdes aux mêmes endroits à chaque fois que cette partie du code est exécutée. Cela éviterait que nous ayons des résultats différents à chaque fois que nous exécutons ce modèle. Nous effectuons le clustering en appelant la méthode `fit()` avec notre matrice TF-IDF générée sur un vocabulaire des 8000 mots les plus fréquents de notre corpus.

4.3 Améliorations

4.3.1 TF-IDF avec ngrams

Nous avons décidé de prendre en compte les bigrammes et les trigrammes dans notre matrice TF-IDF. Cela permettrait de trouver le contexte des mots au lieu de ne chercher que les unigrammes. Par exemple, « New York » serait interprétée comme une ville et non « new » et « york » qui n'ont plus le même sens du tout une fois séparé.

4.3.2 Nombre de clusters idéal

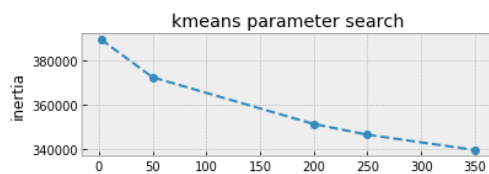


Figure 1: Clusters pour la matrice TF-IDF

Pour trouver le nombre de clusters idéal, nous avons effectué une représentation graphique de notre inertie sur plusieurs modèles k-means avec un différent nombre de clusters. On représente ensuite l'inertie en fonction du nombre de clusters pour trouver le nombre optimal de cluster à l'aide de la méthode du coude. Effectivement, l'inertie indique à quel point les tweets dans un cluster sont proches. Le coude est indiqué par la valeur de l'inertie qui devient peu significative pour les clusters de plus. En regardant la figure 1, nous pouvons voir que notre coude est à 200.

4.3.3 Réduction de dimensionnalité

Nous avons décidé d'utiliser une technique de réduction de la dimensionnalité. Effectivement, chacun de nos tweets est représenté par un vecteur de taille 8000. Il s'agit de nos 8000 ngrams. Il s'agit d'un volume énorme dans l'espace ce qui fait en sorte que nos données sont isolées et éparpillées. Nous décidons de réduire la dimension de notre matrice TF-IDF afin d'obtenir une matrice moins clairsemée. Effectivement, il est très difficile d'effectuer des calculs de distance de similarité entre les vecteurs. Alors, nous avons décidé d'utiliser LSA (Latent Semantic analysis) à l'aide de `TruncatedSVD` de sklearn. Avant d'effectuer la réduction de dimensionnalité, on applique `StandardScaler` sur la matrice TF-IDF. Cela permet de normaliser les caractéristiques de la matrice en supprimant la moyenne et en adaptant la variance à l'unité. Ensuite, lors de l'application de `TruncatedSVD` sur la matrice standardisée, la dimension précisée est de 100.

4.3.4 Allocation de Dirichlet latente (LDA)

LDA est un modèle génératif probabiliste permettant d'expliquer des ensembles de données en se basant sur des données de groupements non-observés qui sont reconnus pour la similarité de leurs données. Il s'agit de fixer un nombre de thèmes et l'algorithme cherche à apprendre les thèmes de chaque tweet, ainsi que les mots associés à ces thèmes. Initialement, le thème associé à un tweet n'est pas très bon. Pour chacun de nos documents (tweets), LDA prend chaque mot du tweet et met à jour le thème lié. Pour chaque mot d'un tweet, deux probabilités sont calculées: qu'un tweet soit assigné à un certain thème et qu'un thème soit associé à un mot. En multipliant ces deux probabilités, on obtient la probabilité qu'un certain thème génère un mot dans un tweet. Effectivement, deux matrices sont produites à

