Détection d'évènements à partir de Twitter

Houssem Eddine DRIDI[1]* — **Guy LAPALME****

* *Druide informatique inc.*
1435 rue Saint-Alexandre, bureau 1040
Montréal, Québec, Canada H3A 2G4
houssemeddine.dridi@gmail.com
** *RALI - Département d'informatique et de recherche opérationnelle*
Université de Montréal
C.P. 6128, Succ Centre-Ville
Montréal, Québec, Canada H3C 3J7
lapalme@iro.umontreal.ca

RÉSUMÉ. *Nous présentons un système pour déterminer, à partir des données de Twitter, les évènements qui suscitent de l'intérêt d'utilisateurs au cours d'une période donnée ainsi que les dates saillantes de chaque évènement. Un évènement est représenté par plusieurs termes dont la fréquence augmente brusquement à un ou plusieurs moments durant la période analysée. Afin de déterminer les termes (notamment les hashtags) portant sur un même sujet, nous proposons des méthodes pour les regrouper: des méthodes phonétiques adaptées au mode d'écriture utilisé par les utilisateurs et des méthodes statistiques. Pour sélectionner l'ensemble des évènements, nous avons utilisé trois critères :* fréquence, variation *et* Tf·Idf.

ABSTRACT. *We present a system for finding, from Twitter data, events that raised the interest of users within a given time period and the important dates for each event. An event is represented by many terms whose frequency increases suddenly at one or more moments during the analysed period. In order to determine the terms (especially the hashtags) dealing with a topic, we propose methods to cluster similar terms: phonetic methods adapted to the writing mode used by users and some statistical methods. In order to select the set of events, we used three main criteria:* frequency, variation *and* Tf·Idf.

MOTS-CLÉS : *Twitter, hashtags, évènement, similarité sémantique,* DBscan.

KEYWORDS: *Twitter, hashtags, event, semantic similarity,* DBscan.

1. travail effectué au RALI-DIRO Université de Montréal

1. Introduction

Plusieurs recherches ont montré que les données publiées par les internautes sur les sites de médias sociaux, notamment Twitter, reflètent presque en temps réel l'intérêt du public. Twitter limite à 140 le nombre de caractères utilisés dans un message incluant possiblement des hyperliens. Dans un tweet, les sujets peuvent être étiquetés avec un mot *hashtag*, un mot précédé par un dièse `#` (*hash* en anglais). En cliquant sur un hashtag, la liste des tweets ayant le même hashtag s'affiche. Voici un exemple de tweet : `les manifestants se sont dispersés. #manifencours #ggi`. Les sujets sont `manifencours` et `ggi`. En cliquant sur `#ggi` la liste des tweets ayant comme sujet `ggi` s'affiche.

Comme tous les médias sociaux, les utilisateurs inscrits sont en mesure d'établir des relations entre eux, un utilisateur pouvant s'abonner à d'autres ce qui lui permet de consulter leurs messages au moment de sa connexion.

Le contenu d'un tweet peut être un avis, une information ou un témoignage. La vaste communauté de Twitter, le haut taux d'utilisation, plus de 500 millions de tweets par jour, et la variété des intérêts des utilisateurs accumulent des informations sur des évènements locaux (par exemple, *grève sur la hausse des frais de scolarité au Québec*) ou internationaux (*décès de Michael Jackson*). Comme nous le détaillons à la section 2, plusieurs études (Sutton *et al.*, 2008 ; Kwak *et al.*, 2010 ; Becker *et al.*, 2011 ; Jianshu et Bu-Sung, 2011 ; Ozdikis *et al.*, 2012a ; Ozdikis *et al.*, 2012b) ont montré que Twitter est une source intarissable pour dégager les intentions ou même les émotions des utilisateurs. Contrairement aux autres plates-formes de médias sociaux (*e.g. Facebook*), le contenu de Twitter est public et accessible *via* des interfaces de programmation. Tous ces facteurs nous ont encouragés à utiliser Twitter pour réaliser notre objectif, soit l'identification d'évènements qui stimulent l'intérêt des utilisateurs à un moment donné.

Nous considérons un évènement comme *quelque chose* qui arrive sur une seule journée et à un seul endroit, par exemple une manifestation, ou bien qui s'étend sur plusieurs jours ou plusieurs endroits, par exemple une épidémie. Un évènement sera représenté par un ensemble de *termes*[1] dont la fréquence augmente brusquement à un ou plusieurs moments durant la période analysée. Comme les hashtags permettent de donner une idée générale sur les sujets discutés dans un tweet, la majorité de nos méthodes utilisent ces éléments afin de déterminer les sujets saillants.

Nous avons expérimenté avec des tweets portant sur la Tunisie, la plupart écrits par des Tunisiens. Nous avons été amenés vers ce type de textes à cause de nos compétences qui nous permettent de comprendre le français et l'arabe, en particulier le mode d'écriture des Tunisiens, détaillé à la section 3, qui comporte des abréviations, des fautes de grammaire et d'orthographe, des mots arabes écrits avec des alphabets français et des chiffres et plusieurs langues à l'intérieur d'un même tweet. Il nous

1. Ici, selon le contexte, un terme peut correspondre à un mot, à un groupe de mots ou à un hashtag

était ainsi plus facile de déterminer la précision de notre système étant donné notre connaissance de l'actualité en Tunisie. Notre corpus est composé de 276 505 tweets collectés pendant 67 jours (du 8 février au 15 avril 2012).

À la différence de travaux portant sur la détection d'évènements à partir de documents longs et structurés tels que ceux analysés dans le cadre du projet TimeML (Pustejovsky *et al.*, 2010) ou du *Topic Detection and Tracking* (TDT) (Allan *et al.*, 1998), notre tâche est compliquée par la taille limitée et le type particulier d'écriture des tweets écrits en dialecte tunisien. La limite de taille des tweets reste toutefois un avantage car un utilisateur ne peut se disperser et ne traite donc que d'un seul sujet, contrairement à des textes plus longs où il peut être plus difficile de déterminer l'évènement relaté. Farzindar et Khreich (2013) présentent un panorama complet de la problématique de la détection d'évènements avec Twitter.

Intuitivement, l'augmentation brusque de la fréquence d'un terme devrait indiquer la présence d'un sujet saillant ou évènement. Pour le vérifier, nous avons calculé les fréquences de différents termes, notamment les hashtags. Effectivement, nous avons constaté que les fréquences de certains termes avaient tendance à augmenter brusquement lors d'un évènement. Si chaque terme se référait à un sujet distinct, nous pourrions distinguer facilement les évènements. Cependant, un sujet est souvent représenté par plus d'un terme. Par exemple, la disparition de l'avion *Boeing 777* du *vol MH370* de la *Malaysia Airlines* le 8 mars 2014, a provoqué l'apparition de plusieurs hashtags référant à cet évènement : *#PrayForMH370*, *#MH370*, *#MH370Flight*, *#MalaysiaAirlines*...

Il ne suffit donc pas de calculer la fréquence de chaque terme séparément, il faut plutôt les regrouper quand ils réfèrent au même sujet, pour ensuite calculer la fréquence de chaque cluster afin de déterminer l'évènement le plus important. Certains termes (hashtags) du même cluster pouvant avoir été créés avant d'autres, le regroupement des termes sert aussi à déterminer la durée de l'évènement.

La figure 1 montre les différentes expériences que nous avons menées et qui sont décrites dans les sections suivantes.

La tâche initiale était d'extraire, d'une façon continue, les tweets à partir de Twitter. La méthode d'extraction du corpus et ses caractéristiques sont décrites à la section 3.

Notre première tâche est donc de regrouper les termes référant à un même sujet. Pour ce faire, nous avons développé trois méthodes[2].

