Classification des posts sur des réseaux sociaux

Ahmed REMADNA

INSA-Lyon
ahmed.remadna@insa-lyon.fr,
LIRIS, Equipe DRIM, INSA-Lyon
Bâtiment Blaise Pascal, Domaine de la Doua
7 Avenue Jean Capelle
69100 Villeurbanne, France
Encadrant: Előd Egyed-Zsigmond

Résumé Dans ce travail nous présentons des méthodes pour l'identification des fils de discussions sur les réseaux sociaux (notamment Twitter) concernant des événements qui déroulent dans le monde réel comme la Fête des Lumières à Lyon, et nous proposons un modèle pour l'enrichissement des requêtes de collectes pour un suivi dynamique et sur une période prolongé d'un événement avec des mots clefs qui évoluent.

Abstract. In this work we present a method for the identification of discussion threads on the social networks (mainly Twitter), cenetrd on events taking place in the real world, for instance: F"ête des lumières in Lyon", and we propose a model for the enrichment of collection queries to thoroughly cover such events over a longer time period.

Keywords: Text mining, Twitter, Clustering, TF-IDF, Z-score.

1 Introduction

Avec les progrès techniques du web et des capacités de stockage et d'échanges sur Internet. Les réseaux sociaux connaissent une explosion en termes de volume de données et en fonction du nombre d'utilisateurs à travers le monde. Cette utilisation quotidienne des réseaux comme Twitter et Instagram a changé l'image du web 2.0 et lui a donné une nouvelle dimension et aussi de nouveaux défis.

Twitter est un réseau social de type microblogging, où les gens peuvent raconter leurs vies, partager leurs expériences, leurs émotions, leurs avis, etc. en postant des courts messages sur un espace public. Ces messages sont appelés des tweets. Chaque message a une taille maximale de 140 caractères. Comme dans les autres réseaux sociaux il est possible d'utiliser un mot-dièse (hashtag) pour annoter le tweet avec un mot-clé de contexte.

Twitter a été créé en 2006 avec le slogan original "what are you doing ?"(qu'est-ce que vous faîtes?) et en novembre 2009 ce slogan a été changé vers "What's

happening ?" (quoi de neuf ?) pour mieux engager les utilisateurs à partager leurs expériences.

Actuellement Twitter est une source géante de données avec 320 millions d'utilisateurs actifs et plus de 500 millions de tweets qui sont publiés chaque jour en 35 langues [5].

Avec ce volume important des données, Twitter est devenu dernièrement un champ de recherche très attractif pour plusieurs acteurs de société, plus particulièrement les agences de presse, les entreprises, les chercheurs en informatique et science de l'information, les psychologues et les sociologues. Ces domaines de recherche concernent principalement l'étude de marché, le suivi de campagnes publicitaires, l'analyse de tendances, analyse de comportement humain, social et individuel, la détection des maladies et l'identification des personnes influentes, etc.

Le suivi des événements locaux sportifs ou culturels, sur les réseaux sociaux suscitent de plus en plus les autorités responsables de ces événements ainsi un objectif majeur pour certains organismes lucratifs comme les entreprises publicitaires qui cherchent à profiter au mieux, est l'extraction des informations pertinentes à partir de ces sites communautaires, dans l'intérêt d'analyser le succès de ces événements et pour pouvoir améliorer leurs actions. Mais cet objectif reste limité par les approches et les méthodes fournies dans le monde scientifique, et plus particulièrement le monde de la recherche en informatique.

Dans ce travail nous présenterons une approche pour l'identification des fils de discussions qui concernent un événement donné sur Twitter et Instagram, et aussi un modèle pour enrichir les requêtes de collecte, ce qui permet de mieux couvrir l'événement sur les réseaux sociaux.

1.1 Problématique

Les posts publiés sur Twitter ou Instagram reflètent l'interaction d'utilisateurs avec les événements réels qui se déroulent dans le monde, comme les élections, les événements sportifs et culturels, les catastrophes naturelles, etc. Ces événements réels ont un impact direct sur la quantité de tweets mises en ligne.

Le suivi de ces événements sur les réseaux sociaux généralement et sur Twitter plus précisément est un défi audacieux pour les chercheurs, tout d'abord parce qu'un sujet sur Twitter est caractérisé par plusieurs termes (ces termes peuvent être des hashtags) qui peuvent changer dynamiquement où certains peuvent devenir moins utilisés et d'autres peuvent apparaître. Donc il est incontournable de trouver un moyen pour couvrir tous ces termes utilisés pendant le processus d'analyse. Cela représente l'un de nos objectifs dans ce travail. Mais avant de chercher les nouveaux termes, il faut pouvoir identifier les ensembles de tweets qui parlent du même sujet et qui représentent un fil de discussion, ce qui définit l'objectif principal de notre travail.

(Figure 1) représente une illustration de cette problématique à travers l'exemple de suivi L'EURO2016 à Lyon. Pour cet objectif nous commençons par une requête qui prend comme entrée les deux mots-clés de départ Lyon et EURO2016.

Cette requête permet de récupérer des tweets qui contiennent l'un de ces deux termes, mais notre objectif est de découvrir d'autres termes qui caractérisent le sujet EURO2016. En prenant l'exemple de l'affaire de Benzema, avec son nom qui devient très utilisé dans les réseau sociaux et appartient effectivement au sujet EURO2016. Notre contribution devrait proposer le mot Benzema comme un nouveau terme à rajouter dans la requête de collecte. Pour cela dés que le regroupement de tweets est effectué, nous essayerons de chercher les mots les plus importants (comme Benzema) dans les fils de discussions correspondantes (qui parlent de EURO2016), ces mots peuvent enrichir la requête de départ. Mais ce n'est pas simple avec la quantité de données énormes mis en ligne chaque jour. En effet, il faut pouvoir traiter ces données d'une manière continue, en utilisant ce qui existe comme méthodes et algorithmes d'analyse. La difficulté de ce traitement est due à plusieurs raisons, entre autres la grande différence entre le style de langage de Twitter par rapport aux médias classiques, avec sa nature fragmentée et ambiguë liée à la contrainte de taille.

L'absence des mots de contexte (hashtags) dans une grande partie du corpus est un autre véritable casse-tête où il faut surmonter ce problème afin de mieux exploiter cette ressource riche par des informations utiles.

Le contexte des réseaux sociaux fournissent d'autres informations qui peuvent être exploitées afin de mieux analyser les sujets. Ces données sont des informations à notre portée qui concernent le tweet lui-même : la date précise de la publication, l'identifiant de l'utilisateur qui a posté le tweet (comme ces relations dans le réseau) et les coordonnées géographiques. Ces informations nous permettront éventuellement de chercher une autre similarité plus que la similarité textuelle, en fonction de la distance temporelle, spatiale et entre utilisateurs des réseaux sociaux.

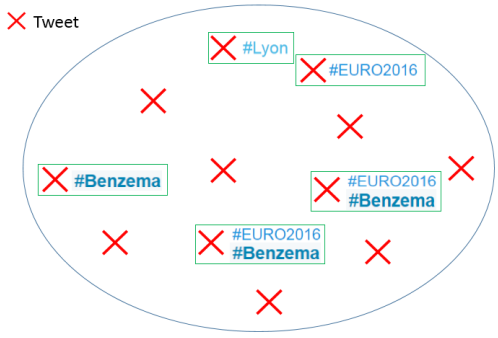

Fig. 1. Illustration de la problématique par un exemple de l'EURO2016.

2 État de l'art

L'analyse des posts dans les réseaux sociaux est une nouvelle discipline qui émerge en informatique. Des contributions comme la détection des événements, la détection des maladies et l'extraction des connaissances depuis Twitter, sont les travaux les plus proches de notre problématique.