partir d'un Bag-of-Words passé en entrée, soit une matrice tweet par thème et une des mots par thème. LDA utilise CountVectorizer de sklearn qui représente chaque tweet par un vecteur indiquant la fréquence de chaque mot dans le tweet. Le constructeur LDA prend en paramètres `n_components`, qui est le nombre de thèmes que nous souhaitons extraire de nos données et `learning_method`, qui prend comme valeur `online` ou `batch`. Avec la valeur `online`, cela permet d'utiliser une méthode Bayes variationnelle permettant de mettre à jour l'attribut `components` de façon progressive en se basant sur le taux d'apprentissage. Nous avons décidé de mettre `n_components` à 200 et de comparer `online` et `batch`. Comme on peut voir dans les figures 2 et 3, les clusters, représentés par les bulles, sont plus dispersés avec `online` et se chevauchent beaucoup plus avec `batch`. On choisit donc la valeur `online`. Par la suite, nous avons décidé d'attribuer à chaque tweet le topic auquel il a la plus grande probabilité d'association. Nous avons déterminé ces attributs après avoir effectué plusieurs tests avec des valeurs de paramètres différents.

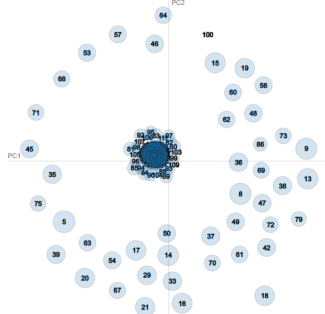


Figure 2: Dispersion des topics pour LDA avec online

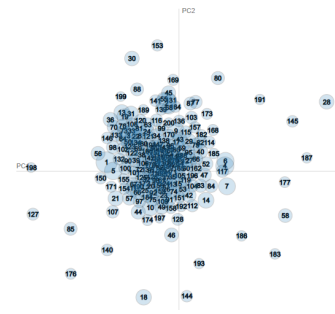


Figure 3: Dispersion des topics pour LDA avec batch

5 Résultats

5.1 Modèle naïf

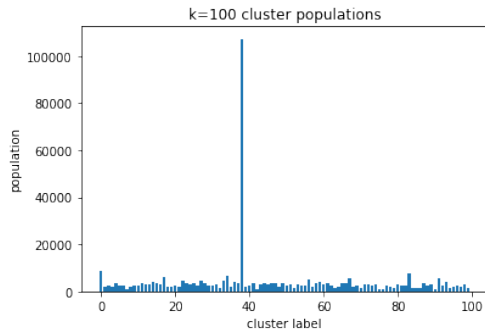


Figure 4

Afin d'analyser nos résultats, nous avons implémenté une fonction qui retourne les 15 mots les plus fréquents dans chaque cluster. Une autre fonction a été implémentée afin de voir quelques tweets d'un cluster particulier. En regardant la figure 4, on remarque que le cluster 36 contient près de 115000 tweets soit près de 30% des tweets. En regardant certains tweets de ce cluster (voir annexe 1 et 2 pour les résultats), on remarque qu'ils n'ont aucun lien entre eux. Il y a des nouvelles sur ISIS, la mort de Castro, le Congrès américain, ... Nous avons regardé un autre cluster choisi aléatoirement (cluster 87 avec 2311 tweets) et nous remarquons que tous les tweets contiennent le mot "American", bien qu'ils parlent de sujets différents. Nous pouvons conclure que ce modèle n'est pas le plus efficace.

5.2 Premier modèle amélioré

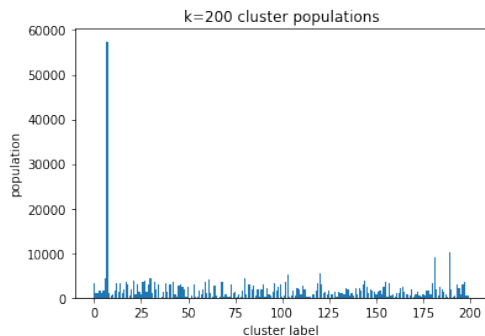


Figure 5

Encore une fois, nous avons fait l'analyse de la population de chaque cluster. En regardant la figure 5, on remarque que notre sommet contient environ 55000 tweets. Ce qui est une très grande amélioration comparé à notre modèle précédent. Ensuite, si on regarde les résultats du cluster 8 (voir Annexe 3), on remarque que les tweets sont beaucoup plus similaires entre eux. En effet, ils ont tous un rapport avec "High School". Cela peut être expliqué par le fait que nous prenons en compte les bigrammes.

