REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ SAAD-DAHLEB-BLIDA

MÈMOIRE DE MASTER

DISCIPLINE : Informatique
OPTION : GL

RATR^{Tweets} : Résumé Automatique des Tweets en Temps Réel

Préparé par :

BOUBAKEUR Oussama
KOUIDRI Hichem

Devant le jury composé de :

Président	: N. CHIKHI
Examinateur	: N. BOUSTIA
Examinateur	: A. MILOUD
Promotrice	: A. MADANI

BLIDA : 2014

RÉSUMÉ

Les événements populaires en temps réel provoquent souvent une recrudescence du trafic sur Twitter. Ces messages contiennent souvent des faits importants et la mise à jour en temps réel de l'événement qui se déroule, et donc un grand nombre d'utilisateurs cherche ces mises à jour en direct pour obtenir un résumé des aspects importants de l'événement. Cependant, les principaux moteurs de recherche sociale, y compris Twitter présentent encore les tweets répondant à la requête dans l'ordre chronologique inverse. Pour avoir une vue d'ensemble des aspects important de cet événement, un utilisateur est obligé de lire des vingtaines de tweets pour pouvoir comprendre l'évènement.

Dans ce mémoire, nous proposons une méthode pour le résumé automatique des événements. Nous avons utilisé un algorithme de récupération à base de graphes qui identifie les tweets avec des points de discussion populaires parmi l'ensemble des tweets retournés par le moteur de recherche Twitter en réponse à une requête comprenant un ou plusieurs événements. Pour assurer une couverture maximale de la diversité d'actualité, nous effectuons le regroupement thématique des tweets avant d'appliquer l'algorithme de recherche.

Mots-clefs : Traitement Automatique des Langues, Recherche d'Information, Résumé automatique, Clustering, Classification.

ABSTRACT

Title: Automatic summarization of tweets in real time

Popular real-time events often cause an increase in traffic on Twitter. These messages often contain important facts and the real-time update of the event that takes place, and thus a large number of users looking for these updates directly to obtain a summary of the important aspects of the event. However, the main drivers of social research, including Twitter still represent tweets responding to the request in reverse chronological order. For an overview of the major aspects of the event, a user is forced to read around twenty of tweets in order to understand the event.

In this memory, we propose a new method for an automatic summarization of events. We used a recovery algorithm based on graph that identifies tweets with popular talking points among all tweets returned by Twitter search engine in response to a request comprising one or more events. To ensure maximum coverage of diversity of news, we perform the thematic grouping of tweets before applying the search algorithm.

Keywords : Natural Language Processing, Information Retrieval, Automatic summarization, clustering, classification.

الملخص

العنوان: التلخيص الأوتوماتيكي للتغريدة في الوقت الحقيقي.

غالبا ما تتسبب الأحداث الشعبية وقت حدوثها أو في الوقت الحقيقي في زيادة أو ضغط حركة المرور على التويتر كما تحتوي غالبا على تحديثات لحقائق و أحداث هامة و بالتالي عدد كبير من المستخدمين يبحث عن هذه التحديثات مباشرة للحصول على الجوانب الهامة لملخص الحدث أو عدة أحداث.

ان المحرك الرئيسي للأبحاث الاجتماعية بما في ذلك التويتر تقدم كذلك تويت بالمواصفات المطلوبة في ترتيب زمني عكسي وحتى تكون لدينا لمحة عن الجوانب الرئيسية لهذا الحدث فان المستخدم مضطر لقراءة عشرات التويت لفهم الحدث.

في هذا البحث نقترح طريقة التلخيص التلقائي للأحداث قمنا باستخدام خوارزمية الانتعاش أو ما تعرف بخوارزمية الاسترجاع تعتمد على الرسم البياني و باستطاعتها التعرف على التغريدة المرجعة من طرف محرك البحث تويتر استجابة لطلب يضم حدث واحد او عدة أحداث و لضمان تغطية شاملة للتنوع في الأخبار نقوم باجراء التجميع المواضيعي لكل تويت قبل تطبيق خوارزمية البحث.

الكلمات الرئيسية: المعالجة الأوتوماتكية للغات، التلخيص الأوتوماتيكي، استخراج المعلومات، استرجاع المعلومات، التصنيف.

REMERCIEMENTS

A l'issue de ce travail, nous remercions, en premier lieu, le bon **Dieu** de nous avoir donné la force et le courage de le mener à terme.

Nous tenons, également, à exprimer notre sincère reconnaissance et notre profonde gratitude à tous ceux qui ont contribué de près ou de loin à la réalisation de ce mémoire, notamment notre promotrice, **Mme. MADANI Amina,** qui grâce à elle nous avons eu l'opportunité de découvrir le domaine du résumé automatique. Ses conseils illuminés et son aide précieux nous ont permis de mener à bien ce modeste travail et on la remercie très chaleureusement.

Nous remercions la promotion de **Master II informatique GL 2013/2014**, et tous les **enseignant**s qui ont contribué à notre formation.

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

LISTE DES EQUATIONS

INTRODUCTION GÉNÉRALE

1. Introduction

DE nos jours, l'Internet se révèle plus que jamais un outil indispensable d'échange d'informations. Une quantité considérable d'informations est offerte à une vitesse inédite d'où ses services s'adaptent de plus en plus aux besoins des internautes. Ceux-ci peuvent consulter l'Internet pour trouver de l'information, envoyer des e-mails, acheter des produits, lire des journaux en ligne, etc. Pendant les dernières années, l'Internet a connu encore une plus vaste portée grâce au développement des médias sociaux. Basés sur des techniques de communication faciles et accessibles pour tous, ces médias favorisent les interactions sociales à travers l'Internet. Les médias sociaux se distinguent des médias traditionnels tels que les journaux, la télévision et la radio car leur utilisation est peu coûteuse et libre, de façon à permettre à tout le monde d'y accéder ou de publier de l'information. Ils subviennent aux besoins des individus d'échanger des opinions, de demander des conseils et de communiquer de façon rapide et facile.

Les médias sociaux qui ont récemment pu bénéficier d'un considérable essor sont les réseaux sociaux tels que Facebook[1], LinkedIn[2] et Twitter[3]. Ce sont des sites web qui rassemblent des identités sociales telles que des individus, des entreprises et des organisations qui peuvent échanger de l'information à travers des interactions sociales. Grâce à leur caractère maniable et leur accès libre, les réseaux sociaux bénéficient d'un succès croissant auprès du grand public.

La popularité des réseaux sociaux est d'autant plus grande que la demande d'informations qui est devenue plus importante dans notre société. En général, les gens aiment être informés de ce qui se passe autour d'eux. Autrefois, ces informations provenaient surtout de leur environnement social direct. Aujourd'hui, les gens aiment également être informés des évènements qui se passent dans le monde entier en temps réel.

[1] https://www.facebook.com
[2] https://www.linkedin.com
[3] https://twitter.com

Dans le cadre des systèmes de résumé automatique, deux aspects sont particulièrement importants : d'une part la découverte des sujets dans les messages retournés pour une requête donnée et, d'autre part, les stratégies d'extraction employées pour trouver les messages pertinents constituant le résumé pour ces sujets. Actuellement, ces deux aspects sont relativement indépendants. Ainsi, à partir d'une requête ou une collection de messages, de tels systèmes génèrent dans un premier temps, une modélisation des sujets. Celles-ci sont ensuite utilisées pour catégoriser les messages selon les sujets découverts. D'autre part il y a le système de résumé qui se charge de trouver les N messages les plus pertinents dans une collection dans lesquels appliquer des stratégies de recherche et d'extraction.

Les problématiques abordées dans ce mémoire sont de définir une adaptation unifiée pour la modélisation des sujets émergents (tendances[5]) à partir d'un ensemble de tweets et des stratégies d'extraction pour résumer le primaire " essentiel " de ce que les utilisateurs disent à propos de ces sujets en temps réel.

3. Domaine : Traitement Automatique des Langues

Notre travail s'inscrit dans le cadre du traitement automatique des langues naturelles (ou TAL). Nous nous situons dans une approche statistique du TAL, au sens où nous nous reposons sur des modèles et méthodes issus de corpus[6], à l'aide d'apprentissage automatique.

Le TAL est un domaine issu de l'Intelligence Artificielle, et s'intéresse particulièrement à des problèmes de génération et de compréhension automatique des langues naturelles avec le support d'approches originaires des domaines de l'apprentissage automatique et de la linguistique computationnelle **[Callison-Burch et Osborne, 2003]**. Ainsi, il n'est pas rare que les approches utilisées en TAL empruntent aux modèles probabilistes, à la théorie de l'information, et à l'algèbre linéaire **[Manning et Schütze, 1999]**.

Parmi les tâches importantes auxquelles s'est attaqué le TAL, on compte l'annotation, l'analyse syntaxique, la traduction, la classification de textes et le résumé automatique. Ces deux dernières tâches sont souvent considérées dans la littérature comme des tâches de Recherche d'Information **[Shakespeare, 1946]**, **[Callison-Burch et Osborne, 2003]** et qui font l'objet de notre étude.

---

[5]Les **tendances** sont les sujets importants discutés sur twitter. (Source : http://en.wikipedia.org/wiki/Twitter#Trending_topics).

[6] Un **corpus** est un ensemble de documents, artistiques ou non (textes, images, vidéos, etc.), regroupés dans une optique précise. . (Source : http://fr.wikipedia.org/wiki/Corpus).

Tous les récents développements dans le domaine d'échange d'informations ont donné une direction importante vers les applications informatiques conçues pour l'analyse et la détection d'évènements exprimés sur Internet. Les messages envoyés via les réseaux sociaux tels que Twitter constitue une source précieuse d'information échangée parmi les multiples internautes. Par conséquent, il est important de concevoir des systèmes automatiques aptes à rechercher et à résumer les évènements qui sont exprimés sur les réseaux sociaux.

Le *résumé automatique* est utilisé pour la détection des évènements importants sur les sites web et les réseaux sociaux ainsi que l'éclaircissement sur un événement particulier en présentant l'information essentielle. Il consiste à rechercher des textes contenant des informations de haute importance et les analyser de façon automatique afin de mieux comprendre cet évènement.

À cet effet, une grande partie de cette étude sera consacrée au résumé automatique exprimé dans les tweets[4].

2. Problématique

Pouvoir, à partir d'un ensemble de documents, extraire une information, est un des plus anciens sujets traités en informatique. Ce sujet a débouché sur le domaine de la « Recherche d'Information » (RI), terme introduit en 1950 par Calvin Northrup Mooers. Il a conduit les travaux fondateurs dans ce domaine avec le développement du système de « Zato coding » **[Mooers, 1948]**.

Il est inutile de rappeler l'importance qu'a de nos jours le domaine de la recherche d'information, en particulier par l'utilisation massive des réseaux sociaux qui ont permis un accès facile et efficace aux quantités d'informations en croissance exponentielle disponibles sur Internet. Cet impact sociétal peut se mesurer en particulier par le fait qu'un réseau social, **TWITTER**, a récemment pu bénéficier d'un considérable essor dans ce domaine, mais les réseaux sociaux ont certaines limitations, en particulier lorsque la requête est faite dans l'intention d'obtenir une réponse précise concernant un événement particulier : Ils renvoient un ensemble de messages pouvant contenir l'information demandée, à charge au demandeur de la trouver dans cette liste de messages. Ceci a conduit au développement des recherches sur le principe des systèmes de résumé automatique où on propose directement à l'utilisateur une réponse précise à sa question en la résumant.

[4]Les **tweets** sont des messages de 140 caractères au maximum qui sont envoyés via le réseau social Twitter. (Source : http://en.wikipedia.org/wiki/Twitter#Tweets).

4. Objectifs

Quelques tentatives ont été réalisées dans le domaine du résumé automatique. Dans la première étape de notre travail, nous avons exposé l'évolution de ces tentatives en faisant une analyse approfondie de l'état de l'art et en parcourant un nombre important des travaux réalisés dans ce domaine. Aussi, nous avons arrêté un certain nombre de critères pour pouvoir faire une comparaison entre ces travaux.

L'étape suivante a permis de proposer une approche capable d'extraire les tendances à partir des tweets en temps réel en utilisant le résumé automatique. Notre objectif est de fournir de bons résultats (avec une précision importante) en utilisant une collection de données appropriées à ce genre de travail.

5. Organisation du mémoire

Ce mémoire s'articule en 4 chapitres principaux : **le chapitre 1** présente Twitter. Ce chapitre nous sert à comprendre Twitter en général ainsi que sa structure.

Le chapitre 2 traite de l'état de l'art du résumé automatique des tweets. Nous proposons une étude approfondie sur un ensemble de travaux réalisés dans le domaine du résumé automatique des tweets.

Le chapitre 3 présente une nouvelle approche de résumé automatique des tweets en temps réel. Cette approche s'appuie sur un modèle de modélisation de sujet, un modèle de regroupement (clustering) et un modèle de résumé.

Le chapitre 4 présente les tests et les évaluations de l'approche.

Nous clôturons ce mémoire par la conclusion générale du travail présenté dans cette étude.

CHAPITRE 1
TWITTER

1. Introduction

TEWITTER[1] un réseau social très populaire, qui repose sur le principe du microblog ou microblogue qui est un dérivé concis du blog et qui permet de publier un court article. Ces articles constituent souvent une mine d'information, qui se rapporte à des événements se produisant en temps réel. Les données provenant de ces articles peuvent être utilisées pour l'extraction des sujets émergents et le résumé automatique. Dans ce chapitre, nous donnons une présentation générale de Twitter ainsi que La structure de ses messages. Ensuite, nous décrivons un aspect important de Twitter et qui fait de lui un réseau social a part pour les études de recherche d'information, communément appelée API-Twitter.

2. Twitter

Twitter[2] est un mix de réseau social et de plateforme de micro-blogging. Des informations qui n'excèdent pas 140 caractères appelés **tweets** sont diffusées sur la plateforme Twitter. Lors de l'écriture d'un **tweet** (l'information postée), **Twitter** nous pose la question « What are you doing ? » (Que faites-vous ?). Ce microblog est donc utilisé pour présenter ce qu'il se passe autour de nous à un moment donné. L'intérêt réside sur le **temps réel,** on poste et c'est tout de suite mis en ligne, à la seconde près. La Figure **1.1** montre une capture d'écran de l'interface de l'utilisateur Twitter. Les mises à jour des statuts peuvent être envoyées via un navigateur Web, SMS, e-mail ou des tierces applications et ils sont affichés sur le profil des utilisateurs.

Les tweets sont visibles par tous et chacun peut commenter, répondre, faire passer, non à ses amis, mais à ses ''followers''.

[1] http://fr.wikipedia.org/wiki/Twitter
[2] http://cyberchemille.org/spip.php?article72

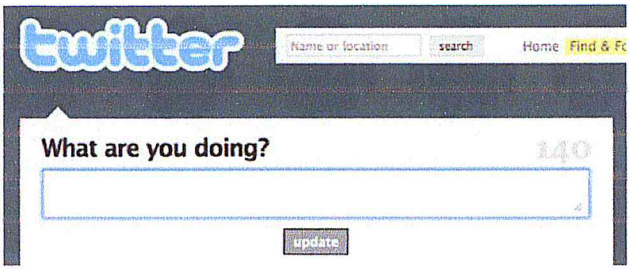


Figure 1.1 : Capture d'écran de l'interface utilisateur de Twitter.

2.1 Historique

L'histoire[3] de **Twitter** a débuté autour de 2005 chez un petit groupe de collaborateurs qui travaillait au sein de l'entreprise de démarrage Odeo, fondée par Noah Glass à San Francisco. Les trois autres sont Jack Dorsey, Biz Stone et Evan Williams. Jack Dorsey, déjà dans la fin de ses vingtaines, il entretient une passion pour la programmation. Inspiré par les communications radio des chauffeurs de taxi, le jeune homme rêve depuis quelques années déjà d'un système de communication par messages textes qu'on pourrait envoyer à un groupe d'amis par l'entremise de son téléphone cellulaire.

Twitter, qui à ses débuts se nomme Twttr en écho au site de photos flickr, voit le jour en 2006. Le 21 mars, Jack Dorsey envoie le premier gazouillis («just setting up my twttr»). L'été suivant, la plateforme est ouverte au public. À l'époque, il n'y a pas de limite au nombre de caractères permis. Twitter compte alors une centaine d'abonnés. En avril 2007, Twitter devient une véritable entreprise et Jack Dorsey en prend les commandes.

Twitter connaîtra son véritable envol un an plus tard, au Festival South by Southwest (SXSW) à Austin, au Texas. Lieu de rencontre de l'avant-garde techno, SXSW récompense Twitter en lui accordant un Web Award. Les participants à la conférence, eux, adoptent Twitter sur-le-champ. C'est le coup de baguette magique qu'il fallait. En l'espace d'un week-end, le nombre de gazouillis envoyés passe de 20 000 à 60 000.

Avec les années, les gens ont développé le réflexe de se tourner vers Twitter en temps de crise (séisme en Haïti, révolution en Égypte) pour y trouver des informations de premier plan.

2.2 Les Followers

Twitter a mis en place un concept de *followers* (**suivre** les gens). Donc, on a des followers (des personnes qui nous suivent) et ont suit les gens (on est leur follower), c'est-à-dire que l'on

[3]http://techno.lapresse.ca/dossiers/le-phenomene-twitter/201103/19/01-4381094-lhistoire-de-twitter-en-un-peu-plus-de-140-caracteres.php

suit les informations qu'ils postent et dès qu'un certain utilisateur met à jour son statut, tous les followers sont informés. Ce résultat est obtenu en ajoutant la nouvelle entrée à leur page personnelle, un aperçu est représenté sur la Figure **1.2**.

Cette opération est réalisée en cliquant sur le bouton **suivre** ou (Follow) sur une page **Twitter** .On peut suivre tous les autres utilisateurs à moins que cet utilisateur a mis son profil en mode privée. Dans ce cas, une demande d'approbation doit être envoyée en premier.

Figure 1.2 : Capture d'écran de la page personnelle Twitter.

2.3 Les tweets

Les utilisateurs de Twitter s'échangent des messages texte courts appelés **tweets** qui ne peuvent excéder 140 caractères. Un **tweet** typique se compose d'un texte et des métadonnées. La Figure **1.3** présente la structure d'un tweet.

Les métadonnées contient des informations sur l'auteur (nom, lieu, la langue, ...etc) et des informations sur le tweet (date de création, numéro d'identification, ...etc). Un texte comprend souvent des **URLs**[4] , les noms d'utilisateurs d'autres auteurs, **hashtags**[5], un signe **retweet**. La Figure **1.4** montre un exemple d'un retweet.

Les retweets sont des réponses à d'autres tweets. Ils se distinguent par le signe retweet (RT) dans le texte, qui indique que le texte suivant est d'un autre message. Hashtags et noms d'utilisateur peuvent également être distingués : hashtags suivent un signe dièse (#) et les noms d'utilisateurs suivent du signe (@).

[4]**URL** (Universal Resource Locator) : une chaîne de caractères spécifique qui constitue une référence à une ressource Internet.
[5] **Hashtag** : une manière spéciale pour marquer des entités.

Liste partielle des informations liées à un *tweet*

```
{ "id_str"=>114749583439036416,    * Numéro d'identification unique du message
"text"=>"Un message sur Tweeter fait moins de 140 caractères. Par contre, les
         informations du code produit par Tweeter fait des dizaines de lignes.",    * Texte du message
"created_at"=>"Thu Dec 13 15:48:32 +0000 2012",    * Date du message en temps universel (UTC)
"entities"=>
 { "hashtags": [ ... ],
   "urls": [ ... ],
   "user_mentions": [ ... ],
   "media": [ ... ] }
```

S'il y a lieu, éléments du message qui sont cliquables ou affichables : mots-clics, adresses URL, mentions d'autres utilisateurs, médias (les informations requises pour les « médias », comme les photos, peuvent exiger de très nombreuses lignes)

```
"retweet_count":=>85,    * Nombre de fois que le message a été retweeté
"user"=>
 { "id_str"=>22189544,    * Numéro d'identification unique de l'utilisateur
   "screen_name"=>"PierrotPeladeau",    * Nom Twitter adopté par l'utilisateur
   "name"=>"Pierrot Péladeau",    * Nom complet choisi l'utilisateur pour s'identifier
   "description"=>"Social assessment of interpersonal information systems –
                   Évaluation sociale de systèmes d'information interpersonnels",    * Description du compte ou note autobiographique
   "entities": =>
    { "url": { "urls": [{ "expanded_url": null,
      "url": " http://pierrot-peladeau.net/section/blog",  "indices": [0,22] }]},    S'il y a lieu, URL (lien internet) inclus dans la description
"lang"=>"fr",    Langue choisie par l'utilisateur pour son interface
"place"=>
 {  "attributes":
     { "street_address": =>"5 Rue de Lobau",    "locality":=>"Paris",
       "region": =>"75004",    "postal_code":=>"75004",    "iso3:=>"FRA",
       "phone": =>"01 42 76 40 40",    "623:id":=>"210176",
       "twitter":=>"Paris",    "url":=>"http://www.paris.fr" },
   "id"=>"7238f93a3e899af6",
   "url"=>"http://api.twitter.com/1/geo/id/2b6ff8c22edd9576.json",
   "name"=>"Paris",    "full_name"=>"Paris, Paris",
   "place_type"=>"city",    "country_code"=>"FR",    "country"=>"France",
   "bounding_box"=>
    { "coordinates"=>
      [[ [2.2241006,48.8155414],    [2.4699099,48.8155414],
         [2.4699099,48.9021461],    [2.2241006,48.9021461] ]],
      "type"=>"Polygon" } },
"source"=>"web"}    L'application numérique d'où provient le message
```

Identification, si l'utilisateur l'a permis, du nom, des coordonnées géographiques et d'autres informations complémentaires à propos d'un lieu qu'il a associé au message

Longitudes et latitudes des intersections des côtés périmètre de ce lieu

Figure 1.3 : La Structure d'un tweet [http://pierrot-peladeau.net/fr/archives/3668].

Figure 1.4 : Anatomie d'un tweet. [https://media.twitter.com/best-practice/anatomy-of-a-tweet].

