Département de géomatique appliquée
Faculté des lettres et sciences humaines
Université de Sherbrooke

Modélisation spatio-temporelle pour la détection d'événements de sécurité publique à partir d'un flux Twitter

Donald Boileau

Mémoire de Maître ès sciences géographiques (M.Sc.),
Cheminement Géomatique

Mars 2017

Identification du jury :

Directeur de recherche :

Prof. Goze Bertin Bénié, PhD., Professeur titulaire au Département de géomatique appliquée, Chercheur au Centre d'applications et de recherches en télédétection (CARTEL) Université de Sherbrooke.

Co-directeur de recherche :

Prof. Francis Fortin, Ph.D., Professeur adjoint, chercheur au Centre international de criminologie comparée et co-chercheur au réseau SERENE-RISC, École de criminologie, Université de Montréal.

Membres du jury :

Mickaël Germain, PhD., Professeur associé - Département de géomatique appliquée, Université de Sherbrooke, Campus Longueuil.

Goze Bertin Bénié, PhD., Professeur titulaire au Département de géomatique appliquée, Chercheur au Centre d'applications et de recherches en télédétection (CARTEL) Université de Sherbrooke.

Francis Fortin, Ph.D., Professeur adjoint, chercheur au Centre international de criminologie comparée et co-chercheur au réseau SERENE-RISC, École de criminologie, Université de Montréal.

Tony Brien, M.Sc., Criminologue et chef de section au Service de Police de Sherbrooke.

Résumé du projet

Twitter est un réseau social très répandu en Amérique du Nord, offrant aux autorités policières une opportunité pour détecter les événements d'intérêt public. Les messages Twitter liés à un événement contiennent souvent les noms de rue où se déroule l'événement, ce qui permet la géolocalisation en temps réel.

Plusieurs logiciels commerciaux sont offerts pour effectuer la vigie des réseaux sociaux. L'efficacité de ces outils pour les autorités policières pourrait être grandement améliorée avec un accès à un plus grand échantillon de messages Twitter, avec un tri préalable pour dégager les événements pertinents en moins de temps et avec une mesure de la fiabilité des événements détectés.

Ce mémoire vise à proposer une démarche afin de détecter, à partir du flux de messages Twitter, les événements de sécurité publique d'un territoire, automatiquement et avec un niveau de fiabilité acceptable. Pour atteindre cet objectif, un modèle informatisé a été conçu, basé sur les quatre composantes suivantes: a) la cueillette de tweets à partir de mots clés avec un filtrage géographique, b) l'analyse linguistique et l'utilisation d'un répertoire de rues pour déceler les tweets localisables et pour trouver leurs coordonnées à partir des noms de rue et de leur intersection, c) une méthode spatio-temporelle pour former des grappes de tweets, et d) la détection des événements en identifiant les grappes contenant au moins deux (2) tweets communs touchant le même sujet.

Ce travail de recherche diffère des articles scientifiques recensés car il combine l'analyse textuelle, la recherche et le géocodage de toponymes à partir d'un répertoire de noms de rue, la formation de grappes avec la géomatique et l'identification de grappes contenant des tweets communs pour détecter localement des événements de sécurité publique.

L'application du modèle aux 90 347 tweets cueillis dans la région de Toronto-Niagara au Canada a résulté en l'identification et la géolocalisation de 1 614 tweets ainsi qu'en la formation de 172 grappes dont 79 grappes d'événements contenant au moins deux (2) tweets touchant le même sujet, soit un taux de fiabilité de 45,9 %.

Mots-clés : détection d'événement, analyse de grappes, Twitter, géolocalisation, sécurité publique, vigie des réseaux sociaux, toponymes.

Project abstract

Twitter is a social media that is very popular in North America, giving law enforcement agencies an opportunity to detect events of public interest. Twitter messages (tweets) tied to an event often contain street names, indicating where this event takes place, which can be used to infer the event's geographical coordinates in real time.

Many commercial software tools are available to monitor social media. The performance of these tools could be greatly improved with a larger sample of tweets, a sorting mechanism to identify pertinent events more quickly and to measure the reliability of the detected events.

The goal of this master's thesis is to detect, from a public Twitter stream, events relative to public safety of a territory, automatically and with an acceptable level of reliability. To achieve this objective, a computer model based on four components has been developed: a) capture of public tweets based on keywords with the application of a geographic filter, b) natural language processing of the text of these tweets, use of a street gazetteer to identify tweets that can be localized and geocoding of tweets based on street names and intersections, c) a spatio-temporal method to form tweet clusters and, d) event detection by isolating clusters containing at least two tweets treating the same subject.

This research project differs from existing scientific research as it combines natural language processing, search and geocoding of toponyms based on a street gazetteer, the creation of clusters using geomatics and identification of event clusters based on common tweets to detect public safety events in a Twitter public stream.

The application of the model to the 90,347 tweets collected for the Toronto-Niagara region in Ontario, Canada has resulted in the identification and geocoding of 1,614 tweets and the creation of 172 clusters from which 79 event clusters contain at least two tweets having the same subject showing a reliability rate of 45.9 %.

Keywords : event detection, cluster analysis, Twitter, geocoding, public safety, social media monitoring, toponyms.

Liste des figures

Liste des tableaux

Glossaire

Terme	Définition
Bigramme	Une séquence de deux éléments adjacents d'une chaîne d'entités lexicales qui sont typiquement des lettres, des syllabes ou des mots.
Centroïde	Point fictif situé à l'intérieur d'un polygone (ou d'une ligne) et dont les coordonnées correspondent généralement au centre de ce polygone.
Decahose	Le flux *Decahose* fournit un échantillon aléatoire de 10% du flux complet en temps réel *Firehose*.
Doublon	Une information redondante, c'est-à-dire elle est présente en double (voire plus) de manière inutile.
Événement	Quelque chose qui arrive à un moment et un endroit spécifiques avec toutes les conditions nécessaires et des conséquences inévitables.
Firehose	Le *Firehose* Twitter est un accès technique aux plus de 500 millions de tweets publiés quotidiennement sur Twitter. Cet accès technique était initialement réservé à quelques très rares acteurs qui revendaient les données aux prestataires et plateformes de social media intelligence sous un accord de licence avec Twitter.
Géocodage	À partir du (des) nom(s) de rue trouvés dans un tweet, recherche de la géolocalisation (latitude, longitude) avec un outil tel que l'API de Google Maps.
Géolocalisation	Coordonnées de latitude et de longitude d'un tweet.
Geonames	GeoNames est une base de données géographiques gratuite et accessible par Internet sous une licence *Creative Common*s. La base de données1 contient plus de 8 millions de noms géographiques qui correspondent à plus de 6,5 millions de lieux existants. La latitude et la longitude sont disponibles pour chaque emplacement.

Grappe	Regroupement d'au moins deux (2) tweets conformes aux paramètres de délai et de distance.
Grappe significative ou grappe d'événement	Une grappe comptant au moins deux (2) tweets portant sur le même sujet (aussi appelée grappe d'événements).
Lemmatisation	La lemmatisation désigne l'analyse lexicale du contenu d'un texte regroupant les mots d'une même famille. Chacun des mots d'un contenu se trouve ainsi réduit en une entité appelée lemme (forme canonique). La lemmatisation regroupe les différentes formes que peut revêtir un mot, soit : le nom, le pluriel, le verbe à l'infinitif, etc.
Taux de fiabilité	Le nombre de grappes d'événement comptant au moins deux (2) tweets portant sur le même sujet par rapport au nombre total de grappes détectées.

Acronymes

Terme	Définition
API	API est un acronyme pour *Applications Programming Interface*. Une API est une interface de programmation qui permet de se « brancher » sur une application pour échanger des données. Une API est ouverte et proposée par le propriétaire du programme.
JSON	JSON ou *JavaScript Object Notation*, est un format de données textuelles dérivé de la notation des objets du langage JavaScript. Il permet de représenter de l'information structurée comme le permet XML par exemple. Un document JSON a pour fonction de représenter de l'information accompagnée d'étiquettes permettant d'en interpréter les divers éléments, sans aucune restriction sur le nombre de celles-ci.
LSH	Locality sensitive hashing (LSH) est une méthode de recherche approximative dans des espaces de grande dimension. C'est une solution au problème de la malédiction de la dimension qui apparait lors d'une recherche du plus proche voisin en grande dimension.
NER	La reconnaissance d'entités nommées (*Named Entity Recognition* ou *NER)* est une sous-tâche de l'activité d'extraction d'information dans des corpus documentaires. Elle consiste à rechercher des objets textuels (c'est-à-dire un mot, ou un groupe de mots) catégorisables dans des classes telles que noms de personnes, noms d'organisations ou d'entreprises, noms de lieux, quantités, distances, valeurs, dates, etc.
NLP	NLP *(Natural Language Processing*) est une discipline à la frontière de la linguistique, de l'informatique et de l'intelligence artificielle, qui concerne l'application de programmes et techniques informatiques à tous les aspects du langage humain.
TF-IDF	Le TF-IDF (de l'anglais *Term Frequency- Inverse Document Frequency*) est une méthode de pondération qui permet d'évaluer l'importance d'un terme contenu dans un document, relativement à une collection ou à un corpus. Le poids augmente proportionnellement au nombre d'occurrences du mot dans le document. Il varie également en fonction inverse de la fréquence du mot dans le corpus.

Dédicace et remerciements

Je dédie ce mémoire à ma conjointe Louise Boucher pour le soutien et les sacrifices consentis sans lesquels ce travail ne serait jamais réalisé. Le présent mémoire de maîtrise est le résultat de multiples efforts déployés et de sacrifices consentis par bon nombre de personnes. Je dois la conduite et l'aboutissement de ce projet à mes directeurs de recherche : Prof. Goze B. Bénié et Prof. Francis Fortin pour leur suivi et pour leur effort d'encadrement.

Mes remerciements vont à l'endroit des membres du Jury : monsieur Tony Brien, criminologue et chef de section au Service de Police de Sherbrooke, Professeur Mickaël Germain, Professeur Goze B. Bénié et Professeur Francis Fortin pour avoir accepté d'évaluer ce travail.

Je manifeste ma gratitude à l'endroit du Dr. Yves Voirin pour m'avoir appris l'outil de programmation python qui a servi au traitement des données et aux professeurs Richard Fournier et Norm O'Neill pour m'avoir guidé dans la modélisation spatiale. Mes remerciements s'adressent également à Tony Brien du Service de Police de Sherbrooke pour m'avoir aiguillonné vers la recherche touchant les réseaux sociaux.

J'adresse mes remerciements à tous les autres enseignants du Département de géomatique appliquée et à toute l'administration de la Faculté pour leurs différents enseignements et services rendus dans le cadre de ce projet.

TABLE DES MATIÈRES

CHAPITRE 1 : Cadre théorique

1.1 Mise en contexte

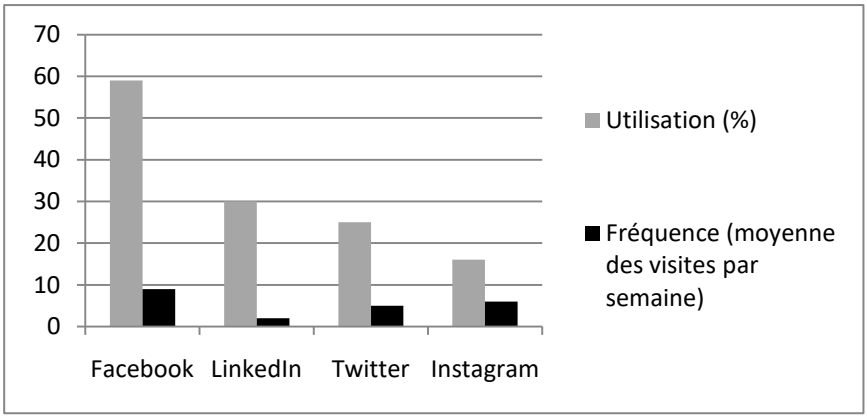
Depuis le lancement de la plateforme Facebook en 2004 et l'introduction du téléphone intelligent vers 2007, les réseaux sociaux ont connu une croissance phénoménale au Canada et dans le monde entier. Selon une enquête de Forum Research effectuée au Canada en janvier 2015 (Forum Research, 2015), Facebook est utilisé par 59 % des Canadiens âgés de 18 ans et plus tandis que les plateformes LinkedIn et Twitter sont visitées par 30 % et 25 % des adultes canadiens respectivement. Plusieurs autres réseaux comme YouTube, Instagram, Flicker, Pinterest et Snapchat sont aussi utilisés par des millions d'usagers à travers le Canada et le monde entier. La Figure 1 affiche le taux d'utilisation et le nombre de visites par semaine recensées par l'enquête pour les principaux réseaux au Canada.