5.3 Second modèle amélioré

Dans le but de voir nos 200 thèmes principaux de nos tweets, nous avons affiché les dix mots les plus représentatifs de chaque thème en utilisant la fonction `display_topics`. Ensuite, nous avons décidé d'imprimer la distribution des thèmes à travers les tweets en affichant les probabilités que le tweet soit assigné à chacun des thèmes trouvés. On peut voir un aperçu du résultat dans le tableau de la figure 6. Dans ce tableau, on peut voir que les tweets 0, 2 et 5 font tous partis du topic 16. En regardant les mots principaux de ce topic, nous remarquons "cuba", "fidel", "castro", ... Afin d'analyser de manière plus poussée, nous avons imprimé les 50 premiers tweets et regardé le topic des tweets parlant de la mort de Fidel Castro. Tous sauf 2 font partis du cluster 16.

	Topic0	Topic1	Topic2	Topic3	Topic4	Topic5	Topic6	Topic7	Topic8	Topic9	Topic10	Topic11	Topic12	Topic13	Topic14	Topic15	Topic16	dominant_topic
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.25	16
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.88	16
3	0	0	0	0	0	0	0	0	0	0	0	0.78	0	0	0	0	0	11
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	74
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.33	16

Figure 6: Association de chaque tweet au thème le plus probable

Afin de mieux visualiser nos différents clusters, nous avons décidé d'utiliser pyLDavis de sklearn pour voir nos clusters et les mots les plus fréquents associés à chacun. En observant la figure 2, nous pouvons voir que plusieurs de nos clusters sont dispersés ce qui est un très bon signe. Cela nous indique que notre modèle LDA a bien su différencier les topics. Par contre, on remarque plusieurs clusters qui se chevauchent autour de l'origine du système d'axe. De plus, si l'on regarde les tailles des bulles, on peut voir que nous possédons des bulles de taille moyenne dans les extrémités de notre système d'axes. Par contre, plus on se rapproche de l'origine, plus nos bulles, c'est-à-dire nos clusters, sont de petites tailles. Cependant, cela reste une amélioration comparé à nos deux premiers modèles où l'on avait un cluster de taille géante. De plus, en regardant les mots les plus fréquents des clusters, nous pouvons voir qu'ils diffèrent d'un cluster à l'autre ce qui nous indique que nos ensembles sont distincts.

Nous avons aussi effectué une étude à travers le temps pour ce modèle. Nous affichons pour chaque cluster les dates de création de ses tweets. La figure 7 représentent la distribution de tous les tweets anglais à travers le temps. La figure 8 représentent la distribution de tous les tweets du cluster 16 à travers le temps. Si l'on compare nos deux figures, on remarque que nous avons la même tendance sur plusieurs dates. Cependant, il y a la présence de nouveaux sommets. Par exemple, on remarque qu'il y a plusieurs tweets le 26 novembre 2016 dans le cluster 16. Cela correspond à la date de la mort de Fidel Castro.