Normalisation des hashtags ① Les utilisateurs des médias sociaux commettent souvent des fautes d'orthographe. Dans notre cas, ce phénomène est amplifié par le fait que les Tunisiens ont tendance à écrire des mots arabes en utilisant l'alphabet latin et des chiffres, chaque utilisateur translittérant le mot de sa façon. Pour normaliser ces hashtags, nous avons eu recours à des algorithmes phonétiques de type *Soundex* pour supporter le dialecte tunisien, afin d'attribuer le même

2. Les nombres encerclés font référence à ceux de la figure 1

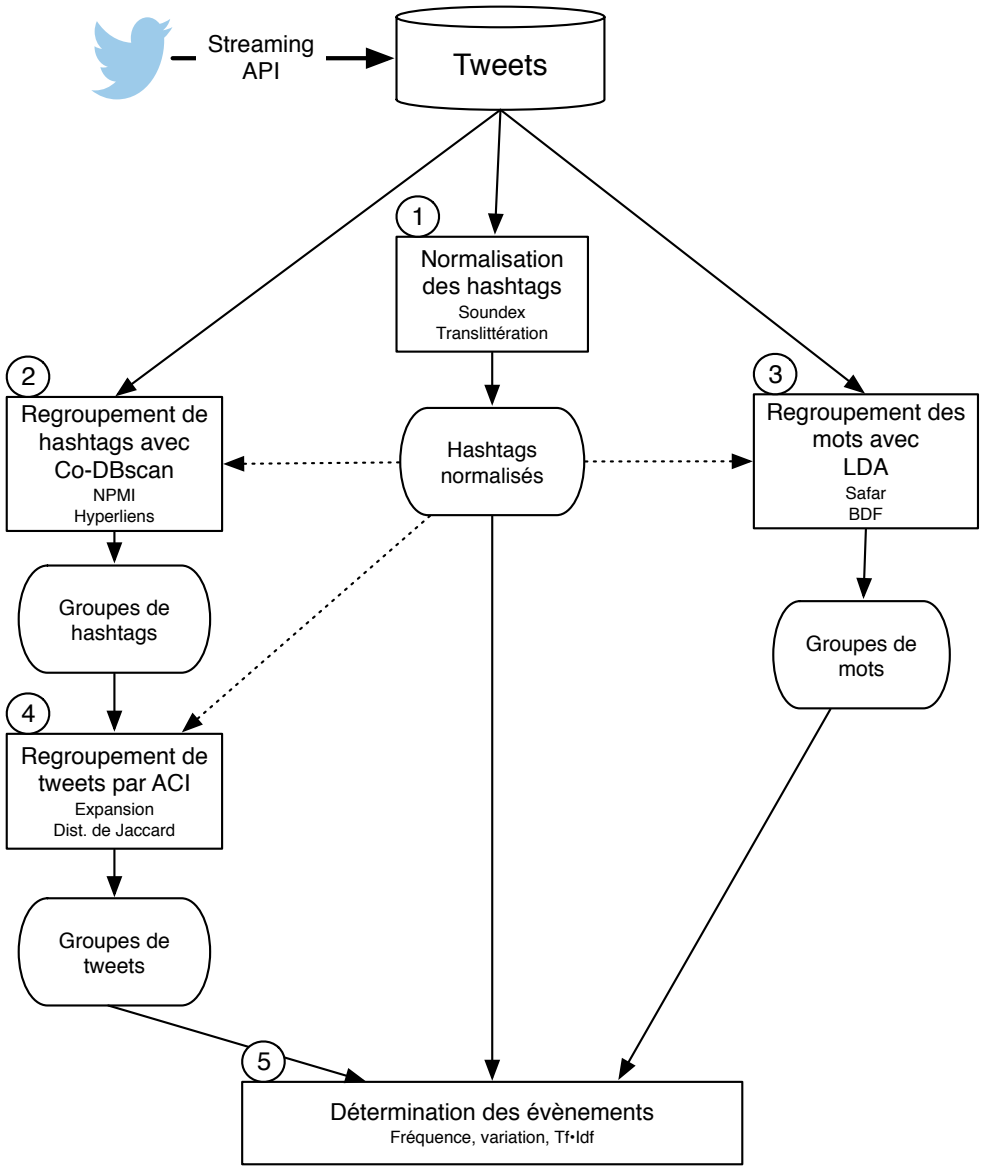

Figure 1. *Organisation des expériences menées avec le système de détection d'évènements : un rectangle représente un traitement identifié par un numéro encerclé détaillé dans le texte ; on indique en italique la méthode impliquée à cette étape. Un rectangle arrondi représente un résultat obtenu après traitement.*

code aux hashtags avec une prononciation semblable. Nous proposons, également, un algorithme de translittération qui permet de coder les hashtags écrits en alphabet arabe avec le même code que leurs semblables écrits en alphabet latin. La section 4.1 détaille ce processus.

Regroupement de hashtags avec $CoDBscan$ ② Nous avons développé une variante de l'algorithme de regroupement *DBscan* qui tient compte de la distance entre les éléments pour regrouper les hashtags similaires. Le calcul de la distance entre les hashtags est fondé sur leur cooccurrence et celle des hyperliens. Nous avons profité également des résultats obtenus par la tâche de normalisation pour améliorer le regroupement. Plus de détails dans la section 4.2.

Regroupement des mots similaires ③ Nous avons utilisé LDA (*Latent Dirichlet Allocation*), une technique statistique pour détecter les sujets d'une collection de documents. LDA considère un document comme un mélange de sujets latents, un sujet étant lui-même représenté comme une distribution de mots ayant tendance à apparaître ensemble. Nous avons profité de cette distribution pour établir les termes d'un même sujet. La section 4.3 explique ce processus.

Regroupement de tweets par un algorithme de clustering incrémental ④ Nous regroupons les tweets similaires en comparant les hashtags trouvés dans les tweets avec la mesure de Jaccard. Dans ce processus, nous avons étendu les tweets par d'autres hashtags similaires, en utilisant des clusters de hashtags, obtenus par des variantes de $CoDBscan$, afin d'améliorer le regroupement. Voir la section 4.4 pour plus de détails.

Détermination des évènements ⑤ Une fois ces regroupements effectués, nous avons utilisé les clusters de hashtags et de mots obtenus pour étiqueter chaque tweet avec un sujet. Nous supposons qu'un tweet porte sur un sujet si au moins un de ses termes est présent. Cette tâche a permis de calculer la fréquence quotidienne de chaque sujet. Nous identifions l'ensemble de sujets référant à des évènements en fonction de trois critères : fréquence, variation (ou écart-type) et *Tf·Idf* pour lesquels des sujets correspondent à des pics. Cette méthodologie est présentée à la section 5.

Nos résultats (section 6) ont été évalués par des personnes familières avec les évènements qui se sont déroulés en Tunisie à l'aide d'une application Web que nous avons développée pour en faciliter l'annotation. La section 7 présente des extensions pour des travaux futurs et la section 8 conclut en rappelant nos principales contributions.

2. Travaux antérieurs en détection des évènements

Les microblogs sont un excellent moyen pour diffuser des informations, discuter des évènements et y donner des avis. Kwak *et al.* (2010) ont constaté que les utilisateurs de Twitter diffusent parfois des nouvelles avant les journaux, la télévision ou la

radio. Sutton *et al.* (2008), dans une étude sur les incendies des forêts en Californie en 2007, ont montré que Twitter a représenté une source d'information importante pour les citoyens ; ils ont même constaté que les médias traditionnels se sont tournés vers Twitter pour obtenir des renseignements.

Les recherches qui s'intéressent à la détection des évènements à partir des articles journalistiques sont fondées sur des techniques de traitement automatique de la langue naturelle (Makkonen *et al.*, 2004 ; Zhang *et al.*, 2007) dont l'extraction d'entités nommées. Toutefois, l'application de ces techniques sur les tweets est plus difficile compte tenu de leur faible quantité d'informations. Plusieurs recherches ont montré que le contenu de ces outils (notamment Twitter) reflète bien l'intérêt et les préoccupations des utilisateurs en temps réel.

Jianshu et Bu-Sung (2011) ont proposé une méthode fondée sur la fréquence quotidienne des termes dans le corpus. La fréquence de chaque terme est représentée sous forme d'un signal. Ces signaux sont analysés par ondelettes pour déterminer quand et comment la fréquence du signal change dans le temps (Kaiser, 2011). Les auteurs ont considéré que les termes avec des signaux similaires représentaient le même évènement, la similarité entre signaux étant calculée par corrélation croisée.

Ozdikis *et al.* (2012a) et Ozdikis *et al.* (2012b) ont effectué une expansion sémantique des termes présentés dans les tweets qui s'appuie sur la cooccurrence des termes afin de regrouper les tweets selon leur similarité. Chaque tweet est représenté sous forme d'un vecteur de termes. La similarité entre deux tweets est calculée par cosinus de similarité. Les auteurs ont considéré que chaque cluster de tweets représente un évènement. Ce travail a montré que les hashtags sont de bons indicateurs pour détecter les évènements à partir des tweets. En outre, l'expansion sémantique augmente le nombre de tweets portant sur un même évènement ce qui augmente également son importance et allonge la durée d'un évènement.