3. L'API-Twitter

Les utilisateurs de Twitter génèrent plus de 400 millions de Tweets tous les jours[6]. Certains de ces tweets sont disponibles pour les chercheurs à travers des API publiques gratuitement. Il y a plusieurs types d'informations à extraire à partir de Twitter et qui sont les suivants :

- Information sur un utilisateur.
- Tweets publiés par un utilisateur, et
- Les résultats de la recherche sur Twitter.

Pour accéder aux données de Twitter les APIs peuvent être classés en deux types en fonction de leur méthode de conception et d'accès :

- API REST sont basés sur l'architecture[7] REST maintenant couramment utilisés pour la conception des API Web. Ces API utilisent la stratégie d'attraction pour la récupération de données. Pour recueillir des informations d'un utilisateur doit explicitement la demande.
- Streaming API fournit un flux continu de l'information publique de Twitter. Ces API utilisent la stratégie de pression pour la récupération de données. Une fois la demande de renseignements est faite, l'API streaming fournit un flux continu de mises à jour sans autre intervention de l'utilisateur.

Ils ont de différentes capacités et limites à l'égard de ce qui est de combien d'informations peuvent être récupérées. Le Streaming API a trois types de paramètres :

- Flux public (Public streams) : Ce sont des courants contenant les tweets publics sur Twitter.

[6] http://articles.washingtonpost.com/2013-03-21/business/37889387_1_ tweets-jack-dorsey-twitter
[7] http://en.wikipedia.org/wiki/Representational_state_transfer

- Les flux de l'utilisateur (User streams) : Ce sont les flux mono-utilisateur, avec pour tous les tweets d'un utilisateur.
- Site flux (Site streams) : Ce sont des flux multi-utilisateurs et destinés à des applications qui accèdent aux tweets de plusieurs utilisateurs.

Comme l'API « flux public » est l'API la plus polyvalente pour la collecte des données à partir l'API Streaming, c'est celle qu'on va utiliser dans notre étude.

4. Statistiques

L'agence DashBurst[8] a étudié et analysé l'activité du réseau social **Twitter** pour finalement rassembler les résultats au sein d'une infographie unique. Cette infographie présente donc les statistiques les plus récentes de Twitter comme le nombre de tweets envoyés, le nombre de followers moyen et le prix d'une tendance sponsorisée.

Voici les principaux chiffres à retenir concernant Twitter pour l'année 2013 :

- L'engagement moyen des followers baisse selon le nombre de tweets par jour. Si beaucoup de messages sont publiés par jour l'engagement moyen a tendance à baisser. **1 à 2 tweets quotidiens** engendrent un engagement plus important de 120% qu'avec 6 ou 7 tweets par jour.
- **170 milliards de tweets** ont été envoyés depuis le 21 mars 2006.
- Un tweet avec une image est **deux fois plus partagé** qu'un tweet sans image.
- **80% des membres** accèdent à Twitter via leur smartphone.
- **200 000 dollars**, c'est le prix d'une tendance sponsorisée à la journée.
- **40 millions** d'internautes utilisent Vine, le service vidéo lancé par Twitter en janvier 2013.
- Les membres de Twitter sont suivis en moyenne par **208 followers**.

5. Conclusion

Dans ce chapitre nous avons présenté Twitter, nous avons vu en détails la structure des tweets qui constituent nos données à traiter dans cette étude ainsi que les APIs-Twitter permettant la collection de ces données.

Nous pouvons donc conclure que Twitter a des certaines particularités qu'ils le rendent très utile pour le résumé automatique des sujets émergents, deux aspects importants de ces particularités sont les suivantes :

[8] http://dashburst.com/infographic/state-of-twitter-in-2013/

- **Twitter** est utilisé par un large public pour exprimer les différents sujets des évènements qui se rapportent en temps réel. Il constitue donc, une source précieuse d'information.
- L'**API-Twitter** permet un accès facile et efficace pour recueillir un très grand nombre de tweets. Le corpus recueilli peut être arbitrairement grand, contenant des mines d'information qui vont être exploité par la suite.

Dans le chapitre suivant, nous allons présenter un état de l'art concernant le résumé automatique des tweets et étudier en profondeur différentes approches réalisées dans ce domaine.

CHAPITRE 2
ÉTAT DE L'ART

1. Introduction

AFIN d'aider les utilisateurs à trier le grand nombre de tweets qui se produisent chaque jour, **Twitter** a ajouté un certain nombre d'outils qui aident les utilisateurs à trouver les sujets les plus importants et leurs tweets associés. La page d'accueil de **Twitter** affiche les sujets importants qui ne sont que des expressions par exemple :"Pleine Lune (full moon)», "Joie (Glee)», ou " David Guetta" - pour trois différentes gammes de temps afin de voir quels sujets sont populaires en ce moment, aujourd'hui, ou au cours de la dernière semaine. Pour la plupart des sujets, les utilisateurs sont obligés de lire les messages connexes (en cliquant sur le sujet) pour essayer de voir les détails de ce sujet. Ce processus est fastidieux et les messages retournés sont triés que par récence.

Par conséquent, pour un sujet donné, les utilisateurs sont susceptibles de rencontrer les spams, les messages dans d'autres langues et d'autres sources de désinformation. Pour aider les utilisateurs de plus, **Twitter** a établi un partenariat avec le site tiers **WhatTheTrend**[1] afin de fournir des définitions sur les tendances. WhatTheTrend[2] permet aux utilisateurs de saisir manuellement les descriptions de la raison pour laquelle un sujet est une tendance. Bien que l'idée est bonne en théorie, dans la pratique **WhatTheTrend** souffre également de spam et les utilisateurs sont libres d'entrer quelques définitions qu'ils préfèrent pour une tendance. Un rapide regard sur l'histoire de définitions pour quelques sujets, **WhatTheTrend** montre souvent des définitions oscillantes entre les utilisateurs qui tentent de spammer le site et d'autres utilisateurs qui tentent de fournir des informations exactes. Le plus grand inconvénient de ce site est que les définitions sont entrées à la main et non automatisées. Par conséquent, il y'a souvent un certain temps de latence avant qu'une nouvelle tendance soit définie par un utilisateur. Alors que **WhatTheTrend**[2] est un pas dans la bonne direction, une meilleure approche est une technique automatisée qui résume les tweets en temps réel et génère ou extrait les tendances.

[1] http://www.whatthetrend.com

Dans ce chapitre, nous donnons un bref historique de résumé automatique de texte. Nous décrivons les nombreuses dimensions et les défis associés à ce domaine diversifié. Suite à cela, nous présenterons la description des approches existantes dans le domaine du résumé automatique des tweets.

2. Les trending topics : Tendances

Les ***trending topics***[2], abrégés « TT » sur **Twitter**, sont les sujets tendances. Ce sont des mots, des **hashtags** ou des phrases qui ont été tweetés de multiples fois à un moment donné, pour un pays donné, voire tous les pays confondus. On parle alors de Worldwide Trends ou tendance mondiale. Le résumé automatique est donc utilisé pour extraire ces tendances des tweets et les résumés.

Les **tendances** sont conduites typiquement par des événements émergents et des histoires intéressantes qui attirent l'attention d'un grand nombre d'utilisateurs de Twitter **[Mathioudakis et Koudas, 2010]**. La découverte des **tendances** temps réel et leur évolution s'avère très importante pour les journalistes, les analystes ainsi que pour les spécialistes du e-marketing qui peuvent trouver une nouvelle information capitale.

3. Le résumé automatique de texte

Le résumé automatique de texte peut être défini comme le problème de génération automatique d'une version condensée de l'essentiel du contenu d'un ou plusieurs documents relatifs à un ensemble particulier d'utilisateurs ou de tâches **[Lin, 2009]**. Comme les tweets sont composés principalement de texte, il est important de comprendre quels progrès ont été accomplis dans le domaine du résumé automatique de texte afin de comprendre son intérêt potentiel pour le résumé des tweets.

3.1 Une brève histoire du résumé automatique

Les travaux de recherche sur le résumé automatique ont commencé il y a maintenant près de 50 ans avec les études menées par **[Luhn, 1958]** qui expérimente des méthodes pour générer automatiquement des extraits d'articles techniques. **[Luhn, 1958]** s'est intéressé à la réduction de la quantité de travail manuel et de l'expertise impliqués dans cette tâche ainsi que l'amélioration de leur "cohérence et objectivité ". Suite aux travaux de **[Luhn, 1958]**, **[Edmundson, 1969]** a développé des techniques supplémentaires pour résumer un corpus de

[2] http://fr.wikipedia.org/wiki/Twitter#Trending_Topic

documents plus diversifié dans le but d'aider les utilisateurs ou évaluer des documents pour de plus amples lectures. Les premières recherches dans le résumé d'un texte ont axé principalement sur des techniques statistiques simples qui comptaient sur les caractéristiques lexicales comme la fréquence des mots **[Luhn, 1958]** ou des indices de formatage tels que les titres et les rubriques **[Edmundson, 1969]**. Les travaux plus tard ont intégré des approches plus sophistiquées, telles que l'apprentissage machine **[Kupiec et al, 1995]**, le traitement du langage naturel **[Barzilay et Elhadad, 1997]**, et les approches hybrides **[Neto et al, 2002]**. Dans la plupart des cas, le résumé du texte est effectué afin de gagner du temps aux utilisateurs en réduisant la quantité de contenu devant être lu **[Luhn, 1958]**, **[Edmundson, 1969]**. Toutefois, le résumé de texte a également été réalisé à d'autres fins telles que la réduction du nombre des fonctions requises pour la classification **[Kolcz et al. 2001]** ou le regroupement (clustering) des documents **[Ganti et al. 1999]**. Cependant, bien que l'intérêt pour le résumé automatique de texte a commencé en 1958, il n'est pas devenu un domaine de recherche actif jusqu'aux années 1990 avec l'introduction du World Wide Web **[Lin, 2009]**.

Avec la croissance de l'Internet, l'intérêt d'améliorer le résumé a grandi pour résumer les nouvelles formes de documents tels que des pages web **[Mahesh, 1997]** et les blogs **[Zhou et Hovy, 2006]**, **[Hu et al, 2007]**. Plus récemment , l'intérêt s'est déplacé du résumé d'un unique document à plusieurs documents en partie grâce à des conférences annuelles telles que la Conférence d'analyse de texte **TAC**[3]qui visent à faire avancer l'état de l' art dans le résumé en fournissant de grandes collections d'essais et d'évaluation commune des systèmes résumant.

3.2 Les différentes approches dans le résumé automatique de texte

En général les systèmes automatiques de résumé de texte peuvent être divisés en deux catégories : extraction ou abstraction.

3.2.1 Les systèmes d'extraction

Ils sont conçus pour résumer un document (ou un ensemble de documents) en sélectionnant et en concaténant les phrases les plus pertinentes du document(s). La sélection de la phrase est habituellement effectuée en pesant les phrases d'une certaine manière, puis en choisissant l'ensemble des phrases avec plus de poids. Les systèmes de pondération de la phrase dans la littérature peuvent être classés en trois types : fonctionnalité basée sur les

[3] http://www.nist.gov/tac/

caractéristiques **[Luhn, 1958]** et **[Edmundson, 1969]**, basée sur les chaînes lexicales **[Barzilay et Elhadad, 1997]** et basée sur les graphes **[Mihalcea et Tarau, 2004]**.

Dans la pondération en fonction des caractéristiques, les phrases sont notées par rapport à un ensemble de fonctionnalités telles que la fréquence des mots, la position de la phrase dans l'ensemble du document (par exemple, dans le premier ou le dernier paragraphe), ou l'inclusion de certains des mots clés (par exemple " en conclusion ").

Dans la pondération en fonction des chaînes lexicales, les phrases sont évaluées sur la base de la force de la chaîne lexicale dans laquelle ils appartiennent, où une chaîne lexicale est une séquence de mots qui sont liés comme par synonymie ou hyponymie.

Enfin, dans les approches fondées sur les graphiques, un graphe est réalisé et représente les différentes unités de texte comme sommets et les relations sémantiques entre ces sommets comme arête. Après que le graphe est construit, un algorithme de rang graphique est appliqué pour déterminer les scores de chaque sommet. Enfin, les sommets sont triés par leurs scores et les unités de textes associés aux sommets de haute notation sont utilisées comme un résumé résultant.

3.2.2 Les systèmes d'abstraction

Ils prennent une approche totalement différente à l'égard d'un résumé. Au lieu de sélectionner des phrases marquantes pour une utilisation comme résumé, une approche d'abstraction génère son propre texte de synthèse grâce à une analyse linguistique détaillée et la transformation du texte source. Ces types de systèmes tentent de représenter soit la structure ou des concepts d'un texte source à l'aide des arbres d'analyse ou des bases de connaissances de texte, respectivement. De ces représentations internes, un système d'abstraction serait tenté de condenser ces représentations et les transformer en un résumé court en utilisant la génération de langage naturel ou des modèles pré peuplés **[Hahn et Mani, 2000]**.

Comme les systèmes d'abstraction nécessitent souvent une connaissance spécifique au domaine, leur application est plus restrictive que les systèmes d'extraction. Cette limitation en plus de leur complexité accrue s'est traduite par quelques systèmes de résumé basé abstraction dans la littérature dans la dernière décennie **[Jones, 2007]**.

3.3 Problème de dimensions

Avant d'examiner les approches les plus récentes du résumé automatique de texte, il est important de comprendre d'abord les différentes dimensions du problème. Jones divise ces

dimensions en trois grandes catégories ou «facteurs» : d'entrée (input), d'usage (purpose) et de sortie (output) **[Jones, 2007]**. Ce qui suit est une liste de la plupart des facteurs qu'elle décrit.

• *Facteurs d'entrées* (Input factors) décrivent la source de l'information à résumer et les dimensions de couverture telles que la langue, le genre, l'enregistrement (c'est à dire le style linguistique), la longueur, la structure (par exemple, l'emplacement, des rubriques, des citations, des hyperliens, ... etc.) et le nombre de sources (document unique contre plusieurs documents).

• *Facteurs d'usages* (Purpose factors) décrivent l'usage prévu des éléments de synthèse et de couverture tels que son audience (par exemple, générale ou spécifique) et l'utilisation envisagée. Il peut y avoir de nombreux usages différents pour un résumé qui **[Hahn et Mani, 2000]** classent en trois utilisations principales : indicatif, informatif, ou critique. Les résumés indicatifs sont destinés à aider les lecteurs à décider si oui ou non un document est pertinent pour une étude plus approfondie (par exemple table des matières). Les résumés informatifs sont destinés à être les remplacements entiers d'un document tel qu'un lecteur serait en mesure de lire le résumé au lieu du document. Ces types de résumés ne peuvent contenir que les faits pertinents contenus dans le document qui se rapportent à une requête spécifique. Enfin, les résumés d'évaluation sont destinés à fournir une sorte de commentaire ou opinion sur le document en plus de résumer le contenu (par exemple, une critique de livre).

• *Facteurs de sortie* (Output factors) décrivent la sortie de synthèse générée et les dimensions de couverture telles que le style, la réduction / couverture (c'est à dire la longueur du résumé par rapport à la source), la cohérence (c'est à dire de la correction grammaticale), et la dérivation. La dérivation est un concept important qui vise à déterminer si le résumé est constitué d'extraits littéraux du document(s) source d'origine ou paraphrases. Ces types de dérivation sont généralement connus que par des résumés soit d'extraction ou d'abstraction, respectivement, et ont une grande influence sur la conception globale du système de récapitulation **[Lin, 2009].**

3.4 Défis décrits dans le résumé automatique des tweets

Alors que **[Jones, 2007]** a décrit plusieurs facteurs à considérer lors de la conception d'un système de résumé automatique, il y'a aussi de nombreuses difficultés associées à ces facteurs mentionnés dans la littérature. L'une de ces difficultés de conséquence particulière est le facteur de rendement de cohérence. La cohérence est une mesure de la maîtrise d'un résumé en termes de celui-ci obéissent aux règles de la grammaire, de la logique, et le discours. La cohérence

peut être difficile à atteindre pour les deux systèmes d'extraction et d'abstraction, mais encore plus pour les systèmes d'extraction.

Pour les systèmes d'extraction, la cohérence peut être perdue si les concepts clés sont répartis entre plusieurs phrases, mais seulement un sous-ensemble de ces phrases est choisi pour l'extraction **[Lin, 2009]**. **[Lin, 2009]** mentionne également des problèmes similaires avec des références anaphoriques et temporelles. L'anaphore est l'utilisation d'un pronom ou un terme équivalent au lieu de répéter un mot utilisé plus tôt. Si un système d'extraction choisit une phrase qui contient une anaphore sans choisir également la phrase précédente qui définit l'anaphore, la référence anaphorique sera perdue. Il peut en résulter soit un résumé incohérent où le terme n'est jamais défini, et/ou une situation pire où le terme défini dans un nouveau contexte est incorrect.

Pour les systèmes d'abstraction, les principaux défis à la cohérence se rapportent à la capacité du système à générer une sortie courante sans compter sur la structure de la phrase des documents sources. Pour ces types de systèmes, ils doivent intégrer les connaissances en dehors des règles du langage naturel afin de traduire leur représentation interne du résumé dans un résumé fluide et cohérent **[Hahn et Mani, 2000]**.

D'autres défis mentionnés dans la littérature sont centrés autour de plusieurs facteurs de production décrits par **[Jones, 2007]**. **[Barzilay et Elhadad, 1997]** ont averti que les systèmes qui dépendent trop de la disponibilité de certains mots clés ou éléments structuraux tels que l'emplacement peuvent avoir différents niveaux de performance sur la base du genre de documents étant résumés. Par exemple, Méthode de localisation (location Method) **[Edmundson, 1969]** repose sur l'apparition de titres prédéfinis des mots tels que «Introduction», «but», et «Conclusions» qui peuvent ou ne peuvent pas être présent dans tous les types de documents.

D'autres difficultés sont mentionnées par **[Lin, 2009]** qui décrit certains problèmes de résumer des documents sources à la fois uniques et multiples. Pour les documents simples, **[Lin, 2009]** mentionne qu'il peut être difficile d'améliorer les techniques simples telles que les méthodes qui utilisent des extraits, des résumés analytiques ou des endroits particuliers des paragraphes d'un résumé. **[Lin et Hovy, 1997]** ont établi empiriquement que de nombreux genres de documents disposent de l'information la plus pertinente dans des endroits prévisibles. Les articles de presse, par exemple, sont écrits de telle sorte que l'information la plus importante est généralement placée au début **[Zhou et Hovy, 2006]**. Pour de multiples sources de documents, à la fois **[Hahn et Mani, 2000]** et **[Lin, 2009]** décrivent les problèmes des questions de la contradiction et de la redondance. Dans ces cas, les systèmes résumant doivent développer

des méthodes permettant d'identifier les similitudes et les différences entre les documents de base et les résoudre dans le résumé de sortie **[Lin, 2009]**.

4. Travaux réalisés dans le domaine du résumé automatique des tweets

Dans ce qui suit, nous allons décrire des diverses méthodes et approches qui sont utiles dans le sujet de résumé automatique des tweets avec leurs propres avantages et limites sur l'autre.

4.1 Les travaux de [Sharifi et al., 2010]

[Sharifi et al., 2010] ont développé une méthode qui permet de résumer en temps réel les sujets tendances des tweets en des résumés courts. Pour cela, ils ont développé un algorithme appelé (PR : Phase Reinforcement) qui prend une phrase tendance ou toute phrase spécifiée par un utilisateur, qui rassemble un grand nombre de tweets contenant la phrase, et fournit un résumé automatiquement créé.

L'algorithme de PR commence par une phrase de départ, qui est le sujet pour lequel on souhaite générer un résumé. Ce sont généralement des tendances, mais peuvent être d'autres sujets. Ainsi, suivant l'ensemble des tweets renvoyés, l'algorithme filtre les messages à supprimer comme les spams ou les messages non pertinents. Le filtrage est une étape importante, car les spams et les autres messages non pertinents peuvent tromper l'algorithme de PR en résumant le spam au lieu du contenu souhaité. Le spam est filtré en utilisant un classificateur Naïve Bayes **[Kalita, 2002]** qu'ils ont formé en utilisant le contenu de spam précédemment recueillies auprès **Twitter** et également supprimer tous les tweets non-anglais, ainsi que les messages en double, car ils vont créer seulement des résumés en anglais.

L'idée centrale de l'algorithme de PR est de construire un graphe acyclique ordonné de tous les mots de l'ensemble des tweets. Le graphe est organisé autour d'un nœud racine central, qui contient la phrase de départ de la synthèse. Les mots adjacents au nœud de départ sont des mots qui se produisent immédiatement avant ou après la phrase de départ à l'intérieur de chaque tweet. Ces mots adjacents sont également placés soit avant, soit après le nœud de départ respectant l'ordre trouvé dans les tweets de la phrase d'apprentissage.

Ensuite, un poids est calculé aux nœuds selon leur fréquence d'occurrences respective de leur ordre des mots de la racine. Par conséquent, si un mot apparait M fois après la phrase de départ dans les tweets, son poids sera proportionnel à M.

Une fois le graphe construit, l'algorithme de PR commence à rechercher le meilleur résumé partiel en additionnant le poids de chaque chemin unique en partant du nœud racine à chaque nœud feuille. Le chemin du poids le plus élevé est considéré comme le meilleur chemin de résumé partiel à partir du nœud racine. Pour le chemin du poids le plus élevé l'algorithme crée un nouveau graphe avec une nouvelle phrase racine qui contient tous les mots dans ce chemin. L'ensemble de tweets est alors filtré pour garder que les tweets contenants la nouvelle phrase racine. Ce processus est répété jusqu'à trouver le résumé final qui est les mots du chemin avec le poids le plus élevé.

Discussion : Les résumés automatiques générés sont comparés contre des résumés produits par des êtres humains pour cinquante sujets. L'algorithme de PR peut s'effectuer mieux quand un sujet a un motif de phrase dominante autour du thème central. Chaque fois qu'un sujet fait naturellement partie d'une phrase plus grande, l'algorithme de PR fonctionne bien et il est capable d'isoler ces phrases dominantes de l'ensemble des tweets d'entrée. Cela est particulièrement vrai pour les sujets #hashtag qui sont une convention que les utilisateurs de **Twitter** ont adoptée afin de rendre certains sujets faciles à trouver via la recherche pour le hashtag. Si le hashtag ne tombe pas naturellement dans une phrase, alors l'algorithme de PR n'est pas en mesure de générer une expression dominante autour du thème.