Dans cette partie nous allons citer quelques travaux connexes que nous avons synthétisés, afin d'inspirer les différentes étapes de notre processus d'analyse en général. Ensuite, nous parlerons des différentes stratégies de représentation du texte et nous conclurons par une étude comparative entre les algorithmes de clustering dont nous allons nous servir.

Dans [17] [14] [6] les auteurs ont proposé différentes méthodes pour l'analyse des posts des réseaux sociaux et l'extraction des informations pertinentes, mais en suivant un ensemble d'étapes très similaires, qui commencent par la collecte des posts à partir de ces réseaux et ensuite une étape de prétraitement ou nettoyage afin de préparer les données avant d'effectuer l'étape de regroupement (Clustering). D'après ces articles nous avons identifié les grandes étapes à faire dans notre contribution pour appliquer le regroupement des posts.

Dans [17], les auteurs ont proposé une approche pour l'analyse continue du flux Twitter en temps réel, en utilisant un clustreing en ligne basé sur la densité, avec les deux algorithmes DBSCAN et OPTICS. Le plus de cet article est l'adaptation de ces deux algorithmes pour supporter le contexte de Clustering en ligne et avec une technique incrémentale pour ignorer les valeurs aberrantes.

Les auteurs de [14] proposent une méthode pour la détection des événements sur Twitter en utilisant un filtrage stricte avec le Clustring hiérarchique, ce travail permet de détecter automatiquement des événements comme, par exemple la guerre en Syrie et aussi ce qui concerne la crise politique en Ukraine.

Le traitement décrit dans cet article est fait presque en temps réel, où chaque 15 minutes l'algorithme rajoute les tweets collecté récemment dans le processus de regroupement. La phase de collecte très détaillé nous a inspiré pour l'implémentation de notre collecteur, pour récupérer les tweets. Dans la phase de filtrage de tweets les auteurs ont étudié l'impact de la variation de filtrage sur les résultats finaux.

Le point faible de ce travail est l'ignorance des milliers de tweets avec le filtrage agressif, ce qui conduit à une perte de la valeur d'une grande partie de données qui peuvent être utiles.

Dans la partie suivante, nous présentons quelques travaux intéressants par rapport aux trois phases principales: prétraitement, transformation du texte et clustering.

2.1 Prétraitement

Dans [6] les auteurs ont proposé une méthode de détection d'événements en utilisant une expansion lexico-sémantique (lexicale et sémantique) sur des mots et des hashtags, cette expansion est intégrée dans l'étape de prétraitement et

permet a des mots comme alexsandra et alexandra d'être considérés comme un mot identique.

Les résultats expérimentaux montrent que l'utilisation de cette expansion apporte plus de précision. Les auteurs suggèrent que les hashtags peuvent être représentatifs par rapport à des événements dans Twitter.

Des techniques simples et intuitives dans la phase de nettoyage ont été évoquées dans [6] [11] [13] comme la suppression des liens, des identifiants Twitter et l'enlèvement des points et virgules dans les mots.

Une étude sur le développement des conversations sur Twitter, qui concerne des sujets publicitaires a été présentée dans [9]. Les auteurs de cet article arrivent à compter le nombre de discussions positives et négatives à propos plusieurs marques de constructions automobiles. Ce travail a été fait en quasi temps réel avec une phase de prétraitement intéressante qui le point important dans ce travail. Cela consiste à l'utilisation d'un moteur à base de règles dans le filtrage, pour éliminer les tweets qui ne parle pas du sujet de publicité. L'étape de recueil de données dans ce travail a été relativement longue (sur 15 jours) est avec un résultat de plus de 7 million de tweets.

D'après tous ces travaux mentionnés, nous avons retenu l'utilisation des types de nettoyage simples qui consiste à supprimer les liens, les identifiants, les points, les virgules et les guillemets comme des choix possibles, aussi nous avons décidé d'appliquer une modalité de filtrage qui permet de ne garder que les hashtags.

2.2 Représentation du texte

La majorité des algorithmes de partitionnement ne prennent pas comme entrée un texte brut, mais des vecteurs numériques. Pour cela il est nécessaire de trouver une transformation représentative qui convertit le texte des tweets vers des vecteurs numériques. Une famille de cette transformation est appelé Bag-of-Words.

Dans la littérature les méthodes de transformation du texte vers des Bag-of-Words peuvent être divisées sur trois approches principales. La première est une approche purement statistique basée sur l'occurrence des termes comme TF(Term Frequency) et TF-IDF(Term Frequency-Inverse Document Frequency). La deuxième est une approche sémantique, qui comporte les deux méthodes LSA (Latent Semantic Analysis) et LDA (Latent Dirichlet Allocation). La troisième est une approche alternative entre les deux, il s'agit principalement de l'ensemble des méthodes de la famille N-gram.

Nous avons exclu la troisième approche qui ne convient pas au cadre des tweets, car nos tweets ont des longueurs différentes au contraire du principe de cette approche qui se base sur la division de de chaque phrase sur un nombre fixe de mots.

Dans [21] et [18] les auteurs proposent l'utilisation de la méthode LDA(Latent Dirichlet Allocation) sur un flux de Twitter, d'après les résultats montrés dans les deux articles l'utilisation d'une approche sémantique comme LDA permet d'améliorer considérablement la qualité de la modélisation du sujet Twitter.

L'utilisation de l'approche statistique comme le TF-IDF a été évidemment utilisé sur le flux Twitter comme dans [15], car elle est la plus simple mais aussi convient parfaitement au type de corpus comme Twitter où l'importance du mot vient de sa fréquence d'utilisation.

Dans le contexte de Bag-of-Words statistique, une nouvelle méthode appelée Z-score à été proposée dans [19]. les auteurs de cet article présentent une comparaison entre cette méthode et TF-IDF. Les résultats obtenus peuvent faire pencher la balance en faveur de Z-score pour mieux identifier les mots les plus importants par rapport au TF-IDF.

D'après les articles déjà synthétisés dans la partie transformation du texte vers des vecteurs numérique, nous avons décidé de commencer par l'utilisation de l'approche statistique sur notre flux de travail et étudier la qualité des résultats, avant de passer vers l'approche sémantique. Cette approche a été utilisée par l'implémentation de deux méthode TF-IDF et Z-score

2.3 Clustering

D'après les articles cités dans le début de ce chapitre, les algorithmes de regroupement (Clustering) utilisés par les chercheurs dans les travaux connexes sont le Clustering hiérarchique et les Clustering basé sur la densité par les deux algorithmes DBSCAN et OPTICS et avec notre volonté d'essayer les trois approches de clustering qui existent (partitionnement, hiérarchique, basée sur la densité) nous avons décidé d'utiliser ces trois algorithmes sur notre flux de travail et en rajoutant l'algorithme de Kmeans qui appartient au type d'algorithme de partitionnement[16].