Becker *et al.* (2011) ont proposé une approche pour identifier, parmi tous les tweets, ceux qui décrivent des évènements. Plusieurs tweets sont tout simplement une conversation entre amis ou des opinions. Les auteurs ont implémenté un algorithme de regroupement en ligne, qui affecte chaque nouveau tweet à un cluster adéquat. Par la suite, ils ont appliqué un algorithme de classification pour déterminer si un cluster (selon le contenu de ses tweets associés) porte sur un évènement ou non.

Comme les autres tâches, la détection d'évènements s'appuie sur les termes pour déterminer les sujets saillants survenus durant une période donnée. Toutefois, pour la détection d'évènements non connus *a priori*, il est difficile d'utiliser un ensemble prédéfini pour déterminer des évènements qu'on ne connaît pas. Les travaux qui s'intéressent à cette tâche cherchent à regrouper les tweets similaires puis à déterminer parmi les clusters obtenus ceux qui réfèrent à des évènements. La similarité entre les tweets s'appuie souvent sur leur contenu textuel. Nous utilisons une approche similaire, mais nous suggérons une méthode plus simple fondée sur la fréquence des clusters de termes. Malgré sa simplicité, notre méthode repère les sujets saillants au

cours d'une période. Notre système détecte non seulement les évènements, mais il en détermine les dates saillantes.

3. Description du corpus

3.1. *Caractéristiques des données*

Nous avons extrait les tweets d'une façon continue pendant plusieurs jours en utilisant la *streaming API*[3]. Pour recueillir des tweets, nous avons utilisé l'ensemble de mots-clés décrit dans le tableau 3.1. Nous avons récupéré 276 505 tweets entre le 8 février 2012 et le 15 avril 2012. Une fois enlevés les hashtags, les utilisateurs mentionnés et les hyperliens, les tweets contiennent en moyenne 12,4 mots, le tweet le plus long en contenant 37. Le tableau 3.1 présente quelques statistiques sur le corpus.

Mots-clés	**Définition**
`marzouki`	Président actuel de la Tunisie
`hammadi jebali`	Premier ministre dans cette période
`Tunisie`, `tounes`, `Tunisia`	`tounes` est la prononciation arabe de Tunisie
`tnelec`	Les élections tunisiennes
`sebsi`	Ex-Premier ministre après la révolution tunisienne
`nahdha`, `ennahdha`	Le parti au pouvoir durant cette période
`ghannouchi`	Chef d'`ennahda`
`sidi bouzid`	La région où la révolution tunisienne a commencé
`14jan`	14 janvier, date de fuite du président déchu Ben Ali

Tableau 1. *Mots-clés utilisés pour extraire notre corpus*

Numbre de tweets	276 505
Nombre des retweets	32 890
Nombre d'utilisateurs distincts	26 093
Nombre de tweets qui contiennent au moins un hashtag	147 395
Nombre de tweets qui contiennent au moins un utilisateur mentionné	88 595
Nombre de tweets qui contiennent au moins un hyperlien	168 309
Nombre de hashtags distincts	12 218

Tableau 2. *Statistiques sur les tweets de notre corpus*

3. `https://dev.twitter.com/streaming/overview`

3.2. *Dialecte tunisien*

Le dialecte tunisien est la langue employée par tous les Tunisiens, appelé *darija*. Il diffère de l'arabe standard, il est très influencé par la langue française, mais il intègre parfois des mots d'autres langues comme l'anglais, le punique, le berbère ou l'italien.

Le mode d'écriture employé par les Tunisiens dans les SMS et les médias sociaux présente d'autres caractéristiques : un même mot peut être écrit en alphabet latin ou arabe et parfois en mélangeant les deux alphabets. Lorsqu'un mot arabe est écrit en alphabet latin, les lettres arabes ne pouvant être transcrites directement sont remplacées par un chiffre dont la forme rappelle vaguement la lettre en arabe ou avec deux lettres qui rappellent la prononciation de la lettre. Le même mot peut donc être écrit de plusieurs façons, d'où la nécessité d'une certaine normalisation. Dridi (2014) présente plusieurs exemples de ce type d'écriture.

Certains travaux se sont intéressés au dialecte tunisien (Boujelbane, 2013), mais ils traitaient des textes plus longs en arabe plutôt que des tweets très courts combinant souvent les alphabets latin et arabe. Nous n'avons pas trouvé de ressources linguistiques permettant de déterminer les relations sémantiques entre des termes du dialecte tunisien, et même si nous en avions eues, nous aurions toujours eu besoin de déterminer les termes sémantiquement similaires à cause de l'évolution dynamique du vocabulaire dans les médias sociaux.

4. Regroupement

Nos méthodes s'appuient sur les termes trouvés dans les tweets pour déterminer les sujets saillants ; or, un sujet est souvent représenté par plus d'un terme. Il est donc important de regrouper les termes référant un même sujet, sinon chaque terme dans le corpus représentera un sujet différent.

Généralement, les textes générés par les utilisateurs sur le Web, notamment dans les microblogs, contiennent des mots non standard, car les utilisateurs commettent souvent des fautes d'orthographe et utilisent des abréviations produisant ainsi plusieurs variantes pour un même terme. Plusieurs travaux (Clark et Araki, 2011 ; Sproat *et al.*, 2001) proposent de normaliser automatiquement les termes, c'est-à-dire transformer toutes les variantes en un terme unique. Cette section présente notre approche à ce problème après avoir décrit quelques particularités de notre corpus.

4.1. *Normalisation des hashtags*

Dans cette section, nous présentons les techniques que nous avons appliquées pour normaliser les termes (① dans la figure 1). Les utilisateurs de microblogs commettent souvent des fautes d'orthographe créant ainsi plusieurs variantes pour un même terme. Ce problème a été déjà traité par les systèmes de correction orthographique à l'aide d'algorithmes phonétiques. Ces algorithmes indexent les mots selon leur prononcia-

tion. Le principe consiste à utiliser la prononciation d'un mot mal écrit pour prédire le bon mot, avec la même prononciation, qui lui correspond. Nous avons utilisé l'algorithme de *Soundex* (Russell, 1918) afin de normaliser les termes qui ont une prononciation similaire.

Comme les prononciations varient d'une langue à une autre, il existe plusieurs variantes de *Soundex*. Nous avons appliqué un *Soundex* pour le français standard puisque la prononciation des Tunisiens ressemble à celle des Français et nous l'avons adapté au dialecte tunisien. Nous avons également proposé un algorithme de translittération afin de regrouper les hashtags écrits en alphabet arabe avec leur équivalent en alphabet latin. Toutefois, les utilisateurs utilisent souvent des hashtags dérivés des dates pour référer à des évènements. Comme le *Soundex* n'est pas adapté à ce type de données, nous avons traité ces hashtags de façon particulière avec des expressions régulières. Dridi (2014) détaille ce processus.

4.2. *Regroupement des hashtags*

La relation sémantique entre deux termes sert à déterminer leurs degrés d'association. Cette information joue un rôle important dans plusieurs domaines du TAL tels que la construction automatique des thesaurus, la recherche d'information... Par exemple, il est utile d'utiliser les termes similaires à ceux spécifiés dans la requête de l'utilisateur pour récupérer les documents pertinents. Plusieurs travaux se sont appuyés sur des bases construites manuellement par des linguistes (*e.g.* WordNet) pour déterminer la relation sémantique entre les termes. Ces bases contiennent des informations indiquant le type de relation (synonyme, antonyme, hyperonyme...) entre les termes. Cependant, elles ne couvrent pas les dialectes ni le mode d'écriture (fautes, abréviations) employés dans les médias sociaux. En outre, le vocabulaire employé dans les médias sociaux est enrichi fréquemment par de nouveaux termes inventés par les utilisateurs (par exemple *produits*, *personnes*, *parti politique*...).

Une autre approche, à base de techniques statistiques, permet de fournir une information quantitative indiquant le degré de la similarité sémantique entre les termes. Cette information est estimée à partir des données observées, en se fondant sur la notion de cooccurrence, soit l'apparition simultanée de deux ou plusieurs termes dans une même fenêtre. Une fenêtre peut être un paragraphe ou une phrase. Deux termes qui cooccurrent appartiennent souvent à un même contexte. Par exemple, les termes *loi* et *avocat* apparaissent fréquemment ensemble dans un même contexte : *la justice*.

Nous avons implémenté des méthodes s'appuyant sur la cooccurrence (② dans la figure 1) afin de regrouper les hashtags d'un sujet commun. Étant donné que les tweets sont courts, nous avons considéré le tweet entier comme fenêtre. Nous avons constaté que les hashtags qui cooccurrent fréquemment sont similaires ou réfèrent à un même sujet.