4.2 Les travaux de [Inouye et Kalita, 2011]

Après les résultats non concluants de l'approche Phrase de renforcement de **[Sharifi et al., 2010]**, **[Inouye et Kalita, 2011]** ont présenté une autre approche basée sur une technique datant de début des travaux de synthèse **[Luhn , 1958]**.C'est l'approche TF-IDF (Terme Fréquence Inverse Document Frequency) Hybride qui permet d'extraire un ou plusieurs résumés pour chaque sujet.

TF -IDF (Terme Fréquence Inverse Document Frequency) [Jones, 1972] est une technique de pondération statistique qui a été appliquée à de nombreux problèmes de recherche d'information. L'idée est d'attribuer à chaque phrase dans un document un poids qui reflète l'importance de la phrase dans le document. Les phrases sont classées par leur poids à partir de laquelle les **m** meilleurs phrases avec le plus de poids sont choisies comme résumé. Le poids d'une phrase est la somme des poids des termes individuels dans la phrase. Les termes peuvent être des mots, des phrases, ou tout autre type. Pour déterminer le poids d'un terme, on utilise la formule :

$$TF\_IDF = \mathrm{tf}_{ij} * \log_2 \frac{N}{\mathrm{df}\,j} \qquad \textbf{(2.1)}$$

Où
- tf_{ij} est la fréquence du terme t_j dans le document d_i.
- N est le nombre total de documents.
- df_j est le nombre de documents au sein de l'ensemble qui contient le terme t_j.

La valeur TF -IDF est composée de deux parties principales. La composante de fréquence de terme (TF) attribue plus de poids aux mots qui se produisent fréquemment dans un document parce que les mots importants sont souvent répétés **[Sharifi et al., 2010]**. La composante de fréquence inverse de document (IDF) compense le fait que certains termes comme mots vides communs sont fréquents. Étant donné que ces mots ne permettent pas de distinguer entre une phrase ou un document sur un autre, ces mots sont proportionnellement pénalisés de leur document de fréquence inverse. Par conséquent, TF -IDF donne le plus de poids aux mots qui apparaissent le plus fréquemment dans un petit nombre de documents et le moins de poids à des conditions qui se produisent rarement ou se produisent dans la majorité des documents.

L'équation **(2.1)** définit le poids d'un terme dans le contexte d'un document. Cependant, ici il ne s'agit pas d'un document traditionnel. Il s'agit d'un ensemble de tweets qui sont chacun lié à un sujet. Le problème est sur la façon de laquelle le document va être défini. Il y a deux façons :

1- définir un document unique qui englobe tous les messages ensemble. Dans ce cas, la définition de la TF composant est simple puisque il s'agit de calculer les fréquences des termes dans tous les messages. Toutefois, cela amène à perdre le composant IDF puisque c'est un seul document.

2- définir chaque tweet comme un document faisant une définition clair de la composante IDF. Mais, la composante de TF aura un problème : Parce que chaque message contient seulement une poignée de mots, donc la valeur des fréquences des termes sera très petite.

Pour gérer cette situation **[Inouye et Kalita, 2011]** ont redéfini TF -IDF en termes d'un document hybride. D'abord un document est défini en une seule phrase et lors du calcul des fréquences du terme, le document est supposé être une collection complète des messages.

Ainsi, une méthode de normalisation est choisie pour que l'algorithme TF -IDF pousse toujours vers des phrases plus longues. Le poids d'une phrase est normalisé en la divisant par un facteur de normalisation. Un poids de zéro est donné aux mots d'arrêt en les comparants à une liste prédéfini.

Pour résumer l'ensemble de messages Twitter, les tweets sont regroupées en k clusters basés sur une mesure de similarité. Chaque groupe est résumé en choisissant le tweet le plus

pondéré déterminé par l'hybride TF-IDF pondération. Donc l'outil de synthèse de cluster tente de créer k sous-thèmes en regroupant les messages. Après, pour chaque groupe d'un sous-thème, l'algorithme TF-IDF hybride sélectionne le tweet le plus pondéré pour chaque sous-thème.

Discussion : L'algorithme TF -IDF hybride définit un document différemment des formules utilisées dans le calcul du poids total d'un tweet. Donc ici il semble que la fréquence des mots simples et la réduction de la redondance sont les meilleures techniques pour résumer les tweets. Ceci est probablement dû aux caractéristiques non structurées et la taille des tweets qui ne sont pas comme des documents traditionnels.

4.3 Les travaux de [Harabagiu et Hickl, 2011]

Alors que les méthodes de **[Sharifi et al 2010]** et **[Inouye et Kalita, 2011]** pourrait être utilisée pour synthétiser le contenu des tweets dans un résumé prose d'une longueur fixe, **[Harabagiu et Hickl, 2011]** se concentrent sur la synthèse des tweets liées à des événements complexes du monde réel. Cette approche se capitalise sur une combinaison de deux types de modèles :(1) un modèle génératif qui induit des structures d'événements complexes et (2) un modèle de comportement de l'utilisateur qui saisit comment les utilisateurs ont communiqué un contenu pertinent.

Pour le premier modèle qui concerne la structure d'événement. Elle est définie comme un graphe S = (ET, RT), constitué de l'ensemble des E événements mentionnés et les relations de l'événement R qui sont pertinents à un sujet T. Un événement complexe est supposé se réfère à un ensemble cohérent de sous-événements qui se produisent sur une période de temps donnée et dans un endroit particulier. Un événement mentionné se compose d'un prédicat et / ou un prédicat nominale qui fait référence à un (ou plusieurs) événement (s) dans le monde réel, alors une relation d'événement se compose d'une propriété sémantique qui peut être attribué à deux ou plusieurs événements mentionnés.

Pour le deuxième modèle concernant le comportement des utilisateurs pour les tweets individuels peut être utilisé pour identifier un contenu pertinent pour un résumé. **[Harabagiu et Hickl, 2011]** mettent l'hypothèse que lorsque les utilisateurs interagissent avec les tweets, ils fournissent des évaluations de pertinence implicites qui peuvent être utilisées dans un résumé. Les actions d'un utilisateur peuvent être représentées comme des tweets individuels qui relient des chaînes de tweet (t1, t2). Ils mettent l'accent sur trois types de chaînes : (1) les chaînes de

retweet (où t2 est le retweet de t1) , (2) les chaînes de réponses (où t2 est une réponse à l'expéditeur du t1) et (3) les chaînes de citation (où t2 cite le texte de t1) .

Discussion : Ici **[Harabagiu et Hickl, 2011]** se sont concentrés sur la synthèse des tweets liées à des événements complexes du monde réel. Alors que les travaux précédents ne reposent que sur la surface des indices lexicaux et de la redondance. **[Harabagiu et Hickl, 2011]** est le premier travail qui a exploré plus d'avantage les techniques d'extraction pour générer les résumés.

4.4 Les travaux de [Liu et al., 2011]

[Liu et al., 2011] ont proposé d'explorer une variété de sources de texte pour résumer les thèmes émergeants de Twitter en utilisant des tweets qui sont normalisés via un système dédié ''tweet normalisation'' et les pages Web liées à partir des tweets pour l'intégration de différentes sources de texte.

Cette approche consiste d'abord à l'extraction d'un ensemble de concepts importants pour chaque sujet, puis sélectionne une collection de phrases qui peuvent couvrir autant de concepts aussi importants que possible, mais dans la limite de la longueur spécifiée.

Les concepts sont sélectionnés par extraction de n-grammes (n = 1, 2, 3) à partir des documents d'entrée correspondant à chaque sujet. Ils suppriment :

1- n-grammes qui n'apparaissent qu'une seule fois dans les documents.
2- n-grammes qui ont un mot composé avec le document inverse fréquence (IDF) de valeur inférieure à un seuil.
3- n-grammes qui sont enfermés par ordre supérieur des n- grammes avec la même fréquence. Ces filtres sont conçus pour exclure les n-grammes significatifs du concept de l'ensemble.

Pour chaque sujet Twitter, Ils recueillent un ensemble de pages Web liées par le sujet des tweets pour les utiliser comme une autre source d'entrée pour la compression. Pour chaque thème, ils sélectionnent n (n = 10) URL qui apparaissent le plus fréquemment dans les sujets tweets et rarement dans les différents sujets de Twitter.

Donc le système de compression bénéficierait des deux (tweets et les contenus Web liés à eu), L'utilisation de différentes sources textuelles peut aider à ressortir le poids des principaux concepts et ainsi éliminer les informations de spam. Dans cette approche ils utilisent la concaténation soit des tweets originaux ou les tweets normalisés avec les pages Web liées

comme entrée pour le système de synthèse à base de concept. Il en résulte deux entrées " Web + OrigTweets'' et ''Web + NormTweets''.

Discussion : Ici **[Liu et al., 2011]** ont proposés d'explorer une variété de sources de texte pour résumer les thèmes Twitter. Ils ont utilisé le cadre de l'optimisation basée sur le concept avec de multiples sources de saisie de texte pour générer les résumés. Une meilleure performance est observée lors de l'utilisation des tweets normalisés en entrée, et le contenu Web liés peuvent fournir des informations supplémentaires de la rubrique utile.

4.5 Les travaux de [Chakrabarti et Punera, 2011]

[Chakrabarti et Punera, 2011] ont proposé une méthode pour résumer les événements hautement structurés et récurrents tels que les matches de football. Ils ont supposé que les nouveaux événements avaient déjà été détectés par d'autres méthodes.

Cette méthode proposée a essayé d'extraire quelques tweets qui décrivent le mieux les sous événements intéressants dans l'événement. Ils ont formé un modèle HMM (Hidden Markov Model) pour identifier des cas de sous- événements basés sur les tweets dans un segment de temps. Pour récupérer les tweets qui sont à proximité de d'autres tweets dans le corpus ils ont utilisé un " tf -idf avec cosinus similitude " basé sur un modèle appelée " modèle caché de markov " pour détecter la chaîne des événements par l'apprentissage qui sous-entend la représentation d'état caché des événements répétés.

Discussion : Ici **[Chakrabarti et Punera, 2011]** ont abordé le problème de la construction des résumés en temps réel des événements de tweets dans Twitter .Ils ont proposé une approche basée sur l'apprentissage d'une représentation sous-jacente de l'état caché d'un événement. Cependant, la disponibilité des événements récurrents très structurés sont rares dans la réalité et donc leur approche ne serait pas capable de gérer grande majorité des événements du monde réel.

4.6 Les travaux de [Wei et al., 2012]

[Wei et al., 2012] ont proposé un cadre basé sur le temps réel pour résumer un sujet sur Twitter. Les sujets sont résumés par sous-thèmes en temps réel afin de saisir pleinement l'évolution du sujet rapide sur Twitter. Ils ont classés et sélectionnés les tweets pertinents et diversifiés comme un résumé de chaque sous-thème.

Cette approche consiste à modéliser et formuler le classement des tweets dans un modèle graphique de renforcement mutuel unifié, où l'influence sociale des utilisateurs et la qualité du

contenu des tweets sont prises en considération. Elle se décompose en trois grandes étapes. Tout d'abord, Ils ont effectués une segmentation thématique qui segmente le flux de tweets sur le sujet en sous-groupes thématiques en termes de temps d'affichage, dans lequel chaque groupe décrit un sous-thème. Deuxièmement, les tweets dans chaque groupe sous-thème sont classés selon la pertinence du tweet par renforcement du modèle de classement en profitant de la qualité du contenu des tweets et l'influence sociale des auteurs. Troisièmement, Ils ont générés le résumé pour chaque sous-thème sur les résultats du classement des tweets en enlevant les tweets redondants au niveau de toute la question.

Discussion : Ici **[Wei et al., 2012]** ont proposés de résumer les flux des tweets à l'égard de long sujets en temps réel pour produire un aperçu de l'évolution du sujet, qui est exprimé par sous-thèmes dans l'ordre chronologique . Pour chaque sous-thème, un ensemble de tweets pertinents est choisi pour produire le résumé en les classant selon leur pertinence. Différent de documents traditionnels, les tweets souffrent de désinformation et de style d'écriture irrégulier. Ils ont donc modélisé la pertinence d'un tweet en utilisant un graphique de renforcement mutuel unifié afin d'intégrer l'influence sociale des utilisateurs et la qualité du contenu des tweets.

4.7 Les travaux de [Ritter et al., 2013]

[Ritter et al., 2013] ont proposé un cadre basé sur l'événement graphique à l'aide des techniques d'extraction d'information qui est capable de créer des résumés de longueur variable pour différents sujets. En particulier, Ils étendent l'algorithme de classement Pagerank **[Brin et Page, 1998]** pour partitionner les graphes d'événements et ainsi détecter les aspects important de l'événement pour être résumés. Cette approche est basée sur 3 étapes :

1- La première étape consiste à extraire les entités nommées, les phrases d'événements grâce au travaux de **[Ritter et al ., 2012]**. Ils regroupent tous les tweets qui parler de la même entité nommée au niveau du même cluster.

2- La deuxième étape consiste à supposer que les entités nommées et les événements de phrases qui se coproduisent au niveau d'un même tweet sont très probablement liés. Étant donnée une collection de tweets, ils représentent ces connexions par un graphe non orienté pondéré.

Où
- Les nœuds : les entités nommées et les phrases d'événements.
- Les arêtes : deux nœuds sont reliés par une arête non orienté s'ils coproduits dans les k tweets, et le poids de l'arête est k.

3- La troisième étape consiste à appliquer un des algorithmes de classement à base de graphe qui sont largement utilisés dans le résumé automatique pour décider de la

pertinence des concepts ou les phrases basées sur l'information mondiale récursive tirée de l' ensemble du graphique . Ils ont adapté pour cela l'algorithme de PageRank qui tient compte des poids des arêtes lors du calcul du score associé à un sommet dans le graphe.

Discussion : Ici **[Ritter et al., 2013]** ont proposé une étude pour générer des résumés compacts de longueurs variables pour le résumé des tweets, par l'extension d'un algorithme de classement. L'évaluation a montré que les techniques d'extraction d'information sont utiles pour générer des résumés de presse digne d'une bonne lisibilité.

4.8 Les travaux de [Khan et al., 2013]

[Khan et al., 2013] ont proposé une méthode pour résumer un sujet sur Twitter basé sur un algorithme de graphe comme pour **[Ritter et al.,2013]**. Sauf que dans cette approche ils ont réalisé d'abord le regroupement thématique (topic modeling) des tweets pertinents, puis d'appliquer l'algorithme proposé sur chaque groupe indépendamment. Cette approche est basée sur 2 étapes :

La première étape consiste à deviser les tweets en différents groupes thématiques grâce au modèle LDA (Latent Dirichlet Allocation) **[Blei et al., 2003]** qui est un modèle génératif probabiliste permettant d'expliquer des ensembles d'observations, par le moyen de groupes non observés, eux-mêmes définis par des similarités de données.

La deuxième étape consiste à identifier les tweets représentatifs de chaque cluster, donc ils construisent d'abord un graphe lexical, puis ils appliqueent une variante de l'algorithme PageRank **[Brin et Page, 1998]** pour déterminer le score des unités lexicales dans le graphe. Le PageRank est basé sur la notion intuitive d'approbation. En PageRank, une page peut avoir un classement élevé si de nombreuses pages pointent vers elle ou une autre page de haut rang pointe vers elle. De même, dans une collection de tweets, si un mot coproduit avec beaucoup de mots différents, Il peut être considéré comme un mot relié aux événements importants qui a été utilisé par de nombreux utilisateurs pour signaler des sous événements. De plus, les mots du graphe lexical s'appuient les uns et les autres par rapport à leur force d'association.

Discussion : Ici **[Khan et al., 2013]** ont proposé une méthode pour résumer en temps réel les événements importants en utilisant un nombre limité des tweets pertinents. La méthode tente d'intégrer les tweets d'événements pertinents les plus informatifs qui couvrent les divers aspects d'actualité de l'événement, tout en minimisant la répétition d'informations. L'évaluation

réalisée sur des données du monde réel montre que la méthode peut résumer les événements en temps réel avec une grande précision et de rappel.

5. Comparaison

Dans cette partie nous allons comparer les travaux du résumé automatique étudiés précédemment en utilisant un tableau comparatif et en se basant sur les critères de comparaison décrits ci-dessous :

1- **Catégorie :** Compte tenu de méthodes proposées dans la littérature et les défis associés, la première décision dans l'élaboration d'un algorithme de compression des tweets est de choisir d'utiliser une approche d'abstraction ou d'extraction. Bien que ces deux approches aient leurs forces et leurs faiblesses, une approche d'extraction est généralement choisie, car ces méthodes sont plus étroitement liées à la structure et la diversité des tweets. Par exemple, les approches d'abstraction sont plus bénéfiques dans les situations où des taux élevés de compression sont nécessaires comme pour la synthèse de documents multiples et longs.

   Toutefois, les tweets sont l'antithèse des documents longs. Comme ils sont si courts, ils sont déjà fortement condensés menant à un plus grand potentiel de trouver un extrait pour servir de résumé. En outre, les systèmes d'abstraction généralement s'effectuent mieux dans des domaines limités, car ils ont besoin de sources de connaissances externes telles que les grammaires, les analyseurs et les ontologies. Ces approches peuvent ne pas fonctionner aussi bien avec les tweets, car ils ne sont pas structurés et diversifiés en matière. Les techniques d'extraction, d'une autre part, sont connues pour une meilleure échelle avec plus de diversité de domaines **[Hahn et Mani, 2000]**.

2- **Facteurs d'entrées et sorties :** Comme décrit dans la section **3.3** les facteurs d'entrées décrivent la source de l'information à résumer et Les facteurs de sorties décrivent la sortie de synthèse générée. Les dimensions de couverture qu'on a choisis sont ''longueur'' et ''genre'' pour les facteurs d'entrées, ''longueur'', ''mesure'' et ''évaluation'' pour les facteurs de sorties.

3- **Temps réel ou non.**

Tableau 3.1 : Comparaison des travaux du résumé automatique des tweets.

Structure et contenu	Approche	Catégorie d'approche		Facteurs d'entrées		En Temps réel	Facteurs de sorties		
		Type	Système de pondération	Longueur	Genre		Longueur	Mesure	Évaluation (couverture / cohérence)
	sharifi et al. 2010	Extraction	Graphiques(le calcul des scores de base sur la fréquence des mots)	1500 tweets	Sujets tendances	non	1 tweet	ROGUE et humaine	Qualité globale comparant au résumé humain
	Inouye et Kalita 2011	Extraction	Caractéristiques (fréquence des mots)	1500 tweets	Sujets tendances	non	4 tweets	ROGUE et humaine	Qualité globale comparant au résumé humain
	Harabagiu et Hickl 2011	Extraction	Caractéristiques (concept, propriété de chaine)	2500 tweets	Sujets d'événements réels	non	250 mots	humaine	Couverture et la cohérence
	Liu et al. 2011	Extraction	Caractéristiques (concepts, fréquence des mots)	1,7 k tweets	Des thèmes généraux et des thèmes de hashtag	non	2 - 3 tweets	ROGUE et humaine	Couverture du contenu grammaticalité, pas de redondance, la clarté référentielle, mise au point
	Chakrabarti et Punera 2011	Extraction	Caractéristiques (fréquence des mots)	1,8 k tweets	Jeux de football spécifique	non	10 - 70 tweets	Précision @ k (intérêt pour le sujet)	bons résumés pour les sports structurés et les événements récurrents
	Wei et al. 2012	Extraction	Graphiques (selon la saillance des tweets)	10 k tweets	Sujets hashtag segmentés	oui	10 tweets	ROGUE, de précision / rappel	une bonne lisibilité et contenu riche
	Ritter et al. 2013	Extraction	Graphiques (Graphe lexical, Version PageRank amélioré)	465 tweets par cluster	Des thèmes hashtag	non	1-4 tweets (chaque cluster)	Humaine	Couverture du contenu important
	Khan et al. 2013	Extraction	Graphiques (Graphe lexical, PageRank)	2,7 k tweets	Des thèmes de hashtag	oui	10 tweets	ROGUE, de précision / rappel	Couverture et précision sur l'aspect principale de l'évènement

6. Conclusion

Dans ce chapitre, nous avons étudié la question à aborder dans ce mémoire à savoir le résumé automatique des tweets. Nous nous sommes intéressés d'abord par le processus visant la production d'un résumé de texte en général, aux types et aux caractéristiques de ce dernier. Ensuite nous avons exploré plusieurs méthodes qui ont été proposées pour résoudre le problème de production de résumés automatiques des tweets. Il semble bien que les méthodes purement statistiques exploite les caractéristiques des tweets (Fréquence des mots, TFIDF,...) ou qui s'appuie sur les graphes s'avèrent être extrêmement bénéfique pour le sujet sensible du résumé automatique des tweets en raison des caractéristiques non structurées et courtes des tweets.

À partir de cette synthèse des méthodes proposées pour le résumé automatique des tweets, nous pouvons conclure que chaque approche a sa propre signification et importance pour la génération du résumé. Chaque approche utilise des différents algorithmes et une technique d'évaluation différente pour mesurer et comparer la précision du résumé produit. Donc, de tous les travaux qu'on a décrits, nous pouvons conclure que plusieurs défis se posent lorsque nous tentons d'effectuer le résumé des tweets :

- Les mots sont souvent mal orthographiés dans les tweets qui signifie que nous ne pouvons pas utiliser un dictionnaire ou base de connaissances.
- Beaucoup de tweets sont bruyants et sans rapport avec le sujet.
- La nature diverse, brève et bruyante des tweets entraîne une mauvaise performance pour le résumé des sujets dans Twitter.

Dans le chapitre suivant, nous allons présenter notre approche de résumé automatique des tweets. Cette approche s'inspire de certains travaux présentés dans ce chapitre.