Étant donné l'ampleur de ce phénomène, les services de police canadiens ont graduellement intégré les réseaux sociaux dans leurs activités quotidiennes pour communiquer avec les citoyens, effectuer une vigie des messages touchant leur territoire et enquêter sur des personnes d'intérêt. Les activités d'enquête occupent une grande partie de leur utilisation des réseaux sociaux : plus de 80 % des policiers canadiens et américains sondés par la firme Nexis en 2014 (LexisNexis, 2014) rapportent consulter les réseaux sociaux pour des fins d'enquête tandis que seulement 12 % déclarent les utiliser pour d'autres activités dont la vigie du territoire.

Cette vigie s'effectue surtout à l'aide de logiciels commerciaux fonctionnant via l'Internet. Plus de 230 fournisseurs (Burbary, 2016) offrent une solution commerciale permettant d'afficher les messages à partir d'une recherche basée sur la localisation du message, des mots-clés ou le réseau d'un usager, d'effectuer le suivi de comptes de personnes d'intérêt et d'afficher la provenance de ces messages sur une carte géographique.

La vigie des réseaux sociaux est nécessaire pour prendre connaissance des activités pertinentes à la sécurité publique sur le territoire et pour prévenir les actions qui pourraient mettre en danger les individus et les biens privés et publics. Les dirigeants des

services de police reconnaissent ces enjeux mais sont encore hésitants à assigner davantage de ressources pour leur faire face.



Tiré de Forum Poll™, Janvier 2015

Figure 1- Les réseaux sociaux au Canada en 2015

Lors d'un récent sondage mené par l'association internationale des chefs de police auprès de ses membres (IACP, 2015), 76,6 % des 553 répondants déclaraient être assez ou très préoccupés par l'utilisation criminelle des réseaux sociaux mais 80,2 % rapportent que moins de 10 heures par semaine sont investies par leur personnel pour les enquêtes et la vigie des réseaux sociaux. Au total, 32,7 % des répondants ont déclaré utiliser une application commerciale reliée aux réseaux sociaux et 71,2 % mettent à profit les informations disponibles sur Twitter.

Plusieurs facteurs limitent l'efficacité des outils commerciaux dans ce domaine. D'abord, ils sont principalement conçus pour surveiller l'impact du marketing d'une marque de commerce ou celle d'une entreprise. De plus, pour la vigie générale du territoire, ils utilisent surtout les messages qui sont auto-localisés, c'est-à-dire ceux où l'usager a accepté de transmettre sa localisation en latitude et en longitude lors de l'envoi du message. Selon Ajao *et al*. (2015), seulement 0,5 % des usagers de Twitter activent cette fonction d'auto-localisation sur leur appareil numérique. Ces contraintes réduisent donc de beaucoup la portée de la vigie sur le territoire.

De plus, les messages captés par ces outils contiennent énormément de textes non-pertinents, étant donné que l'on fournit tous les messages contenant au moins un des

mots-clés demandés; beaucoup de temps est donc requis à l'analyste chargé de la vigie pour identifier les messages pertinents.

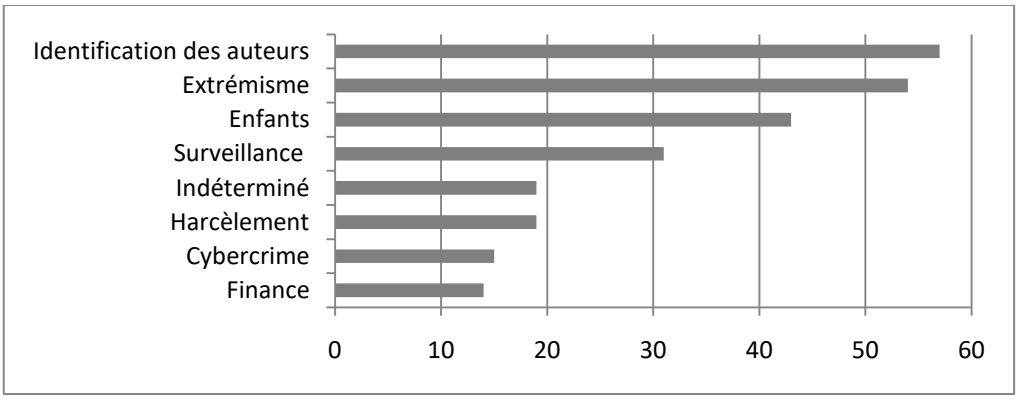
Les résultats obtenus par la recherche scientifique depuis 2010 permettent de transformer cette vigie en détection d'événements à partir des données disponibles sur les réseaux sociaux tout en améliorant grandement l'efficacité du processus de vigie.

1.2 État des connaissances

Des milliers d'articles ont été publiés sur le thème du crime, de la police et des forces de l'ordre, mais un nombre très limité touche l'analyse automatique des données en ligne ou la détection d'événement. Lors d'une enquête effectuée par Edwards *et al.* (2015), trois critères de sélection ont été appliqués aux 13 246 articles trouvés sur ce thème : a) l'utilisation de données disponibles en ligne, b) l'application de processus automatisés de traitement de données, et c) la concentration du sujet sur la criminalité. Parmi les 206 articles répondant à ces trois critères, seulement 31 articles (15 %) touchaient la surveillance des données en ligne ou la détection d'événement. La Figure 2 affiche la distribution de ces 206 articles par sujet.

Devant cette rareté d'articles alliant le thème du crime et des forces de l'ordre à la surveillance des données en ligne, notre revue de littérature doit être élargie à l'ensemble du domaine de la détection d'événement sans égard au type d'événement visé.

La majorité des articles scientifiques étudiant la détection d'événements puisent leurs données à partir du flux Twitter car, en plus d'être utilisé par plus de 320 millions de personnes et de constituer un médium facilitateur pour la transmission de nouvelles, ce réseau fournit un outil de requête public donnant accès aux messages de ses usagers.



Tiré de ACM Computing Surveys 2015 [4]

Figure 2- Répartition des articles par sujet

1.2.1 Le flux Twitter

Twitter met plusieurs flux à la disposition des usagers qui désirent analyser et tirer avantage des nombreux messages circulant sur son réseau. Pour les fournisseurs d'outils commerciaux de vigie, il est possible d'obtenir un accès à 100 % du flux avec le *Firehose* ou à 10 % du flux avec le *Decahose,* moyennant un forfait mensuel. Depuis 2015, ces flux sont seulement disponibles à la société Gnip contrôlée par Twitter qui exerce un contrôle assez sévère sur le type d'organisation qui demande le service et sur l'utilisation des données obtenues.

La grande majorité des articles recensés utilisent le flux public de Twitter qui représente un échantillon de 1 % du flux total du *Firehose*. Mais la recherche par mot-clé peut résulter en la cueillette d'une partie plus grande que cette balise de 1 %. Par exemple, si le flux total sur Twitter au moment d'une requête est de 350 000 tweets à la minute et que le nombre total de tweets contenant le mot-clé est de 2 000 soit 0,6 %, tous ces tweets seront transmis au demandeur à ce moment.

L'API public de Twitter peut fonctionner selon deux modes : la requête ponctuelle de type *REST* et la requête en continu de type *streaming*. Dans cette étude, nous nous intéresserons au mode *streaming* étant donné que l'objectif ultime est l'exercice d'une vigie du réseau social en mode continu.

En mode *streaming*, trois catégories de requête sont disponibles : la requête géographique, la requête par usager et la requête par mot-clé. La requête géographique est exécutée en fournissant la latitude et la longitude du coin gauche inférieur et du coin droit supérieur d'un rectangle de capture. Tous les tweets se trouvant à l'intérieur du polygone où l'usager a accepté de signaler son positionnement (tweets auto-localisés) sont alors transmis en temps réel.

La requête par usager peut s'appliquer à plusieurs usagers simultanément et fournit les messages émis par ces derniers. La requête avec mot-clé peut contenir jusqu'à 400 mots-clés; les tweets émis à travers le monde contenant un de ces mots-clés sont alors transmis en continu en autant que leur nombre ne dépasse pas 1 % du flux total en cours.

Morstatter *et al.* (2013) ont comparé les données obtenues de l'API *Streaming* et du *Firehose* avec chacune des trois catégories de requête pendant une période de 28 jours. Ils ont cueilli 528 592 tweets et 1 280 344 tweets respectivement avec l'API *Streaming* et le *Firehose*. Durant cette période ils ont noté que l'API *Streaming* recevait, en moyenne, 43,5 % des tweets disponibles via le *Firehose* pendant une même journée, ce qui dépasse de beaucoup la balise de 1 %.

Les données transmises par Twitter en réponse à une requête sont retournées en format JSON; ce format peut facilement être traité avec le langage Python pour extraire les informations des tweets obtenus.

1.2.2 Perspective sur la recherche scientifique

Étant donné que nous nous intéressons à la détection d'événements pour les services policiers surtout au niveau municipal, la localisation exacte des événements et leur occurrence dans le temps revêtent une importance primordiale. De plus, il importe de pouvoir corroborer un tweet décrivant un événement avec au moins un autre tweet différent préférablement en provenance d'un autre usager. Ces contraintes ont donc orienté notre recension de la recherche scientifique actuelle. La Figure 3 illustre un schéma qui décrit l'état des connaissances scientifiques pour la détection d'événements basé sur le flux Twitter.

Les articles recensés utilisent un ensemble de mots-clés (événement non-spécifié) ou un titre d'événement spécifique (événement spécifié) pour cueillir les tweets avec l'API public de Twitter. Dans le premier cas, le modèle parcourt le flux Twitter à la recherche de tweets contenant un des mots-clés recherchés et décèle les événements selon l'algorithme utilisé.

Dans le cas d'un événement spécifié, le titre de l'événement est saisi comme intrant principal du modèle et celui-ci cueille et traite tous les tweets pour en faire ressortir des informations spécifiques aux besoins de l'article scientifique. Les différents traitements de détection pour ce dernier type de requête sont mis en relief dans la Figure 3 à l'aide de cases pointillées.

La Figure 3 illustre aussi que la plupart des articles touchant la détection traitent un seul des trois types de données suivants: le texte du tweet, l'auto-localisation de l'usager (latitude et longitude) et les autres contenus audio-visuels du tweet (photo, vidéo, audio). Dans le cadre de ce mémoire, l'analyse des contenus audio-visuels ne sera pas considérée.

Par contre, près de 65% des articles recensés utilisent l'analyse du texte du tweet (n=31 sur un total de 48); ceux-ci se subdivisent en deux catégories soient l'analyse linguistique (NLP) et l'analyse de localisation. Ces sous-catégories sont traitées dans la section 1.2.3.1.

Plus de 31% des articles recensés (n=15) font appel à l'auto-localisation par l'usager (latitude et longitude) pour leurs données d'analyse. Deux catégories se dégagent par le type de modélisation utilisé avec ces données : la modélisation géo-spatiale et la modélisation spatio-temporelle. La section 1.2.3.2 touchant l'auto-localisation traite plus en détail chaque méthode de modélisation.

Enfin, quelques articles ont été retenus sur le développement de systèmes dédiés à la détection d'événements à partir du flux Twitter pour la gestion de crise par les agences publiques. Nous traiterons ceux-ci dans la section 1.2.3.3 intitulée 'Systèmes scientifiques de vigie pour la sécurité publique'.