Ce choix est lié a un intérêt de profiter des points forts de chaque algorithme, ces points sont mentionnées dans l'étude comparatif qui suit dans cette section et ainsi pour une perspective de passer vers le clustering en temps réel où il faut bien choisir l'algorithme correspondant à cette mission, c'est à dire l'algorithme le plus optimal, ce dernier il doit avoir un bon compromis entre la qualité de résultats et le temps d'exécution.

le tableau 1 présente une étude comparative entre les quatre algorithmes de Clustering que nous allons utiliser :

	Forces	Faiblesses
Kmeans	- relativement extensible pour traitement des grands ensembles de données[16]. - relativement rapide par rapport aux clustering Hiérarchique, DBSCAN et OPTICS.	- sensible face aux données aberrantes. - la spécification du nombre de clusters à priori. - les clusters sont construits par rapport à des centres inexistants dans les données traitées[16]. - Si on enlève des données ou on rajoute d'autres il faut refaire tout le calcul.
Hiérarchique	- pas besoin de spécifier le nombre de clusters à priori. - un partitionnement par niveau (on peut s'arrêter à un niveau souhaité). - applicable sur du texte brut.	- sensible face aux données aberrantes. - très gourmand en terme de mémoire et il devient assez lent avec les grands ensembles - L'algorithme fait un seul passage à travers l'ensemble de données. Par conséquent, les individus qui sont attribués par erreur ne seront pas réaffectés plus tard. - les clusters ne sont pas très stables, un petit changement dans la population peut impliquer un grand changement du résultat final. - Si on enlève des données ou on rajoute des autres il faut refaire tout le calcul.
DBSCAN	- élimination des valeurs aberrantes[1]. - pas besoin de spécifier le nombre de clusters à priori. - la possibilité de réintégrer les nouvelles données dans les clusters sans refaire tout le calcul[14].	- incapable de construire des clusters de densités différentes - la spécification de rayon maximale et le minimum de points par cluster à priori. - relativement lent par rapport le Kmeans.
OPTICS	- élimination des valeurs aberrantes[4]. - construction des clusters de densités différentes. - pas besoin de spécifier le nombre de clusters à priori. - la possibilité d'intégrer de nouvelles données dans les clusters sans refaire tout le calcul[14].	- la spécification du minimum de points par cluster à priori. - relativement lent par rapport le Kmeans.

Table 1. Comparaison des méthodes de clustering.

3 Méthodologie

Afin de valider nos méthodes d'étude de l'évolution syntaxique des sujets et faire émerger des fils de discussion, nous avons mis en place un cadre d'expérimentation ayant l'architecture de la (Figure 2) qui représente les grandes parties de notre contribution. Chaque phase possède une entrée, une sortie et une série de paramètres que nous allons faire varier afin de trouver les meilleurs valeurs. Ce cadre est découpé en phases de collecte (1), de récupération de tweets (2) et de nettoyage (3). pour un objectif de regrouper des tweets nous les avons transformés en vecteurs numériques (4) avant d'appliquer des méthodes de clustering (5). Les résultats de ces méthodes sont ensuite analysés et visualisés en (6).

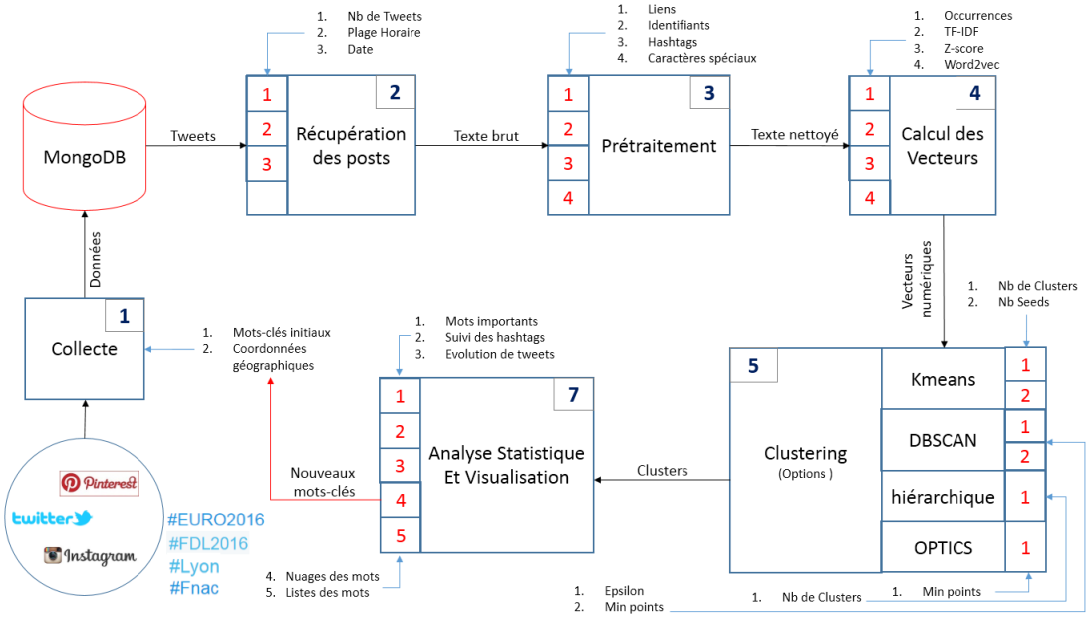

Fig. 2. Architecture du cadre d'expérimentation.

Dans ce qui suit, nous allons spécifier chaque phase de l'architecture.

3.1 Phase 1 : Collecte

Dans ce projet nous allons travailler sur plusieurs collections de tweets, soit collectées par nous-même, soit un jeu d'essai trouvé sur Internet. Dans un premier lieu on a commencé par la collection 'Twitter_FDL2015' décrite ci-après:

Twitter_FDL2015 Est une collection récoltée pendant la fête des lumières 2015 (c'est un événement culturel qui a lieu chaque année à Lyon du 8 au 11 décembre). Cette récolte a été faite par l'équipe dans lequel j'ai effectué le stage.

L'application de collecte prend comme entrée une liste de mots-clés à chercher et un rectangle de coordonnées géographiques, afin de cibler cette recherche. pour cette collection la requête initiale à comme mots-clés Lyon, FeteDesLumieres2015, FDL2015, lumignon et les coordonnées géographique de la grande Lyon.

La collecte se fait d'une manière continue sur plusieurs jours et les données sont injectées directement dans une base de données NoSQL, de type orienté documents et en utilisant le format de données JSON.

Pour la gestion de cette base de données on a opté l'utilisation de MongoDB qui utilise lui-même le format de données JSON.

Description de la collection Cette collection contient :

- 31 008 Tweets.
- collectés sur 4 jours (du 07 Décembre à 13H jusqu'à 11 Décembre à 07H).
- 1559 Tweets géolocalisés avec les coordonnées.
- d'une taille de 90,3 Mo (json non compressé).

(Figure 3) représente l'évolution du collecte par plage horaire sur les 4 jours.

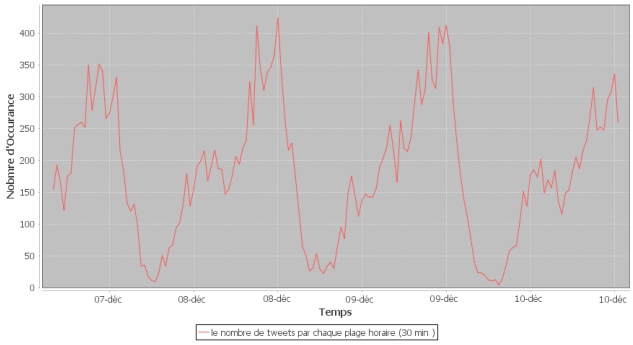

Fig. 3. L'évolution du collecte par plage horaire

Pour chaque Tweet collecté on a :

- des attributs qui concernent le tweet lui-même :
  - le contenu (du texte).
  - les hashtags s'ils existent.
  - la date de la publication.

    - les propriétés textuelles du mots et hashtags.
    - URL du tweet.
- des attributs qui concernent la position géographique :
    - les coordonnées géographiques de l'utilisateur lors de publication de tweets s'ils existent.
    - le pays.
    - le code du pays.
- des attributs qui concernent le profil d'utilisateur :
    - la date de création de compte.
    - une description de l'utilisateur.
    - un URL vers l'image de profil.
    - le pseudonyme.
    - le nom d'utilisateur.
    - le fuseau horaire.
    - le nombre de Tweets précédents.

Un exemple d'un tweet qui appartient à la collection Twitter_FDL2015 en format Json est dans Annexe A.