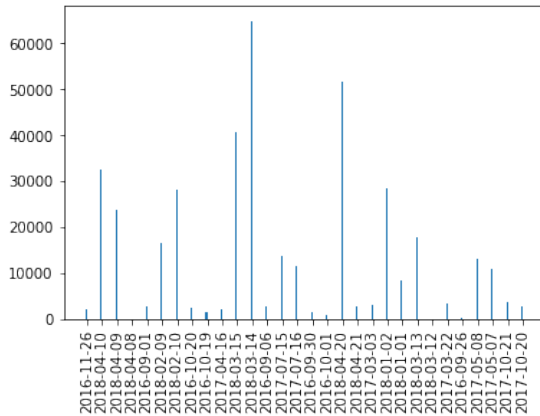


Figure 7: Distribution de tous les tweets anglais à travers le temps

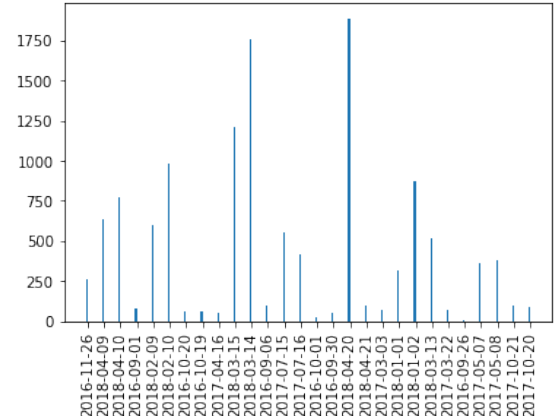


Figure 8: Distribution de tous les tweets du cluster 16 à travers le temps

6 Discussion

6.1 Faiblesses et avantages de l'algorithme K- mean

L'algorithme K-means est un algorithme simple et rapide comparativement aux autres algorithmes de clustering. Cependant, il s'agit principalement d'un algorithme de partitionnement. Il faut aussi préciser la quantité de clusters avant de démarrer l'algorithme. Cela est une difficulté supplémentaire si nous ne connaissons pas bien l'ensemble de données avec lequel on travaille. Le plus grand avantage de cet algorithme est sa performance et sa simplicité pour fit un grand ensemble de données. De plus, cet algorithme utilise la distance euclidienne pour calculer la distance des points avec les centroides. Cette distance prend en considération la norme des vecteurs et n'est pas nécessairement la manière la plus efficace pour évaluer nos données.

6.2 Faiblesses et avantages de l'algorithme LDA

LDA nous permet de voir toutes les probabilités d'association de chaque tweet à un certain topic. Cela facilite énormément l'analyse de nos données ainsi que la compréhension des résultats. Avec les nombreuses manières de visualiser nos données, il devient facile de détecter les incohérences et de voir les liens entre les tweets. Une des faiblesses de l'algorithme LDA est le fait que nous devons choisir le nombre de topics à catégoriser et le learning rate du modèle. Cependant, sklearn possède la classe GridSearchCv nous permettant de tester plusieurs combinaisons de valeurs d'attributs différentes. Il est possible de voir dans notre code que nous avons essayé de trouver la meilleure combinaison d'attributs. Malheureusement, en se basant sur plusieurs forums cette manière ne fonctionne pas bien puisqu'elle retourne toujours la combinaison avec la plus petite quantité de clusters passée à GridSearchCv. Le bug a été signalé sur Github, mais n'est toujours pas réglé. Nous ne pouvons pas être certains des valeurs que nous utilisons comme paramètres.

Aussi, nous avons remarqué que lorsque nous utilisons un nombre de topics plus faibles, nos clusters étaient bien dispersés dans la représentation avec pyLDAvis. Par contre, les clusters contenaient des tweets qui n'avaient aucune similarité entre eux. Nous avons essayé avec un nombre de topic à 75. L'annexe 4 contient notre représentation graphique. Nous avons alors opté pour un nombre de topics plus élevé ce qui permet d'avoir une meilleur similarité entre les tweets des topics.

6.3 Améliorations

Une première amélioration aurait été de pouvoir utiliser la classe GridSearchCV. Cela nous aurait permis de trouver le nombre de topics le plus optimal afin d'éviter d'avoir autant de chevauchement de clusters autour de l'origine (voir figure 2).

On aurait pu effectuer plusieurs modèles LDA avec des paramètres différents. Ensuite, on les aurait entraînés sur une partie du corpus (environ 10%). Ces données seraient prises de manière aléatoire afin de s'assurer que ces sous-ensembles sont bien représentatifs de nos données. Cela nous permettrait d'économiser beaucoup de temps.

Aussi, il aurait été intéressant de voir le résultat de l'application de K-Means sur le résultat du modèle LDA. En effet, nous avons pris la probabilité d'appartenance à un topic pour classifier chaque tweet.

Une autre amélioration possible est l'utilisation de la distance cosinus au lieu de la distance euclidienne pour l'algorithme K-Means.