Pour mesurer le degré de relation entre deux hashtags, nous avons utilisé la mesure *Pointwise Mutual Information* (*PMI*) (Church et Hanks, 1989). *PMI* mesure la quan-

tité d'informations apportée pour la présence simultanée d'une paire de termes, dans notre cas les hashtags H_i.

$$PMI(H_i, H_j) = \log(\frac{P(H_i\&H_j)}{P(H_i)P(H_j)}) = \log(\frac{N * a}{(a+b) * (a+c)})$$

$P(H_i\&H_j)$ est la probabilité que H_i et H_j apparaissent ensemble dans un même tweet. $P(H_i)P(H_j)$ est la probabilité que H_i et H_j apparaissent ensemble, s'ils sont statistiquement indépendants. Le ratio $P(H_i\&H_j)$ et $P(H_i)P(H_j)$ mesure le degré de dépendance entre H_i et H_j. *PMI* est maximisé lorsque H_i et H_j sont parfaitement associés. *N* est le nombre de tweets considérés ; a est le nombre de fois que H_i et H_j apparaissent ensemble, b le nombre de fois que H_i est présent, mais que H_j est absent alors que c est le nombre de fois que H_j est présent, mais où H_i est absent.

Pour la plupart des algorithmes de regroupement, l'utilisateur doit disposer de suffisamment de connaissances sur les données pour déterminer le nombre de clusters. À titre d'exemple, l'algorithme *k-moyenne* nécessite de spécifier à l'avance le nombre *k* de clusters à utiliser. Comme il est difficile de déterminer la bonne la valeur pour *k*, il faut en tester plusieurs.

Dans notre cas, la détermination *a priori* du nombre de clusters n'est guère envisageable, puisque chaque cluster représente un sujet dont nous ne connaissons pas le nombre dans le corpus. En outre, le *k-moyenne* est incapable de gérer les bruits et les exceptions, car chaque objet doit être associé à un cluster. D'autres algorithmes de regroupement n'exigent pas de spécifier à l'avance le nombre de clusters, par exemple l'algorithme *Density-Based Spatial Clustering of Applications with Noise* (*DBscan*) qui s'appuie sur la notion de densité (Ester *et al.*, 1996).

DBscan nécessite un seuil ϵ. Comme l'intervalle des valeurs de *PMI* est inconnu puisqu'il dépend de caractéristiques du corpus, ceci complique le choix de ϵ. Nous avons donc normalisé les valeurs de *PMI* en utilisant la méthode proposée par Bouma (2009).

$$\textit{NPMI}(e_i, e_j) = \log(\frac{P(e_i\&e_j)}{P(e_i)P(e_j)})/ - \log P(e_i\&e_j) = \textit{PMI}/ - \log P(e_i\&e_j)$$

P(e_i&e_j) est la probabilité que e_i et e_j apparaissent ensemble. La valeur de *NPMI* est donc comprise dans l'intervalle $[-1, 1]$. *NPMI*(e_i, e_j) vaut 1 lorsque e_i et e_j sont entièrement dépendants et -1 s'ils sont complètement indépendants. Cet intervalle fermé facilite la détermination de ϵ.

Nous avons appliqué *DBscan* pour regrouper les hashtags similaires. Nous avons considéré que les hashtags, qui se trouvent dans un même cluster, représentent un même évènement. Cependant, nous avons constaté que *DBscan* regroupe des hashtags qui ne sont pas vraiment similaires. Ceci est dû à un phénomène de transition lors de la fusion. Par exemple, le hashtag *#manifestation* pourrait appartenir à deux voisinages

$C \leftarrow 0$ *// initialiser le nombre de clusters à 0*
for chaque hashtag *h* non visité **do**
  ϵ-$voisinage \leftarrow$ epsilonVoisinage(*h*, ϵ)
  if tailleDe(ϵ-$voisinage$) <*MinHstgs* **then**
    marquer *h* comme $NOISE$ *// h n'appartient à aucun cluster*
  else {*il y a au moins* MinHstgs *points voisins à* h}
    $C \leftarrow C + 1$
    ajouter *h* au cluster *C*
    for chaque hashtag *h'* de ϵ-$voisinage$ **do**
      ϵ-$voisinage' \leftarrow$ hashtags de *H* de valeur *PMI* supérieure ou égale à ϵ
      if tailleDe(ϵ-$voisinage'$) $\geq$ *MinHstgs* **then**
        ϵ-$voisinage \leftarrow \epsilon$-$voisinage \cup \epsilon$-$voisinage'$
      if $Cohesion(C, h') > \lambda$ **then**
        ajouter *h'* au cluster *C*

Figure 2. *L'algorithme* $CoDBscan$ *où* ϵ *est la valeur minimale de similarité entre deux hashtags),* MinHstgs *est le nombre minimal de hashtags dans* ϵ-$voisinage$*) et* H *est l'ensemble de hashtags.* λ *est fixé à 0,1.*

représentant deux évènements différents, pendant lesquels des manifestations ont été organisées. L'application de la fusion regrouperait des hashtags portant sur deux sujets différents. Afin d'améliorer la cohésion à l'intérieur d'un cluster, nous ne les fusionnons pas directement, mais nous vérifions le degré de similarité de chaque élément d'un cluster avec tous les éléments d'un autre avant de les regrouper. Cette variante de *DBscan*, notée $CoDBscan$ pour *CohesiveDBScan*, est présentée à la figure 2.

Pour déterminer le degré de similarité, nous utilisons la fonction $Cohesion(C, e)$ déterminée comme suit :

$$Cohesion(C, e) = \frac{1}{|C|} \sum_{i=1}^{|C|} Similarity(h_i, e) \quad [1]$$

e est un hashtag dont on va mesurer le degré de similarité avec les hashtags trouvés dans un cluster *C*. *Cohesion(C,e)* est la moyenne des valeurs de *PMI* entre le hashtag *e* et ceux trouvés dans *C*. Plus la valeur de *Cohesion(C,e)* est grande plus le hashtag *e* est similaire aux hashtags trouvés dans *C*. Pour ajouter *e* à *C* la valeur de *Cohesion(C,e)* doit dépasser un certain seuil λ défini manuellement, 0,1 dans nos expériences. Une illustration du fonctionnement de cet algorithme est donné dans (Dridi, 2014).

Étant donné la taille réduite d'un tweet, les utilisateurs ne peuvent pas décrire des évènements, échanger des informations ou exprimer leur avis d'une manière efficace. Pour contourner cette limite, Twitter offre à ses utilisateurs la possibilité d'ajouter des hyperliens vers des pages externes permettant de mieux expliquer et détailler le contenu d'un tweet. Un hyperlien peut désigner une page qui contient un texte, des

images, de l'audio ou de la vidéo. Dans notre corpus, 62 % des tweets contiennent au moins un hyperlien qui constitue une information intéressante pour déterminer le degré d'association entre les termes, notamment les hashtags. Certaines URL sont trop longues et dépassent la taille permise pour un tweet, c'est pourquoi l'interface de Twitter raccourcit automatiquement les URL avec un service de réduction d'URL. Comme un tel service génère toujours la même URL pour la même entrée, même tapée par des utilisateurs différents, nous avons comparé les URL courtes plutôt que les URL finales.

Nous avons développé une autre méthode de regroupement à base d'hyperliens : nous avons supposé que deux hashtags qui apparaissent avec les mêmes hyperliens représentent le ou les mêmes sujets détaillés par ces hyperliens. Pour regrouper les hashtags, nous avons utilisé l'algorithme $CoDBscan$ de la figure 2 en ajustant la mesure de similarité entre deux hashtags pour tenir compte du fait qu'ils cooccurrent avec les mêmes liens, cette méthode est notée $CoDBscan_{hyper}$.