CHAPITRE 3
RATR^{TWEETS} : RÉSUMÉ AUTOMATQIUE DES TWEETS EN TEMPS RÉEL

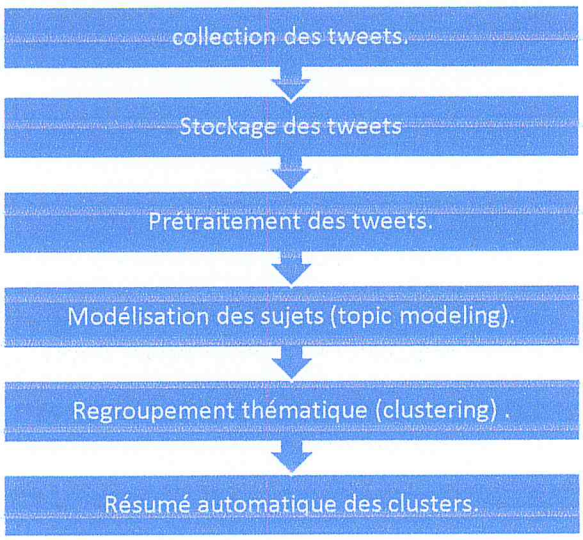
1. Introduction

CE chapitre présente une nouvelle approche de résumé automatique des *tweets*. Nous avons pris comme base les trois travaux de **[Ritter et al., 2013]**, **[Khan et al., 2013]** et **[Sharifi et al., 2010]** étudiés dans le chapitre précédant et qui se basent sur une approche graphique. Ici nous avons proposé une nouvelle approche basée aussi sur les graphes et qui s'appuie sur l'extraction d'informations contenues dans la structure des *tweets* pour résumer les événements en temps réel.

2. L'approche du résumé automatique des tweets RATR^{Tweets}

Notre approche **RATR^{Tweets}** consiste à extraire les informations essentielles incluses dans la collection des *tweets* afin de les traiter, effectuer un résumé et présenter en temps réel le contenu primaire dans cette collection. Tout d'abord, il faut découvrir les sujets(Topics), effectuer un regroupement des *tweets* suivant les topics découverts et enfin effectuer le résumé de chaque topic.

Donc globalement, l'approche est constituée de 6 grandes phases :



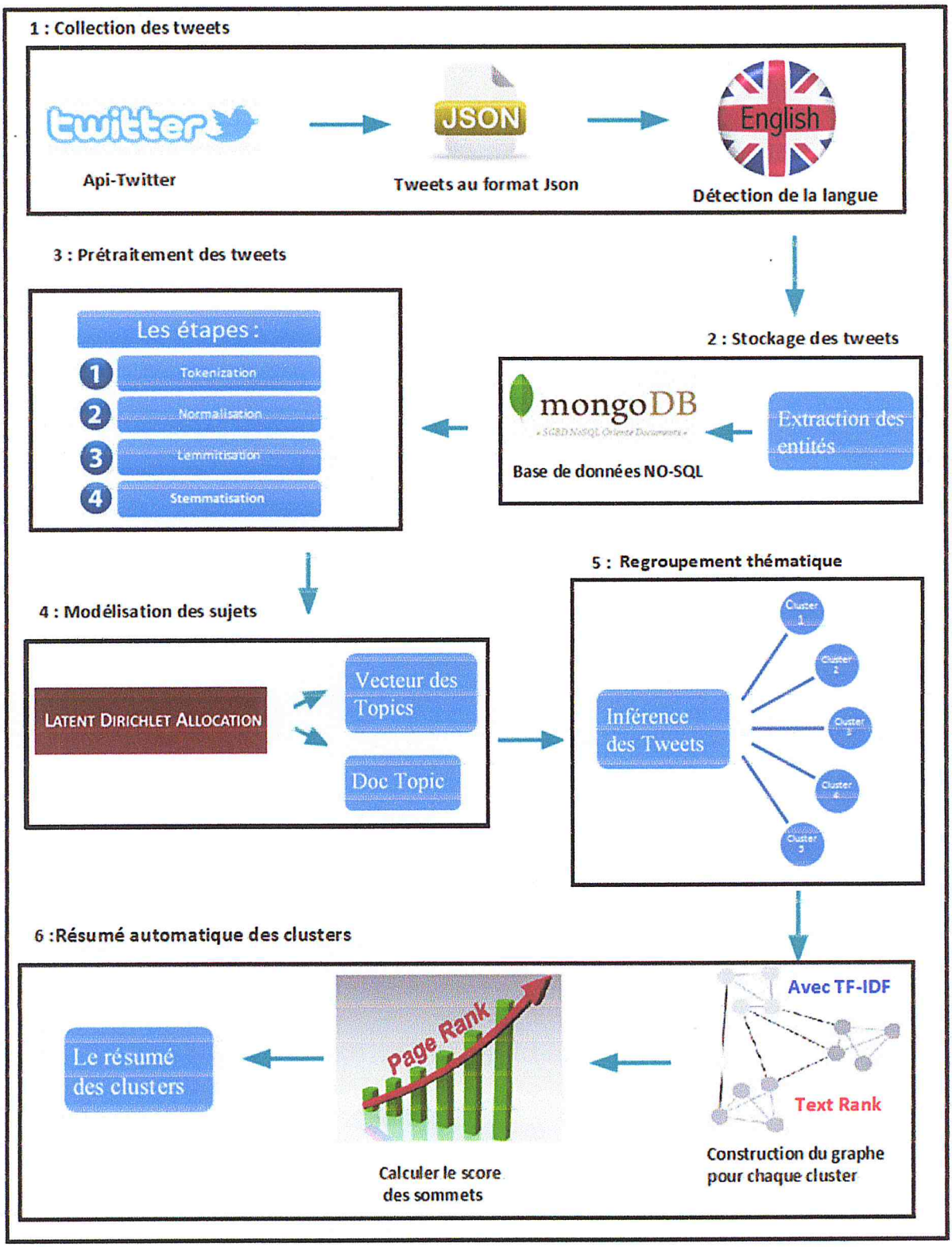


Figure 3.1 : Schéma global de l'approche RATR[tweets].

2.1 La collection des tweets

Cette phase consiste à collectionner l'ensemble des données à traiter dans notre approche.

2.1.1 L'accès à l'API

Comme décrit dans la section **2.3** du chapitre **2** concernant l'API *twitter*, il y a plusieurs méthodes pour collecter les *tweets*. Ici nous avons utilisé le ''flux public'' (Public streams) où les données retournées par la nouvelle API[1] 1.1 de T*witter* sont en format *Java Script Objet Notion* (JSON). **JSON**[2] est un format populaire qui est largement utilisé comme notion objet sur le web.

Les API T*witter* ne sont accessibles que via des requêtes authentifiées. *Twitter* utilise l'authentification ouverte Oauth (Open authentification) et chaque demande doit être signée avec des informations d'identification valable d'un utilisateur donné. L'accès à l'API *twitter* est également limité à un certain nombre de demandes dans un laps de temps réel appelé la limite de vitesse (*rate limit*). Une fenêtre de limite de vitesse est utilisée pour renouveler le quota d'appels de l'API autorisé périodiquement. La taille de cette fenêtre est actuellement 15 minutes. Pour chaque requête. Nous avons limité le nombre de *tweets* retournés à un nombre raisonnable de 200 *tweets*.

On commence par enregistrer l'application dans *twitter*[3], les applications sont connues en tant que consommateur. Après l'enregistrement, l'API fournit 4 codes : (consumer key) et (consumer secret) pour l'authentification, (access token) et (access secret) pour la vérification de l'authentification. Ce protocole fournit une alternative plus sure au niveau de la sécurité des mots de passe. Ces 4 codes par la suite sont utilisés via une bibliothèque spéciale pour l'authentification et la récupération des données. La recherche est effectuée par mot clé ou hashtag et à chaque requête on récupère les 200 *tweets* les plus populaires en temps réel contenant le mot de recherche.

2.1.2 Détection de la langue

Lors de la récupération des *tweets*, nous voulons les filtrer selon la langue anglaise, pour cela il faut effectuer la détection de la langue de chaque contenu textuel d'un *tweet*. L'outil NLTK (*Naturel Langage Toolkit*[4]) contient des outils pour l'analyse de texte. Nous avons utilisé

[1]http://dev.twitter.com/docs/api/1.1
[2]http://wikipedia.org/wiki/json
[3]http://dev.twitter.com
[4]http://www.nltk.org

une méthode qui s'appuie sur le compactage de mots vides. Cette méthode est constituée de 4 étapes :

1- Extraction du contenu textuel du *tweet* : consiste à récupérer le contenu de la balise « text ».
2- Découpage du texte : le texte est découpé en segments, dont le mot ici est privilégié comme unité textuelle de base. Le découpage en mots (*tokenization* en anglais) est réalisé par un découpage en tokens séparés par des espaces en blanc.
3- Calculer le nombre des mots vides : On compare le texte avec la liste des mots vides de chaque langue et on calcule leur nombre.
4- Détection de la langue : La langue avec le plus de mots vides est alors choisie comme langue d'origine pour le *tweet*.

Donc à la fin si le *tweet* est détecté comme un *tweet* non écrit en anglais, on l'élimine de la collection.

2.1.3 Extraction des entités

En plus du contenu textuel du *tweet*, nous extrayons 5 autres entités comme c'est décrit dans la Figure **3.2**.


```
"screen_name": "whuggins",
"text": "The #BigSurprise here is that BOTH rags have been 100% in #POTUS #ImpeachObama's court.
"created_at": "2014-05-29T11:32:26",
"hashtags": [
 "BigSurprise",
 "POTUS",
 "ImpeachObama"
],
"user_mentions": [],
"urls": [
 "http://t.co/9qSEzkqgzH"
],
"retweet_count": 0
}
```


Figure 3.2 : Représentation d'un *tweet* en format json.

screen_name : le nom de l'utilisateur dans *twitter*.

text : le contenu textuel du message.

created_at : la date du message en temps universel :

hashtags : les mots clés (#).

urls : les urls mentionnées.

user_mentions : les utilisateurs mentionnés.

2.2 Stockage des tweets

À chaque extraction d'un tweet dans sa forme json, il est directement enregistré dans la base de données, le nom de la base de données est choisi par l'utilisateur et le nom de la collection est choisi par défaut en tant que mot clé de la recherche.

Chaque collection contient plusieurs tweets contenant le même mot de recherche et chaque base de données contient plusieurs collections.

Avec la recrudescence du trafic sur Twitter, garder la trace de chaque requête exige un nouveau paradigme pour le stockage des données. Cela nous a fait pencher pour le NoSQL (Not only SQL) **[Redmond et Wilson, 2012]**, qui permet de stocker de grandes quantités de données d'une manière plus accessible que le modèle relationnelle traditionnel.

Il existe différents types de base de données NoSQL spécifiques à différents besoins (bases de données orientées Clés-Valeurs, orientées Colonnes, orientées Documents et orientées Graphes). Nous avons utilisé les bases de données orientées documents. Dans ces dernières, un document en format semi-structuré hiérarchique est stocké. Un document possède une structure arborescente, il peut contenir plusieurs valeurs et d'autres documents, qui peuvent à leur tour en contenir d'autres et ainsi de suite.

Il existe plusieurs implémentations pour les bases de données orientées documents. Nous avons choisi MongoDB[5] pour les raisons suivantes :

- Stockage orienté document : MongoDB stocke ses données au format JSON. Cela rend très facile de stocker des documents de base directement à partir de l'API-Twitter.
- Simplicité des requêtes : Les requêtes de MongoDB, tandis que syntaxiquement sont bien différentes du SQL, ils sont très similaires sémantiquement.

2.3 Le prétraitement

Le contenu textuel des *tweets* contient des mots mal orthographiés ou collés, des incohérences typographiques, des phrases agrammaticales, des caractères spéciaux ou d'un encodage différent ... etc. Donc, les *tweets* sont en quelque sorte bruités et ils doivent subir un processus de nettoyage et de normalisation approprié. Cette phase est appelée le prétraitement des tweets et elle est très essentielle. Si elle est négligée ou réalisée de façon simpliste, les

[5]http://www.mongodb.org/

systèmes risquent de fausser leurs résultats. En particulier les systèmes de résumé automatique sont très sensibles à la quantité du bruit présent dans le texte.

De façon classique, le prétraitement est constitué d'un découpage approprié du texte (en unités textuelles telles que les mots), d'une normalisation de mot, d'un filtrage adéquat de certains termes et de symboles de ponctuation.

2.3.1 Découpage du texte

C'est le même processus décrit dans l'étape précédente pour la détection de la langue.

2.3.2 La normalisation et le filtrage des termes

Afin de réduire la complexité de la représentation du texte, des processus de normalisation et de filtrage du lexique doivent être effectués. Ces processus ont pour bénéfice immédiat la diminution du nombre de mots à traiter. De façon générale, l'opération qui consiste à identifier une représentation canonique pour un ensemble de mots — semblables — est appelée la normalisation.

Le concept de normalisation est très général. Il englobe, dans notre cas le fait d'associer les variantes de casse d'un mot à une seule forme en minuscules et de substituer les dérivations par leur racine. On peut parler des traitements morphologiques de racinisation (lemmatisation et stemmatisation[6]) et de normalisation de formes graphiques.

Les mots les plus fréquents n'apportent pas, généralement une grande quantité d'informations. On les appelle mots ou termes vides de sens[7]. Dans un cas comme le nôtre où on s'intéresse à la recherche d'information, les termes vides comme les conjonctions, les chiffres, la ponctuation et les symboles spéciaux peuvent éventuellement être supprimés lors d'un filtrage. Les mots vides sont appelés en anglais stop-words. Nous avons utilisé une liste présente dans l'outil NLTK (*Naturel Langage Toolkit*).

Le filtrage élimine les termes ou mots vides de sens, mais aussi les symboles de ponctuation et les caractères étranges. Il y a aussi une suppression des url (http,https), ainsi que les références des utilisateurs (@).

---

[6]Stemming en anglais.

[7]« Les mots vides (ou stop-words, en anglais) sont des mots qui sont tellement communs qu'il est inutile de les indexer ou de les utiliser dans une recherche. Un mot vide est un mot non significatif figurant dans un texte. On l'oppose mot plein » (source Wikipédia français : http://fr.wikipedia.org/wiki/Mot_vide).

2.3.2.1 Lemmatisation

La lemmatisation consiste à trouver la racine des verbes fléchis et à ramener les mots pluriels et féminins à la forme masculine singulière. Pour cela nous avons utilisé le WordNet. Le WordNet **[Miller, 1995]** est un vaste thésaurus de données lexical en anglais. Noms, verbes, adjectifs et adverbes sont regroupés en un ensemble de synonymes cognitifs (des synsets) où chacun exprime un concept distinct. Les Synsets sont reliés entre eux par des relations sémantiques conceptuelles et lexicales.

2.3.2.2 Stemmatisation

La stemmatisation ou stemming est le processus d'élimination de suffixes des mots afin d'obtenir leur racine commune. Cela permet de générer la forme de base (souvent tronquée) appelée le stem (souche en français). Un des algorithmes les plus populaires de stemmatisation qu'on a utilisé est celui de PORTER. L'algorithme PORTER **[Porter, 1980]** se compose d'une cinquantaine de règles de dé-suffixation classées en sept phases successives (traitement des pluriels et verbes à la troisième personne du singulier, traitement du passé et du progressif,...). Les mots à analyser passent par tous les stades et, dans le cas où plusieurs règles pourraient leur être appliquées, c'est toujours celle comprenant le suffixe le plus long qui est choisie. La dé-suffixation est accompagnée, dans la même étape, de règles de recodage. Par exemple, "troubling" deviendra "troubl" par enlèvement du suffixe marqueur du progressif -ing et sera ensuite transformé en "trouble" par application de la règle "bl" devient "ble". Cet algorithme comprend aussi cinq règles de contexte, qui indiquent les conditions dans lesquelles un suffixe devra être supprimé. La terminaison en -ing, par exemple, ne sera enlevée que si le radical comporte au moins une voyelle. De cette manière, "troubling" deviendra "troubl", nous l'avons vu, alors que "sing" restera "sing".

- **Détail de l'algorithme de Porter :**

Soit v représente une voyelle (y est considéré comme une voyelle s'il est précédé par une consonne), c représente une consonne; et soit V représente une suite de voyelles, C représente une suite de consonnes.

Alors un mot en anglais peut être de l'une des 4 formes suivantes **[Lecroq]** :

- CVCV... C
- CVCV... V
- VCVC... C
- VCVC... V

Ce qui peut se représenter par :

- [C]V CV C … [V] , Où : [C](V C)^m[V]

Où m est appelée la mesure d'un mot.

m = 0: tree, by.

m = 1: trouble, oats, trees, ivy.

m = 2 : troubles, private, oaten, orrery.

Tableau 3.1 : Les déférentes étapes et Racines obtenus de l'algorithme Porter **[Lecroq]**.

Étape		**Règles**	**Exemples**
1	**a**	□ *SSES* → *SS* □ *IES* → *I* □ *SS* → *SS* □ *S* →	caresses → caress ponies → poni caress → caress cats → cat
	b	□ (m>0) *EED* → *EE* □ (*v*) *ED* → □ (*v*) *ING* →	feed → feed, agreed → agree plastered → plaster, bled → bled motoring → motor, sing → sing
	c	□ (*v*) *Y* → I	happy → happi, sky → sky
2		□ (m>0) *ATIONAL* → *ATE* □ (m>0) *TIONAL* → *TION* □ (m>0) *ENCI* → *ENCE* □ (m>0) *ANCI* → *ANCE* □ …	relational → relate conditional → condition, rational → rational valenci → valence hesitansi → hesitance …
3		□ (m>0) *ICATE* → *IC* □ (m>0) *ATIVE* → □ (m>0) *ALIZE* → *AL* □ (m>0) *ICITI* → *IC* □ …	triplicate → triplic formative → form formalize → formal electriciti → electric …
4		□ (m>1) *AL* → □ (m>1) *ANCE* → □ (m>1) *ENCE* → □ (m>1) *ER* → □ …	revival → reviv allowance → allow inference → infer airliner → airlin …
Étape 5		□ (m>1) *E* → □ (m=1 and not *o) *E* → □ (m>1 and *d and *L) → lettre non doublée	probate → probat, rate → rate cease → ceas controll → control, roll → roll

Les règles de désuffixation sont exprimées sous la forme $(condition)S_1 \rightarrow S_2$ ce qui signifie que si un mot se termine par S_1 et que le préfixe satisfait la condition alors le suffixe S_1est remplacé parS_2

- $*c$: le préfixe se termine par la lettre c.
- $*v*$: le préfixe contient une voyelle.
- $*d$: le préfixe se termine par une consonne doublée.

- $*o$: le préfixe se termine par cuc où le second c n'est ni w, ni x, ni y.

Il est possible d'utiliser des opérateurs booléens : et, ou, non.

2.4 La modélisation des sujets (Topic Model)

Un **topic model**[8] est un modèle génératif[9] et une méthode populaire permettant de déterminer des sujets ou thèmes abstraits dans un document. L'idée de base est de décrire un document comme un mélange de différents sujets. Un sujet est simplement une collection de mots qui se produisent fréquemment avec l'autre.

- **Prospérités du topic model :**

Comme indiqué dans **[Kireyev et al., 2009]**, les **topic models** ont certaines propriétés qui le rendent appropriés pour analyser les données de Twitter. Celles-ci sont résumées ci-dessous :

- Les Topic models ne font aucune hypothèse sur l'ordre des mots **[Steyver et Griffiths, 2007]**. Ceci est connu comme modèle de sac-de-mots[10] (en anglais bag-of-words). Ainsi, ils ne tiennent pas compte de la grammaire. Ils sont particulièrement adaptés pour gérer la langue et la grammaire des irrégularités dans les messages Twitter.
- Chaque document est représenté comme un vecteur numérique qui décrit sa répartition sur les sujets. Cette représentation est commode pour calculer la similarité du document et effectuer le clustering.
- L'application d'un modèle de sujet est facile, car il utilise l'apprentissage non supervisé. Il permet d'économiser l'effort nécessaire à la création des données étiquetées et des classificateurs en utilisant ces données étiquetées.
- Les Topic models sont utiles pour identifier des relations non observées dans les données. Cela rend le traitement avec les abréviations et les fautes d'orthographe facile en utilisant des topic models.

[8]http://en.wikipedia.org/wiki/Topic_model
[9]Un modèle génératif est un modèle pour générer aléatoirement des données observables. Il spécifie une distribution de probabilité conjointe sur cette observation. (Source Wikipédia : http://en.wikipedia.org/wiki/Generative model).
[10]http://en.wikipedia.org/wiki/Bag of words model

2.4.1 Allocation latente de Dirichlet (LDA)

L'un des modèles les plus populaires pour le topic model est le LDA « *Latent Dirichlet Allocation*». Le LDA **[Blei et al., 2003]** est un modèle Bayesien hiérarchique à 3 couches (voir la Figure **3.4** pour une représentation graphique) où chaque document est modélisé par un mélange de *topics*(thèmes) qui génère ensuite chaque mot du document.

- **Le Modèle :**

La Figure **3.4** représente le modèle graphique de LDA et la Figure **3.3** en donne une intuition. Nous commençons par expliciter les différents termes et paramètres du modèle :

- Un mot w est la donnée discrète, correspondant à l'indice d'un mot dans un vocabulaire fixe de taille V. On peut considérer que w est un vecteur de taille V de composantes toutes nulles sauf pour la composante i où i est l'indice du mot choisi (w^i=1).

- Un document est un N-uplet de mots, $\mathbf{w} = (w_1, \ldots w_N)$.
- Un corpus est une collection de D documents, $\mathbf{D} = (w_1, \ldots, w_D)$.
- Les variables $z_{d,n}$ représentent le topic choisi pour le mot $w_{d,n}$.