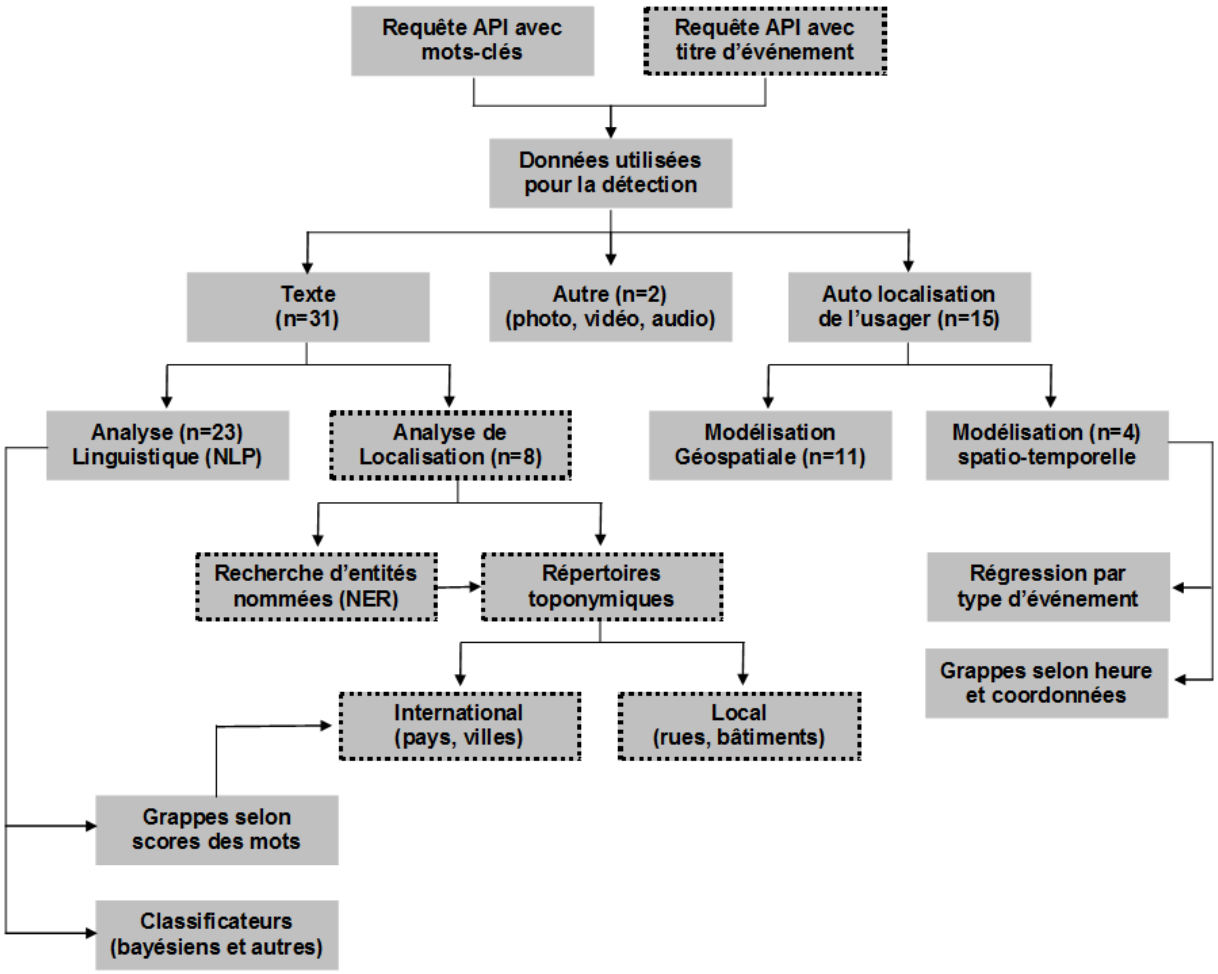


Figure 3- Schéma d'état des connaissances
(n=nombre d'articles recensés)

1.2.3 Revue des articles scientifiques

1.2.3.1 Détection à partir du texte du tweet

Tel que mentionné dans la synthèse de la littérature touchant l'analyse des médias basée sur les événements effectuée par Tzelepis *et al.* (2015), la majorité des études traitant la détection d'événements sociaux utilisent des détecteurs d'événements basés sur l'information textuelle des flux des médias sociaux.

De façon similaire à la détection d'événements à partir de médias conventionnels tels les services de nouvelles publiées sur l'Internet, la plupart des techniques pour la détection

d'événements non-spécifiés à partir du flux de Twitter dépendent d'approches de regroupement en grappes ou *clustering*. Parmi les premiers à utiliser le *clustering*, en 2002, E. Lim a étudié la fréquence rapprochée des messages et leur texte pour former des grappes; en 2009, J. Sankaranarayanan, a développé un algorithme permettant de traiter le texte des tweets pour en dégager des grappes partageant des caractéristiques textuelles communes. M. Cheong et V. Lee, ont détecté des comportements répétitifs en étudiant les textes des usagers de Twitter et en les regroupant en grappes.

Ces approches sont bien adaptées à la détection d'événements non-spécifiés car elles ne sont pas dirigées et ne requièrent pas de données préalables pour entraîner le modèle. De plus, elles augmentent la fiabilité de détection, car on accumule plusieurs messages pour confirmer un même événement. Néanmoins, elles présentent des défis importants quant à la localisation exacte de l'événement.

1.2.3.1.1 Analyse linguistique du texte

La détection d'événement non-spécifiés, où les données sont cueillies à l'aide d'une liste de mots-clés et traitées avec l'analyse linguistique du texte du tweet domine la littérature du domaine en accaparant près de 65% des articles recensés tel que décrit dans la Figure 3. De nombreux articles utilisent l'analyse des mots selon les techniques NLP (*Natural Language Processing*) puis appliquent une méthode TF-IDF (*Term Frequency-Inverse Document Frequency*) qui donne un score à chaque nom, verbe et adjectif d'un tweet par rapport à sa fréquence dans le message et à sa rareté dans un corpus de tweets de référence. Dans leur enquête sur les techniques de détection d'événement basées sur Twitter, Atefeh et Khreich (2015) énumèrent et comparent plusieurs articles utilisant cette méthode de détection (voir le Tableau 1).

Les tweets sont regroupés par grappe selon le vecteur formé par le score de chaque mot pour ensuite être classifiés. On recherche ensuite, dans les champs des tweets faisant partie d'une grappe, des éléments géographiques (nom de ville, coordonnées du tweet ou toponymes dans le texte du message) permettant la localisation au niveau d'un pays ou d'une ville.

Atefeh et Khreich (2015) ont recensé 16 articles utilisant l'analyse linguistique du texte et les grappes pour détecter les événements avec le flux de Twitter. Le Tableau 1 indique les sept articles utilisés par les auteurs pour qualifier les techniques traitant les événements nouveaux de type non-spécifié selon une méthode non-dirigée. On remarque dans la colonne 'Application' qu'ils touchent uniquement la détection générale d'un événement ou la détection de dernières nouvelles : aucune application ne concerne le domaine policier ou la sécurité publique.

Tableau 1- Articles traitant les événements nouveaux

Tiré de Computational Intelligence,
Atefeh et Khreich (2015)

Références	Application
Sankaranarayanan et al. (2009)	Détection de dernières nouvelles
Phuvipadawat et Murata (2010)	Détection de dernières nouvelles
Petrovic et al. (2010)	Détection générale d'événement non-connu
Becker et al. (2011)	Détection générale d'événement non-connu
Long et al. (2011)	Détection générale d'événement non-connu
Weng et Lee (2011)	Détection générale d'événement non-connu
Cordiro (2012)	Détection générale d'événement non-connu

Dans un article de cette enquête qui résume bien cette démarche, Sankaranarayanan *et al.* (2009) utilisent le flux Twitter pour détecter des nouvelles au niveau mondial. Ils regroupent automatiquement les tweets faisant référence à des nouvelles en des grappes de tweets afin que chaque grappe contienne les tweets concernant un sujet spécifique.

Utilisant différentes façons de cueillir les tweets dont une liste de 40 mots-clés, les auteurs appliquent un algorithme de traitement en ligne pour détecter les tweets contenant des nouvelles et les regrouper dans des grappes ayant des mots communs ou similaires. Pour séparer les tweets contenant des nouvelles des autres types de tweets, cet algorithme fait appel à un classificateur naïf bayésien qui traite les mots du message et à deux corpus de tweets (nouvelles et autres types d'information) pour entraîner le classificateur. La localisation des tweets est effectuée à l'aide d'une analyse de reconnaissance d'entités (*Named Entity Recognition)* portant sur le texte du tweet afin de déceler les toponymes ou

noms de lieu. Enfin, la géolocalisation approximative (latitude et longitude) des tweets est obtenue en passant le toponyme identifié dans le texte de tweets dans le répertoire toponymique international, GeoNames, qui permet de situer le pays ou la ville du tweet. Les auteurs illustrent une carte géographique du monde démontrant que les tweets se rapportant à une même grappe ou événement sont largement diffus.

Shakira et Abdolreza (2014) appliquent une méthode similaire en la bonifiant avec la technique LSH (*Locality Sensitive Hashing)* pour un événement spécifié lors de la requête de cueillette de tweets : la 'fin du monde' (*End of world*). Dans leur article, ils effectuent une analyse détaillée des grappes obtenues en traitant le million de tweets cueillis durant quatre jours en mai 2011. La qualité d'une grappe est mesurée par son taux de pureté, soit le nombre de tweets (ou documents) correctement assignés à la grappe (i.e. ayant le même sujet) divisé par le nombre total de tweets de la grappe. Par exemple, une grappe formée de cinq tweets dont un des tweets n'a aucun rapport aux quatre autres aurait un taux de pureté de 0,8 (4/5 x 100). Rappelons que le mot-clé utilisé dans la requête (*End%of%World*) augmente beaucoup la probabilité qu'un tweet soit bien assigné à une grappe, étant donné que chaque tweet traité contient au moins ces trois mots.

Dans ce même article, les grappes obtenues avec la technique LSH sont comparées à celles de la technique de *clustering* K-Moyennes (*K-Means*) avec un taux de pureté moyen de 0,75 et 0,65 respectivement. La taille moyenne d'une grappe durant la journée la plus achalandée, le 21 mai, se situe à 47 tweets par grappe pour un total de 22 grappes identifiées (tiré de la figure 8 de l'article). La localisation des grappes était impossible à effectuer étant donné que les tweets formant chaque grappe provenaient de partout sur le globe et que seulement une infime minorité des tweets pouvaient être localisés car seuls les tweets auto-localisés étaient utilisables. Notons que la plupart des articles pour la détection d'événements non-spécifiés appliquant la méthode NLP *(Natural Language Processing)* avec formation de grappes, ont une portée mondiale et non locale, dû à leur algorithme de traitement basé uniquement sur les caractéristiques du texte ce qui limite considérablement la localisation précise d'une grappe.

1.2.3.1.2 Analyse de localisation à partir du texte

Contrairement aux articles précédents faisant appel aux grappes d'événements, les articles qui visent à localiser les tweets à partir du texte sont basés sur la détection d'événements spécifiés où les données sont cueillies à partir d'une requête axée sur le titre d'un événement. Ils utilisent surtout la recherche de toponymes dans le texte du tweet pour localiser l'événement et pour effectuer une analyse spatiale des données.

Gu *et al.* (2014) utilisent un classificateur naïf bayésien pour traiter le texte des messages et un répertoire toponymique pour la géolocalisation des tweets afin de détecter les incidents de route tels que les accidents, les blocages de circulation et les travaux de voirie dans les villes de Pittsburg et de Philadelphie. Ils ont trouvé que l'échantillon de tweets obtenu de Twitter couvrait la plupart des incidents réels qui ont eu lieu dans ces villes. La technique de géocodage utilisée, combinée avec un répertoire local de rues, a résulté en un taux de géocodage de 4,9 % des tweets traités.

Dans le même article de Gu *et al.* (2014), la plupart des tweets détectés provenaient d'usagers d'influence tels que les médias de nouvelles et les agences publiques comme les départements de transport des États. De plus, une tendance nette a été décelée au niveau hebdomadaire avec un plus grand nombre de tweets émis la fin de semaine et durant le jour. Enfin, plus d'activités Twitter ont été notées dans le centre-ville qu'en banlieue éloignée, ce qui s'explique par la plus grande utilisation du médium en milieu densément peuplé.

Afin d'évaluer les méthodes d'inférence géographique à partir du texte d'un tweet, Jurgens *et al.* (2015) ont appliqué neuf différentes méthodes à un échantillon de 1,3 milliard de tweets et ont géolocalisé ceux-ci avec l'aide de quatre répertoires toponymiques dont Google Places et GeoNames. Avec GeoNames, ils ont atteint un taux de géocodage de 3,4 % sur l'ensemble des tweets traités.