4.3. *Regroupement des mots*

Afin de regrouper des mots dans les tweets (③ dans la figure 1), nous avons considéré la technique du *Topic Model (TM)* qui identifie des sujets abstraits à partir d'une collection de documents. *TM* considère que chaque document peut être représenté comme un mélange de sujets latents, où un sujet est lui-même représenté comme une distribution de mots ayant tendance à cooccurrer, les mots fortement liés à un sujet ayant des valeurs plus grandes. Nous avons profité de cette distribution pour obtenir les termes qui portent sur un même sujet. Les algorithmes de *TM* utilisent la modélisation de sac de mots qui représentent chaque document par les fréquences des mots qui le composent. Cela permet d'ignorer la syntaxe des phrases pour se concentrer sur les termes trouvés dans le document. Le nombre de sujets est un paramètre qui doit être donné avant l'application de *TM*. Dans ce travail, nous avons appliqué l'algorithme *Latent Dirichlet Allocation* (LDA) (Blei *et al.*, 2003) à l'aide de la librairie *Mallet* (McCallum, 2002). Nous avons considéré chaque tweet comme étant un document différent. Après la tokenization des tweets, nous avons appliqué des prétraitements : nous avons supprimé les *usernames* (identifiant d'un utilisateur dans Twitter) et les hyperliens ; nous avons supprimé les mots-clés utilisés pour extraire les tweets ; nous avons éliminé les mots vides anglais, français, arabes et leurs équivalents en dialecte tunisien.

Pour obtenir des données plus riches, nous avons regroupé au sein d'un seul document les tweets partageant un même hashtag, même s'ils sont en différentes langues (*e.g.* en arabe ou en français). Chaque cluster va représenter un document, sous forme de plusieurs tweets. Avec cette technique, les termes qui sont sémantiquement proches seront regroupés, ce qui permet d'obtenir un vocabulaire beaucoup plus riche qu'en se limitant à un simple tweet. Cette méthode amène LDA à retourner des sujets plus significatifs quand il est utilisé sur un long document que sur un court tweet. LDA est toutefois incapable d'identifier les sens des termes, par exemple *parlent* et *parle* sont

considérés comme différents même s'ils correspondent au même verbe. Nos tweets étant écrits en français et en arabe (incluant le dialecte tunisien), nous avons utilisé *BDF*, un lemmatiseur français de notre laboratoire, pour remplacer un mot en français par son lemme alors que pour les mots écrits en alphabet arabe nous avons appliqué le *stemmer* de la librairie *Safar* [4].

4.4. *Regroupement des tweets*

Les mesures de similarité s'appuient sur les hashtags qui sont en commun dans deux tweets. Cependant, les termes, sémantiquement similaires ou écrits dans une forme différente (fautes d'orthographe, abréviation...), ne sont pas considérés comme des termes en commun. Par exemple, si un tweet contient le terme `Ghnem` et un autre contient le terme `Ghanim`, ces deux termes ne seront pas considérés comme des termes en commun même s'ils sont similaires.

Pour améliorer les résultats obtenus par les mesures de similarité afin de détecter les termes en commun entre deux documents même s'ils sont écrits d'une manière différente, nous avons décidé d'étendre les tweets par des hashtags sémantiquement similaires. Cette technique, couramment en recherche d'information afin d'améliorer les résultats de recherche, étend la requête avec d'autres termes reliés. Certaines méthodes utilisent des ressources externes telles que des thesaurus (par exemple WordNet) pour enrichir la requête.

Nous adoptons le même principe en enrichissant les tweets par d'autres hashtags. Cependant, le recours à des ressources externes prédéfinies est difficile étant donné le peu de ressources pour le dialecte tunisien. Un thesaurus contient un ensemble fini de termes, mais le vocabulaire utilisé dans les médias sociaux est évolutif car les internautes produisent souvent de nouveaux termes (abréviations, conventions, etc.) à mesure qu'apparaissent de nouvelles personnes et technologies. Pour étendre les tweets, nous avons exploité les clusters de hashtags sémantiquement similaires obtenus par les méthodes de regroupement et de normalisation présentées à la section précédente. Chaque hashtag est remplacé, le cas échéant, par le cluster de hashtags auquel il appartient. Si un hashtag n'appartient à aucun cluster, nous utilisons le hashtag lui-même.

Les travaux sur la détection des évènements utilisent des techniques de regroupement afin de répartir le corpus en clusters dont les documents sont supposés discuter sur le même sujet (évènement). Les méthodes de regroupement existantes représentent un document sous forme d'un vecteur de traits ainsi que l'importance de chaque trait dans le document. Les traits sont généralement les termes du vocabulaire utilisé dans le corpus. Dans cette représentation vectorielle, l'importance d'un terme dans un document est donnée par le produit *Term Frequency-Inverse Document Frequency* (*Tf·Idf*).

Les algorithmes de regroupement s'appuient sur ces mesures de similarité afin de comparer chaque paire de documents. Étant donné la richesse du vocabulaire de notre

4. `http://sibawayh.emi.ac.ma/safar/index.php`

corpus due principalement à la variété d'écriture de plusieurs termes et aux différentes langues utilisées, le nombre de traits utilisés pour représenter un document (tweet) est immense. Un tweet contenant peu de termes, dans notre corpus environ 7, la plupart des traits d'un tweet sont donc à 0. Nous avons constaté que la plupart des tweets ne contiennent pas de termes qui se répètent. En appliquant *Tf·Idf*, la valeur *Tf* vaut souvent 1 pour les termes présents dans les tweets. Étant donné ces deux caractéristiques, nous avons décidé de représenter les tweets par des vecteurs binaires en considérant seulement les termes, dont la valeur est plus grande ou égale à 1, et nous utilisons l'indice de Jaccard pour comparer une paire de tweets. La mesure du cosinus de similarité pourrait être plus efficace sur des grands textes, mais nous n'avons pas besoin d'utiliser les poids (voir *supra*), l'indice de Jaccard est plus simple et donne de très bons résultats.

Un algorithme de clustering incrémental (④ dans la figure 1) est appliqué pour regrouper les tweets similaires sans fixer *a priori* le nombre de clusters à obtenir.

Le cluster C^* auquel est affecté un tweet tw_i est déterminé par

$$C^* = \arg\max_{C_j \in C} \frac{1}{|C_j|} \sum_{tw_k \in C_j} Similarity(tw_i, tw_k)$$

où

– C_j est le $j^{\text{ième}}$ cluster déjà créé ;

– $Similarity(tw_i, tw_k)$ calcule la similarité, entre tw_i et un tweet tw_k de C_j, en utilisant l'indice de Jaccard ;

– $|C_j|$ est le nombre de tweets dans C_j.

Pour chaque cluster C_j existant, nous comparons tous ses tweets avec tw_i. C_j est considéré comme candidat pour tw_i si la moyenne de la similarité ($\frac{1}{|C_j|} \sum_{tw_k \in C_j} Similarity(tw_i, tw_k)$) de tw_i et des tweets de C_j dépasse un seuil ϵ que, suite à nos expériences préliminaires, nous avons fixé à 0,5.

Le candidat, qui a la valeur $\frac{1}{|C_j|} \sum_{tw \in C_j} Similarity(tw_i, tw)$ la plus élevée, est le cluster auquel tw_i va appartenir. Si l'ensemble de candidats est vide, nous créons un nouveau cluster contenant tw_i. Il n'est pas indispensable d'affecter tous les tweets à des clusters.

Nous obtenons ainsi des tweets avec plus d'informations ce qui permet aux mesures de similarité de mieux détecter les termes en commun entre une paire de tweets. Chaque cluster ne devrait contenir que des tweets discutant du même sujet, mais possiblement écrits à des dates différentes.

5. Détermination des évènements

Allan *et al.* (1998) ont défini un évènement par quelque chose d'unique qui arrive à un certain point dans le temps. Kleinberg (2003) fait remarquer qu'une tendance

ou un évènement qui suscitent de l'intérêt dans un flux de documents sont signalés par un regain d'activité signalé par une augmentation marquée de la fréquence de certains termes associés à l'évènement en question. La détermination des évènements (⑤ dans la figure 1) est un champ de recherche étudié depuis des années, il est souvent appelé *Topic Detection and Tracking* (TDT). La TDT est facilitée par l'existence de flux quotidiens des journaux sur le web. La motivation de cette tâche est l'implémentation d'un système d'alerte qui permet de détecter et d'analyser, à partir d'un flux de documents, les évènements majeurs (Allan, 2002). Les recherches, qui ont abordé ce thème, ont utilisé principalement des techniques de TAL (*e.g.* lemmatisation, détermination des parties du discours...) qui sont efficaces pour des documents contenant des informations bien structurées.

Nous pouvons distinguer deux types de travaux qui s'intéressent à l'identification des évènements.

Évènements connus *a priori* Ces travaux se concentrent sur les évènements dont les caractéristiques (type, nom, emplacement...) sont connues au préalable. Certains travaux (Sakaki *et al.*, 2010) s'intéressent à l'identification de documents (messages) qui discutent d'un évènement particulier (*e.g.* tremblement de terre, concert...) en formulant des requêtes contenant ses caractéristiques. Chakrabarti et Punera (2011) et Shamma *et al.* (2010) s'intéressent à la génération des résumés des messages qui discutent d'un évènement particulier. Petrovic *et al.* (2010) ont essayé de trouver le premier message qui discute d'un évènement précis.