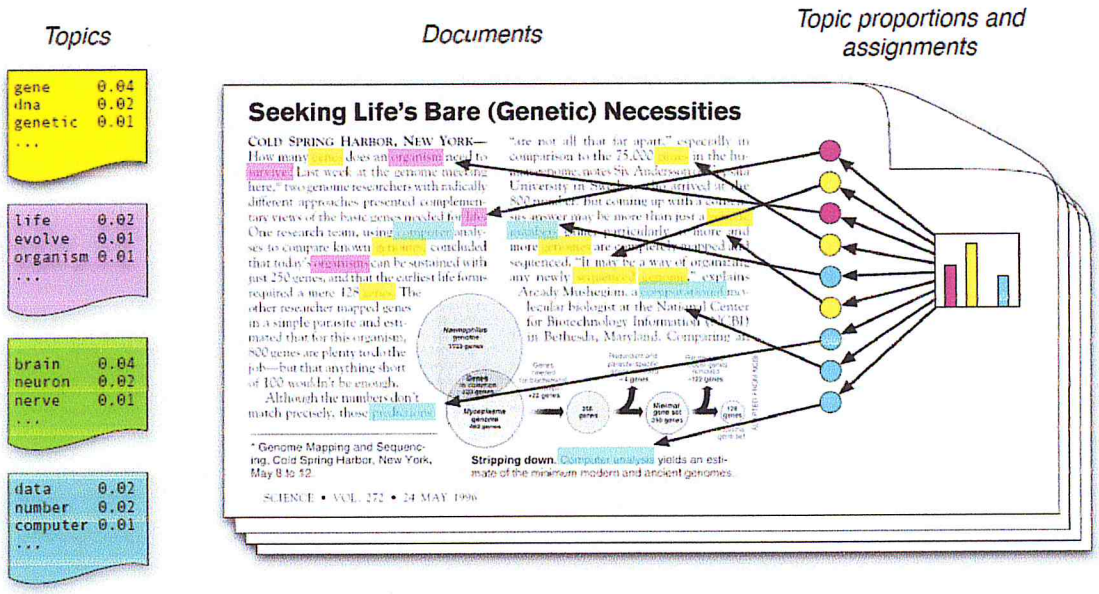


Figure 3.3 : Schéma décrivant LDA. À gauche, on peut voir la structure de chaque *topic*, donnant une probabilité à chaque mot d'un vocabulaire fixe. Pour un document donné, l'histogramme à droite décrit la distribution de *topics* dans ce document. Pour chaque mot du document, on choisit d'abord un sujet depuis cette distribution (les bulles), puis on tire un mot depuis le sujet choisi **[Blei et al., 2003]**.

- Les paramètres θ_d représentent la distribution de topics du document d.
- α et η définissent les distributions à priori sur θ et β respectivement, où β_k décrit la distribution du topic k.

Processus de génération : Le processus génératif suivi par LDA pour un document **w** est le suivant (voir le modèle graphique de la Figure **3.4**) :

1. Choisir $\theta \sim$ Dirichlet(α).
2. Pour chaque mot w_n :
   - Choisir un topic $z_n \sim$ Multinomial(θ).
   - Choisir un mot $w_n \sim$ Multinomial (β_k), avec $k=z_n$.

Loi de Dirichlet : La loi de Dirichlet permet de tirer une variable θ *telle* que $\forall i,\ \theta_i \geq 0$et $\sum_{i=0}^{k} \theta_i = 1$ (θ est dans le $(k-1)$-simplexe). Sa densité est de la forme : $\theta^{\alpha 1-1} \ldots \theta^{\alpha k-1}$

$$p(\theta|\alpha) = \frac{\Gamma\left(\sum_{i=1}^{k} \alpha_i\right)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \theta_1^{\alpha 1-1} \ldots \theta_k \theta^{\alpha k-1} \qquad \textbf{(3.1)}$$

Avec $\alpha \in R^k$, $\alpha_i > 0$ et $\Gamma(x)$ la fonction Gamma. Cette distribution permet donc d'obtenir une distribution multinomiale de paramètre θ, correspondant pour LDA au mélange de topics d'un document.

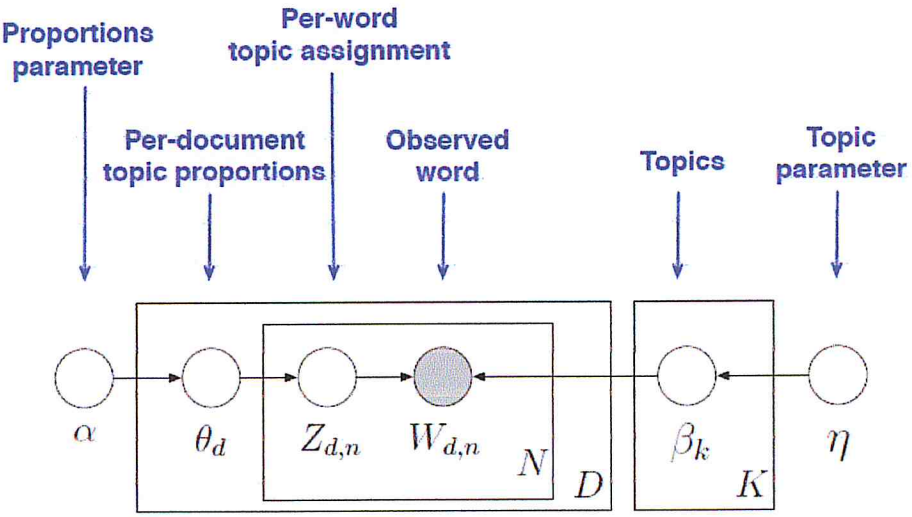


Figure 3.4 : Représentation de LDA sous forme de modèle graphique. Les boîtes représentent des répliques du modèle qu'elles contiennent (par exemple, il y a une boîte N pour chaque document D) **[Blei et al., 2003]**.

2.4.2 Les entrées du système LDA

Afin d'appliquer la méthode LDA sur les tweets, nous définissons d'abord les entrées du système qui sont : le corpus, le nombre de topics et le dictionnaire.

2.4.2.1 Le nombre de topics

L'un des problèmes rencontrés dans la LDA est l'estimation du **nombre de topics** qui doit être donné comme paramètre pour l'algorithme, et comme il est difficile de connaitre ce nombre à l'avance, plusieurs solutions s'offrent. Voilà les plus connues :

- Certains essaient avec des différents nombres et suivent l'évolution des sujets jusqu'à obtenir un nombre avec des résultats satisfaisants.
- D'autres utilisent une mesure spéciale appelée la perplexité[11] pour estimer le nombre idéal de topics.

Une solution plus intéressante sera d'automatiser ce processus et cela a conduit à la création d'un modèle inspiré de la LDA et qu'on a utilisé ici appelé la HLDA **[Teh et al., 2006]** (*Hierarchical Latent Dirichlet Allocation*). La HLDA est un modèle non paramétrique de la LDA, où le **nombre de topics** est appris à partir des données.

2.4.2.2 Le corpus

Le **corpus** est une collection de tweets. Dans la LDA ces tweets sont représentés par des documents et comme l'idée de base de la LDA est de considérer un document comme un mélange de différents sujets, il est important de comprendre où se situe ce document dans cette collection de tweets. Prendre chaque tweet comme un document n'est pas intéressant, car le tweet avec sa limite de 140 caractères est très petit pour contenir des sujets variés. Une solution plus intéressante est que ce document soit représenté par un groupe de tweets.

Comme notre base de données est constituée de plusieurs collections, chaque collection regroupe des tweets autour du même mot clé, donc on a pris chaque collection comme un document pour le LDA.

[11] La perplexité est une mesure de la façon dont une distribution de probabilité ou d'un modèle de probabilité prédit pour un échantillon. Il peut être utilisé pour comparer les modèles de probabilité. Source : http://en.wikipedia.org/wiki/Perplexity.

2.4.2.3 Le dictionnaire

Le **dictionnaire** est une collection de termes ou chaque mot possède un identifiant unique, ces identifiants sont ensuite utilisés dans la représentation vectorielle des tweets.

2.4.3 Les sorties du système LDA

Après avoir appliqué la méthode LDA sur les tweets, nous aurons un vecteur des topics et un doc topics comme résultat.

2.4.3.1 Le vecteur des topics

Le **Vecteur des topics** donne une distribution sur chaque topic avec les mots les plus fréquents et leurs probabilités associées. Nous avons choisi de nommé le Topic par le mot qui a la plus grande probabilité dans le vecteur. La Figure **3.5** donne un aperçu de ce vecteur. Ce vecteur va être utilisé pour calculer la probabilité de chaque tweet d'appartenir à l'un des topics.

TOPIC 0: ALGERIA	0.045*algeria + 0.043*day + 0.041*big + 0.039*tomorrow' + 0.039*eye + 0.039*fennec + 0.039*algeri + 0.036*5vdsnr8ezw + 0.017*worldcup2014 + 0.01
TOPIC 1: WORLDCUP	0.066*worldcup + 0.057*game + 0.046*team + 0.041*win + 0.040*score + 0.030*support + 0.028*play + 0.025*goal + 0.024*match + 0.022*world + 0.02
TOPIC 2: FIFA2014	0.042*fifa2014 + 0.041*forget + 0.036*ronaldo + 0.035*soccer + 0.035*game + 0.034*germani + 0.034*cristiano + 0.029*cup + 0.027*meme + 0.026*bra
TOPIC 3: WORLD	0.027*world + 0.027*team + 0.027*cup + 0.026*brazil2014 + 0.021*argentina + 0.018*fifateams2014 + 0.017*footbal + 0.017*soccer + 0.016*photo + 0.0

Figure 3.5 : Un aperçu du vecteur pour 5 Topics.

2.4.3.2 Le doc topic

Nous avons estimé qu'il est intéressant de savoir où se situe chacun de ces topics dans le corpus. Pour cela nous avons exploité un fichier texte appelé le ''**doc topic** '' qui constitue l'une des sorties du système LDA-Mallet[12] (Machine Learning for Language Toolkit).Il contient la distribution de chaque topic dans un document, et donc, pour chaque collection. La figure **3.6** donne un aperçu de ce fichier.

```
#doc name topic proportion ...
0       0       13      0.4713740458015267      19      0.16603053435114504     0       0.04389312977099236     16      0.04007633587786259     17
0.02862595419847328     2       0.02862595419847328     14      0.02480916030534351     6       0.02480916030534351     15      0.02099236641221374     11
0.02099236641221374     12      0.01717557251908397     8       0.01717557251908397     9       0.013358778625954198    7       0.013358778625954198    5
0.013358778625954198    3       0.013358778625954198    1       0.013358778625954198    18      0.009541984732824428    10      0.009541984732824428    4
0.009541984732824428
1       1       3       0.22118910424305918     18      0.21149816657936094     9       0.1321372446306967      7       0.1179937139863803      15
0.1143268727082242      6       0.07294394971189104     8       0.061157674174960715    14      0.05225248821372446     2       0.003273965426925092    11
0.00248821372446307     16      0.0014405447878470404   13      0.0014405447878470404   12      0.0014405447878470404   5       0.0014405447878470404   1
0.0014405447878470404   19      9.167103195390257E-4    17      6.547930853850184E-4    10      6.547930853850184E-4    4       6.547930853850184E-4    0
6.547930853850184E-4
2       2       17      0.5835280373831776      0       0.11740654205607477     19      0.06950934579439252     13      0.03446261682242990     14
0.032126168224299006    6       0.02511682242990654     16      0.02394859813084112     11      0.02161214953271028     15      0.013434579439252336    8
0.012266355140186916    2       0.009929906542056074    18      0.008761682242990653    12      0.007593457943925234    7       0.007593457943925234    10
0.006425233644859813    4       0.006425233644859813    1       0.006425233644859813    3       0.005257009345794392    9       0.0040887850467289715   5
0.0040887850467289715
3       3       18      0.16802959501557632     10      0.09521028037383178     5       0.0948208722741433      4       0.09092679127725857     12
0.07963395638629284     16      0.07145638629283489     11      0.06873052959501558     14      0.062110591900311526    1       0.05938473520249221     2
0.055490654205607476    6       0.046534267912772584    7       0.03835669781931464     8       0.01966510903426791     15      0.012655763239875389    0
0.011098130841121495    19      0.00837227414330218     9       0.007204049844236676    3       0.006035825545171339    13      0.0029205607476635513   17
0.0013629283489096573
```

Figure 3.6 : Un aperçu du fichier doc Topic.

[12]http://mallet.cs.umass.edu

Nous allons construire une matrice à partir de ce fichier pour pouvoir la visualiser à travers des graphes. Les Figures **3.7** et **3.8** présentent des graphes pour les proportions des topics dans chaque collection, un selon la longueur et l'autre selon la densité des barres en couleur.

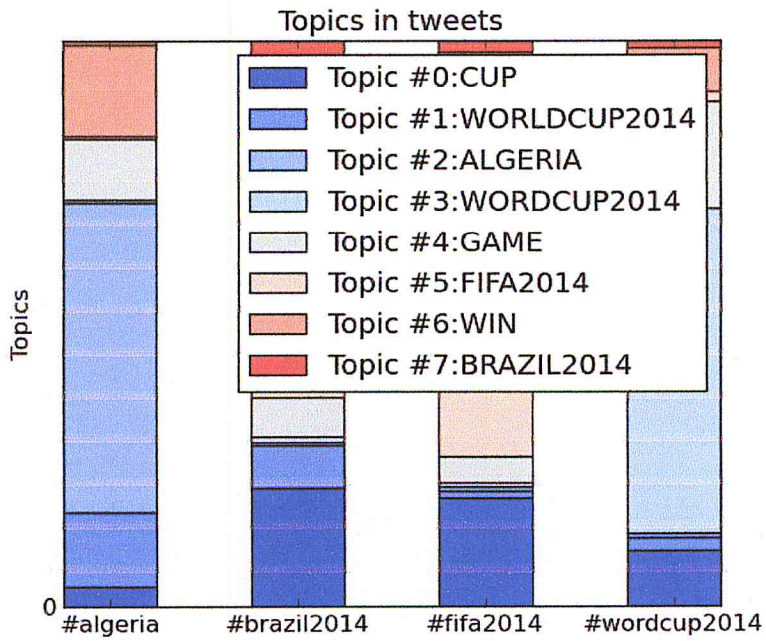


Figure 3.7 : Les proportions des topics selon la longueur.

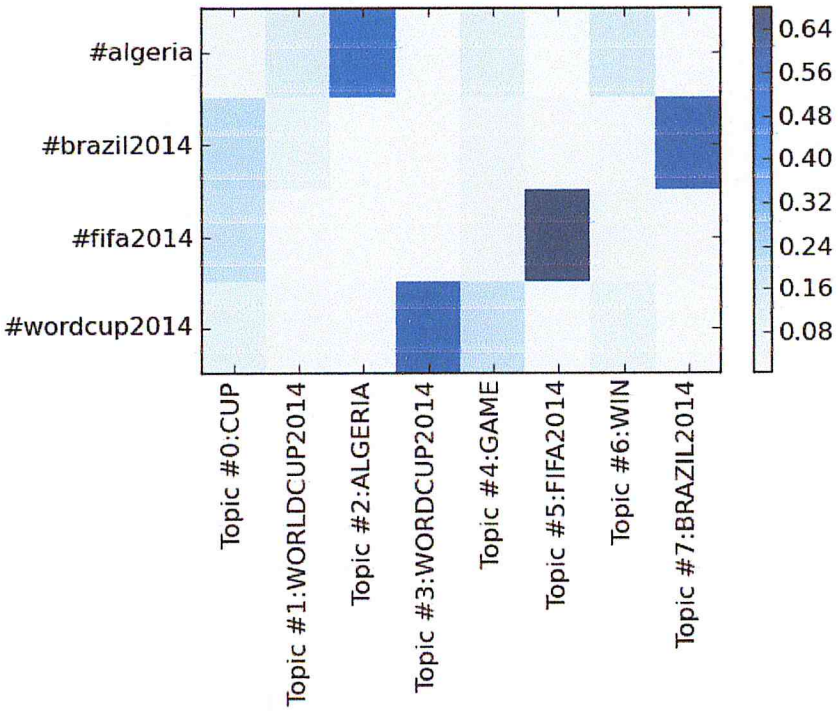


Figure 3.8 : Les proportions des topics selon la densité.

2.5 Le regroupement thématique

Le **regroupement thématique** (**clustering**[13]) est une des techniques d'apprentissage non supervisé qui prend une collection d'objets tels que les tweets et les organise en groupes en fonction de leur similitude. Les groupes qui sont formés sont appelés **clusters**. Pour effectuer cette opération, nous allons nous appuyer sur le **Vecteur des topics** et le **dictionnaire** qu'on a décrit respectivement dans la section **2.4.3.1 et 2.4.2.3,** trois étapes sont nécessaires :

- La construction des vecteurs des tweets.
- La découverte des différentes distributions des topics pour chaque tweet.
- La détermination des topics pour chaque tweet.

2.5.1 La construction des vecteurs / tweets

Cette étape consiste à représenter chaque tweet par un vecteur. La taille du vecteur est déterminée par le nombre des mots contenus dans ce tweet. Chaque case représente le nombre des occurrences d'un mot déterminé à partir du **dictionnaire**.

2.5.2 La découverte des différentes distributions des topics pour chaque tweet

Dans cette étape nous allons utiliser le vecteur des topics et les vecteurs des tweets pour réaliser ce qu'on appelle ''**inférence**'' dans la LDA. L'**inférence**[14] permet de découvrir les différentes distributions (l'ensemble des sujets, leurs probabilités de mots associés) pour chaque tweet. Un exemple de cette distribution pour deux tweets est représenté dans la figure **3.9**.

tweet N° 1	topic0* 0.0909090909091 + topic1* 0.111111111111 + topic2* 0.0909090909091 + topic3* 0.0909090909091 + topic4* 0.0909090909091 + topic5* 0.1
tweet N ° 2	topic0* 0.0833333333333 + topic1* 0.116666666667 + topic2* 0.0833333333333 + topic3* 0.0962962962963 + topic4* 0.0981481481481 + topic5* 0.1

Figure 3.9 : Un exemple d'une distribution des topics pour 2 tweets.

2.5.3 La détermination des topics pour chaque tweet

Pour déterminer le topic pour chacun des tweets dans la collection, le topic avec la **plus grande probabilité** est choisi pour chaque tweet.

Ensuite, Les tweets du même topic sont regroupés dans des **clusters**. Chaque **cluster** donc, contient une collection de tweets qui parle d'un même sujet.

[13]http://en.wikipedia.org/wiki/Cluster_analysis
[14]http://en.wikipedia.org/wiki/Latent_Dirichlet_allocation#Inference

2.5.4 La Visualisation des informations textuelles d'un cluster via le Word cloud (nuage de mot)

Le **Word Cloud** permet de mettre en évidence les mots importants dans le cluster. Typiquement, la fréquence d'un mot est utilisée comme une mesure pour représenter l'importance de ce mot. Le **Word Cloud** est donc, très efficace pour avoir une première vue du résumé.

Dans le **Word cloud**, l'importance d'un mot est mise en évidence par sa taille de la police. Cependant, La phase du prétraitement du texte est très essentielle pour que le résultat soit performant. Les Figures **3.10** et **3.11** démontrent un résultat pour un cluster avec et sans prétraitement. Pour générer un **word cloud** d'abord, nous découpons chacun des tweets du même cluster en mots. Ensuite, on calcule la fréquence de chaque mot dans ce cluster.

Pour éviter la surcharge d'information, nous choisissons généralement les k meilleurs mots pour créer un **word cloud**. On a pris la valeur 30 comme valeur par défaut pour k.

Figure 3.10 : Word Cloud pour un cluster sans prétraitement.

Figure 3.11 : Word Cloud pour un cluster avec un prétraitement.

2.6 Le résumé automatique des clusters

Lorsqu'il s'agit de résumer automatiquement un document, l'approche la plus répandue consiste à pondérer les phrases selon leur contenu. Ici nous avons utilisé le même principe. Puisque chacun des clusters contient une collection de tweets, et ces tweets sont autour du même sujet. Alors, on peut considérer que le cluster représente un document et chaque tweet représente une phrase de ce document. Donc, on peut effectuer des calculs sur une matrice phrases (tweets) x mots et le résumé est généré avec les n tweets les plus lourdes dans l'ordre de leur occurrence. Cependant, comme il s'agit de matrices creuses, il est naturel de les représenter par des graphes et cela nous a conduits à appliquer une approche graphique. Ainsi, nous avons utilisé l'un des algorithmes les plus répandus de résumé automatique par extraction appelé le ''TEXTRANK'' **[Mihalcea et Tarau, 2004]** qui a été dérivé de l'algorithme PAGERANK **[Brin et Page, 1998]**. Celui-ci est utilisé par le moteur de recherche Google[15] pour calculer l'importance des pages web.

Dans ce qui suit, nous illustrons les différentes étapes de cette phase du résumé automatique :

2.6.1 Le Text Rank

L'importance d'un tweet par rapport au cluster dans lequel il appartient est estimée avec la méthode **TextRank [Mihalcea et Tarau, 2004].** Chaque cluster est représenté sous la forme d'un graphe pondéré non dirigé G dans lequel les nœuds V correspondent aux tweets, et les arêtes E sont définies en fonction d'une mesure de similarité.

Cette mesure détermine le nombre de mots communs entre les deux tweets. Pour éviter de favoriser les tweets longues, cette valeur est normalisée par les longueurs des tweets. Soit freq(m, S) la fréquence du mot m dans le tweet S, la similarité entre les tweets Si et Sj est définie par :

$$Sim(Si, Sj) = \frac{\sum_{m \in Si,Sj} \text{freq}(m,Si) + \text{freq}(m,Sj)}{\log(|Si|) + \log(|Sj|)} \quad \textbf{(3.2)}$$

La mesure utilisée pour calculer la fréquence (freq) c'est la **TF-IDF [Jones, 1972]**.

2.6.2 Le Page Rank

Pour évaluer l'importance d'un tweet, il faut en tenir compte de l'intégralité du graphe. Pour cela, nous utilisons une adaptation de l'algorithme **PageRank [Brin et Page, 1998]** qui

[15] https://www.google.com

inclut les poids des arêtes. Le score de chaque sommet V est calculé itérativement jusqu'à la convergence par :

$$p(Vi) = (1 - d) + d \times \sum_{Vj \in \text{voisins}(Vi)} \frac{Sim(Si,Sj)}{\sum_{Vk \in \text{voisins}(Vi)} Sim(Sk,Si)} p(Vi) \quad \textbf{(3.3)}$$

Où d est un « facteur d'amortissement » (typiquement dans l'intervalle [0.8, 0.9]) et voisins(Vi) représente l'ensemble des nœuds connectés à Vi. Le score de la phrase S correspond au score du nœud qui la représente dans le graphe.

$c1 = p(S)$

2.6.3 Le résumé des clusters

Le résumé est constitué de n (n=5) tweets dont les scores sont les plus lourdes dans chacun des clusters. Pour sélectionner les tweets pertinents d'un cluster trois étapes sont effectué :

Premièrement, on classe les tweets par leurs scores associés donné par le Page Rank.

Deuxièmement, on élimine les tweets doubles.

Troisièmement, on sélectionne les 5 tweets les plus lourdes.