Dans l'un des articles combinant les données de Twitter et les données criminelles, Wang *et al.* (2012) démontrent qu'ils peuvent prédire sommairement les délits de fuite pour la

ville de Charlottesville en Virginie à l'intérieur d'un intervalle de confiance assez permissif, à partir du flux Twitter d'une station de télévision locale cueilli sur une période de six mois en 2011. Pour atteindre leur but, ils utilisent un algorithme de traitement linguistique appelé *Semantic Role Labeling (SRL)* qui extrait les événements mentionnés dans les tweets, les entités impliquées dans les événements et les rôles des entités par rapport aux événements. Celui-ci est appliqué au texte de chaque tweet pour en dégager une liste de sujets puis un modèle prédictif est exécuté à l'aide de la régression linéaire.

D'autres articles visant des événements spécifiés révèlent que l'usager mentionne souvent un lieu précis lorsqu'il émet un tweet touchant un événement; Vieweg *et al.* (2010) ont analysé le flux Twitter durant les inondations de la Red River et des feux de forêt de l'Oklahoma en 2009. Les auteurs ont conclu que la plupart des usagers locaux de Twitter citaient un emplacement dans au moins un tweet relatif à un événement; de même, les informations sur la localisation différaient selon la nature d'un événement de crise et l'étape où celui-ci se trouvait dans le cycle d'une crise. Hormis les toponymes tels que les noms de comté et de ville, 24 % des tweets d'Oklahoma et 12 % des tweets de la Red River contenaient des noms de route, de place, de rue et des adresses.

Gelernter et Balaji (2013) ont sélectionné 2 000 tweets liés à un tremblement de terre ayant eu lieu à Christchurch en Nouvelle Zélande et aux feux de forêt d'Austin au Texas en 2011. Utilisant un algorithme basé sur un arbre décisionnel, ils ont trouvé que 28,7 % des tweets contenaient un nom de rue ou un toponyme dans le texte.

1.2.3.2 Détection à partir de l'auto-localisation de l'usager

On peut répartir la recherche sur la détection d'événements utilisant les tweets auto-localisés en deux groupes principaux selon la méthode utilisée: l'analyse de la localisation en utilisant des outils géo-spatiaux et la formation de grappes à partir de tweets auto-localisés selon la proximité en temps et en distance (méthode spatio-temporelle).

À l'aide de la modélisation géo-spatiale et de la régression statistique, Gerber (2014) a tenté de prédire les incidents criminels à Chicago en utilisant les tweets auto-localisés du flux Twitter et en les comparant aux points chauds (*hotspots)* des crimes réels identifiés

sur ce territoire durant la même période. L'analyse linguistique des messages des tweets a été utilisée pour les catégoriser par sujet relié à un type de crime. Pour les 25 types de crime traités, la modélisation des tweets auto-localisés a permis d'améliorer la prédiction de crimes pour 19 types de crime; les crimes liés au vol et au trafic de drogues ont obtenu la meilleure performance prévisionnelle.

Comme Gerber, les autres articles traitant de la détection d'événements dans le domaine policier utilisent les tweets auto-localisés. Bendler *et al.* (2014) ont cartographié les tweets à San Francisco afin de savoir si la dimension spatio-temporelle des tweets peut expliquer les différents types d'incidents criminels. En classifiant les quartiers de la ville selon le type d'activité (zone de restaurants, centre commercial, terrains de jeu, etc.) et en comparant le volume de tweets avec les statistiques criminelles de chaque quartier par type, ils ont identifié quatre types de crime pouvant être prédits à partir du flux Twitter avec un niveau de confiance acceptable : les vols par effraction, les vols de véhicules moteur, les vols qualifiés et les vols mineurs.

Quelques articles regroupent les grappes selon une méthode spatio-temporelle. Cheng et Wicks (2014) utilisent la technique STSS (*Space Time Scan Statistics)* qui recherche les grappes dans un ensemble de tweets auto-localisés à travers le temps et l'espace, peu importe le contenu du message. La détection d'événement est basée sur le volume plus grand de tweets émis autour du site de chaque événement, durant une courte période, par les gens qui en sont témoins. Les auteurs vérifient leur méthode en détectant l'écrasement d'un hélicoptère survenu à Londres en janvier 2013.

La technique DBSCAN (*Density Based Spatial Clustering of Applications with Noise*) proposée par Martin Ester en 1996 a été adaptée et appliquée par Chung-Hong (2012) au flux Twitter. La technique utilise le texte, l'heure et les coordonnées des tweets pour détecter avec succès des grappes de tweets ayant diverses caractéristiques spatio-temporelles pour quatre événements : les embouteillages routiers, le tournoi de tennis US Open en 2013, les inondations et la coupe mondiale de soccer en 2014.

Au lieu d'employer la technique TF-IDF pour traiter les tweets auto-localisés, l'auteur fait appel aux rafales (*bursts*) qui sont définies comme un nombre non habituel de

messages émis durant une courte période. L'attribution d'un score ou d'une pondération selon la fréquence d'apparition d'un mot dans une courte période permet de former des grappes de tweets. Nous avons aussi recensé quatre autres articles employant l'algorithme DBSCAN de façon similaire pour détecter des événements à l'aide des grappes spatio-temporelles.

1.2.3.3 Systèmes scientifiques de vigie pour la sécurité publique

Contrairement aux articles décrits plus haut qui se concentrent uniquement sur les méthodes de détection d'événement sans en dégager une application qui pourrait servir dans un milieu particulier, nous avons relevé quelques articles qui présentent un système de vigie du flux Twitter.

Sycora *et al.* (2013) présentent le système de vigie *EMOTIVE* qui met l'accent sur la détection d'émotions à partir du texte des tweets. Ce système situe les tweets à partir de l'auto-localisation de l'usager ou des toponymes trouvés dans le texte avec le répertoire toponymique international GeoNames. Leur article décrit aussi cinq autres systèmes de vigie avec localisation sur le flux Twitter dont quatre sont énumérés au Tableau 2 (plus *EMOTIVE)* avec leurs caractéristiques principales. À l'exception de SENSEPLACE2, ces systèmes sont du type à événements non-spécifiés car ils cueillent leurs tweets à partir d'une liste de mots-clés saisis par l'usager.

Afin de bien cerner les besoins pour les usagers potentiels de leur système SENSEPLACE2, MacEachren *et al.* (2011) ont effectué un sondage auprès des membres de l'association internationale de gestionnaires de crise (*International Association of Crisis Managers*). Sur les 46 répondants, 39,1 % ont déclaré qu'ils ont utilisé Twitter ou d'autres outils liés aux microblogs pour amasser des informations du public dans le contexte d'une crise.

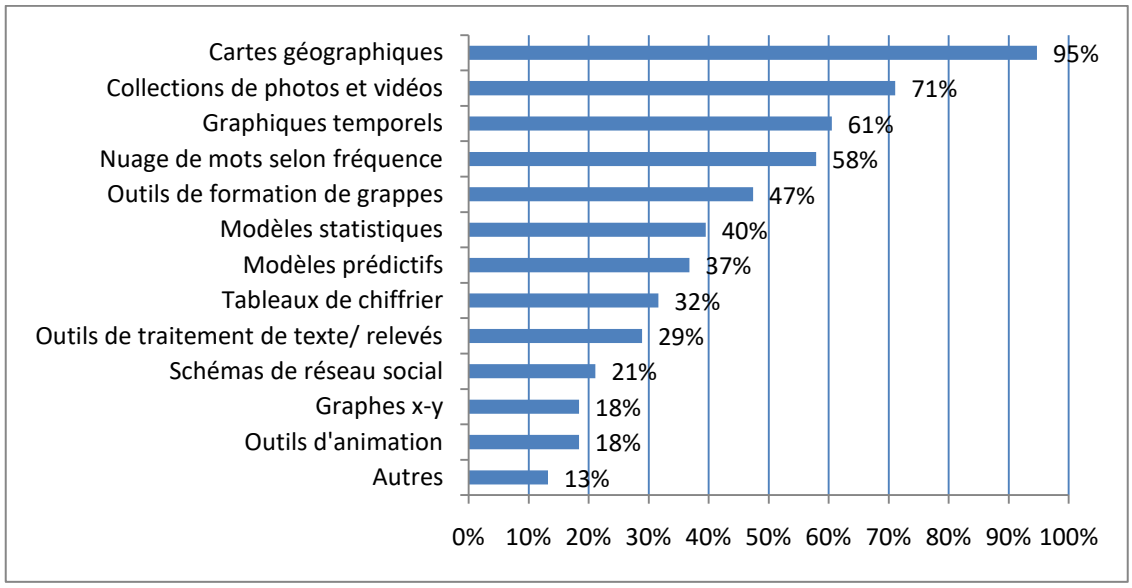
Tableau 2- Systèmes de vigie avec localisation basés sur Twitter

Système de vigie	Localisation des tweets	Organisation des résultats
SENSEPLACE2 MacEachren *et al.* (2011)	Tweets auto-localisés et noms de villes à partir du texte avec répertoire GeoNames	Classification selon sujet, heure et localisation Interface cartographique
CROSSTRACKER Rogstadius *et al.* (2011)	Tweets auto-localisés seulement	Formation de grappes à partir du score de mots Interface cartographique
CRISEES Maxwell *et al.* (2012)	Tweets auto-localisés seulement	Simple liste des tweets visés Interface cartographique
TWITCIDENT Abel *et al.* (2012)	Tweets auto-localisés et noms de villes à partir du texte avec NER et répertoire GeoNames	Classification par sujet, heure et localisation selon requête demandée
EMOTIVE Sycora *et al.* (2013)	Tweets auto-localisés et noms de villes à partir du texte avec NER et répertoire GeoNames	Accent sur la mesure des émotions identifiées dans le texte. Interface cartographique

La Figure 4 démontre l'importance que revêt la localisation des incidents et des tweets avec 95 % des répondants qui affirment que les cartes sont une des caractéristiques désirées d'une application interactive conçue pour incorporer les médias sociaux dans le processus de gestion de crise. De même, la dimension temporelle (graphiques temporels) est la troisième en importance avec 61 % de l'appui des répondants.

Il est intéressant de noter qu'aucun des cinq systèmes énumérés dans le Tableau 2 ne permet de localiser les tweets sur un territoire municipal à partir des toponymes contenus dans le texte. Au mieux, le répertoire GeoNames utilisé par trois des systèmes localise un tweet au niveau d'une ville ou d'une place très connue comme la tour Eiffel de Paris ou la statue de la liberté à New York.

Une autre enquête effectuée par Studies *et al. (2016)* auprès de 761 personnes intervenant dans les services d'urgence européens sur leurs attitudes envers les médias sociaux, a révélé les types d'informations partagées par le public sur les médias sociaux qui pourraient être utiles pour la gestion de crise. Une proportion de 73 % des répondants ont déclaré que les mises à jour de la situation générale seraient très utiles, accompagnées de photos et de vidéos localisées si possible.



Tiré de *SensePlace2: GeoTwitter Analytics Support for Situational Awareness*
MacEachren *et al.* (2011)

Figure 4- Caractéristiques désirées pour une application interactive

1.2.4 Sommaire de l'état des connaissances

En résumé, la vigie des réseaux sociaux préoccupe les chefs de police en Amérique mais peu de ressources sont allouées à cette activité (LexusNexus et IACP, 2015) qui gagnerait beaucoup en efficacité avec l'aide de la détection automatique d'événements touchant la sécurité publique à partir du flux Twitter.

Le Tableau 3 décrit les différentes méthodes recensées pour la détection d'événements à partir du flux Twitter avec leurs avantages et inconvénients par rapport à leur application aux besoins de vigie des réseaux sociaux d'un service de police municipal.

La détection avec l'analyse linguistique et la formation de grappes utilisant les scores de mots du texte de tweet, domine la recherche mais la dispersion des tweets formant les grappes permet rarement la localisation exacte de l'événement ce qui rend cette méthode peu intéressante pour la détection locale. Par contre, l'analyse détaillée de grappes formées avec cette méthode par Shakira et Abdolreza (2014) permet de fixer des balises quant à la taille et au nombre de grappes ainsi qu'à la proportion de celles qui contiennent des tweets pertinents au même événement.