3.2 Phase 2 : Récupération de posts

Cette phase consiste à récupérer les posts des réseaux sociaux (dans le cas de Twitter il s'agit des tweets) à partir de la base de données avec des requêtes, en variant les paramètres en fonction :

- qu'est-ce qu'on veut récupérer ? (quelle collection, tweets ,textes, hashtags, coordonnées géographiques, noms d'utilisateurs ou toutes les données possibles, etc).
- quelles contraintes ?(dans un intervalle de temps, dans un espace géographique donné, avec un nom d'utilisateur spécifique).
- un nombre donné de tweets.

Dans le cas de la collection 'Twitter_FDL2015' à la fin de cette étape on aura la liste des tweets ou la liste des Hashtags sélectionnés.

En premier lieu, nous avons commencé par la récupération du texte des tweets uniquement, avec un nombre donnée de tweets fixé à 1000. Dans ce cas, la sortie de la première étape est une liste qui contient tous les tweets sélectionnés en texte brut. et avant de passer à l'étape de pré-traitement, il fallait extraire chaque mot tout seul, tweet par tweet pour pouvoir appliquer le nettoyage avec les expressions régulières et d'autres.

Pour cela on utilise une étape de transformation des tweets vers des mots, qui prend comme entrée la liste des tweets en texte et qui retourne une liste des mots.

L'ensemble des paramètres modifiables lors de cette phase :

- le nom de collection.
- le type de post (Twitter ou instagram).
- le type de données (Texte, Hashtags, coordonnées géographiques, les noms d'utilisateurs).
- la dimension temporelle (entre deux dates, par plage horaire).
- le nombre de tweets.

3.3 Phase 3 : Prétraitement

L'analyse des tweets représentera un défi majeur pour nous et pour les autres chercheurs qui travaillent sur le flux Twitter. Cela découle de la nature de ces messages publiés, où les tweets sont totalement différents par rapport à d'autres documents comme les articles des journaux, les discours officiels, les pages web,etc. Pour les méthodes de nettoyage, nous, nous sommes inspirés des travaux évoqués dans l'état de l'art que nous avons complété par des méthodes originaux. Parmi les caractéristiques qu'on peut trouver dans les tweets on cite :

- les utilisateurs utilisent un langage qui n'est pas formel, et un mélange entre le jargon, abréviations et plusieurs langues dans le même tweet.
- les Tweets sont pleins d'erreurs d'orthographe, d'erreurs lexicales et d'erreurs syntaxiques.
- l'existence des liens et des identifiants compliquent l'opération d'analyse.

Pour ces raisons on a décidé d'appliquer un nombre minimum de nettoyages et filtrages sur les tweets durant cette phase, qui consiste à :

- supprimer les liens : dans ce contexte, un lien dans un tweet n'a pas un poids sémantique pour le sujet suivi (sauf si on analyse le lien s'il y a plus de détails éventuellement dans la page pointé par le lien).
- supprimer les identifiants: on a implanté ce type de nettoyage optionnel parce qu'un identifiant Twitter n'est pas un mot clé qui concerne le sujet. En même temps, il peut donner plus de similarité entre deux tweets du même sujet qui sont souvent proches et postés par la même personne.
- supprimer les hashtags : l'un des types de nettoyage possible est la suppression des hashtags pour pouvoir travailler uniquement sur du texte en langue naturelle.
- mise à jour du hashtags : cette opération consiste à mettre un dièse (#) devant chaque mot qui est identique à un hashtag (le mot devient un hashtag aussi) pour un intérêt de donner plus de contexte à un tweet, ce qui permet probablement d'améliorer l'identification du sujet auquel il appartient ce tweet.
- suppression des mots non hashtags (ce nettoyage peut être combiné avec le précédent)
- pour résoudre le problème des majuscules et des minuscules on a décidé d'appliquer deux types de traitements, le premier est la conversion de tous les mots en minuscule. Cette transformation pose un problème pour les abréviations identiques à des mots existants qui vont être changés.
- le deuxième traitement, c'est la conversion du premier caractère en majuscule et le reste en minuscule (comme le cas du premier mot d'une phrase en français) de tous les mots qui ne contiennent que des caractères alphabétiques et ne sont pas tout en majuscules, et cela évite qu'un mot comme 'La' (en début de la phrase) et 'la' soient considérés comme des mots différents, mais en même temps on garantit de ne pas convertir les abréviations en minuscule.

– un autre type de traitement consiste à nettoyer les mots par la suppression des guillemets, des virgules, des points, des points d'exclamation et des points d'interrogation, etc.
– supprimer les mots qui contiennent uniquement un seul caractère.
– pour le nettoyage des chiffres ce n'est pas simple, car un numéro comme 3 ou 2016 sont totalement différents. et ça demande une étude préalable des conséquences de cet effet. Par exemple la suppression de numéro 3 au milieu de la phrase peut être utile car ce 3 n'a pas une grande signification dans la phrase, mais est-ce que c'est le même cas pour 2016 qui représente l'année courante, un autre contre exemple dans un contexte spécifique, ou dans un événement donné où le chiffre seul a une signification importante, comme le jour de la demi finale de la coupe du monde en 2014, dans ce jour le Brésil a perdu le match 7 à 1, et à la fin de rencontre le fameux "7" a apparu partout sur les réseaux sociaux, dans ce cas le numéro '7' fait une grande partie du sujet.

Tous ces types de nettoyage ne sont pas tout le temps appliqués à la fois, mais pour une expérimentation on peut préciser l'ensemble de nettoyages à appliquer, afin étudier l'impact de la variation des prétraitements sur le résultat final.

L'entrée de cette étape est la liste des listes des mots bruts extraits de chaque tweet, et la sortie est la liste des listes des mots nettoyés. Si le nettoyage supprime tous les mots d'un tweet, le tweet sera supprimé.

3.4 Phase 4 : Transformation des textes vers des vecteurs numériques

D'après les travaux existants et ce que nous avons évoquer dans de l'état de l'art, on a décidé d'appliquer un regroupement (clustering) sur les tweets, pour savoir quels sont les tweets similaires et pouvant représenter un fil de discussion.

Notre idée est d'appliquer un ensemble d'algorithmes sur le flux de twitter et de voir quels sont les algorithmes qui donneront des meilleurs résultats.

Nous rappelons que les algorithmes de Clustering ne prend pas directement du texte mais des vecteurs numérique pour cela on nous effectuons une transformation à base statistique du texte.

Cette étape prend comme entrée :

– la liste des mots uniques existant dans le flux à analyser après prétraitement.
– la liste des listes des mots de chaque tweet après le pré-traitement.
– le type de méthode à appliquer.

Cette étape fournit comme sortie des vecteurs numériques correspondants à chaque tweet qui est une représentation statistique des différents mots de chaque tweet sélectionné en fonction de la formule utilisé.

Nous avons utilisé différentes manières des de créer ces vecteurs sur la base de la technique des bag-of-words :

Occurrence Il s'agit de la formule la plus simple, qui se base sur le nombre de fois où apparaît le terme, dans notre cas c'est l'occurrence d'un terme dans le tweet. Le vecteur en sortie a comme taille le nombre de mots uniques de flux étudié. La valeur de chaque élément correspond au nombre d'occurrences du mot dans le tweet. Le valeurs des mots qui ne sont pas présents dans le tweet sont égaux à 0. Les vecteurs obtenus sont éparses, avec peu de valeurs non nulles.

l'inconvénient de cette méthode est que ne prend pas en compte l'importance du mot dans le corpus, c'est à dire combien de fois le mot existe dans tout le corpus.