3. Conclusion

Dans ce chapitre nous avons présenté une approche qui tire parti de l'extraction d'information réalisée par un système de résumé automatique des tweets. Nous avons appliqué cette approche sur des tweets en anglais.

Cette approche évalue la pertinence intrinsèque des tweets trouvés par le système et fonctionne en 3 temps : estimation des sujets émergents, puis classification des tweets selon ces derniers et enfin, l'extraction de tweets pertinents constituant le résumé. Pour ce faire, on utilise un modèle de sujets construit à l'aide de la LDA, qui fournit a priori l'estimation des sujets dans la collection des tweets. Ces sujets sont une collection de mots et leurs probabilités associées. Puis, un modèle de classification qui repose sur ces probabilités pour classifier les tweets. En dernier, un modèle de résumé automatique qui permet d'extraire la pertinence intrinsèque des tweets pour chaque sujet.

Dans le chapitre suivant, nous allons présenter l'implémentation et les tests de cette approche.

CHAPITRE 4
EVALUATION

1. Introduction

DANS ce chapitre, nous présentons la partie pratique qui constitue une mise en œuvre d'une plateforme pour notre approche RATR^{Tweets} concernant le résumé automatique des *tweets*. Nous commençons par introduire les outils utilisés, puis nous donnons une présentation de l'application et enfin nous présentons l'évaluation de notre nouvelle approche.

2. Environnement de développement

Pour la réalisation de notre application, nous avons opté pour l'environnement de développement suivant : langage de programmation **Python** 2.7, l'éditeur de texte **Geany** 1.24.1 et base de donnés NoSQL **MongoDB** 2.6.2 sous le système d'exploitation Linux Mint 16 Cinnamon. Nous avons utilisé différentes bibliothèques associées à python, qui sont illustrées ci-dessous :

- **Twitter** (1.14.3) : Twitter[1] est un paquet qui fournit l'accès à l'API-Twitter via des requêtes authentifiés.
- **NumPy** (1.8.1) et SciPy (0.14.0) : NumPy[2] est un paquet de Python qui offre le soutien de tableaux multidimensionnels hautement optimisées. SciPy[3] utilise ces tableaux à fournir un ensemble de recettes numériques rapides.
- **Matpotlib** (1.3.1) : Matplotlib[4] est la bibliothèque la plus pratique et riche en fonctionnalités pour tracer des graphiques de haute qualité en utilisant Python.
- **Pandas** (0.14.0) : Pandas[5] est un paquet python qui offre de haute performance, des structures de données faciles à utiliser et des outils d'analyse de données. Sa

[1]https://pypi.python.org/pypi/twitter
[2]http://www.numpy.org
[3]http://www.scipy.org
[4]http://matplotlib.org
[5]http://pandas.pydata.org

particularité est qu'il permet de stocker des données directement en format json (le format d'extraction des tweets).

- **NLTK** (2.0.4) : Nltk[6] offre des outils très avancés pour le traitement de texte. Elle offre tous les algorithmes nécessaires pour l'étape du prétraitement.
- **Gensim** (0.10.0) : Gensim[7] est un vecteur de modélisation. Il est spécialement conçu pour la manipulation de grandes collections de textes en utilisant des algorithmes en ligne efficaces. Il comprend l'implémentation de tf-idf, processus de Dirichlet hiérarchiques (HDP) et l'allocation de Dirichlet latente (LDA).
- **NetworkX** (1.7) : NetworkX[8] est une bibliothèque Python pour l'étude des graphes et des réseaux. Elle comprend des outils pour l'exploitation des graphes et implémente aussi l'algorithme PageRank.
- **Pytagcloud** (0.3.5) : Pytagcoud[9] est utilisée pour la création des nuages de tags simples (Word Cloud).
- **PyGtk** (2.24.0) : PyGtk[10] permet de créer facilement une interface graphique en utilisant le langage de programmation Python. La bibliothèque GTK 2+ sous-jacente fournit toutes sortes d'éléments visuels et des utilitaires pour elle.

3. Présentation de l'application « RATR^{Tweets} »

Après l'exécution de l'application **RATR^{Tweets}**, une interface simple est affichée comportant la description de l'application (voir Figure **4.1**).

[6]http://www.nltk.org
[7]http://radimrehurek.com/gensim/
[8]https://networkx.github.io
[9]https://pypi.python.org/pypi/pytagcloud/0.3.5
[10]http://www.pygtk.org

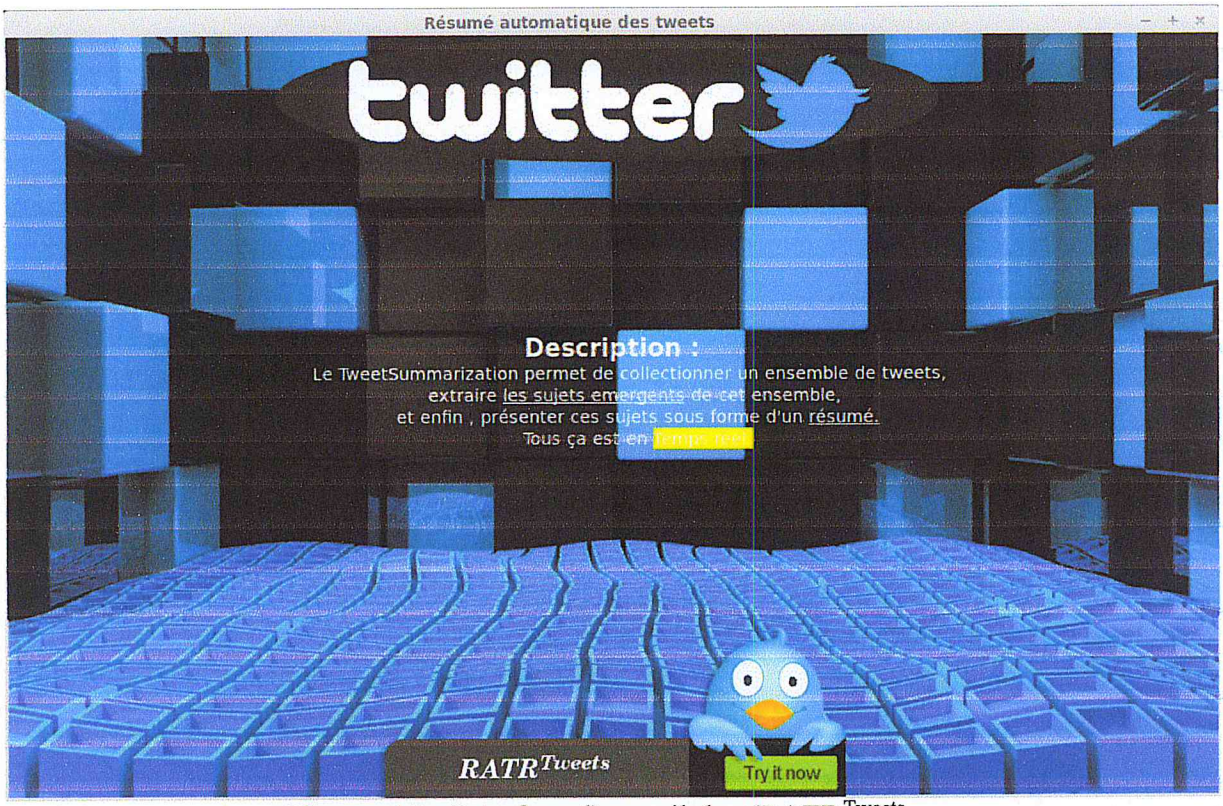


Figure 4.1 : Interface d'accueil de « RATR^{Tweets} ».

3.1 Collection des tweets

La fenêtre collection des tweets contient deux onglets : « Streaming » et « Loading ».

L'onglet « Streaming » permet de collectionner un ensemble de tweets en temps réel, il contient deux champs : « KeyWord » le mot clé pour la collecte des tweets et « DataBase Name » le nom de la base de donnés pour stocker la collection (voir Figure **4.2**).

L'onglet « Loading » permet d'importer une base de données déjà existante et comporte un seule champ « DataBase Name » (voir Figure **4.3**).

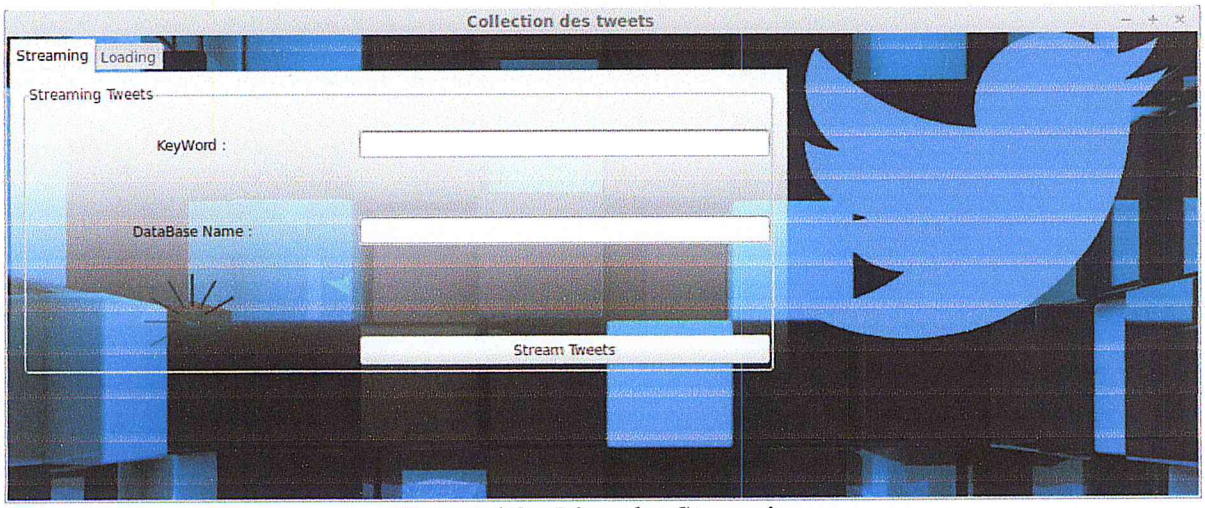


Figure 4.2 : L'onglet Streaming.

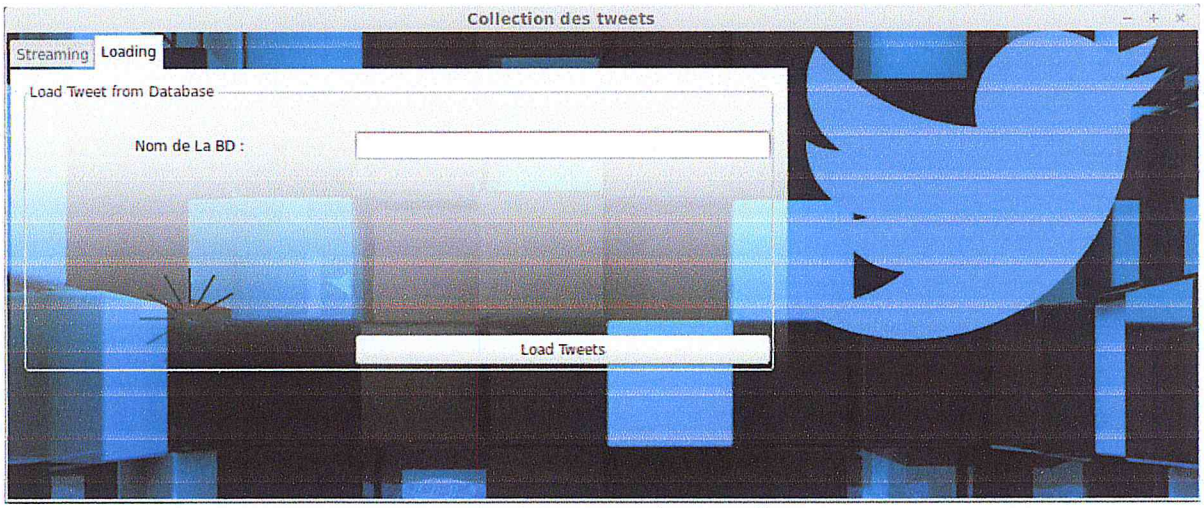


Figure 4.3 : L'onglet Loading.

3.2 Prétraitement et Statistiques

Après la collection des tweets, l'utilisateur est redirigé vers la fenêtre de prétraitement et statistiques qui contient 2 onglets. « Prétraitement » et « Statistiques ».

L'onglet « Prétraitement » nous donne un visualisation sur les étapes du prétraitement, et il contient 5 autre onglets « Text Tweet », « Elim http et @User », « Normalisation », « Lemmitisation » et « Stemmitisation ».

L'onglet « Text Tweet » permet de visualiser le contenu textuel des tweets collectés (voir Figure **4.4**).

L'onglet « Elim Http et @User » permet de visualiser le contenu textuel des tweets après l'élimination des urls (http) et les références des utilisateurs(@) (voir Figure **4.5**).

L'onglet « Normalisation » permet de visualiser le contenu textuel normalisé des tweets (élimination de la ponctuation aussi) (voir Figure **4.6**).

L'onglet « Lemmitisation » permet de visualiser le contenu textuel des tweets après l'étape de Lemmitisation (Word Net) (voir Figure **4.7**).

L'onglet « Stemmitisation » permet de visualiser le contenu textuel des tweets après l'étape de Stemmitisation (Porter) (voir Figure **4.8**).

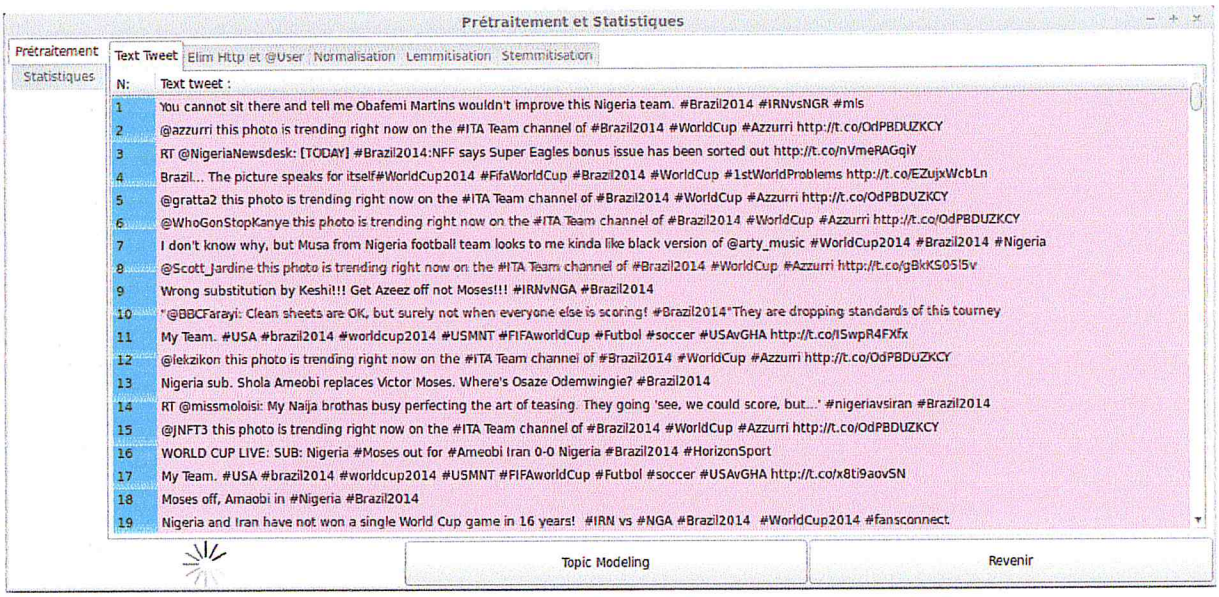


Figure 4.4 : L'onglet Text Tweet.

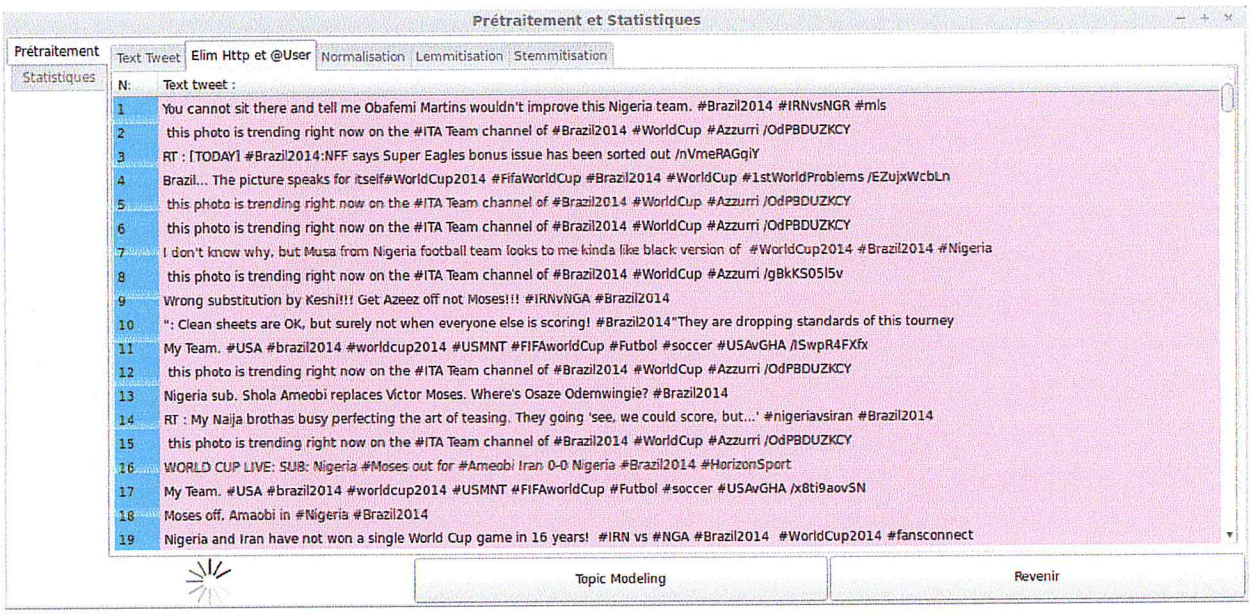


Figure 4.5 : Elim Http et @User.

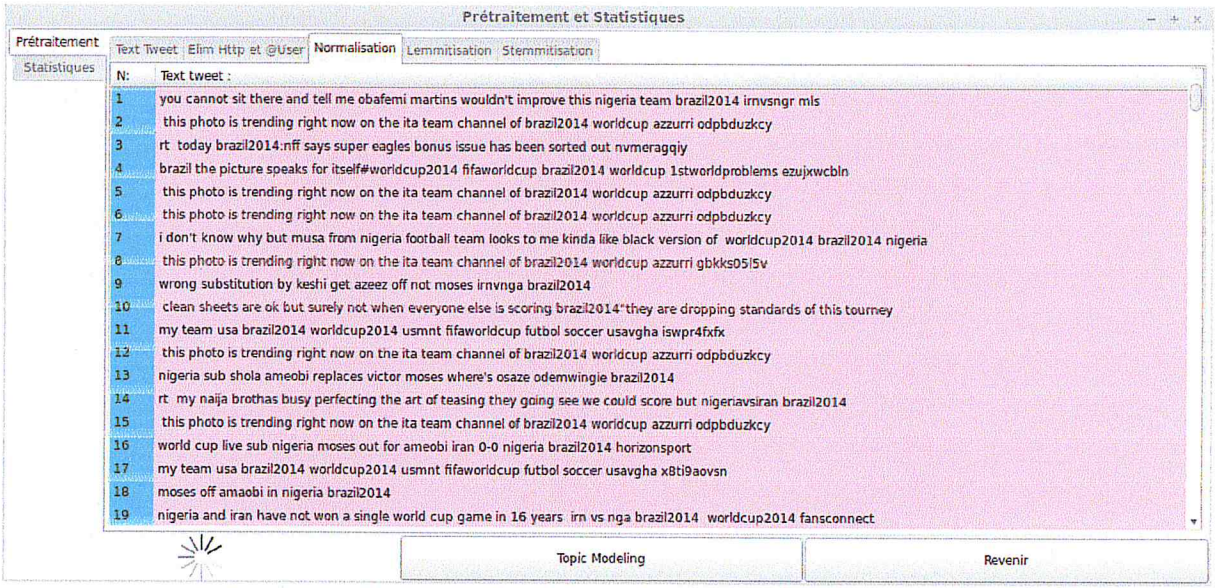


Figure 4.6 : L'onglet Normalisation.

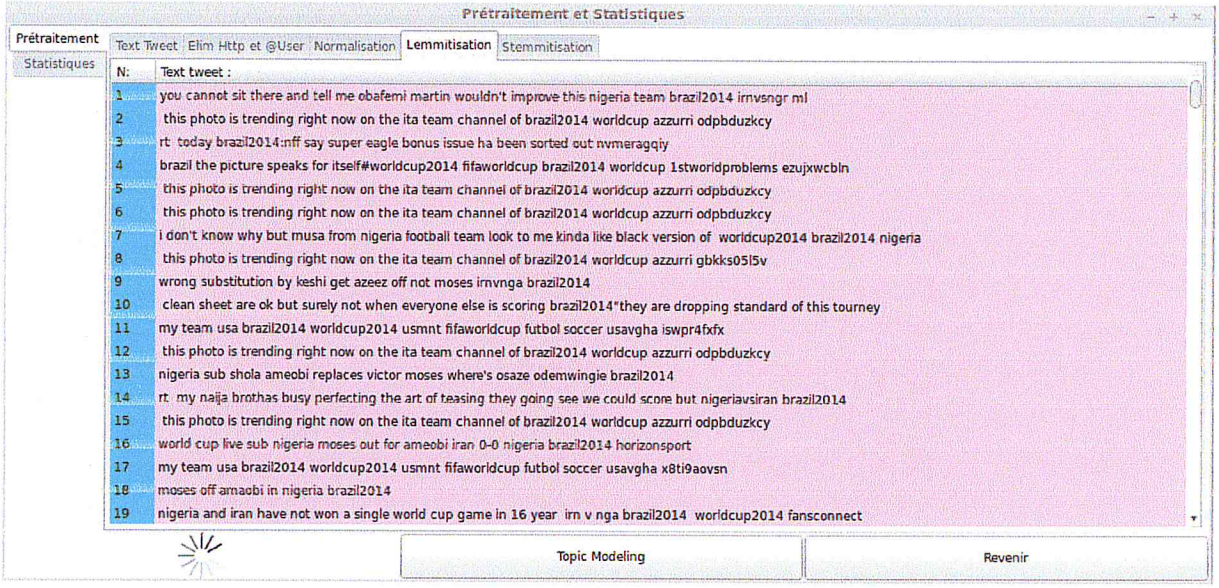


Figure 4.7 : L'onglet Lemmitisation.