Tableau 3- Avantages et inconvénients des méthodes de détection recensées

Méthode	Avantages	Inconvénients
Analyse linguistique du texte et formation de grappes (NLP)	• Grand nombre de grappes pertinentes	• Localisation imprécise des événements • Aspect temporel négligé
Modélisation géospatiale à partir de l'auto-localisation	• Fonctionne bien au niveau local • Précision géographique élevée	• Échantillon de tweets très limité localement • Peu d'articles utilisent les grappes pour la détection
Analyse de localisation à partir du texte	• Fréquence de mention de lieu élevée dans tweets d'événement • Localisation plus pertinente avec texte qu'avec auto-localisation par l'usager	• Utilisée surtout avec requête d'événements spécifiés • Niveau de localisation souvent insuffisant (villes)
Modélisation spatio-temporelle avec ou sans grappes	• Tient compte des 2 dimensions les plus importantes (temps et espace) • Application locale intéressante	• Utilise les tweets auto-localisés, donc échantillon limité • Emploi insuffisant du message du tweet

1.2.5 Importance de la localisation et de l'évolution temporelle

Dans le domaine de la sécurité publique, deux des plus importantes dimensions de la détection d'événement sont la localisation précise et l'évolution temporelle de chaque événement. Dans MacEachren *et al.* (2011) les gestionnaires de crise sondés ont confirmé cette évidence lorsqu'ils ont été invités à prioriser les caractéristiques d'un logiciel de détection via l'Internet : la cartographie et les graphiques temporels figuraient parmi les plus importantes caractéristiques.

Le texte de tweet peut être très utile pour localiser précisément un événement : Vieweg *et al.* (2010) ont conclu qu'un nom de lieu est souvent mentionné dans un tweet lors de la description d'un événement. Dans leur étude, 24 % des tweets cueillis en Oklahoma contenaient des noms de routes, de places, de rues et des adresses. De même, Gelernter et Balaji (2013) ont déclaré que 28,7 % des tweets cueillis contenaient un nom de rue ou un toponyme dans le texte.

Quelques articles tels que celui de Chung-Hong (2012) traitent de la dimension temporelle en mesurant la fréquence des mots dans le temps pour identifier des grappes de tweets et détecter des événements.

Gu *et al.*(2016) offrent une façon intéressante de localiser un incident à partir de Twitter par le géocodage à partir de noms de rue cités dans le texte des tweets. Les auteurs ne font pas appel à la technique des grappes pour détecter les accidents de la route mais obtiennent un taux de géocodage intéressant sur l'ensemble des tweets traités.

Quelques articles traitent des deux dimensions de temps et d'espace en utilisant les tweets auto-localisés. Cheng et Wicks (2014) emploient la technique STSS pour former des grappes spatio-temporelles afin de détecter un accident d'hélicoptère à Londres sans traiter le texte des tweets cueillis. Employant les algorithmes DBSCAN et BURST, Chunghong (2012) analyse le flux de tweets liés aux séismes de Christchurch en Nouvelle Zélande et du Japon en 2011 en comparant les résultats au niveau de la localisation et de la chronologie.

1.3 Problématique

Les services de police effectuant une vigie des réseaux sociaux utilisent présentement surtout des outils commerciaux qui dépendent davantage des tweets auto-localisés pour identifier des événements d'intérêt de sécurité publique. Déjà limitant étant donné que ces messages auto-localisés constituent environ 0,5 % du flux total de Twitter (Ajao *et al*., 2015), un enjeu additionnel est l'accès de ces fournisseurs aux données de Twitter pour des fins d'enquête.

Depuis avril 2015, les fournisseurs d'outils de vigie de réseaux sociaux ont vu leur accès au flux à grand débit de Twitter diminuer de façon importante suite à l'abolition du

réseau de revendeurs de données (TheNextWeb.com, 2015). Plus récemment, l'accès au flux à débit moyen offert à ces fournisseurs a finalement été discontinué suite aux pressions du groupe de protection de libertés civiles le plus influent, l'*American Civil Liberties Union* (Motherboard.com et RT.com, 2016).

Il appert donc essentiel de pouvoir tirer le maximum du flux public disponible gratuitement de Twitter qui représente un échantillonnage d'au moins 1 % du flux total du réseau tout en effectuant une vigie efficace d'un territoire. Les outils commerciaux actuels ne pourront atteindre cet objectif à partir d'un flux limité aux tweets auto-localisés, ce qui laissera les services de police municipaux avec des moyens assez limités.

La recherche scientifique fait avancer la vigie vers la détection automatique d'événements la rendant plus efficace mais il n'existe pas de méthode ou de systèmes directement applicables pour les services de police. Le Tableau 3 et l'état des connaissances présentés ci-haut mènent à conclure qu'aucune méthode recensée ne permet la détection adéquate d'événements, à partir des tweets cueillis en utilisant une requête d'événements non-spécifiés, sur le territoire relativement limité d'un service de police. Ou l'une des deux dimensions de temps et d'espace est négligée ou la précision de la localisation et de la détection est inadéquate.

Sycora *et al.* (2013), en plus de présenter leur système EMOTIVE, ont étudié quatre autres systèmes de vigie développés par les chercheurs afin de répondre aux besoins des gestionnaires de crise. Malheureusement, ces systèmes ne sont pas utiles pour les services de police desservant une municipalité étant donné qu'ils dépendent de tweets auto-localisés ou du répertoire international GeoNames pour détecter les événements.

Les chercheurs sont davantage intéressés dans la modélisation linguistique et la formation de grappes à partir du texte des tweets pour détecter des événements non spécifiés, ce qui diminue grandement la précision de la localisation de l'événement. Par contre, il appert de tirer avantage des résultats de recherche touchant la formation de grappes et des toponymes contenus dans le texte pour rassembler les tweets qui sont émis dans un espace et une période de temps rapprochés.

Ainsi, en combinant l'analyse de localisation du texte des tweets d'un flux de l'API public de Twitter, obtenu selon une requête d'événement non-spécifié c'est-à-dire avec un ensemble de mots-clés, avec une formation de grappes spatio-temporelles basées sur la distance et la période de temps entre les tweets, nous estimons être en mesure d'améliorer grandement la vigie des réseaux sociaux des services de police municipaux. L'analyse de localisation fera appel à la recherche et au géocodage de toponymes dans le texte du tweet.

Quelques-uns des articles recensés nous permettent d'établir des barèmes préliminaires d'évaluation qui peuvent agir comme cible pour ce mémoire. En particulier, le taux de géocodage de 4,9 % obtenu par Gu *et al.* (2016) à partir du texte des tweets cueillis et le nombre et la qualité des grappes formées par l'analyse linguistique NLP et la modélisation de Shakira et Abdolreza (2014).

1.4 Objectifs

Face à la problématique énoncée plus haut, ce mémoire vise à répondre à la question suivante :

> À partir du flux public de Twitter, est-il possible de détecter avec un niveau de fiabilité acceptable, les événements de sécurité publique d'un territoire de service de police en utilisant des grappes spatio-temporelles formées à partir du géocodage des toponymes tirés du texte des tweets?

L'objectif principal du mémoire est de détecter et de localiser précisément un nombre suffisant d'événements de sécurité publique d'un territoire à partir du flux de Twitter avec un taux de fiabilité acceptable. Ce taux se mesure avec le nombre de grappes comptant au moins deux (2) tweets portant sur le même sujet par rapport au nombre total de grappes formés par le modèle.

Les objectifs spécifiques permettant d'atteindre les résultats sont les suivants :

1. cueillir une quantité suffisante de tweets touchant la sécurité publique dans un site canadien afin de pouvoir appliquer un modèle informatisé de détection;

2. trouver la géolocalisation d'au moins 5 % de ces tweets jugés pertinents à l'aide de la recherche et du géocodage des toponymes dans le texte du tweet;
3. obtenir un nombre suffisant de grappes pour pouvoir identifier plusieurs événements de sécurité publique ayant eu lieu sur le site visé; et
4. réaliser un taux de fiabilité de 50 % pour les grappes détectées par le modèle.

Le deuxième objectif spécifique s'apparente aux résultats obtenus par Gu *et al.* (2016) pour le géocodage des toponymes obtenus du texte des tweets touchant les accidents en Pennsylvanie. Le taux de fiabilité de 50 % énoncé dans le quatrième objectif est établi en relation au taux de pureté obtenu par Shakira et Abdolreza (2014).

1.5 Limites de l'étude

Les outils de traitement linguistique les plus performants étant disponibles seulement pour l'analyse des messages en langue anglaise et la plus grande présence d'usagers Twitter en Ontario qu'au Québec (26 % vs. 20 % des répondants adultes au sondage Forum Research de 2015), ont été les principaux facteurs du choix de Toronto-Niagara comme région d'étude.

De même, les difficultés subies lors de la cueillette de tweets avec l'API Streaming de Twitter (décrochage du flux, limites de téléchargement) ont limité l'échantillonnage à une durée d'une semaine, ce qui représente néanmoins un cycle habituel dans l'utilisation du medium.

Enfin, la vitesse du flux Twitter (environ 8 à 10 tweets par seconde) exige beaucoup de mémoire vive d'un ordinateur de puissance normale pour traiter ce flux et pour identifier et sauvegarder les tweets pour la région visée. Ceci ne permet pas d'exécuter simultanément les opérations de cueillette de tweets et de détection d'événements, ce qui oblige un traitement des données cumulées qui se fera hors ligne et par étape.

La principale limite inhérente au cadre utilisé pour le projet est l'échantillonnage maximal d'environ 1 % de l'API public de Twitter.

CHAPITRE 2 : Cadre expérimental

2.1 Données du projet

Les données utilisées dans le cadre de ce projet seront des messages Twitter ainsi que les métadonnées leur étant associés. Le langage de programmation Python et une bibliothèque de code appelée Tweepy seront utilisés pour la cueillette de données avec le flux de Twitter (voir l'encodage dans la section A de l'annexe 2). Les données proviennent de l'API *Streaming* de Twitter, un outil public de requête permettant d'obtenir un échantillon d'environ 1 % du flux total de tweets selon des paramètres définis.

Lors de tests utilisant la requête API effectuée avec les coordonnées d'un rectangle géographique de capture couvrant la région de Toronto-Niagara, le taux de géocodage obtenu avec les tweets cueillis était très faible (0,2 %). La première colonne du Tableau 4 illustre le nombre de tweets récoltés par cette méthode et géocodés en identifiant leur latitude et leur longitude.

Afin de pouvoir obtenir un bassin plus important de tweets, un autre type de requête test fut exécuté avec l'API à partir d'une liste de 87 mots clés (voir Annexe 1). Ces mots-clés parviennent d'une analyse qui identifie les mots les plus fréquents d'un corpus constitué de 2 232 tweets d'incidents criminels émis par les autorités policières des villes de Chicago, New York et Seattle et de 1 313 tweets d'accidents émis par le Ministère du transport et la Police provinciale de l'Ontario. Ces mots-clés rendent compte d'événements de sécurité publique en général et vise à refléter les tweets qui pourraient être d'intérêt pour une organisation policière.

Les tweets obtenus avec la requête par mots-clés furent ensuite filtrés pour sauvegarder ceux contenant les mots « Toronto » ou « Ontario » car certains tweets mentionnent seulement « Toronto, Canada » tandis que d'autres citent la ville et la province ou seulement la province. Quoique le nombre de tweets cueillis soit 13 fois moindre qu'avec la première méthode, le taux de géocodage à partir de ce type de requête a augmenté considérablement pour atteindre 5,7 % tel qu'illustré au Tableau 4.

Ces requêtes de test furent aussi utilisées pour roder le modèle et pour effectuer une analyse de sensibilité des paramètres. Pour obtenir les résultats du projet, on a de nouveau exécuté la requête de 87 mots-clés auprès de l'API de Twitter du 16 au 22 juillet 2016 et obtenu 90 347 tweets. Notons que quelques mesures aléatoires de la vitesse du flux reçu de Twitter avant filtrage ont révélé un flux variant de 5 à 11 tweets à la seconde, ce qui se traduit en un flux moyen d'environ 700 000 tweets par jour, soit 0,14 % du flux total déclaré par Twitter de 500 millions par jour.