TF-IDF (Term Frequency-Inverse Document Frequency) Cette formule reflète l'importance du mot dans le document avec le TF et son importance dans le corpus par l'IDF. La valeur finale présente le rapport entre les deux.

il y a une variation de cette formule mais on a opté l'utilisation de la suivante avec ces paramètres :

la formule tf-idf d'un terme t est :

$$TF - IDF(t) = \left(\frac{a}{b}\right) * log\left(\frac{c}{d}\right) \tag{1}$$

où :

- a : le nombre d'occurrences du terme t dans le tweet.
- b : la taille de tweet (en nombre de mots).
- c : le nombre total des tweets analysés.
- d : le nombre de tweets où ce terme t est présent.

Z-Score est une autre méthode pour déterminer les mots les plus important dans un corpus et elle appartient au modèle bag-of-words statistique avec une principe similaire au TF-IDF.

Pour cette formule on prend les considérations suivantes :

- P_0 dénote la partie 0 du corpus et P_1 dénote tout le corpus sauf la partie 0 donc le $P_{corpus} = P_0 + P_1$.
- dans notre cas on prend P_0 comme le tweet concerné et P_1 comme le reste des tweets.
- tfi_0 représente l'occurrence du terme dans la P_0, et tfi_1 son occurrence dans P_1 par conséquent l'occurrence de ce terme dans tout le corpus tfi = tfi_0 + tfi_1.
- n_0 représente le nombre de termes dans P_0 et n_1 le nombre de termes dans P_1, et par conséquent n = n_0 + n_1.
- $P(t_i)$: représente la probabilité que le terme t_i étant choisi au hasard dans l'ensemble du corpus, en se basant sur le maximum de vraisemblance.

la formule Z-score d'un terme (ti) :

$$Z - score(t_{i0}) = \frac{tf_{i0} - n_0 \cdot P(t_i)}{\sqrt{n_0 \cdot P(t_i) \cdot (1 - P(t_i))}} \tag{2}$$

où :

– tfi_0 : l'occurrence du terme dans le tweet.
– n_0 : le nombre de terme dans le tweet.
– $P(t_i)$: cette probabilité serait estimée comme $P(t_i) = (tfi_0 + tfi_1)$ / n.

TD-IDF et Z-score sont faciles à implémenter et elles peuvent être utilisés pour détecter la similarité statistique entre deux documents, mais elle prennent pas en compte la position des mots dans le texte. L'autre inconvénient comme toutes les méthodes statistiques, il n'y a pas de sémantique associée.

Pour tous nos vecteurs les attributs sont l'ensemble des mots uniques et les individus sont des vecteurs qui reflète la présence de les mots unique dans chaque tweet.

A la fin de cette étape on génère les fichiers associés à chaque type de transformation pour pouvoir les passer vers les algorithmes de clustering.

3.5 Phase 5 : Clustering

Comme a été évoquée lors de l'état de l'art, dans cette phase nous allons utiliser les quatre algorithmes de clustering suivants :

kmeans [3] Un des algorithmes les plus connus en partitionnement de données qui prend comme paramètre obligatoire le nombre de clusters. La construction de clusters est basé sur la distance entre les individus.

Clustering hiérarchique [2] Une méthode de clustering ascendante basée sur la dissimilarité, où initialement chaque individu forme un cluster, et on cherche a réduire le nombre de clusters niveau par niveau, en fusionnant des clusters proches, jusqu'à l'obtention d'un seul cluster.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [1] Un algorithme de clustering basé sur la densité, il prend comme paramètres obligatoires le rayon maximum de voisinage (qui représente une estimation de la densité) et comme deuxième paramètre le minimum des points pour considérer un partitionnement comme un cluster. Le point faible de cet algorithme est qu'il n'est pas capable de construire des clusters de densités différentes.

OPTICS (Ordering Points To Identify the Clustering Structure) [4] Est un autre algorithme basé sur la densité, et représente le successeur de DBSCAN, il a été développé pour surmonter l'inconvénient principal de DBSCAN et permettre de construire des clusters de densités différentes.

Quelques considérations sur l'utilisation des méthodes de clustering :

– on utilise le Kmeans si on veut avoir un clustreing avec un nombre prédéfini de clusters, et en temps d'exécution raisonnable.
– pour le DBSCAN malgré le temps d'exécution très lourd, on utilise cet algorithme si on ne sait pas a priori le nombre de clusters.

Pour ces quatre algorithmes on a d'autres paramètres qui ne sont pas obligatoires comme :

- le type de la fonction de distance (distance Euclidienne , distance de Manhattan).
- la distance entre les centres de clusters (minimale, maximale, moyenne ....etc) pour le kmeans et le clustering hérarchique.

3.6 Phase 6 : Analyse statistique et visualisation

Dans cette partie nous avons effectué une analyse statistique sur notre flux d'entrée. Pour des objectifs simples comme :

- le suivi de l'utilisation d'un hashtag donné.
- le suivi des fréquences de nos tweets (combien de tweets par plage horaire).
- les hashtags les plus utilisés dans une collection donnée.
- calculer la corrélation entre deux Clustering différents.

Cette phase a également un autre objectif, qui est la détermination des mots les plus importants dans le flux du texte (les mots les plus importants peuvent être les nouveaux entrées de la requête du collecte pour notre deuxième objectif).

Pour cela on a utilisé le TF-IDF et le Z-score sur les résultats de Clustering, mais avec un changement de paramètres.

TF-IDF le document devient le Cluster et le corpus est l'ensemble des autres clusters.

Z-Score la partie P_0 prend un Cluster donné. La partie P_1 est l'ensemble des autres clusters.

Pour les mots les plus importants, nous recommandons les termes qui ont des valeurs TF-IDF ou Z-score les plus élevés dans le fils discussion étant des nouveaux mots-clés.

Après l'extraction des statistiques sur les tweets et l'identification des mots les plus importants, on passe les résultats vers l'étape de visualisation, qui la deuxième partie de cette phase, où nous montrerons nos résultats avec des courbes, histogrammes, listes des mots, nuages de tags et des matrices de corrélations.

4 Implémentation

Dans cette section nous allons détailler l'implémentation de chaque partie de notre application.

Dans la phase de collecte nous avons utilisé une API Twitter qui s'appelle 'API Streaming' à l'aide de la bibliothèque Twitter4j en java. Selon Wikipédia, une API est un ensemble de classes et méthodes mise à disposition des

développeurs tiers, leur permettant d'interagir avec le service Web afin de faciliter l'utilisation de certaines fonctionnalités ou pour permettre un accès direct aux données du site.

Twitter dispose de plusieurs APIs permettant de requêter sa base de données, mais aussi de construire des services au-dessus de sa plate-forme. Parmi ses APIs, l'API Streaming. Cette dernière ne renvoie pas de données historiques, mais elle retournera les tweets postés en quasi temps-réel, en mode streaming et en fonction de la requête élaborée. Cette API permet l'accès aux gros volumes de données Twitter avec peu de contraintes. principalement la difficulté d'atteindre un volume équivalent à 1% (plafond d'utilisation) des messages publiés sur Twitter à un instant t. Cette API retourne les résultats d'une requête dans un format brut JSON (JavaScript Object Notation). Pour pouvoir récupérer le flux de données Twitter, il suffit d'avoir un compte développeur et de créer une application Twitter pour s'authentifier à la plate-forme.

Les données retournées par l'application de collecte sont stockées directement dans MongoDB, qui est un système de gestion de base de données (SGBD) NoSQL de type orienté document. Récemment le MongoDB devient l'un des SGBD les plus populaires dans le monde Selon [10]. Aussi, de nombreux sites géants ont adopté ce système tels que eBay, Craigslist, SourceForge et le New York Times.

Une base de données MongoDB contient un ensemble de collections, chacune d'elles est équivalente à une table dans le SQL. Mais son point fort réside dans la manipulation des données sans schéma prédéfini.