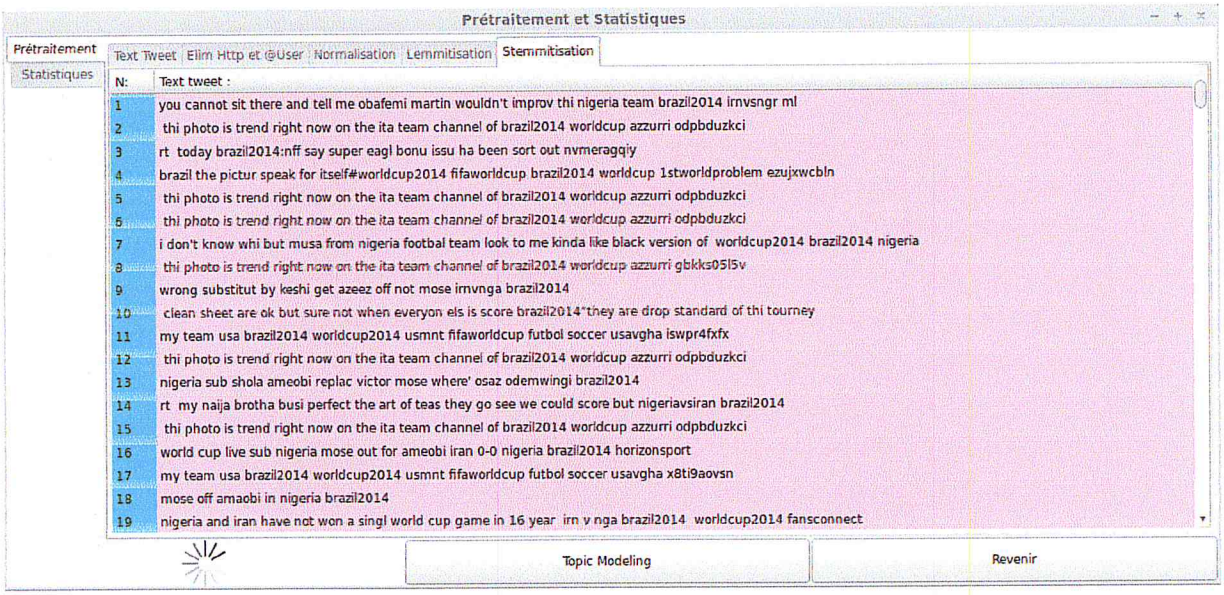


Figure 4.8 : L'onglet Stemmitisation.

Pour l'onglet « Statistiques » il permet de visualiser des statistiques sur les différentes entités des tweets (Voir les détails de cette étape dans la section **4.1)** et il contient 3 autres onglets « Fréquences », « Users », « Hashtags » et « Urls ».

L'onglet « Fréquences » permet de visualiser un tableau de fréquences sur les entités des tweets (''users '', ''hashtags'' et ''urls'') (voir Figure **4.9**).

Chacun des onglets « Users », « Hashtags » et « Urls » permet de visualiser deux graphes un histogramme et un repère log-log sur cette entité (voir Figure **4.10**).

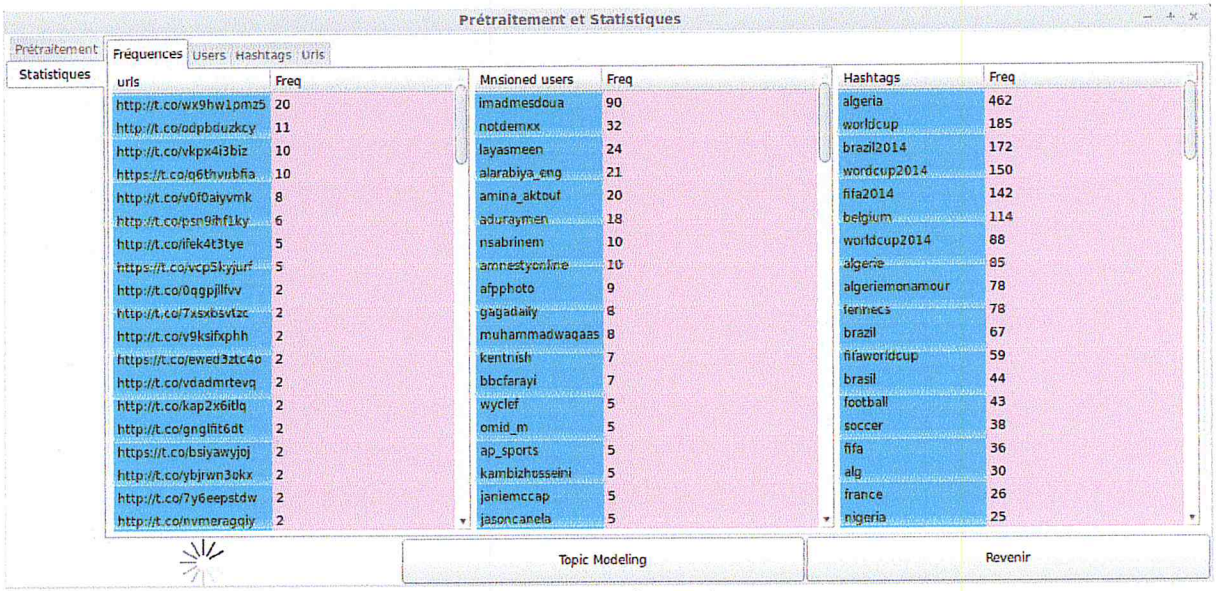


Figure 4.9 : L'onglet Fréquences.

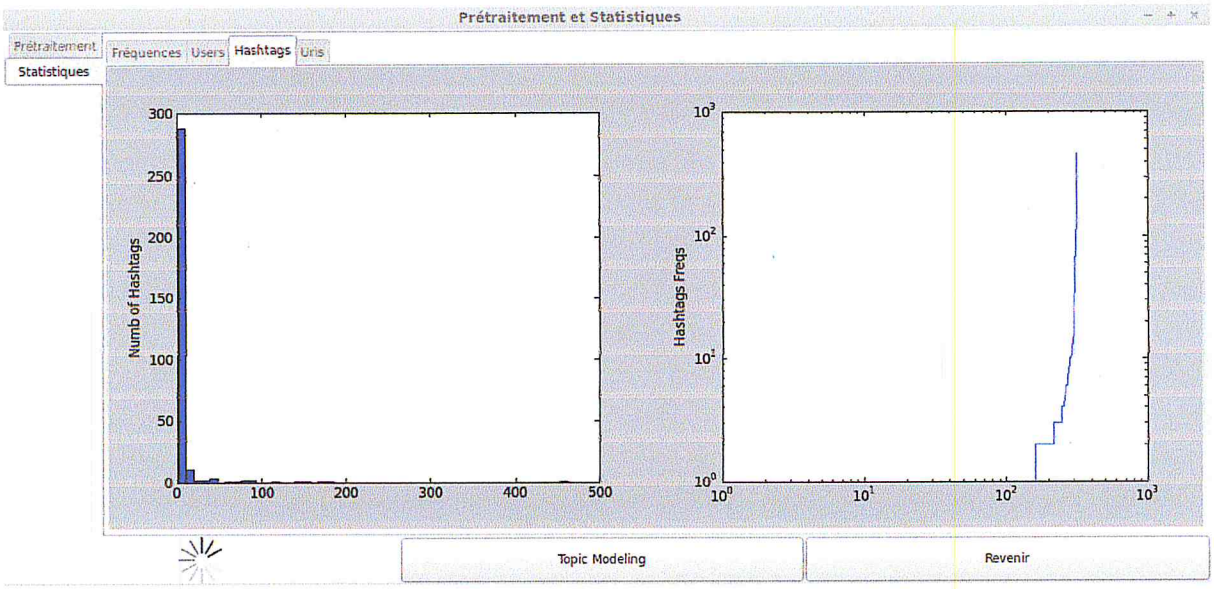


Figure 4.10 : L'onglet Hashtags.

3.3 La modélisation des sujets (Topic Modeling)

A ce niveau, la fenêtre de Topic Modeling nous permet de visualiser les différents topics dans la collection des tweets. Elle contient 3 onglets « Topics », « Prop.Long » et « Prop.Dens ».

L'onglet « Topics » affiche les vecteurs des topics, les mots constituant ces topics et leur probabilité associée (voir Figure **4.11**).

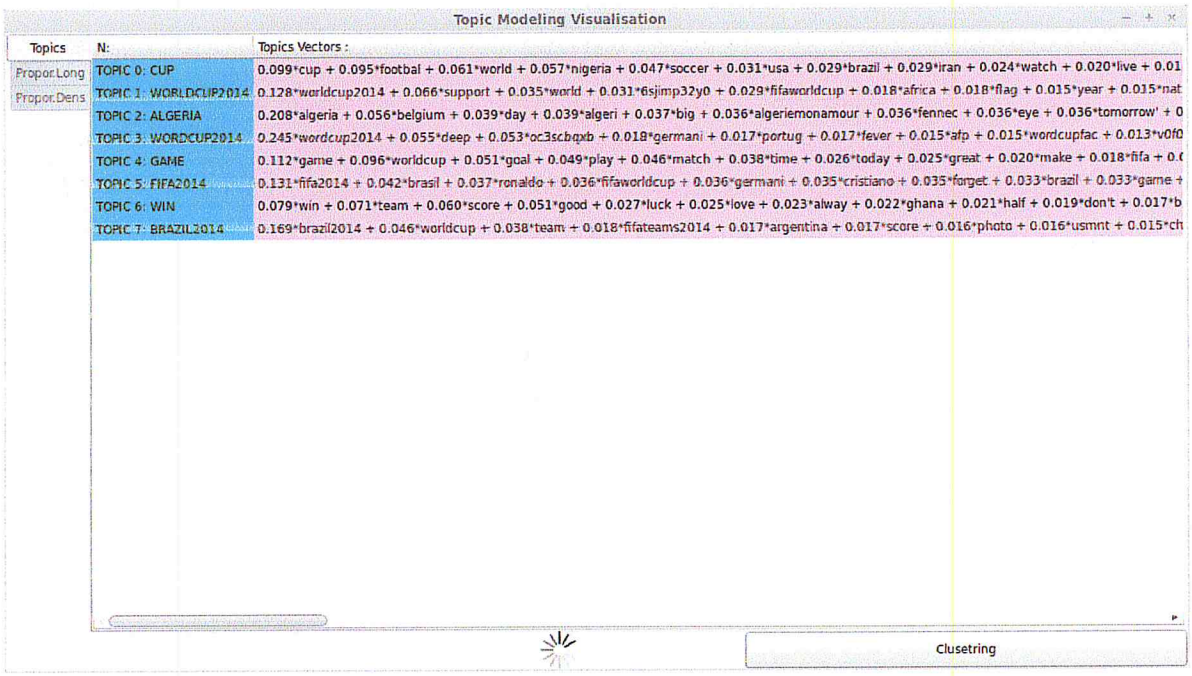


Figure 4.11 : L'onglet Topics.

Les onglets « Propor.Long » et « Propor.Dens » chacun nous permet de visualiser les proportions des topics dans nos collections de tweets sous forme de graphe, un selon la longueur et l'autre selon la densité (voir Figure **4.12** et **4.13**).

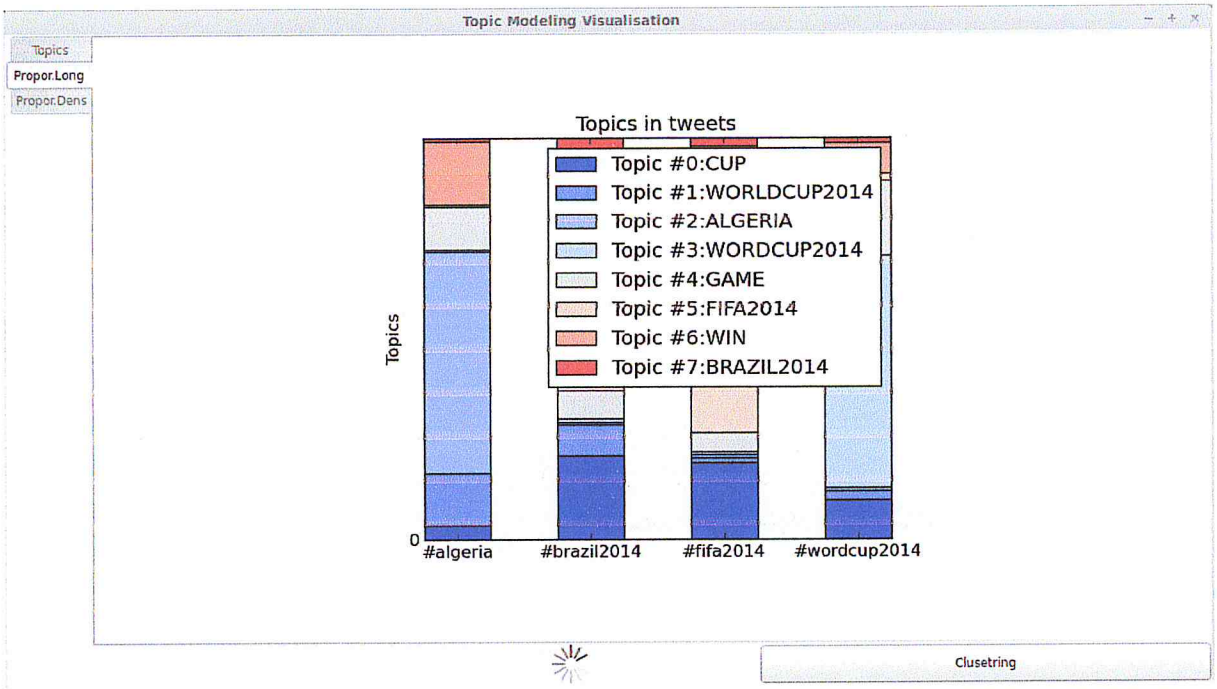


Figure 4.12 : L'onglet Propor.Long.

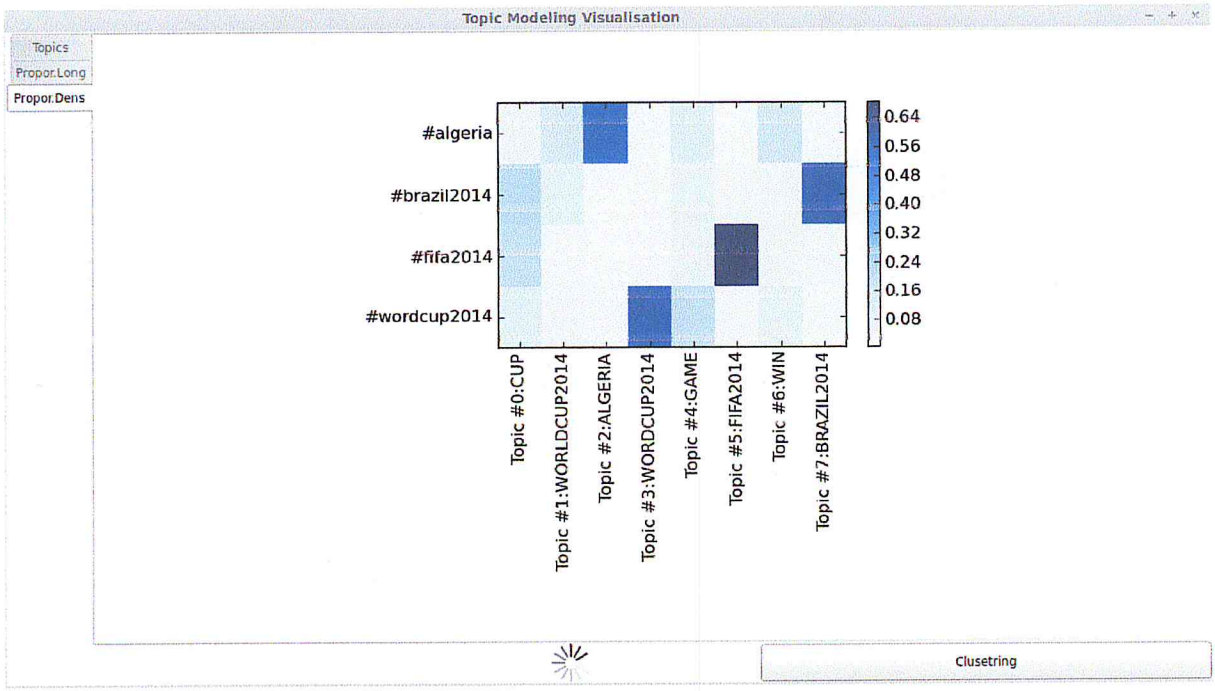


Figure 4.13 : L'onglet Propor.Dens.

3.4 Le Clustering

Une fois la modélisation des topics terminée, on procède au clustering des tweets selon les vecteurs des topics. La fenêtre de Clustering contient 2 onglets « clusters » et Probs ».

L'onglet « clusters » nous donnes les résultats du clustering pour chaque tweet (voir Figure **4.14**).

L'onglet « Probs » nous donne le résultat de la distribution des topics dans chaque tweet (voir Figure **4.15**).

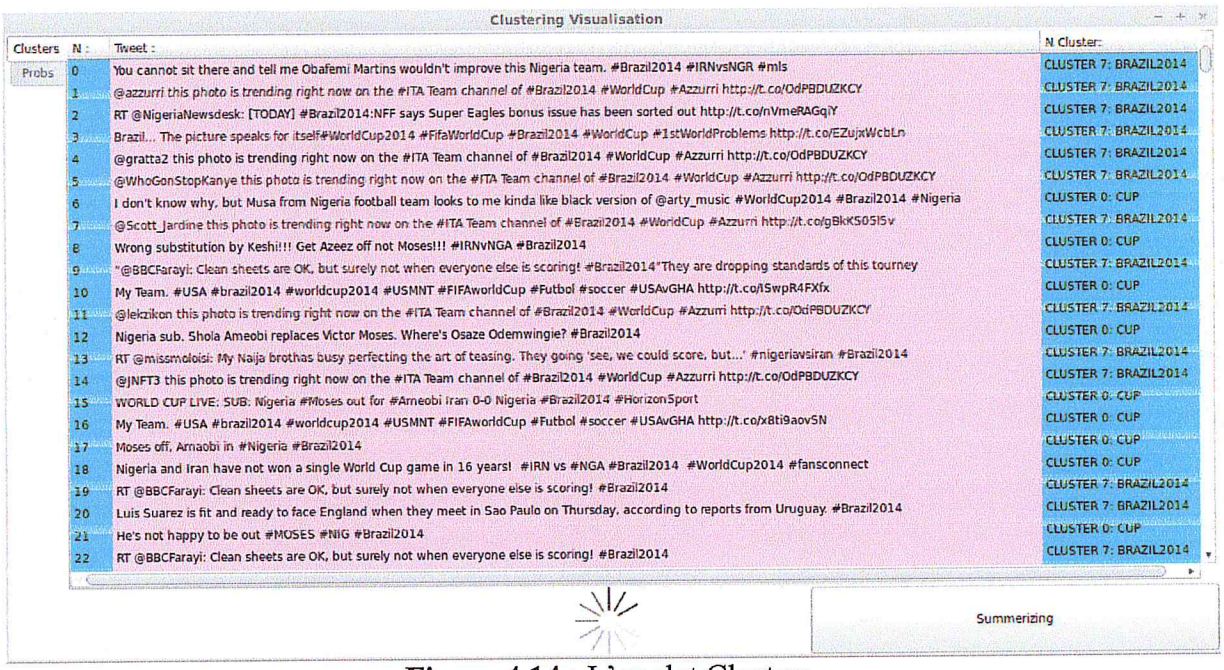


Figure 4.14 : L'onglet Clusters.

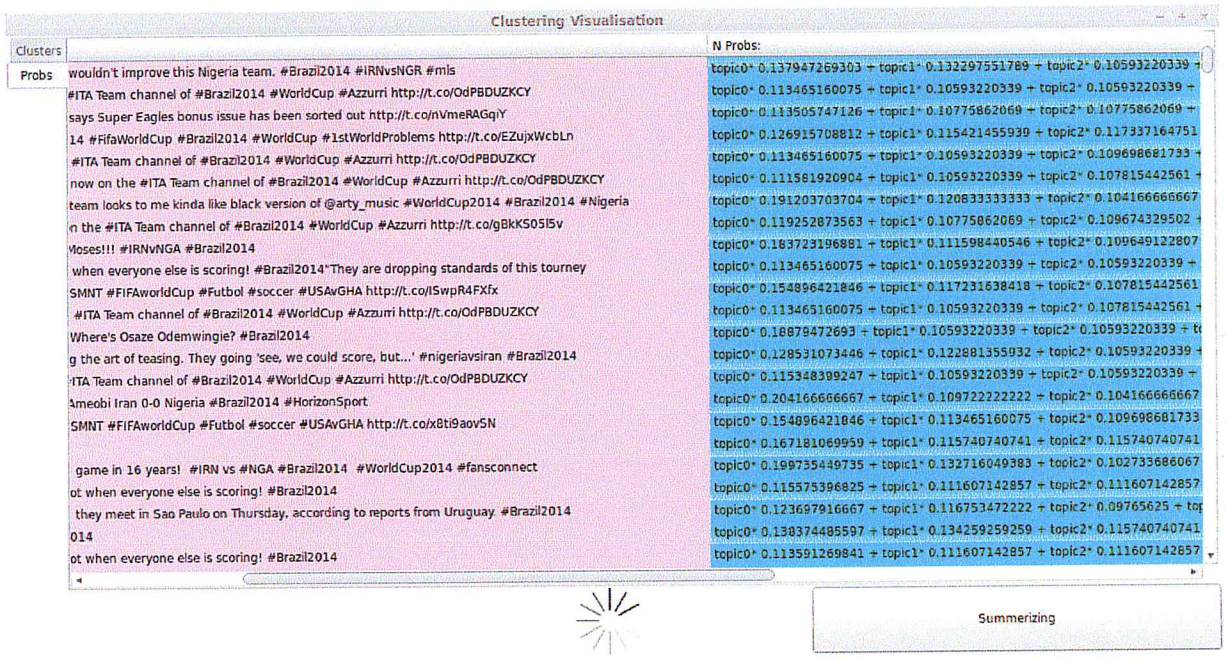


Figure 4.15 : L'onglet Probs.