Tableau 4- Comparaison de tests de requêtes avec l'API de Twitter

	Polygone	**Mots Clés**
Tweets récoltés	421 622	31 999
Tweets géocodés	1 159	1 822
Taux de géocodage	0,2 %	5,7 %

Les métadonnées associées à chaque tweet fourni par l'API de Twitter contiennent plus de 30 champs. Pour chaque tweet cueilli, seuls les cinq (5) champs suivants ont été sauvegardés dans un fichier de format CSV: date et heure, texte du message, nom de l'usager, nombre d'adeptes (*followers*) et emplacement de l'usager (champ texte rempli par l'usager). Le Tableau 5 illustre des exemples de tweets cueillis.

Tableau 5- Exemples de tweets cueillis

Type	**Date/ heure**	**Texte**	**Nom d'usager**	**Nombre d'adeptes**	**Emplacement de l'usager**
Accidents et incendies	2016-05-14 05:40:02	"Medical (Trouble Breathing) Steeles Avenue b\/w Jane St \/ Peter Kaiser Gt, North York (2 Trucks)"	tofire	6 409	Toronto, Ontario
Incidents criminels	2016-07-17 00:09:00	"Person With A Knife | Dundas St W &amp Bathurst St [14 Div.] 07\/17 00:04 #Downtown #Toronto";	TPSCalls*	787	Toronto, Canada

*non affilié au Service de police de Toronto ou à la Ville de Toronto

La Figure 5 illustre la zone géographique de la grande région de Toronto-Niagara en Ontario visée par le projet. Selon le recensement de Statistiques Canada de 2011, la population résidant dans la région de Toronto-Niagara (appelée le *Golden Horseshoe*) est

de 8,1 millions d'habitants, ce qui représente environ le quart de la population canadienne. Étant donné que 78 % de la population canadienne a plus de 19 ans (Statistique Canada 2013) et que 26 % des adultes ontariens ont un compte sur Twitter (Forum Research, 2015), on peut estimer qu'environ 1,6 millions d'usagers de Twitter résident dans la région visée.

Figure 5- Territoire visé par le projet (voir encadré)

Le Tableau 6 illustre la répartition des tweets cueillis : moins de 2,3 % des usagers (5 premières catégories) représentent 31,5 % de tous les tweets. Notons que la première catégorie comportant 13 usagers est formée uniquement de services publics tels les agences gouvernementales et les médias.

Tableau 6- Répartition des tweets cueillis

Tweets par usager	Usagers		Tweets	
250 à 750	13	0,0 %	6079	6,8 %
100 à 249	25	0,1 %	3811	4,2 %
50 à 99	51	0,1 %	3558	4,0 %
25 à 49	146	0,4 %	4800	5,4 %
10 à 24	701	1,7 %	9961	11,1 %
2 à 9	10 820	27,0 %	33 093	36,9 %
1	29 045	70,7 %	29 045	31,6 %
TOTAL	40 137	100,0 %	90 347	100.0%

La grande majorité des tweets cueillis attribuables aux usagers générant moins de 10 tweets durant la période de cueillette contiennent des messages qui n'ont aucun intérêt dans le cadre de l'étude. Le Tableau 7 illustre un échantillonnage de tweets non-pertinents cueillis avec le mot-clé mis en relief.

Tableau 7- Exemples de tweets non-pertinents

its sooo hot outside **shoot** me
The month you were born in decides the **weapon** #Weapons #MonthOfBirth https:\/\/t.co\/oRf3MkWN61
Slash's bday also.means it's been 3 years since I last saw my king of life @PaulMcCartney. I miss you, pol
Carpet **bomb** the whole world.
Another lie from US **police** officers. Can you believe? This is ridiculous https:\/\/t.co\/qnvLCOHJyN

Pour géolocaliser les tweets, nous avons utilisé le fichier de rues de la province de l'Ontario (Gouvernement de l'Ontario, 2015) et y avons administré un prétraitement afin de minimiser les ambiguïtés et d'augmenter la capacité de géocodage. Les ambiguïtés sont surtout causées par les noms de rue utilisant un mot commun (ex. *East, Way, Church*) ou une faute d'orthographe. Ainsi, les noms de rue correspondant aux 2 000

mots les plus communs de la langue anglaise tels que décrits par le *British National Corpus* et le *Corpus of Contemporary American English* (TalkEnglish.com, 2016) furent supprimés du fichier et les numéros de route provinciale furent ajoutés, résultant en un fichier de 45 520 rues et routes.

2.2 Méthodologie

Le langage de programmation Python et une bibliothèque de code pour l'analyse linguistique des messages (NLTK) sont utilisées pour le traitement des données obtenues du flux de Twitter (voir l'encodage dans les sections B et C de l'annexe 2).

Seuls, trois des cinq (5) champs cueillis sont utilisés dans le traitement des données : la date/heure, le nom d'usager et le texte du message. Le champ du nombre d'adeptes s'est révélé non significatif étant donné que le nombre d'adeptes d'un usager n'avait aucune corrélation avec le nombre de tweets cueillis ou géocodés pour cet usager ou avec les tweets de cet usager faisant partie des grappes d'événements. Quant au champ de localisation (ville, province, pays), celui-ci est devenu redondant, le géocodage ayant été effectué à partir des noms de rue trouvés dans le texte des tweets.

Tel qu'illustré dans le schéma méthodologique de la Figure 6, la méthode comprend quatre étapes distinctes de traitement:

- la cueillette des données;
- le filtrage et le géocodage des tweets;
- la recherche et la formation de grappes;
- la détection d'événements.

À la fin de chacune des trois premières étapes, un fichier est sauvegardé et traité dans l'étape ultérieure. La cueillette de données ayant déjà été abordée dans les sections précédentes, les trois autres étapes sont détaillées dans cette section.

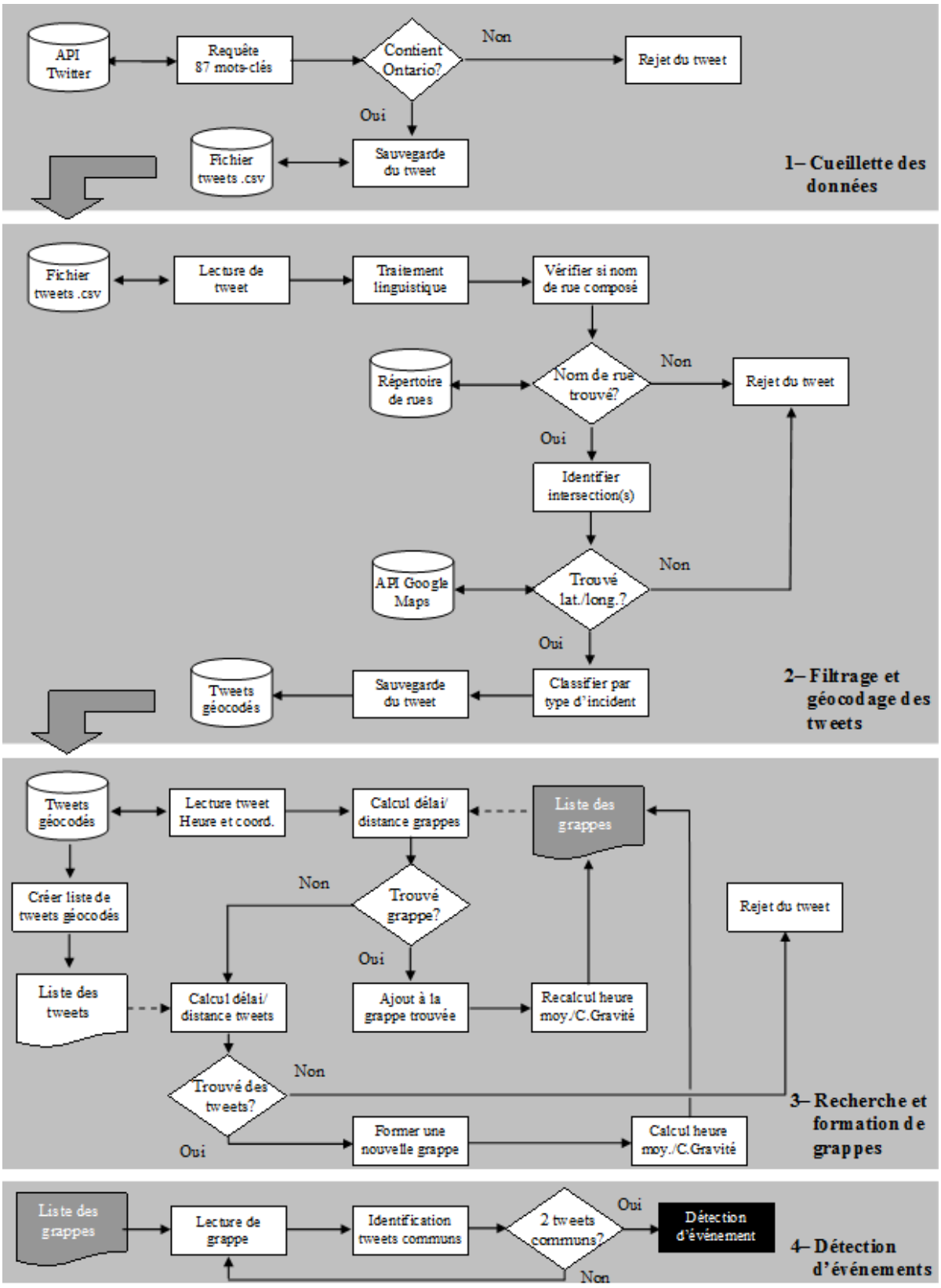


Figure 6- Schéma méthodologique

2.2.1 Filtrage et géocodage des tweets

2.2.1.1 Filtrage initial

Pour le filtrage initial, nous avons gardé les tweets originaux. Les tweets ayant '#RT' comme premiers caractères du texte sont rejetés, car ce sont des retweets, i.e. des tweets qui ont été renvoyés à d'autres usagers par un usager autre que l'expéditeur original. Les tweets contenant des adresses url sont aussi rejetés étant donné qu'ils sont en majorité des commentaires effectués en référence à une page Web donnée.

Par la suite, le texte de chacun des 90 347 tweets est analysé par le module NLTK du modèle informatisé afin de supprimer les mots accessoires (adverbes, articles, etc.) pour conserver uniquement une liste abrégée contenant les noms, les verbes et les adjectifs. Les mots restants sont tous traités dans l'ordre comme des bigrammes (ensembles de deux mots) de façon à comparer chaque bigramme au répertoire prétraité des rues de l'Ontario pour trouver une correspondance.

Si aucun nom de rue composé identique au bigramme n'est trouvé, le programme du modèle passe à un autre bigramme de la liste abrégée en cours de traitement. Si le programme atteint la fin de la liste sans trouver de nom de rue composé, le programme reprend la lecture du texte dès le début, en ignorant les bigrammes, et en comparant chaque mot de la liste abrégée. Si aucune correspondance n'est trouvée avec le répertoire de rues, le tweet en cours est rejeté et le programme lit le prochain tweet dans le fichier de tweets cueillis.

Si le programme identifie un nom de rue (simple ou composé) dans le tweet en cours de traitement, il cherche par la suite d'autres noms de rues dans la liste abrégée afin de trouver une ou des intersections, ce qui augmente la précision du géocodage.

2.2.1.2 Géocodage

Le géocodage des tweets ne fait pas appel aux noms de lieu ou toponymes: seulement les noms et les intersections de rues sont utilisés pour définir la localisation d'un événement. Le traitement de toponymes améliorerait sensiblement le taux de géocodage des tweets traités mais il n'est pas jugé essentiel à l'atteinte des objectifs visés.

Le géocodage est effectué à l'aide de l'API de Google Maps disponible en ligne. Suite à la soumission de(s) nom(s) de rue trouvés, celui-ci retourne: a) la latitude et la longitude ou, b) un message de lieu non trouvé ou, c) un message d'erreur. Dans les deux derniers cas, le tweet est rejeté et le programme passe au tweet suivant.

Si les coordonnées sont trouvées par l'API, le programme vérifie d'abord si les coordonnées se trouvent dans le territoire visé par le projet (Figure 5) car plusieurs tweets ont été collectés à cause qu'ils mentionnaient le mot 'Ontario' dans un des champs du tweet. Si le tweet se situe à l'extérieur de ce territoire, il est rejeté. Ensuite, le programme effectue un traitement pour classifier le type d'incident représenté. Une analyse linguistique est donc appliquée pour lemmatiser le texte original du message. Par exemple, les mots *living* et *weaponry* seront transformés en *live* et *weapon* respectivement.