References

- [1] Josh Montague Data Scientist at Twitter. (Février 2018). Do More with Twitter Data. [En ligne]. Disponible: https://twitterdev.github.io/do_more_with_twitter_data/clustering-users.html.
- [2] Machine Learning Plus. (2017). LDA in Python – How to grid search best topic models? [En ligne]. Disponible: <https://go.shr.lc/2SJkK3p>.
- [3] Aneesha Bakharia. (2016). Topic Modeling with Scikit Learn. [En ligne]. Disponible: <https://medium.com/mlreview/topic-modeling-with-scikit-learn-e80d33668730>.

7 Annexes

7.1 Annexe 1

Nombre de tweets dans le cluster 36: 115431
 Here's what your member of Congress can actually do for you <https://t.co/dEyd05vohZ> <https://t.co/X1G9cc0b7k>
 Wave of Mexico violence reveals hidden graves, severed heads <https://t.co/1zB7LsGdjh> <https://t.co/9nEBe2QGLQ>
 \$20K reward offered for 2 men who escaped California jail <https://t.co/10wWITEMQG> <https://t.co/fLZw4KZudV>
 Smithsonian National Zoo's panda Bei Bei recovering after "lemon-sized mass of bamboo" removed from small intestine... <https://t.co/HMCyUrvnG>
 Iraq's Najaf cemetery swells as fight against ISIS escalates <https://t.co/8mx1gcumwn> <https://t.co/JzGasQk5wp>
 Demonetisation has reduced crime rate in Mumbai: Defence Minister Manohar Parrikar. <https://t.co/Rjsz1e9i3v> <https://t.co/OT6HB9yf7Z>
 Plan to create biodiversity map identifying aquatic species in every river and stream in western U.S. is underway.... <https://t.co/8PXFx0LsSg>
 #1eWorld Syrian government advances into rebel-held east Aleppo <https://t.co/CK3t17X4AM> <https://t.co/kFnmIwceLX>
 "History will absolve me": Fidel Castro's quotes through five decades <https://t.co/rgGQ1F3jWY> <https://t.co/LHRb2ni6D1>
 Judicial appointments: CJI vs govt again, Prasad 'disagrees' on vacancies | <https://t.co/LMRzPJOGHz> <https://t.co/mWtK19uTta>
 The department store "door buster" used to lure Black Friday shoppers? \$19.99 women's boots:... <https://t.co/xHk4i2CjFm>
 Bold reforms like demonetisation to boost economic growth: NITI Aayog. <https://t.co/quJR9fGJp7> <https://t.co/WhzBkAKaUZ>
 Runaway teen from Odisha 'recovered' from Delhi hotel. <https://t.co/i3lTF1332l> <https://t.co/YAzzEYnSGH>
 RT @IExpressSports: #INDvENG Stumps on Day 1 in Mohali and it belongs to India. England reach 268/8 <https://t.co/H0shc4i4q9>
 RT @utkarsh_aanand: #ConstitutionDay Law Min @rsprasad: Legislation must remain preserve of the legislature..problems will resolve if all o...
 Hindu pilgrims cancel trip to Pakistan due to tension | <https://t.co/MSAWShYgvs> <https://t.co/LevZeStwzk>
 Trinamool Congress to oppose Left's strike against demonetisation <https://t.co/UXnkHkKa1A> <https://t.co/UCiJRpC2ev>