Évènements inconnus Dans ce cas, les recherches s'intéressent à la détection des tendances en identifiant les sujets inédits, ou en croissance rapide, au sein d'un flux de documents (Kontostathis *et al.*, 2004). Plusieurs travaux ont montré que les utilisateurs de Twitter discutent et partagent des nouvelles à propos d'évènements imprévus (*e.g.* tremblement de terre). Pour cette raison les travaux qui s'intéressent à la détection des tendances (ou des évènements inconnus) ont eu recours à d'autres indices permettant de signaler la présence d'un évènement dans une période.

Un évènement *e*, au cours d'une période *T*, est représenté par un ensemble de traits F_e dont les fréquences augmentent brusquement à un ou plusieurs points t_e inclus dans *T*. Dans nos expériences, nous nous appuyons sur des fréquences quotidiennes.

Twitter offre déjà un service qui affiche à tout moment les dix tendances les plus fortes (sous forme de termes) dans la page d'accueil d'un utilisateur. Les tendances sont, par défaut, personnalisées par l'emplacement de l'utilisateur et changent régulièrement à un intervalle de quelques minutes. L'algorithme utilisé par ce service n'est pas diffusé, mais nous avons constaté que les tendances affichées semblent référer aux termes les plus fréquents à un moment donné. Les termes peuvent être des hashtags ou des mots dans les tweets. Ce service ne regroupe pas les termes qui représentent la même tendance, par exemple, lors du décès de Michael Jackson, la plupart des

tendances étaient autour de ce sujet : *Michael Jackson*, *MJ*, *King of Pop*... (Kwak *et al.*, 2010). Cependant, Twitter n'affiche pas les tendances pour toutes les régions, nous aurions aimé afficher les tendances pour la Tunisie, mais elle ne fait pas partie de la liste des régions géolocalisées par Twitter.

Étant donné le nombre important de tweets qui contiennent au moins un hashtag et la difficulté d'analyser les tweets, nous avons d'abord essayé de nous appuyer sur les hashtags pour identifier les sujets les plus discutés par les internautes. Nous avons constaté qu'en effet les hashtags donnaient une bonne idée des préoccupations des utilisateurs. Comme un sujet est souvent représenté par plus d'un hashtag, il est nécessaire d'identifier les hashtags sémantiquement similaires, c'est-à-dire discutant du même sujet, sinon chaque hashtag représentera un évènement différent.

Au début, nous avons calculé les fréquences de différents termes (hashtags et mots) trouvés dans notre corpus ; or nous avons constaté que cette méthode n'est pas très efficace parce qu'un évènement peut être référé par plus d'un terme. Pour cette raison, au lieu de calculer la fréquence quotidienne de chaque terme, il est préférable de calculer la fréquence quotidienne des termes d'un cluster représentant un évènement. Nous avons simplement incrémenté la fréquence quotidienne F_{gj} d'un cluster *g* au jour *j*, si au moins l'un des termes de *g* se trouve dans TW_{ij}, où TW_{ij} est le tweet i au jour j. Nous obtenons ainsi une fréquence quotidienne pour chaque évènement représenté par son cluster de termes.

Afin de détecter les évènements majeurs et de déterminer les dates où ces évènements ont eu lieu, nous avons utilisé la méthode proposée par Palshikar (2009) permettant de détecter les dates saillantes. Cette méthode détecte les pics dans une série temporelle. Elle prend comme entrée une série $f(s)$ et retourne les indices I qui correspondent aux pics. Dans notre cas, les indices sont les jours et les fréquences correspondent au nombre d'éléments d'un cluster pour cette journée.

Soit S la fonction qui permet de détecter les pics dans une période donnée :

$$S_i(k, f(s)) = \frac{\max\limits_{1 \leq j \leq k}(f_i(s) - f_{i-j}(s)) + \max\limits_{1 \leq j \leq k}(f_i(s) - f_{i+j}(s))}{2}$$

où *k* est un entier positif indiquant le nombre de voisins à considérer autour de chaque point $f_i(s)$ dans τ, les valeurs les plus appropriées de *k* étant comprises entre 3 et 5. Pour $f_i(s) \in f(s)$, S_i calcule la moyenne de la différence maximale entre les valeurs de *k* voisins à gauche et à droite de $i^{\text{ème}}$ élément. $f_i(s)$ est considéré comme un pic si :

$$S_i > 0 \textbf{ et } (S_i - mean) > stdv$$

où *mean* et *stdv* sont respectivement la moyenne et l'écart-type des valeurs positives de S_i. Dans nos expériences, *k* a été fixé à 3.

Pour déterminer les évènements saillants, nous avons testé trois critères : la fréquence quotidienne, la variation définie par l'écart-type de la fréquence quotidienne et une mesure inspirée du *Tf·Idf* où :

– Tf : correspond à la fréquence quotidienne du sujet S durant la période ;
– $Idf = \log \frac{\text{Nombre de jours}}{\text{Nombre de jours où } S \text{ est présent}}$

Nous avons supposé que les sujets avec les plus grandes valeurs *Tf·Idf* correspondaient à des évènements. L'intuition derrière ce critère est de donner plus d'importance aux sujets qui ne sont pas fréquemment discutés au cours d'une période. Cette mesure pénalise les sujets fréquents apparaissant sur plusieurs jours.

6. Expérimentations

Dridi (2014) détaille l'ensemble des expériences menées pour évaluer les techniques présentées aux sections précédentes ainsi que la détermination des seuils à utiliser dans les algorithmes. Nous présentons maintenant les principaux enseignements que nous en avons tirés.

6.1. *Regroupement*

Les algorithmes de *Soundex* et de translittération (section 4.1) regroupent efficacement les hashtags (écrits en alphabets latin et arabe) de même prononciation sauf pour certains tags courts que l'algorithme regroupait trop agressivement. Nous avons utilisé des expressions régulières afin de normaliser les hashtags correspondant à des dates. Ces méthodes ont regroupé les 12 218 hashtags en 9 033 clusters, soit une réduction d'environ 26 %.

Comme expliqué en section 4.2, avec $CoDBscan$, nous avons réussi à regrouper des hashtags sémantiquement similaires même s'ils n'avaient pas une prononciation semblable, ce qui était impossible avec *Soundex*. Nous avons proposé deux variations de $CoDBscan$: la première s'appuie sur la mesure de *NPMI* pour déterminer le degré de similarité entre deux hashtags (ou codes *Soundex*) tandis que la deuxième utilise les hyperliens partagés par une paire de hashtags (ou codes *Soundex*) comme mesure de similarité. Les résultats obtenus ont montré que ces deux mesures (*NPMI* et les hyperliens partagés) sont de bons indices pour déterminer la similarité entre les hashtags.

DBscan n'était pas efficace dans notre cas parce qu'il construisait un énorme cluster. Pour cette raison, nous avons proposé la fonction de la cohésion qui vérifie la similarité entre un hashtag et un cluster avant de l'ajouter. Afin de regrouper un grand nombre de hashtags, nous avons effectué des améliorations avec *Soundex*. L'évaluation des résultats était coûteuse en temps, il nous fallait des heures ou même des jours pour calculer les précisions des différentes variantes de $CoDBscan$: $CoDBscan_{npmi}$ qui intègre les valeurs de *NPMI* et $CoDBscan_{npmiWithSndx}$ qui y ajoute les résultats du regroupement avec le *Soundex* ; nous avons de plus testé avec $CoDBscan_{hyper}$ et $CoDBscan_{hyperWithSndx}$ qui intègre l'information des hyperliens possiblement regroupés avec le *Soundex*.

Nos variantes de $CoDBscan$ utilisent trois paramètres : ***MinPts***, ϵ, λ. Pour les variantes $CoDBscan_{npmi}$ et $CoDBscan_{npmiWithSndx}$, il est recommandé d'utiliser une valeur $\epsilon \in$ [0,6, 0,8], car avec une telle valeur nous avons réussi à regrouper le plus grand nombre de hashtags avec une forte valeur de précision. Tandis que pour les variantes de $CoDBscan_{hyper}$ et $CoDBscan_{hyperWithSndx}$, il recommandé d'utiliser une valeur $\epsilon \in$ [0,4, 0,6] où la valeur de $epsilon$ correspond à la mesure hyper qui considère les hyperliens partagés entre deux hashtags.