La classification est réalisée en comparant les lemmes obtenus aux deux listes de mots-clés compilées qui sont énumérées dans l'annexe 1. Les tweets sont donc classés comme accident/incendie ou comme incident criminel. Dans l'exemple du tweet contenant live and weapon, il serait classé comme incident criminel. Enfin, le tweet classé est sauvegardé avec ses coordonnées et son type dans un fichier de tweets géocodés pour traitement lors de la troisième étape.

2.2.2 Sensibilité du modèle et formation de grappes

Nous avons choisi une méthode spatio-temporelle pour regrouper les tweets en grappes selon la distance et la durée de temps qui les séparent. L'application de ces deux paramètres pouvant faire varier les résultats significativement, une analyse de sensibilité devient donc essentielle.

2.2.2.1 Analyse de sensibilité

L'analyse de sensibilité permet d'alléger le modèle en fixant les entrées dont la variabilité n'affecte pas la variable de sortie. Afin de vérifier la sensibilité du modèle aux variations des paramètres de délai et de distance, des tests ont été effectués à partir de 538 tweets d'accident géocodés avec les paramètres de délai fixés à 30, 60, 90 minutes et de distance

fixés à 0,5, 1, 2 et 3 kilomètres. Tel qu'attendu, le nombre total de grappes augmente avec les paramètres mais le nombre de grappes ayant au moins 2 tweets communs sans doublons demeure le même quel que soit le délai utilisé. Ceci signifie que le modèle n'est pas sensible au paramètre de délai. De plus, ce sont les mêmes sept (7) grappes qui se répètent quelque soit le paramètre utilisé.

Le Tableau 8 affiche la sensibilité du modèle au paramètre de distance : c'est la distance de 3 km qui détecte le plus de grappes significatives. Ceci s'explique par la technique de géocodage utilisée par l'API de Google. Pour les tweets contenant trois (3) noms de rue, qui représentent 43 % de tous les tweets géocodés (voir 'Géocodage' dans la section d'analyse des résultats), ceux-ci sont géocodés selon la première intersection qui retourne des coordonnées, ce qui peut couvrir facilement deux à trois km lorsque l'accident se trouve sur une autoroute. Pour les tweets contenant seulement un nom de rue sans adresse, ceux-ci sont géocodés avec la coordonnée située au centroïde de la rue visée. Plus la rue est longue, plus le centroïde pourrait être éloigné de l'emplacement réel du tweet.

Tableau 8- Sensibilité de la détection d'événement au paramètre de distance

Grappe	Tweets communs sans doublons	0,5 km	1 km	2 km	3 km
1	3				
2	2				
3	7				
4	2				
5	2				
6	4				
7	2				

2.2.2.2 Formation des grappes

Suite à l'analyse de sensibilité décrite plus haut, les paramètres de délai et de distance utilisés pour la formation des grappes ont été fixés à 30 minutes et à 3 km.

Les grappes sont formées en cherchant d'abord les grappes existantes répondant aux paramètres; si une grappe est trouvée, le tweet traité est ajouté à la grappe, sinon une nouvelle grappe est formée avec les tweets répondant aux paramètres. Avant qu'un tweet en lecture soit ajouté à une grappe, le texte du tweet est comparé à celui des tweets de la grappe visée. Si le même texte est trouvé, c'est un doublon et on passe à la lecture du tweet suivant dans la liste de tweets actifs afin d'éviter la duplication de tweet dans la même grappe.

Les grappes sont formées lorsque deux tweets ou plus ayant été émis à l'intérieur d'un délai de détection de 30 minutes se trouvent à l'intérieur d'une distance de 3 km. Le centre de gravité de la grappe est calculé en effectuant la moyenne des latitudes et longitudes; l'heure moyenne de la grappe est aussi calculée avec la moyenne des heures des tweets de la grappe. Si un tweet est ajouté à une grappe existante, le modèle recalculera le centre de gravité et l'heure moyenne de la grappe.

Afin d'expliquer la formation de grappes, la Figure 7 illustre les opérations qui se déroulent lors la configuration spatiale d'une grappe ayant 5 tweets dont le tweet en traitement et les quatre tweets répondant aux paramètres de délai de détection et de distance. Les tweets qui se rendent à cette étape seront tous enregistrés dans une liste de tweets actifs géocodés qui servira de base de comparaison pour chaque nouveau tweet géolocalisé qui s'avère pertinent.

Le processus de formation de grappes se déroulera pour chacun des deux types d'événements, les accidents et les incidents criminels, jusqu'à la fin de la lecture du fichier de tweets géocodés, c'est-à-dire le fichier de données complet. Par la suite, les grappes identifiées seront sauvegardées par type d'événement détecté. Ceci permettra de mesurer le nombre de grappes contenant des événements pertinents par rapport à l'ensemble des grappes identifiées afin de calculer le niveau de pertinence par type d'événement.

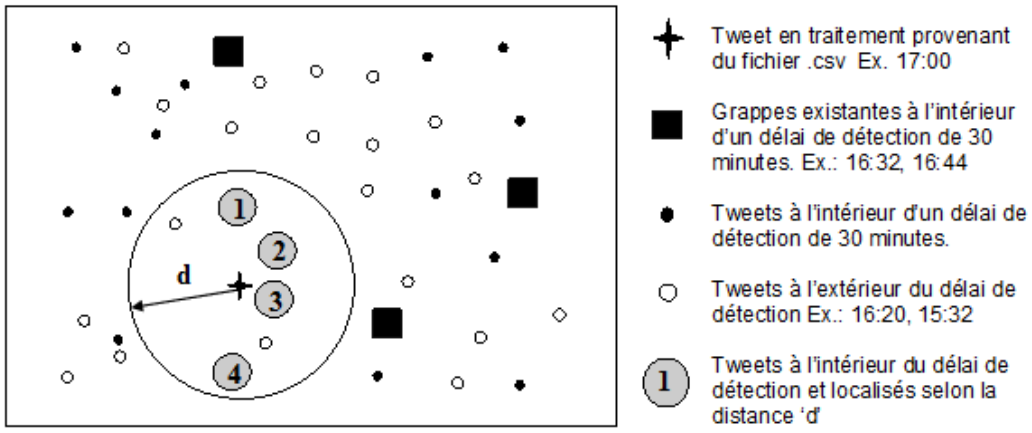


Figure 7- Représentation spatiale de formation des grappes

2.2.3 Détection d'événements.

Pour chaque type d'événement (accident/incendie et crime), les grappes identifiées par le modèle sont examinées par l'auteur afin de déterminer si elles contiennent au moins deux tweets ayant un message touchant le même sujet.

Le Tableau 9 affiche deux grappes : la première contenant quatre tweets n'ayant aucun sujet en commun et la deuxième avec cinq tweets dont trois partagent le même thème avec le texte commun affiché en italique. La deuxième grappe est donc reconnue comme un événement détecté.

Tableau 9- Exemple de détection de grappe d'événement

Tweets	Noms d'usager	Tweets communs	Messages
4	tofire, tofire, tofire, tofire,	0	• Medical (Unconscious) - York St b\\/w Lakeshore West York St Ramp \\/ Bremner Boulevard, Toronto (2 Trucks) • Check Call - Distillery Lane b\\/w Parliament St \\/ Trinity St, Toronto (2 Trucks) • Alarm Highrise (Commercial) - Carlton St b\\/w Reverend Porter Lane \\/ Church St, Toronto (6 Trucks) • Medical (Unconscious) - Wellesley St b\\/w Yonge St \\/ Lane East Yonge South Wellesley, Toronto (2 Trucks)
5	RyanCalma3, tofire, 1010traffic, , tofire, tofire,	3	• 401 east starting at Bathurst is heavy traffic in both collectors and express #401 • *Medical (Trouble Breathing) - Bathurst St b\\/w Newbury Lane \\/ York Downs Dr, North York (2 Trucks)* • WB 401 express at Yonge, collision cleared. Theres still one EB express at the DVP. Two right lanes and the off-ramp blocked • *UD: Tems Transferred (Read Remarks) - Bathurst St b\\/w Newbury Lane \\/ York Downs Dr, North York (2 Trucks)* • *[1] NY TEMS TRANSFERRED - READ REMARKS Bathurst St b\\/w Newbury Lane &amp York Downs Dr Pumper-143 [143],*

CHAPITRE 3 : Présentation et analyse des résultats

Dans ce chapitre, nous avons voulu mettre à l'épreuve notre modèle qui vise à détecter avec un niveau de fiabilité acceptable, les événements de sécurité publique du territoire visé.

Le graphique de la Figure 8 résume les résultats obtenus par le modèle en traitant les 90 347 tweets cueillis avec notre approche séquentielle de filtrage des tweets, de géocodage et de formation de grappes. Le modèle a permis de trouver les coordonnées de 5 150 tweets pour un taux de géocodage de 5,7 % (5 150/90 347). Le modèle a permis d'exclure les tweets géocodés situés hors du territoire ainsi que les doublons de tweet avant de traiter les tweets pour la formation de grappes.

Au total, 79 grappes d'événements ont été détectées sur un total de 172 grappes créées représentant un taux de fiabilité global de 45,9 %. La proportion de grappes d'événements détectées est beaucoup plus grande pour les accidents et incendies avec 75 grappes sur l'ensemble des 79 grappes d'événements laissant seulement 4 grappes d'événement pour les incidents criminels.

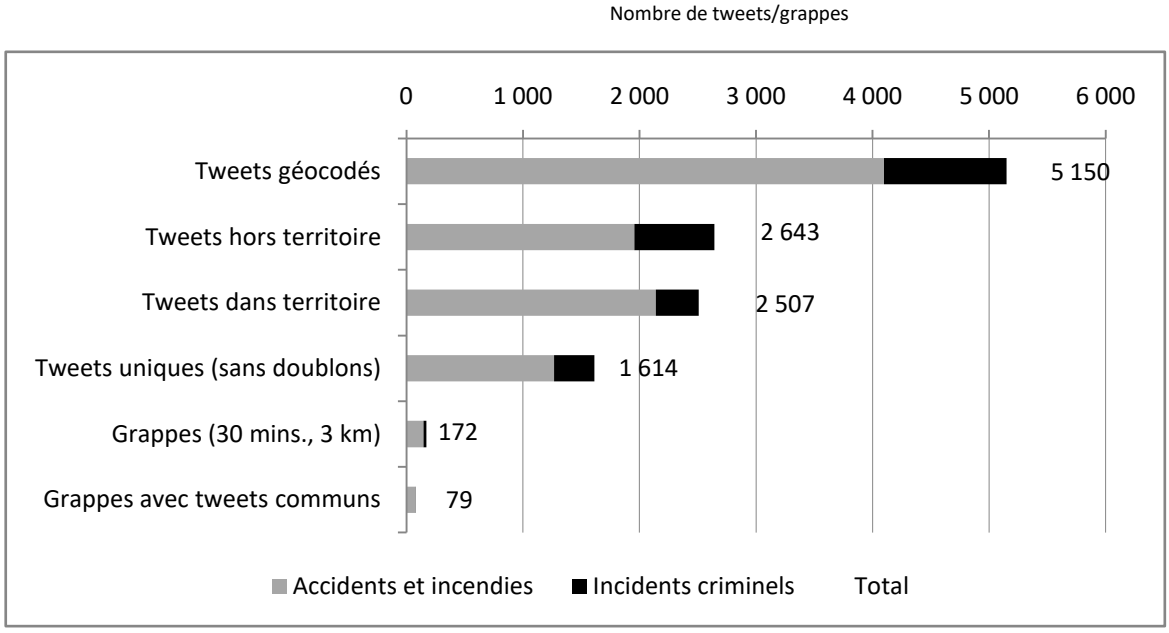


Figure 8- Sommaire des résultats (90 347 tweets traités)

La méthodologie utilisée s'appuyant sur la localisation des tweets à partir du texte et sur le traitement spatio-temporel des tweets, les résultats sont analysés en deux étapes : l'opération de géocodage et la formation de grappes.

3.1 Analyse du géocodage

Sur le total de 90 347 tweets cueillis, le modèle a expédié 6 530 demandes de géocodage à l'API de Google. Celui-ci a généré une latitude et une longitude pour 5 150 tweets et retourné un message d'absence de coordonnées ou d'erreur pour 1 380 tweets, ce qui représente un taux de rejet de 21 %.