Nous avons implanté notre système en utilisant le langage Java, en utilisant la bibliothèque Weka pour les opérations de partitionnement. Pour récupérer les données à partir de MongoDB avant de les exploiter en java, nous faisons interagir la base de données via l'intermédiaire de package 'Java MongoDB Driver'.

Durant la phase de nettoyage, nous avons utilisé les expressions régulières 'Regex' en java pour identifier les chaînes de caractères à nettoyer.

La phase de transformation des textes vers des vecteurs numériques nous a posé un nouveau défi qui est la grande taille mémoire utilisée et le temps de calcul relativement lent, nous le montrons sur l'exemple suivant :

Un vecteur numérique de type Z-score avec 2000 tweets représente :

- 8516 mots uniques.
- des valeurs Double où chaque variable est de taille 8 octets.
- une matrice de 2000 * 8516 variables Double.
- une taille mémoire de la matrice de 136 Mo.
- avec un fichier Arff (format de fichier de la bibliothèque Weka) généré de 341 Mo.

Afin d'optimiser l'utilisation de la mémoire et le temps de traitement nous avons mis en place un certain nombre d'astuces :

- utilisation de la structure HashMap : c'est une structure de données en java qui permet de stocker des données sous la forme clé-valeur, avec un mécanisme d'indexation par hachage qui rend l'accès aux éléments très rapide.

– malgré l'inconvénient du manque d'ordre dans les HashMaps on a choisi de les utiliser pour profiter de la rapidité d'accès avec l'utilisation des itérateurs pour les parcourir si nécessaire.
– sauvegarder les mots uniques dans un fichier pour éviter le re-traitement plus tard dans d'autres étapes.
– sauvegarder les vecteurs d'occurrences, de TF-IDF et de Z-Score dans des fichiers pour éviter de refaire le calcul dans d'autres expérimentations.

Dans la phase de clustering nous avons utilisé l'implémentation des quatre algorithmes en Weka [20] mais pour le DBSCAN on a aussi utilisé sa implémentation en R [7] pour voir si elle est plus rapide.

Dans la phase de visualisation nous avons utilisé la bibliothèque JFreeChart [12] en java pour tracer les courbes et les histogrammes. Ainsi nous avons utilisé la bibliothèque D3.js [8] pour dessiner les listes de mots et les nuages de termes. Finalement nous utilisons R pour visualiser la matrice de corrélation entre les clusters.

5 Résultats et Discussion

Avec les quatre algorithmes de clustering, les différents stratégies de représentation du texte et le grand nombre de paramètres dans la phase de prétraitement, il était nécessaire de mettre en place un protocole d'expérimentation qui permet de varier ces paramètres et de trouver l'influence sur les résultats finaux.

Pour concevoir ce protocole d'expérimentation nous avons défini les valeurs possibles de chaque paramètre dans des vecteurs, pour l'exploitation automatique. Aussi nous avons utilisé une base de données qui contient la configuration de chaque expérimentation exécuté (les valeurs de toutes les paramètres utilisées) ainsi le répertoire généré de cette dernière, ce répertoire contient les fichiers du résultats de l'expérimentation courante. La base d'expérimentation permet d'éviter les redondances des testes avec les mêmes paramètres.

La (Figure 4) représente une partie des exécutions effectuées en fonction de notre protocole d'expérimentation. Chaque étape possède un code (E1, N1, ...) qui vont identifier les paramètres d'une expérience. Ils vont également permettre de nommer sans ambiguïté les fichiers temporaires liés à cette expérience. Nous avons défini un tableau de jeux de paramètres et valeurs possibles pour chaque étape, et nous avons leurs attribuées des noms.

Par exemple : le code E1N1V1C1 identifie l'expérimentation où en utilisant comme entrée les textes de tous les tweets (E1: correspond à la requête retournant ces textes), sur lesquelles nous n'appliquons aucun nettoyage (N1: pas de nettoyage) et nous allons les transformer en vecteur en utilisant TF-IDF (V1) et ils vont être regrouper selon la méthode Kmeans (C1) avec un jeu de paramètres.

Durant ce projet nous avons tourné des centaines d'expérimentations, ce qui nous a permis de trouver le bon paramétrage pour avoir les résultats souhaité et aussi pour comprendre l'influence de chaque paramètre.

Entré	Nettoyage	Vecteurs	Clustering	Répertoire
E1	N1	TF-IDF V1	KMeans C1	1111
			DBScan C2	1112
		Z-Score V2	KMeans	1121
			DBScan	1122

Fig. 4. Expérience à appliquer en suivant le protocole d'expérimentation.

Lors des premiers expérimentations effectuées sans nettoyage, nous avons constaté que notre regroupement donne des clusters non équilibrés (Figure 5) pour les quatre algorithmes et on arrive pas à identifier les fils des discussions.

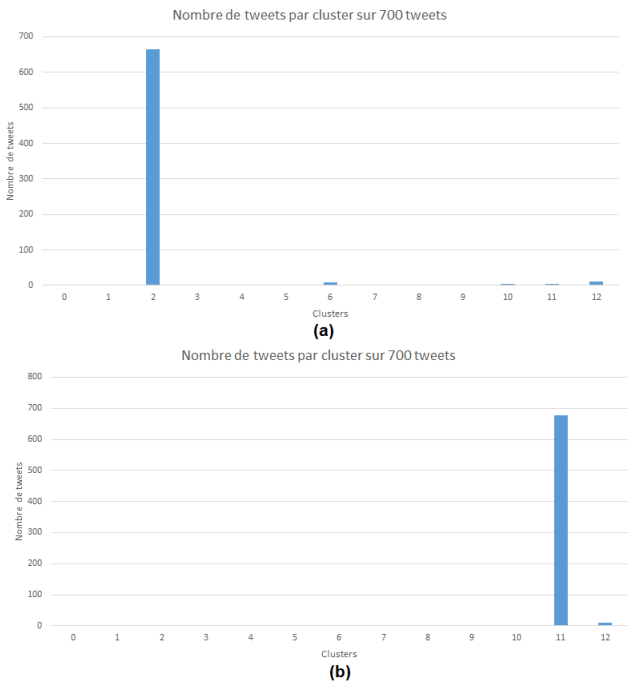

Fig. 5. Clustering non-équilibré sans aucun nettoyage avec (a) Kmeans, (b) DBSCAN.

L'utilisation combinée des types de nettoyage parmi : la suppression des liens, la suppression des identifiants, la conversion en minuscule et la suppression des caractères seuls ne conduit que vers un impact minimal sur les résultats finaux de clustering et cela donne aussi des partitions non-équilibrées (Figure 6).

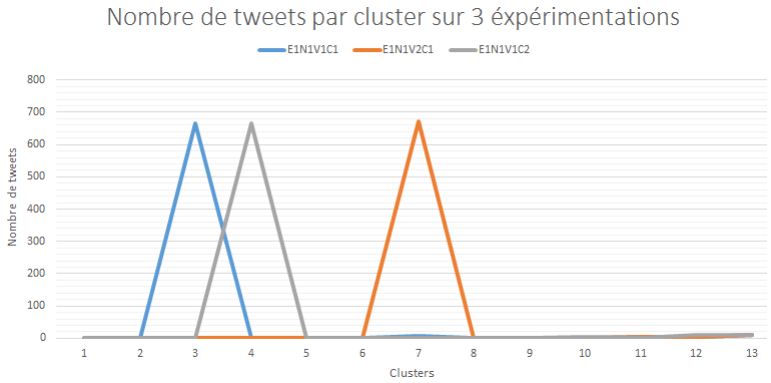

Fig. 6. Clustering non-équilibré avec la variation de prétraitement (trois expérimentations).