Nos résultats ont été évalués en les comparant avec des annotations manuelles effectuées par des experts. Ces annotations pourraient être exploitées par d'autres travaux dans le futur sous forme de données de référence pour déterminer la similarité entre certains hashtags. Cependant, certains termes peuvent toujours être regroupés car ils n'ont pas de dépendance temporelle. D'autres sont regroupés seulement par période. Par exemple, une relation sémantique entre *université de manouba* et *drapeau tunisien* ne sera valide qu'au cours de la période analysée dans ce travail. En dehors de cette période, il y a peu de chances qu'il faille considérer ces termes comme similaires.

Tâches	#H	# groupes	#H regroupés	*Pr*
Normalisation *Soundex* et date	12 218	9 033	12 218	96 %
$CoDBscan_{npmi}(\epsilon = 0{,}7)$	9 973	928	5 011	94 %
$CoDBscan_{npmiWithSndx}$ ($\epsilon = 0{,}6$)	10 929	908	6 633	90 %
$CoDBscan_{hyper}$ ($\epsilon = 0{,}5$)	6 008	1 213	3 556	92 %
$CoDBscan_{hyperWithSndx}(\epsilon = 0{,}4)$	7 431	925	4 660	91 %

Tableau 3. *Résultats obtenus par les méthodes de regroupement de hashtags : #H indique le nombre de hashtags,* Pr *est la précision calculée pour la normalisation sur les 20 clusters plus importants.*

Le tableau 3 récapitule les principaux résultats obtenus. Pour la tâche de normalisation, le nombre de hashtags regroupés est le même que celui de nombre de hashtags considérés car chaque hashtag possède un code *Soundex*. Par contre pour les variantes de $CoDBscan$, il est possible qu'un hashtag n'appartienne à aucun cluster.

6.2. *Détection*

Nous avons testé trois méthodes afin de déterminer l'ensemble des évènements intéressants au cours d'une période :

– clusters de hashtags obtenus par une normalisation à base de prononciation (flèche du centre sur le rectangle ⑤ dans la figure 1) ;

– clusters de mots obtenus par LDA à base de techniques statistiques (flèche de droite sur le rectangle ⑤ dans la figure 1) ;

– clusters de tweets obtenus par un algorithme qui détermine lui-même le nombre de clusters de tweets similaires (flèche de gauche sur le rectangle ⑤ dans la figure 1).

Nous avons testé trois critères (voir section 5) : la fréquence quotidienne, la variation définie par l'écart-type de la fréquence quotidienne et une mesure inspirée du *Tf·Idf*.

Nous ne retenons que les sujets dont la valeur correspond à un pic identifié par la formule donnée en section 5. Par exemple, en appliquant cette méthode sur les 9 033 hashtags normalisés nous en avons retenu 123 selon le critère de fréquence, 88 selon la variation et 81 selon le *Tf·Idf*.

Dans ce type de problème, il est difficile de déterminer la pertinence des résultats obtenus, car il n'y a pas de données de référence. Pour évaluer l'adéquation de chaque critère et vérifier si les sujets détectés correspondaient à des évènements, nous avons demandé à dix experts tunisiens au courant des évènements en Tunisie de valider nos résultats. L'annotation des évènements a été faite par l'intermédiaire d'un site Web [5] que nous avons créé. Une fois connecté, l'expert a reçu une liste de sujets et les dates saillantes de chaque sujet et il a dû reconnaître le sujet référé par des hashtags et juger s'il s'agissait d'un évènement ou non.

Le tableau 4 résume les résultats des annotations effectuées sur les regroupements d'évènements :

– la normalisation des hashtags donne une bonne précision (nombre de vrais évènements détectés), mais un même sujet est souvent représenté par plusieurs clusters (tableau 3), ceci conduit à l'obtention de sujets similaires. Ce problème provient du fait que le sujet n'est représenté que par les termes (hashtags) avec une prononciation similaire. Le rappel n'est donc pas très bon, même s'il est difficile à évaluer précisément faute de référence. Un groupement typique est : `rvolution, revoliton2; revoltion, revolution, rrevolution; révolution, revolution2` ;

– les groupes de termes par LDA sont moins nombreux, mais un des problèmes reste le choix du nombre de sujets discutés. Le vocabulaire utilisé dans notre corpus est très riche, beaucoup de mots sont inventés par les utilisateurs. Malgré le stemming et la lemmatisation utilisés pour normaliser les termes, nous n'avons pas réussi à reconnaître certains mots, tels que des mots vides censés être supprimés. Un groupe typique est : `drapeau, faculté, salafistes, étudiant, manouba, lettre, ali, salafiste, police` ;

– le groupement de tweets par $CoDBscan$ combinés avec *NPMI* et les hyperliens nous semble être la méthode est la plus efficace pour déterminer l'ensemble des évènements intéressants. Presque tous les sujets identifiés sont distincts sans devoir fixer au préalable le nombre de sujets. Un exemple de ces groupes est : `#concours, #emploi, #interim, #jeune, #jobs, #programme, #rec, #recrutement, #chômage, #travail`.

On constate également que le critère inspiré du *Tf·Idf* est le plus efficace pour déterminer l'ensemble des évènements intéressants, et ce pour tous les types de regroupements.

5. `http://rali.iro.umontreal.ca:8080/dridihou/`

	Fréquence		Variation		*Tf·Idf*	
	# év	*Pr*	# év	*Pr*	# év	*Pr*
Hashtags normalisés	123	0,64	88	0,82	81	0,95
Groupes termes par *LDA*	53	0,73	67	0,87	74	0,92
Gr. tweets + $CoDBscan_{npmiWithSndx}$	98	0,61	52	0,80	46	0,93
Gr. tweets + $CoDBscan_{hyperWithSndx}$	104	0,54	51	0,69	46	0,94

Tableau 4. *Résultats des annotations effectués par 10* experts*. Pour chaque groupe et méthode d'identification des évènements, on indique le nombre d'évènements annotés (# év) avec la précision obtenue (*Pr*).*

7. Travaux futurs

À l'avenir, nous comptons tester ces méthodes sur d'autres types de données. Nous avons déjà commencé à extraire des données envoyées à partir du Québec. Cette fois-ci, nous avons profité de paramètres de géolocalisation pour extraire ces tweets. Comme mentionné précédemment, ces paramètres ne sont pas fonctionnels pour le cas de la Tunisie. Grâce à l'option de géolocalisation, nous n'avons pas eu à définir des mots-clés pour extraire les données et nous avons pu extraire un grand nombre de tweets, 4 millions entre février et juin 2014.

Sauf pour les méthodes utilisées dans la tâche de normalisation, notre système pourrait en principe fonctionner sur le corpus du Québec quoiqu'il faudra optimiser les temps de calcul qui seront prohibitifs sur des données aussi volumineuses. En travaillant sur ce corpus qui devrait contenir des tweets écrits en français et en anglais, il sera préférable d'utiliser une autre variante de *Soundex*, car celui que nous avons présenté ici est adapté aux textes écrits par des Tunisiens. Il sera toutefois inutile de translittérer les hashtags pour un corpus écrit entièrement en alphabet latin. En travaillant uniquement en anglais et français, nous pourrons profiter de ressources linguistiques (dictionnaires monolingues et bilingues, thesaurus, etc.) supportant ces langues.

Les méthodes proposées pour déterminer les termes similaires faisaient une distinction entre les mots et les hashtags trouvés dans les tweets. Aucune méthode ne servait à la création des clusters contenant à la fois des mots et des hashtags similaires. L'intuition derrière nos méthodes est de vérifier l'apport de chacun de ces éléments (mots et hashtags) pour déterminer les évènements au cours d'une période. Cependant, il est évident qu'il existe des hashtags et des mots sémantiquement similaires. Ainsi, nous pourrions proposer une méthode permettant de regrouper les hashtags et les mots.

Les résultats présentés dans ce travail prouvent que les hyperliens sont des éléments importants pour déterminer la similarité entre les hashtags. Rappelons que nous avons considéré que plus les hashtags partagent les mêmes hyperliens, plus ils sont similaires. Néanmoins, tout comme il y a des termes similaires, il y a aussi des hy-

perliens similaires (référant à un même sujet). Il serait donc intéressant de regrouper les hyperliens similaires ce qui permettait d'améliorer le regroupement de hashtags. De même, ils peuvent être utilisés comme indices pour identifier les évènements où chaque groupe d'hyperliens réfère à un sujet.