7.2 Annexe 2

Nombre de tweets dans le cluster 87: 2311
 The first American to die in combat in Syria: Scott Dayton, a 42-year-old Navy sailor <https://t.co/P6zjbQN901>
 A hot souvenir from Cuba for some Americans: A cancer vaccine <https://t.co/0lRYv4um51>
 Scott Dayton Identified as First American to Die in Syria Combat <https://t.co/f0pd6hJBEF>
 Breaking News: Fidel Castro has died at 90. The Cuban revolutionary was a nemesis to 11 American presidents. <https://t.co/01wGISur9L>
 Disney archivist Dave Smith spent 40 years preserving the history of an American icon <https://t.co/QmHAjR1B1L> <https://t.co/0QrNzWo9ze>
 The keeper of Disney's legacy, archivist Dave Smith spent 40 years preserving the history of an American icon <https://t.co/wlE2n72Coc>
 American President-Elect Trump and Third World Concerns: <https://t.co/043FVznN39> #Africa <https://t.co/tDoYgs0Tvf>
 Naval officer Scott Dayton identified as first American killed in Syria <https://t.co/oh4Ir8qhH1> <https://t.co/GwmW9Gicay>
 Why the American left should beware of choosing to fight the right on populist ground <https://t.co/iPnzWNSuf2>
 #Thanksgiving online sales surge as American shoppers choose to stay at home <https://t.co/gmCF8j5OM9> <https://t.co/YYZW3K1Po>
 A map of the world according to American stereotypes <https://t.co/RLlgHtcrnU>
 A map of the world according to American stereotypes <https://t.co/RLlgHsUPYk>
 RT @indy100: A map of the world according to American stereotypes <https://t.co/mxntLkUe2x>
 How the US destroyed Native American land - in one animated map <https://t.co/SQvEFWoh96>
 American Nazis are so stupid they don't even know how to draw swastikas properly <https://t.co/T5ixgQ0YrH>
 RT @HeerJeet: 153 years ago a vicious regime committed to human bondage was defeated. Americans should celebrate it as a holiday: <https://t.co/3AMhDVxgWw>
 "The success of our government depends on maintaining the trust of the people we serve. The American public needs t... <https://t.co/3AMhDVxgWw>

7.3 Annexe 3

Nombre de tweets dans le cluster 7: 57438

In Havana, music stops as Cubans mourn 'father' Castro <https://t.co/rKzhGyabn6> <https://t.co/USkbiCbAXL>

Marianne Faithfull pays homage to Paris victims at Bataclan <https://t.co/kchA0HcuiA> <https://t.co/000peneev0>

Afghan, Pakistani Leaders Discuss Tense Bilateral Ties <https://t.co/KwITCjR0CD> <https://t.co/JjZThztAno>

Fidel Castro's military forays into Africa <https://t.co/404zG25NEa> <https://t.co/FfXOCdtsCr>

Bay of Pigs: America's Waterloo in anti-Castro fight <https://t.co/CwdVZAjIqa> <https://t.co/IcamsAycur>

The 37 iconic bridges that span the Seine in Paris <https://t.co/SZFISqJe4x>

Rooney 'cleared over drinking storm' <https://t.co/REE1pX63nV> <https://t.co/sdXRU5vtLK>

More casualties reported among Turkish soldiers in Syria <https://t.co/TRRc7rD28p> <https://t.co/VDd75fuNvc>

Bargain Hunters Excited About Black Friday <https://t.co/JgOUFmKC32> <https://t.co/rUHRpI4Htk>

Credit Suisse estimates Turkmenistan has the most wealth per adult citizen: <https://t.co/3vf40EJNTI>

Gorbachev hails Castro for 'strengthening' Cuba <https://t.co/aUKpsHvghb> <https://t.co/J2RpTU0e1p>

Obama Moved Aggressively to Restore Relations with Cuba <https://t.co/gVzAG3yjr4> <https://t.co/05T0i9tteK>

Obituary: Cuban Revolutionary leader Fidel Castro, 1926-2016 <https://t.co/jXlT0TC3JZ> <https://t.co/icSKUzSofD>

Here's what your member of Congress can actually do for you <https://t.co/dEyD05vohZ> <https://t.co/X1G9cC0b7k>

Iraq's Najaf cemetery swells as fight against ISIS escalates <https://t.co/8mx1gcumwn> <https://t.co/JzGasQk5wp>

Demonetisation has reduced crime rate in Mumbai: Defence Minister Manohar Parrikar. <https://t.co/Rjsz1e9i3v> <https://t.co/OT6HB9yf7Z>

#ieWorld Syrian government advances into rebel-held east Aleppo <https://t.co/CK3t17X4AM> <https://t.co/kFnmIwceLX>

7.4 Annexe 4