La variation des paramètres des différents algorithmes de clustering permet d'identifier des fils de discussions clairement. (Figure 7) montre l'identification des fils de discussions où les cases contiennent des termes ayant des scores TF-IDF les plus élevés pour un cluster donné. Par exemple, le (Cluster12) concerne la fête des lumières, un autre fil de discussion parle des sujets politiques (les élections et Front National dans le Cluster8) et dans d'autres expérimentations on arrive à trouver des fils de discussion autour du football et l'Olympique Lyonnais.

Fig. 7. Fils de discussions trouvés en utilisant le Kmeans et TF-IDF.

Un autre indicateur positif de notre processus de Clustering et les stratégies de représentation du texte que nous avons utilisées est de regrouper les Tweets en anglais dans le même cluster. (Figure 8) représente des tweets en anglais trouvé dans le même Cluster TF-IDF et Z-score.

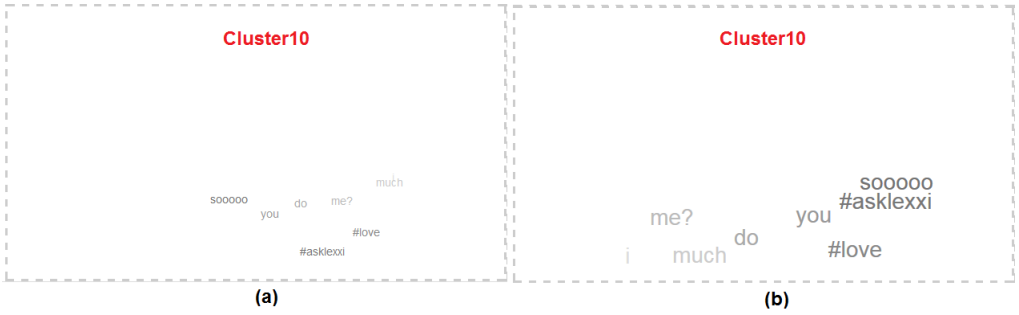

Fig. 8. Clusters contenant des tweets en anglais en fonction de : (a) TF-IDF, (b) Z-Score.

La combinaison de bons paramètres de clustering avec le paramètre le plus important de prétraitement (trouvés par expérimentation) qui sont :

– Clustering avec les 2 types de nettoyage qui sont la mise a jour des hashtags par des dièses et la suppression du reste des mots.

– Transformation en vecteurs par conversion vers des vecteurs numériques Z-score.

– Clustering avec l'algorithme Kmeans.

Cela a permet d'avoir un clustering équilibré (Figure 9), et d'améliorer considérablement l'identification des fils de discussions et aussi les termes les plus importants dans chaque cluster. (Figure 10) décrit les résultats du Clustering sur les hashtags en fonction du Z-score où on trouve 3 Clusters parmi 12 qui parlent de la fête de lumières 2015. Cet événement a été annulé et transformé en hommage aux victimes des attentats terroristes de 13 novembre à Paris. Du coup des termes comme solidaire et victimes deviennent une partie du sujet cet événement. Ces mots pourront être utilisés pour enrichir la requête de collecte afin de bien couvrir le sujet fête des lumières 2015.

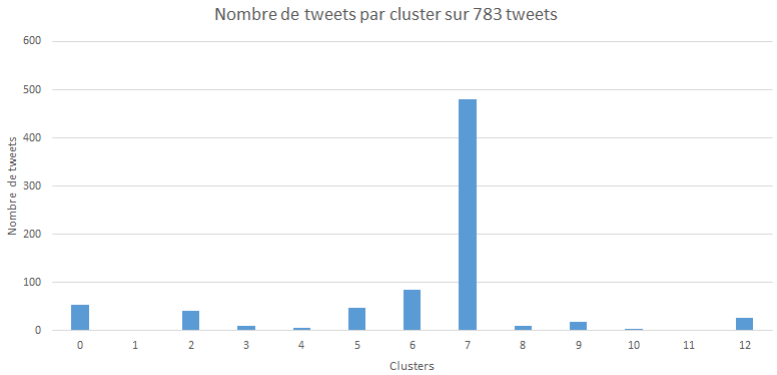

Fig. 9. Regroupement sur des hashtags qui donne des clusters équilibrés.

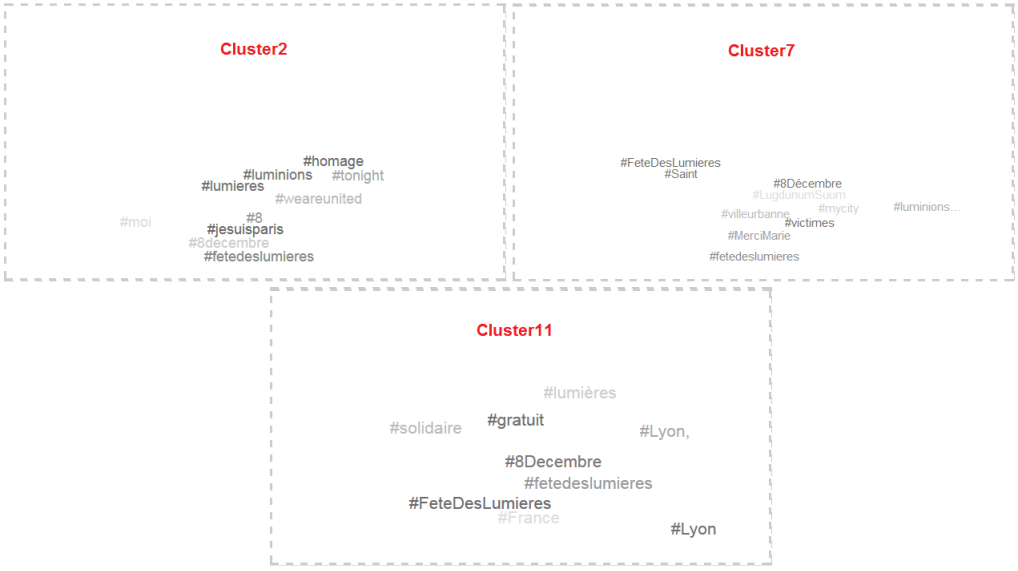

Fig. 10. Clusters concernent la fête des lumières 2015 avec leurs termes les plus importants.

6 Conclusion et perspectives

A partir d'une requête de collecte qui contient les cordonnées géographiques de la région Lyonnaise et les mots : "Lyon, FeteDesLumieres2015, FDL2015 et lumignon" nous avons pu identifier des fils de discussions qui parlent de cet événement culturel sur Twitter, ce qui représente notre objectif principale. Aussi à la fin de notre processus nous arriverons à suggérer automatiquement d'autres termes liés à cet événement qui peuvent enrichir la requête de collecte, pour le deuxième objectif de ce travail.

Notre contribution consiste à effectuer un regroupement (Clustering) sur le flux de Twitter et Instagram en passant par une phase de nettoyage et une phase de conversion du texte vers des vecteurs numériques, en fonction du présence des termes statistiquement.

Comme perspective, nous pouvons évoquer:

- Créer un jeu de test et de validation de grande taille (>1k tweets)
- Le passage vers le Clustering en ligne (en temps réel par des fenêtres temporelles).
- L'utilisation d'une méthode de transformation du texte à base sémantique comme LDA(Latent Dirichlet Allocation) ou LSA(Latent Semantic Analysis) est une autre perspective scientifique pour pouvoir la comparer avec l'approche statistique déjà appliquée.

– L'invention d'une fonction de distance entre les tweets qui comporte des distances spatiales, temporelles, sémantiques, syntaxiques et entre utilisateurs est un défi ambitieux pour nous dans le futur, où il va nous permettre de conquérir d'autre approche pour le calcul de similarité dans le contexte d'analyse du flux Twitter, et éventuellement élargir l'utilisation de ce travail vers d'autres objectifs comme l'analyse des tendances et l'extraction de connaissance à partir des réseaux sociaux.