3.5 Le Résumé

La fenêtre du résumé présente les résultats du résumé automatique, chaque cluster représente un sujet émergent et les 5 tweets les plus pertinents qui sont choisies POUR représenter le résumé du cluster. Cette fenêtre contient 2 onglets « Word Cloud » et « résumé».

L'onglet « Word Cloud » permet de visualiser un word cloud pour chaque cluster (voir Figure **4.16**).

L'onglet « Résumés » présente le résumé (voir Figure **4.17**).

Figure 4.16 : L'onglet Word Cloud.

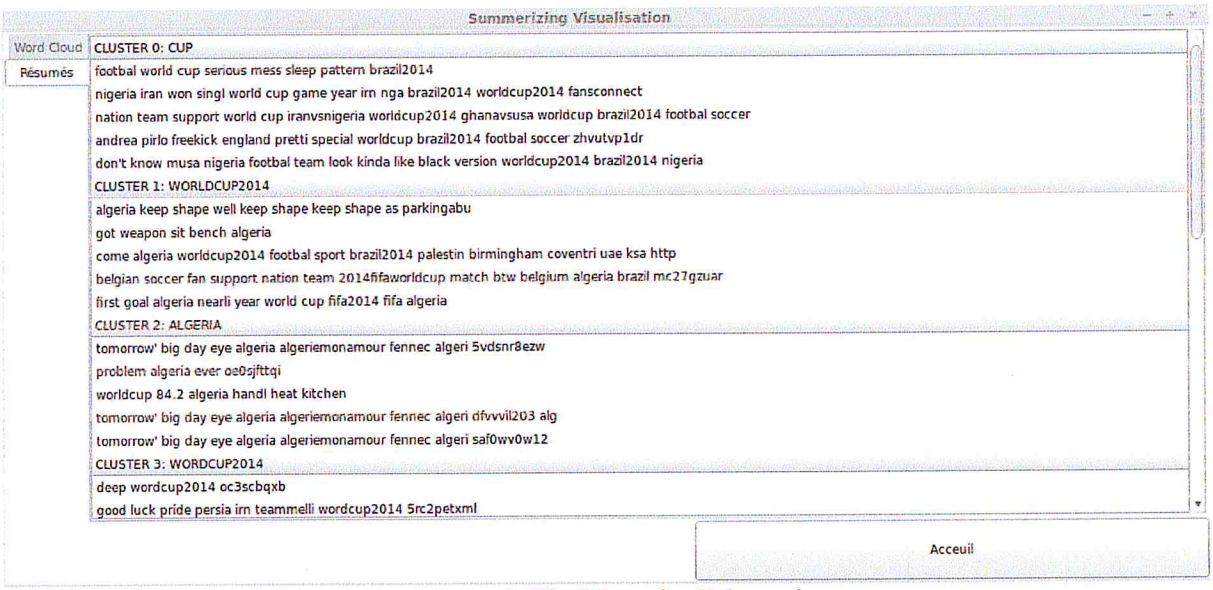


Figure 4.17 : L'onglet Résumés.

4. Test

4.1. Collections de tests

Une collection de tweets est utilisée pour l'évaluation de notre approche. Nous avons utilisé la coupe du monde 2014 comme un événement en temps réel pour tester la performance de notre approche. Pour cela, Nous avons utilisé l'API-Twitter de type ''public streaming'' avec les mots clés présentés dans le tableau 4.1.

Tableau 4.1 : Mots clés utilisées pour la collection des tweets.

Mot clés
#wordcup2014
#algeria
#brazil2014
#fifa2014

4.2 Les statistiques

Dans cette étape nous allons exploiter trois entités des tweets : les utilisateurs (@), les urls (http) et les hashtags(#) afin de calculer leur fréquence. Cela permet de voir les entités qui ont été largement mentionnées dans la collection des tweets. Nous visualisons les résultats de deux manières :

4.2.1 Un tableau des fréquences

Chaque entité est présentée dans un tableau contenant les mots et leurs fréquences associées. Ce tableau est représenté dans la Figure **4.18**.

4.2.2 Histogramme et repère log-log

En statistiques, un histogramme[11] est un graphique permettant de représenter la répartition d'une variable continue. Ici cette variable est la mesure de la fréquence.

Pour le repère log-log[12], c'est un repère dans lequel les deux axes sont gradués[13] selon une échelle logarithmique[14]. Cette échelle place les valeurs sur l'axe (fréquence) en progression exponentielle.

[11]http://fr.wikipedia.org/wiki/Histogramme
[12]http://fr.wikipedia.org/wiki/Repère_log-log
[13]Gradué(s) : Sur lequel sont portées des marques à intervalles réguliers afin de permettre la mesure. Source : http://fr.wiktionary.org/wiki/gradué
[14]http://fr.wikipedia.org/wiki/Échelle_logarithmique

Les Figures **4.19** et **4.20** présentent respectivement un histogramme et un repère log-log pour l'entité ''hashtags''.

urls	Freq
http://t.co/odpbduzkcy	11
http://t.co/vkpx4i3biz	10
https://t.co/q6thvubfia	10
http://t.co/v0f0aiyvmk	8
http://t.co/psn9ihf1ky	6
http://t.co/ifek4t3tye	5
https://t.co/vcp5kyjurf	5
http://t.co/0qgpjllfvv	2
http://t.co/7xsxbsvtzc	2
http://t.co/v9ksifxphh	2
http://t.co/vdadmrtevq	2
http://t.co/kap2x6itlq	2
http://t.co/gnglfit6dt	2
https://t.co/bsiyawyjoj	2
http://t.co/ybjrwn3okx	2
http://t.co/7y6eepstdw	2
http://t.co/nvmeragqiy	2
http://t.co/lgi1qmvpoo	2
http://t.co/mppzumvirr	2
http://t.co/ogpgj9qq2o	2

Mnsioned users	Freq
imadmesdoua	76
notdemxx	32
layasmeen	24
amina_aktouf	20
aduraymen	18
amnestyonline	10
afpphoto	9
gagadaily	8
kentnish	7
bbcfarayi	6
omid_m	5
ap_sports	5
kambizhosseini	5
janiemccap	5
jasoncamela	5
o_lucky_me	4
tiesto	4
cristiano	4
sushmaswaraj	3
ussoccer	3

Hashtags	Freq
algeria	168
brazil2014	164
wordcup2014	150
fifa2014	137
worldcup	113
algerie	82
algeriemonamour	78
fennecs	78
brazil	52
fifaworldcup	46
brasil	43
worldcup2014	41
football	40
soccer	38
fifa	31
france	26
nigeria	21
argentina	17
fifateams2014	16
usmnt	14

Figure 4.18 : Tableau des fréquences des entités des tweets.

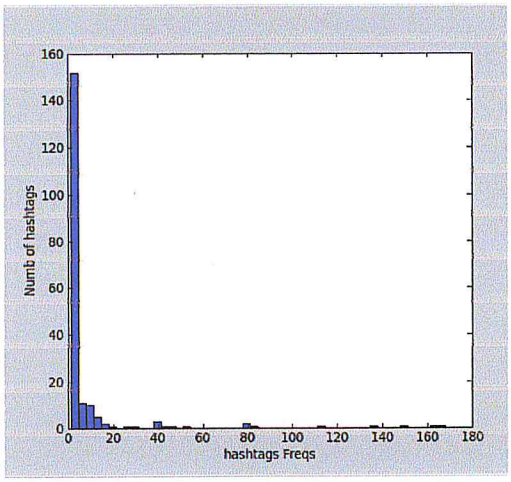


Figure 4.19 : Histogramme pour l'entité hashtags.

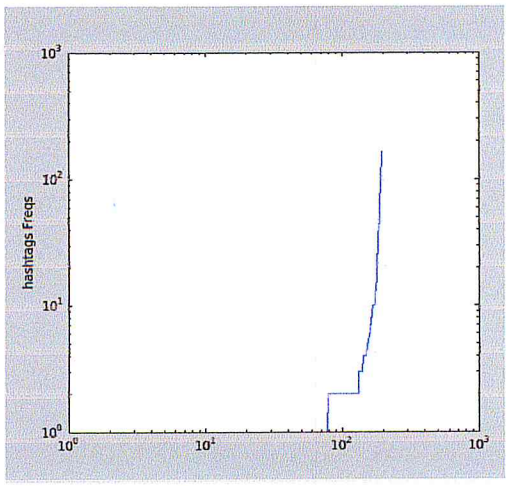


Figure 4.20 : Repère log-log pour l'entité hashtags.

Les tableaux ci-dessous présentent d'autres statistiques sur la collection de test :

Tableau 4.2 : Analyse des tweets.

Nombre total de tweets	905
Nombre de mots distincts	1910
Nombre de topics	8
La moyenne de nombre de tweets par topic	70

Tableau 4.3 : Analyse des entités

Entité	Nombre d'apparition
Utilisateur mentionné (@)	201
Hashtag (#)	455
Url (http)	314

Tableau 4.4 : Nombre d'apparitions des mots d'arrêts les plus utilisées.

Terme	Nombre d'apparition
the	474
to	206
a	159
in	157
on	133
for	126
of	110
and	76
at	48

4.3 Les mesures d'évaluation

Pour évaluer l'approche sur la collection de tests, nous utilisons les mesures d'évaluation décrites ci-dessous.

Le rappel (*Recall*), la précision (*Precision*) et la *F-mesure* (*F-measure*) sont utilisés pour évaluer les résultats obtenus. Ces mesures sont fréquemment utilisées pour évaluer le degré de la pertinence des résumés qui constituent les points focaux mentionnés dans les tweets. Le rappel R et la précision P sont définis par les formules suivantes **[van Rijsbergen, 1979]** :

$$P = \frac{\sum_i a_i}{\sum_i a_i + \sum_i b_i}, \quad R = \frac{\sum_i a_i}{\sum_i a_i + \sum_i c_i} \qquad \textbf{(4.1)}$$

Pour un résumé d'un cluster construit C_i :

- a_i (TP : *True Positive*) est le nombre de tweets pertinents mentionnés dans le résumé du cluster C_i.

- b_i (FP : *False Positive*) est le nombre de tweets non pertinents mentionnés dans le résumé du cluster C_i.

- c_i (FN : *False Negative*) est le nombre de tweets pertinents mais non mentionnés dans le résumé du cluster C_i.

La *F-mesure* mesure un équilibre entre la précision et le rappel. Elle calcule la moyenne harmonique des deux précédentes valeurs **[Knijf, 2008]** :

$$F1 = \frac{2 \times P \times R}{P + R} \quad \textbf{(4.2)}$$

Dans la section suivante, nous allons discuter les résultats de l'évaluation.

4.4 Résultats

Pour l'évaluation, Nous appliquons notre approche $RATR^{Tweets}$ avec des différentes valeurs pour le nombre des tweets qui représentent le résumé pour chaque cluster.

Comme l'indique le tableau **4.5**, selon les mesures d'évaluation nous pouvons dire qu'avec notre approche les résultats de *résumé* obtenus sont de bonne qualité.

Tableau 4.5 : Les résultats du *résumé de l'approche* $RATR^{Tweets}$.

Nombre de tweets du résumé par cluster	Rappel	Précision	F-mesure
1	0.80	0.98	0.88
5	0.96	0.96	0.96
10	0.97	0.91	0.93
15	1.00	0.80	0.88

Nous constatons qu'en augmentant la valeur du nombre de tweets qui représente le résumé du cluster le rappel augmente alors que la précision diminue. Ceci a également été rapporté dans les travaux de **[Chakrabarti et Punera, 2011]**.

5. Conclusion

Dans ce chapitre, nous avons appliqué notre approche sur une collection de tweets. D'abord Nous avons présenté l'environnement de développement ainsi que le fonctionnement de l'application. Par la suite, nous avons présenté qu'elle que statistiques de note collection de test, et pour terminer nous avons évalué notre approche.

CONCLUSION GÉNÉRALE

NOTRE travail de mémoire s'est déroulé dans le cadre des systèmes de résumé automatique(RA) des tweets, autrement dit les systèmes de recherche et d'extraction d'information qui constitue les sujets émergeants dans la collection des tweets. Pour cela, nous avons choisi d'abord d'employer un modèle de modélisation des sujets (Topic Modeling), technique largement utilisée en RI (Recherche d'information) puis un modèle de résumé automatique sur le résultat du premier modèle.

Dans ce mémoire, nous avons parcouru l'état de l'art des systèmes de résumé automatique des tweets. Nous avons vu quels étaient leurs fondements et leurs évaluations. Nous avons pu constater que parmi les étapes clés de la chaîne RA (analyse des tweets, recherche d'information et extraction des tweets pertinents), une phase de traitement importante consistait à sélectionner les tweets pertinents dans le contexte d'un sujet bien défini.

À cette occasion, nous avons vu que cette sélection doit se faire parmi une collection de tweets qui parlent du même thème utilisé pour guider le système de résumé automatique dans ses sélections à l'intérieur des tweets. Notre travail s'est orienté essentiellement dans deux directions : l'utilisation du Topic Modeling pour découvrir les sujets émergeants, effectuer le regroupement (clustering) des tweets et enfin la sélection des tweets pertinents pour chaque sujet. Étant données les techniques d'extraction d'information utilisées en RA, nous considérons qu'un tweet pertinent est tout d'abord un tweet proche, dans sa forme, d'un tweet bien écrit. Le présupposé est que les tweets trop bruités issus du réseau social Twitter conduisent à une extraction de résumé non pertinent, et ne sont donc pas pertinents. Nous avons donc mis en place une méthode de prétraitement capable de réduire la quantité du bruit dans les tweets. Pour la modélisation des sujets et le clustering, nous avons utilisé le modèle LDA, qui s'adapte au données a traitées (les tweets). Pour résumer chaque sujet, nous avons utilisé le TextRank c'est une méthode qui s'appuie sur une approche graphique.

Le travail présenté débouche sur plusieurs perspectives de recherche. Il serait intéressant de :

- Évaluer l'approche sur des collections de tests comportant un nombre plus élevé de tweets, et sur des événements autre que le sport.
- Il est clair que dans notre approche, le topic modeling est utilisé pour assurer une couverture maximale de la diversité des événements, donc l'utilisation de d'autres modèles pour le regroupement thématique peut améliorer le résultat.
- Une perspective incontournable, est d'appliquer ce qu'on appelle la contextualisation des tweets pour pouvoir présenter le résumé sous forme d'un texte au lieu des tweets.

BIBLIOGRAPHIE

[Barzilay et Elhadad, 1997] Barzilay R., et Elhadad M., 1997. Using lexical chains for text summarization. In *Proceedings of the ACL workshop on intelligent scalable text summarization*, pages 10–17.

[Blei et al., 2003] Blei D. M., Ng A. Y., et Jordan M. I., 2003. Latent dirichlet allocation. *Journal of Machine Learning Research.* 3:993–1022.

[Brin et Page, 1998] Brin S. et L. Page L., 1998. "The anatomy of a large-scale hypertextual web search engine," Computer networks and ISDN systems, vol. 30, no. 1, pp. 107–117.

[Chakrabarti et Punera, 2011] Chakrabarti D., et Punera K., 2011. Event summarization using tweets. In Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media, pages 66–73.

[Callison-Burch et Osborne, 2003] Callison-Burch C., et Osborne M., 2003. Statistical Natural Language Processing. In A Handbook for Language Engineers, A. Farghaly, Ed. CSLI.

[Edmundson, 1969] Edmundson H. P., 1969. New Methods in Automatic Extracting Journal of the ACM (JACM) 16(2), 264–285).

[Ganti et al., 1999] Ganti V., Gehrke J., et Ramakrishnan R., 1999. Cactus—clustering categorical data using summaries. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on knowledge discovery and data mining*, pages 73–83, New York, NY, USA. ACM Press.

[Hahn et Mani, 2000] Hahn U., et Mani I., 2000. The challenges of automatic summarization. *Computer*, 33(11) : 29–36.

[Harabagiu et Hickl, 2011] Harabagiu S., et Hickl A., 2011. Relevance modeling for microblog summarization. In Fifth International AAAI Conference on Weblogs and Social Media.

[Hu et al., 2007] Hu M., Sun A., et Lim E. P., 2007. Comments-oriented blog summarization by sentence extraction. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 901–904, New York, NY, USA. ACM Press.

[Inouye et Kalita, 2011] Inouye D., et Kalita J. K., 2011. Comparing twitter summarization algorithms for multiple post summaries.In Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing(socialcom), pages 298–306. IEEE.

[Jones, 1972] Jones K. S, 1972. A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation.

[Jones, 2007] Jones K. S., 2007. ``Automatic summarising: the state of the art", *Information Processing and Management.*

[Kalita, 2002] Kalita J. K., 2002. Naïve Bayes Classifiers for Spam Detection. MXLogic, Inc. Colorado Springs, CO.

[Khan et al., 2013] Khan M. A. H., Bollegala D., Liu G., et Sezakiy K., 2013“ Multi-Tweet Summarization of Real-Time Events,” Graduate School of Information Science and Technology.The University of Tokyo, Tokyo 113–8656, Japan, Center for Spatial Information Science he University of Tokyo, Tokyo 153–8505, Japan.

[Kireyev et al., 2009] Kireyev K., Palen L., et Anderson A., 2009. Applications of topics models to analysis of disaster-related twitter data. *NIPS Workshop.*

[Knijf, 2008] Knijf J.De., 2008. "Studies in Frequent Tree Mining", UU Universiteit Utrecht (169 pag.) (Utrecht: Utrecht University).

[Kolcz et al., 2001] Kolcz A., Prabhakarmurthi V., et Kalita J., 2001. Summarizing as feature selection for text categorization, pages 365-370. CIKM '01.

[Kupiec et al., 1995] Kupiec J., Pedersen J. O., et Chen F., 1995. A trainable document summarizer. In *Research and Development in Information Retrieval*, pages 68–73.

[Lecroq] Lecroq T. L'algorithme de PORTER.Universite de Rouen.FRANCE.

[Lin, 2009] Lin J., 2009. Summarization. In *Encyclopedia of database systems*. Heidelberg, Germany: Springer-Verlag.

[Lin et Hovy, 1997] Lin C., et Hovy E., 1997. Identifying topics by position.

[Liu et al., 2011] Liu F., Liu Y., et Weng F., 2011. Why is"SXSW" trending? Exploring multiple text sources for twitter topic summarization. In Proceedings of the ACL Workshop on Language in Social Media (LSM).

[Luhn, 1958] Luhn H. P., 1958. The Automatic Creation of Literature Abstracts. IBM Journal of Research and Development 2(2), 159–165.

[Mahesh, 1997] Mahesh K., 1997. Hypertext Summary Extraction for Fast Document Browsing, Working Notes of the AAAI Spring Symposium for the WWW, pages 95-103.

[Manning et Schütze, 1999] Manning C. D., et Schütze H., 1999. Foundations of statistical natural language processing. MIT Press, Cambridge, MA, USA.

[Mathioudakis et Koudas, 2010] Mathioudakis M., et Koudas N., 2010. Twitter Monitor : Trend detection over the Twitter Stream, Proceedings of SIGMOD Conference ? p. 1115-1158.

[Mihalcea et Tarau, 2004] Mihalcea R., et Tarau P., 2004. Textrank: Bringing order into texts. In *Conference on empirical methods in natural language processing*, Barcelona, Spain.

[Miller, 1995] Miller A. G., 1995. WordNet: A Lexical Database for English. Communications of the ACM Vol. 38, No. 11: 39-41.

[Mooers, 1948] Mooers C. N., 1948. Application of Random Codes to the Gathering of Statistical Information. PhD thesis, MIT.

[Neto et al., 2002] Neto J. L., Freitas A. A., et Kaesterner C. A. A., 2002. Automatic text summarization using a machine learning approach. In *SBIA '02: Proceedings of the 16th brazilian sympoisium onartificial intelligence*, pages 205-215, London, UK.

[Porter, 1980] Porter M. F., 1980. An Algorithm for Su_x Stripping Program.

[Redmond et Wilson, 2012] Redmond E., et Wilson J. R., 2012. Seven Databases in Seven Weeks. Pragmatic Programmers.

[Ritter et al., 2012] Ritter A., Mausam, Etzioni O., et Clark S., 2012. Open domain event extraction from twitter. In KDD, pages 1104–1112. ACM.

[Ritter et al., 2013] Ritter A., Xu W., Grishman R., et Meyers A., 2013. "A Preliminary Study of Tweet Summarization using Information Extraction," Computer Science and Engineering University of Washington Seattle, WA 98125, USA.Computer Science Department New York University New York, NY 10003, USA.

[Shakespeare, 1946] Shakespeare W., 1946. The Tragedy of Hamlet, Prince of Denmark. Crofts, New York.

[Sharifi et al., 2010] Sharifi B., Hutton M. A., et Kalita J., 2010. "Automatic Summarization of Twitter Topics," in National Workshop on Design and Analysis of Algorithm, Tezpur, India.

[Steyver et Griffiths, 2007] Steyver M., et Griffiths T., 2007. *Probabilistic Topic Models.* Lawrence Erlbaum Associates.

[Teh et al., 2006] Teh Y. W., Jordan M. I., Bealet M. J., et M.Blei D. M., 2006. Hierarchical Dirichlet Processes Journal of the American Statistical Association 101: pp. 1566-1581.

[Wei et al., 2012] Furu Wei F., Liu X., Zhou M., et Shum H. Y., 2012. Quickview : Nlp-based tweet search. In Proceedings of the ACL System Demonstrations, pages 13–18. Association for Computational Linguistics.

[Xing et Xia, 2006] Xing G., Xia Z., 2006. "Classifying XML documents based on structure/content similarity", In: Workshop of the INitiative for the Evaluation of XML Retrieval, Springer, pp. 444-457.

[Zhou et Hovy, 2006] Zhou L., et Hovy E., 2006. On the summarization of dynamically introduced information: Online discussions and blogs, AAAI-2006 Spring Symposium on Computational Approaches to Analyzing Weblogs.