Dans cet article, nous n'avons pas distingué les tweets discutant d'un évènement de ceux qui donnent d'autres types d'informations. Comme extension, nous avons imaginé développer un classificateur permettant de déterminer si un tweet porte sur un évènement ou non. Pour ce faire, nous pourrions nous appuyer sur d'autres aspects que le seul contenu du tweet, par exemple tenir compte des utilisateurs mentionnés dans le tweet, est-ce une réponse à un autre tweet ? Est-ce un retweet ?

8. Conclusion

Dans ce travail, nous nous sommes intéressés à la détection des sujets ayant suscité l'intérêt des utilisateurs au cours d'une période donnée. Pour ce faire, nous avons eu recours aux données de médias sociaux qui constituent un excellent moyen pour les internautes pour partager leurs idées, donner leur avis et diffuser des nouvelles. Ces aspects conduisent à l'accumulation de données qui reflètent les préoccupations des utilisateurs en temps réel.

Notre première tâche dans le processus de détection d'évènements consiste à regrouper les termes similaires ou discutant du même sujet. Nous avons proposé des méthodes pour regrouper les hashtags avec une prononciation semblable. Pour ce faire, nous avons utilisé l'algorithme phonétique *Soundex*, que nous avons adapté au dialecte tunisien, pour attribuer le même code aux termes qui se prononcent de la même façon. Comme un même hashtag peut être écrit soit en alphabet latin soit en alphabet arabe, nous avons proposé un algorithme pour regrouper un hashtag écrit en alphabet arabe avec ses correspondants en alphabet latin. Pour réunir les hashtags référant à un même sujet, mais ne se prononçant pas de la même façon, nous avons modifié l'algorithme *DBscan* pour regrouper les hashtags indépendamment de la langue avec laquelle ils étaient écrits. Pour déterminer la similarité entre les hashtags, nous avons profité d'informations trouvées dans les tweets telles que la cooccurrence et les hyperliens partagés entre les hashtags.

Nous avons utilisé l'algorithme LDA pour regrouper les mots qui portent sur le même sujet. Nous avons regroupé dans un seul document les tweets partageant les mêmes hashtags afin d'améliorer les résultats. L'absence de données de référence nous a obligés à évaluer manuellement les résultats de nos méthodes de regroupement de hashtags. L'évaluation, effectuée par des experts familiarisés avec les évènements qui se sont déroulés en Tunisie, a révélé une précision qui dépassait souvent 90 %.

Par la suite, nous avons utilisé ces clusters pour déterminer les sujets saillants ou les évènements. Pour ce faire, nous avons proposé deux méthodes. À partir des clusters obtenus par la normalisation des hashtags et de LDA, nous avons considéré que chaque cluster correspondait à un sujet. Un tweet porte sur un sujet si au moins l'un de ses

termes est présent. Cette tâche a permis d'obtenir la fréquence quotidienne de chaque sujet. Nous avons également expérimenté en regroupant les tweets similaires avec un algorithme de regroupement incrémental déterminant le nombre de clusters.

Comme tous les sujets ne sont pas nécessairement des évènements, nous avons adapté la méthode de Palshikar (2009) pour sélectionner les sujets saillants en évaluant trois critères : fréquence, écart-type, *Tf·Idf*. Notre évaluation a montré que le critère *Tf·Idf* est le plus adéquat pour cette tâche.

9. Bibliographie

Allan J., *Topic detection and tracking : event-based information organization*, vol. 12, Kluwer Academic Publishers, 2002.

Allan J., Carbonell J., Doddington G., Yamron J., Yang Y., « Topic detection and tracking pilot study final report », 1998.

Becker H., Naaman M., Gravano L., « Beyond trending topics : Real-world event identification on Twitter », *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM11)*, 2011.

Blei D., Ng A., Jordan M., « Latent Dirichlet Allocation », *Journal of Machine Learning Research*, vol. 3, p. 993-1022, 2003.

Boujelbane R., « Génération des corpus en dialecte tunisien pour la modélisation de langage d'un système de reconnaissance », *Actes de la 15e Rencontres des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RECITAL'2013)*, Les Sables d'Olonne, France, p. 206-216, 2013.

Bouma G., « Normalized (pointwise) mutual information in collocation extraction », *Proceedings of the Biennial GSCL Conference*, p. 31-40, 2009.

Chakrabarti D., Punera K., « Event summarization using tweets », *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, p. 66-73, 2011.

Church K., Hanks P., « Word association norms, mutual information, and lexicography », *Proceedings of the 27th annual meeting on Association for Computational Linguistics*, Association for Computational Linguistics, p. 76-83, 1989.

Clark E., Araki K., « Text normalization in social media : progress, problems and applications for a pre-processing system of casual English », *Procedia-Social and Behavioral Sciences*, vol. 27, p. 2-11, 2011.

Dridi H. e., Détection d'évènements à partir de Twitter, PhD thesis, Université de Montréal, Montréal, Canada, oct, 2014.

Ester M., Kriegel H.-P., Sander J., Xu X., « A density-based algorithm for discovering clusters in large spatial databases with noise. », *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, p. 226-231, 1996.

Farzindar A., Khreich W., « A survey of techniques for event detection in Twitter », *Computational Intelligence (Early View)*p. 33 pages, sept, 2013.

Jianshu W., Bu-Sung L., « Event Detection in Twitter », *in* L. Adamic, R. Baeza-Yates, S. Counts (eds), *ICWSM*, The AAAI Press, p. 401-408, 2011.

Kaiser G., *A friendly guide to wavelets*, Springer, 2011.

Kleinberg J., « Bursty and hierarchical structure in streams », *Data Mining and Knowledge Discovery*, vol. 7, n° 4, p. 373-397, 2003.

Kontostathis A., Galitsky L., Pottenger W., Roy S., Phelps D., « A survey of emerging trend detection in textual data mining », *Survey of Text Mining*, Springer, p. 185-224, 2004.

Kwak H., Lee C., Park H., Moon S., « What is Twitter, a social network or a news media ? », *Proceedings of the 19th international conference on World wide web*, ACM, p. 591-600, 2010.

Makkonen J., Ahonen-Myka H., Salmenkivi M., « Simple semantics in topic detection and tracking », *Information Retrieval*, vol. 7, n° 3, p. 347-368, 2004.

McCallum A., « Mallet : A machine learning for language toolkit », 2002, http ://mallet.cs.umass.edu.

Ozdikis O., Senkul P., Oguztuzun H., « Semantic Expansion of Tweet Contents for Enhanced Event Detection in Twitter », *Advances in Social Networks Analysis and Mining (ASONAM), 2012 IEEE/ACM International Conference on*, IEEE, p. 20-24, 2012a.

Ozdikis O., Senkul P., Oguztuzun H., « Semantic Expansion of Tweet Contents for Enhanced Event Detection in Twitter », *ASONAM*, IEEE Computer Society, p. 20-24, 2012b.

Palshikar G., Simple algorithms for peak detection in time-series, Technical report, TRDDC, 2009.

Petrovic S., Osborne M., Lavrenko V., « Streaming First Story Detection with application to Twitter », *HLT-NAACL*, The Association for Computational Linguistics, p. 181-189, 2010.

Pustejovsky J., Lee K., Bunt H., Romary L., « ISO-TimeML : An International Standard for Semantic Annotation », *in* E. L. R. A. (ELRA) (ed.), *Proceedings of the 7th International Conference on Language Re- sources and Evaluation (LREC'10)*, Valletta, Malta, May, 2010.

Russell R., « Soundex coding system », *United States Patent*, 1918.

Sakaki T., Okazaki M., Matsuo Y., « Earthquake shakes Twitter users : real-time event detection by social sensors », *Proceedings of the 19th international conference on World wide web*, ACM, p. 851-860, 2010.

Shamma D. A., Kennedy L., Churchill E., « Statler : Summarizing media through short-message services », *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work (CSCW10)*, p. 551-552, 2010.

Sproat R., Black A., Chen S., Kumar S., Ostendorf M., Richards C., « Normalization of non-standard words », *Computer Speech & Language*, vol. 15, n° 3, p. 287-333, 2001.

Sutton J., Palen L., Shklovski I., « Backchannels on the front lines : Emergent uses of social media in the 2007 southern California wildfires », *Proceedings of the 5th International ISCRAM Conference*, Washington, DC, p. 624-632, 2008.

Zhang K., Zi J., Wu L., « New event detection based on indexing-tree and named entity », *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, p. 215-222, 2007.