7 Références

References

1. DBSCAN. https://en.wikipedia.org/wiki/DBSCAN. 2016-06-12.
2. Hierarchical clustering. https://en.wikipedia.org/wiki/Hierarchical_clustering. 2016-06-13.
3. k-means clustering. https://en.wikipedia.org/wiki/K-means_clustering. 2016-06-13.
4. OPTICS algorithm. https://en.wikipedia.org/wiki/OPTICS_algorithm. 2016-06-13.
5. Twitter, les chiffres de l'entreprise. https://about.twitter.com/fr/company. 2016-06-13.
6. Semantic Expansion of Tweet Contents for Enhanced Event Detection in Twitter. In *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 20–24. IEEE, aug 2012.
7. Douglas Bates. The R Project for Statistical Computing. https://www.r-project.org/. 2016-06-12.
8. Mike Bostock. Data-Driven Documents library. https://d3js.org/. 2016-06-12.
9. Changhyun Byun, Yanggon Kim, Hyeoncheol Lee, and Kwangmi Ko Kim. Automated Twitter data collecting tool and case study with rule-based analysis. In *Proceedings of the 14th International Conference on Information Integration and Web-based Applications & Services - IIWAS '12*, page 196, New York, New York, USA, dec 2012. ACM Press.
10. DB-Engines. DB-Engines Ranking. http://db-engines.com/en/ranking. 2016-06-13.
11. Dehong Gao, Wenjie Li, Xiaoyan Cai, Renxian Zhang, and You Ouyang. Sequential summarization: A full view of twitter trending topics. *IEEE Transactions on Audio, Speech and Language Processing*, 22(2):293–302, feb 2014.
12. David Gilbert. JFreeChart. http://www.jfree.org/jfreechart/. 2016-06-12.
13. Youngsub Han, Hyeoncheol Lee, and Yanggon Kim. A real-time knowledge extracting system from social big data using distributed architecture. In *Proceedings of the 2015 Conference on research in adaptive and convergent systems - RACS*, pages 74–79, New York, New York, USA, 2015. ACM Press.
14. Georgiana Ifrim, Bichen Shi, and Igor Brigadir. Event detection in Twitter using aggressive filtering and hierarchical tweet clustering. *CEUR Workshop Proceedings*, 1150:33–40, 2014.
15. Kathy Lee, Diana Palsetia, Ramanathan Narayanan, Md. Mostofa Ali Patwary, Ankit Agrawal, and Alok N. Choudhary. Twitter trending topic classification. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on, Vancouver, BC, Canada, December 11, 2011*, pages 251–258, 2011.

16. Marc Plantevit. Différentes techniques de clustering. "Cours" 2016.
17. Robert Popovici, Andreas Weiler, and Michael Grossniklaus. On-line Clustering for Real-Time Topic Detection in Social Media Streaming Data, 2014.
18. Daniele Quercia, Harry Askham, and Jon Crowcroft. Tweetlda: supervised topic classification and link prediction in twitter. In *Web Science 2012, WebSci '12, Evanston, IL, USA - June 22 - 24, 2012*, pages 247–250, 2012.
19. Jacques Savoy. Text representation strategies: An example with the State of the union addresses. *Journal of the Association for Information Science and Technology*, 66(8):1645–1654, jun 2015.
20. Ian Witten. Weka 3: Data Mining Software in Java. 2016-06-10.
21. Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. Comparing twitter and traditional media using topic models. In *Advances in Information Retrieval - 33rd European Conference on IR Research, ECIR 2011, Dublin, Ireland, April 18-21, 2011. Proceedings*, pages 338–349, 2011.

A Annexe : Exemple d'un tweet collecté en format JSON


```
{
    "_id" : ObjectId("566582f4bc16ea0554d4ec89"),
    "contributors" : null,
    "coordinates" : null,
    "created_at" : "Mon Dec 07 13:00:36 +0000 2015",
    "entities" : {
        "hashtags" : [],
        "symbols" : [],
        "urls" : [],
        "user_mentions" : [
            {
                "id" : 145176185,
                "id_str" : "145176185",
                "indices" : [
                    0,
                    13
                ],
                "name" : "AccorHotels Arena",
                "screen_name" : "AccorH_Arena"
            }
        ]
    },
    "favorite_count" : 0,
    "favorited" : false,
    "filter_level" : "low",
    "geo" : null,
    "id" : NumberLong(673849591943532544),
    "id_str" : "673849591943532544",
    "in_reply_to_screen_name" : "AccorH_Arena",
    "in_reply_to_status_id" : null,
    "in_reply_to_status_id_str" : null,
    "in_reply_to_user_id" : 145176185,
    "in_reply_to_user_id_str" : "145176185",
    "is_quote_status" : false,
    "lang" : "fr",
    "link" : [
        "https://twitter.com/_Vivi69/status/673849591943532544"
    ],
    "place" : {
        "attributes" : {},
        "bounding_box" : {
            "coordinates" : [
                [
                    [
```

```
                    4.771831,
                    45.707363
                ],
                [
                    4.771831,
                    45.808281
                ],
                [
                    4.898367,
                    45.808281
                ],
                [
                    4.898367,
                    45.707363
                ]
            ]
        ],
        "type" : "Polygon"
    },
    "country" : "France",
    "country_code" : "FR",
    "full_name" : "Lyon, Rhône-Alpes",
    "id" : "179b8df9e368044d",
    "name" : "Lyon",
    "place_type" : "city",
    "url" : "https://api.twitter.com/1.1/geo/id/179b8df9e368044d.json"
},
"retweet_count" : 0,
"retweeted" : false,
"source" : "<a href=\"http://twitter.com/download/android\" rel=\"nofollow\">Twitter for
"text" : "@AccorH_Arena bonjour, sera-t-il possible d'apporter un appareil photo compact
"timestamp_ms" : "1449493236896",
"truncated" : false,
"user" : {
    "contributors_enabled" : false,
    "created_at" : "Sat Feb 12 13:02:32 +0000 2011",
    "default_profile" : false,
    "default_profile_image" : false,
    "description" : "Fan de foot, joueuse de badminton et supporter de l'#OL #TeamOL...
    "favourites_count" : 967,
    "follow_request_sent" : null,
    "followers_count" : 73,
    "following" : null,
    "friends_count" : 336,
    "geo_enabled" : true,
```

```
        "id" : 251104275,
        "id_str" : "251104275",
        "is_translator" : false,
        "lang" : "fr",
        "listed_count" : 9,
        "location" : "Lyon - France",
        "name" : "Vivi",
        "notifications" : null,
        "profile_background_color" : "022330",
        "profile_background_image_url" : "http://abs.twimg.com/images/themes/theme15/bg.png"
        "profile_background_image_url_https" : "https://abs.twimg.com/images/themes/theme15/
        "profile_background_tile" : false,
        "profile_banner_url" : "https://pbs.twimg.com/profile_banners/251104275/1440602912"
        "profile_image_url" : "http://pbs.twimg.com/profile_images/629252559165915136/bEarCx
        "profile_image_url_https" : "https://pbs.twimg.com/profile_images/62925255916591513
        "profile_link_color" : "4682B4",
        "profile_sidebar_border_color" : "A8C7F7",
        "profile_sidebar_fill_color" : "C0DFEC",
        "profile_text_color" : "333333",
        "profile_use_background_image" : true,
        "protected" : false,
        "screen_name" : "_Vivi69",
        "statuses_count" : 5999,
        "time_zone" : "Paris",
        "url" : "http://www.betaseries.com/membre/Vivi_69",
        "utc_offset" : 3600,
        "verified" : false
    }
}
```