Les mots « Toronto » ou « Ontario » ayant été utilisés pour filtrer les tweets cueillis de l'API de Twitter, 51 % des tweets géocodés (2 643) se trouvent hors du territoire étudié (régions autres que Toronto-Niagara et tweets d'origine américaine).

Pour les 2 507 tweets situés dans le territoire visé, le Tableau 10 illustre les résultats de géocodage selon le nombre de rues trouvées dans le répertoire de l'Ontario. Ainsi, la grande majorité des tweets géocodés provient des intersections de tweets ayant le nom de 2 ou 3 rues dans le message. Les tweets où seulement une rue sans adresse a été détectée (8 % des cas) ont été géocodés par l'API de Google à partir du centroïde de la rue visée.

Tableau 10- Rues détectées pour le géocodage

	Accidents et incendies	**Incidents criminels**	**Total**	
1 rue avec adresse	15	5	20	1 %
1 rue sans adresse	155	35	190	8 %
2 rues	1 014	208	1 222	48 %
3 rues	957	118	1 075	43 %
Total	2 141	366	2 507	100 %

Il existe une concentration importante des tweets géocodés lorsqu'on compile ceux-ci par nom d'usager et selon les heures de la journée. Le Tableau 11 indique que la grande majorité des tweets provient de 6 usagers et que ceux-ci sont surtout des services publics comme les postes de radio et les services d'incendie (91 % et 85 %). La concentration des

61 % tweets durant la journée pour les accidents et les incendies est à l'opposé de celle des incidents criminels qui se produisent (65 %) en soirée et durant la nuit.

Tableau 11- Concentration des usagers et des heures de tweets (1 614 tweets géocodés sans doublons)

	Accidents et incendies	Incidents criminels
Usagers majeurs	5	1
Part des tweets (usagers majeurs)	70 %	71 %
Part des tweets (services publics)	91 %	85 %
Tweets de jour (6h à 18h)	61 %	
Tweets du soir (18h à 6h)		65 %

La Figure 9 illustre la distribution géographique des tweets uniques géocodés pour les deux types d'incident. On note que les incidents se produisent davantage dans les zones de grande densité de population; de même, malgré un nombre de tweets assez différent entre les deux types, la distribution géographique demeure proportionnelle.

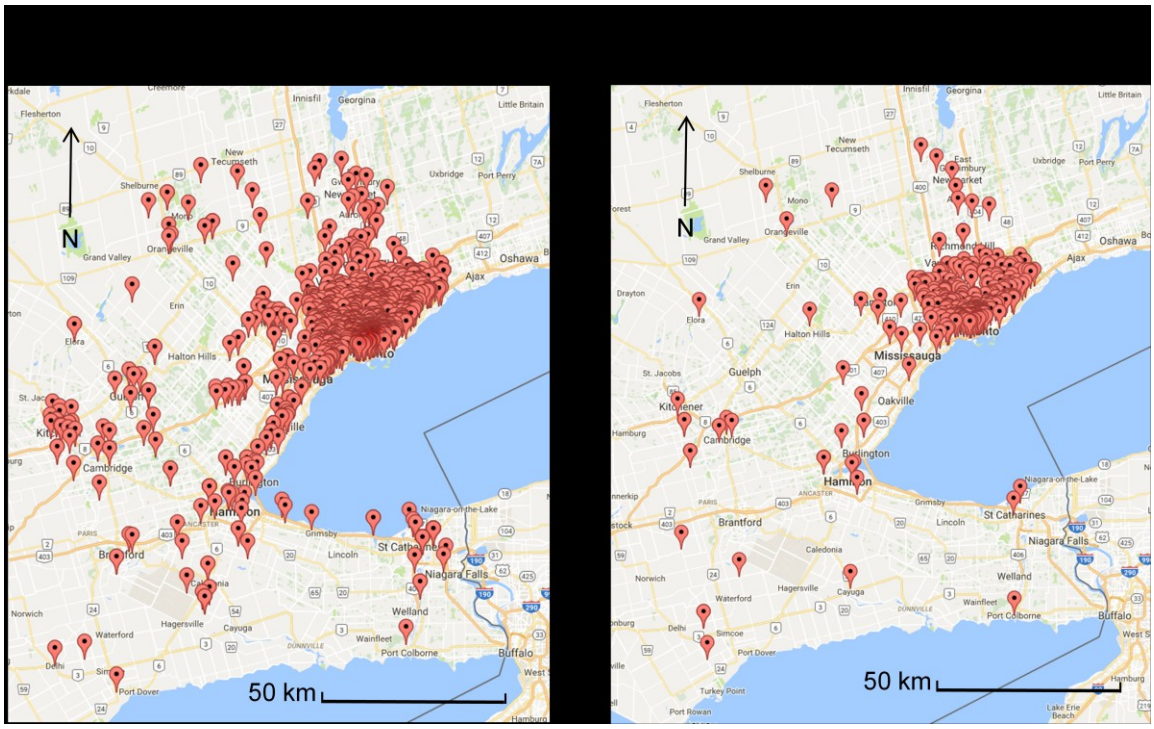


Figure 9- Cartes des tweets uniques géocodés

3.2 Analyse de la formation de grappes

En appliquant aux 1 614 tweets uniques géocodés les paramètres de 30 minutes de délai de détection et de 3 km de distance entre le tweet traité et les grappes ou les tweets déjà traités, on obtient un total de 172 grappes dont 146 pour les accidents et incendies. Le Tableau 12 décrit une grappe incluant le message de chacun de ses tweets.

Tableau 12- Exemple de grappe d'événement

Tweets	Heure moyenne de grappe	Centre de gravité de grappe	Noms d'usager	Usagers communs	Mots clés	Mots clés communs	Tweets communs
7	2016-07-16 14:08	(43,6943,-79,3447)	tofire, , TPSOperations, , tofire, tofire, tofire, , CTVToronto, , , CP24, ,	4	lane, fire, truck, , fire, , lane, fire, truck, , lane, fire, truck, , fire, , fire, , fire, truck, ,	6	7

Écart type des heures de tweets: 15,5 minutes
Distance moyenne du centre de gravité: 40 mètres
Description des tweets:

2016-07-16 13:50	tofire	Elmsdale Road bw Oconnor Dr Lane North O Connor East Wolverton, East York (7 Trucks)
2016-07-16 13:54	TPSOperations	Fire: Elmsdale Rd OConnor Dr..smoke coming from an apt. No ins reported at this time.
2016-07-16 13:56	tofire	UD: [2 Alm] Fire (Residential) - Elmsdale Road bw Oconnor Dr Lane North O Connor East Wolverton, East York (16 Trucks)
2016-07-16 14:00	tofire	UD: [2 Alm] Fire (Residential) - Elmsdale Road bw Oconnor Dr Lane North O Connor East Wolverton, East York (19 Trucks)
2016-07-16 14:23	tofire	[2] EY Fire - Residential Elmsdale Rd bw Oconnor Dr & Ln N O Connor E Wolverton R224 P322 P323 R321 A322 C32 R235 A321
2016-07-16 14:27	CTVToronto	Toronto Fire says that no injuries have been reported after a two-alarm blaze at a building on Elmsdale Rd.
2016-07-16 14:27	CP 24	Toronto Fire says that a two-alarm blaze at a building on Elmsdale Rd. has been contained to a closet. Three trucks remain on scene.

Tel qu'illustré au Tableau 13, le modèle a assigné 93 tweets géocodés à des grappes existantes au lieu de former une nouvelle grappe avec d'autres tweets. Le taux d'assignation de 22 % est beaucoup plus haut pour les accidents et incendies (92/410) que le taux de 2 % noté pour les incidents criminels (1/54).

Tableau 13- Analyse des grappes

	Accidents Incendies	Incidents criminels	Total
Tweets uniques géocodés traités	1 268	346	1 614
Tweets contenus dans grappes	410	54	464
Tweets assignés à grappe existante	92	1	93
Grappes (30 minutes, 3 km)	146	26	172
Tweets par grappe	2,8	2,1	
Grappes d'événements	75	4	79
Taux de détection d'événement	5,9 %	1,2 %	4,9 %
Taux de fiabilité	51,4 %	15,4 %	45,9 %
Écart type d'heure (en minutes)	7,9	4,4	
Distance moyenne au centre de gravité de grappe (en mètres)	645	183	

Le nombre de tweets par grappe est plus élevé pour les accidents et incendies (2,8) par rapport aux incidents criminels (2,1). De même, le taux de détection d'événement, qui se calcule avec le nombre de grappes d'événements par rapport au nombre de tweets traités, varie de façon importante selon le type de tweet. Pour les accidents et incendies, celui-ci est de 5,9 % (75/1 268) comparé aux incidents criminels qui affichent un taux de 1,2 % (4/346). Ces deux résultats s'expliquent par le fait que beaucoup plus de personnes sont témoins d'un accident ou d'un incendie que d'un incident criminel.

L'écart type observé pour l'heure des tweets par rapport à la moyenne de leur grappe est presque deux fois plus grand pour les accidents/incendies (7,9 minutes) que pour les incidents criminels (4,4 minutes). La différence est encore plus importante pour la distance moyenne au centre de gravité de la grappe pour les accidents et incendies (645 m) comparativement aux incidents criminels (183 m). Ces écarts s'expliquent par la plus grande concentration d'usagers pour les incidents criminels : 71 % des tweets de ce type proviennent du même usager. Ce dernier émet des tweets à intervalles de temps plus rapprochés pour un même événement et la précision dans la description des rues est plus précise et constante, ce qui améliore le géocodage.

La distribution des grappes d'événements selon leur nombre de tweets est illustrée à la Figure 10 pour les accidents et les incendies. La conversion de grappes en grappes

d'événements est faible pour les grappes ayant 2 tweets soit de 68 à 17, un taux de conversion de 25 %, ce qui est normal étant donné que la grappe contient seulement 2 tweets. Le taux de conversion augmente à 67 % pour les grappes à 3 tweets, à 86 % pour 4 tweets et à 100 % pour 5 tweets et plus. Quant aux incidents criminels, seules, 4 grappes sur 26 ont été converties en grappes d'événements contenant toutes 2 tweets.

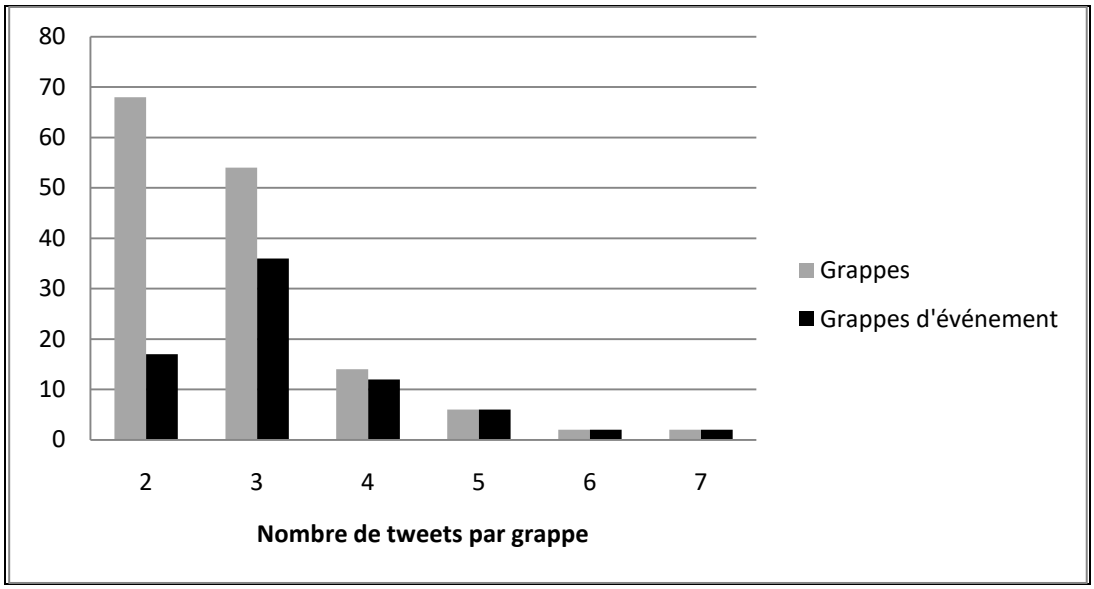


Figure 10- Distribution des grappes d'accident et d'incendie

Le Tableau 14 affiche encore les grappes spatio-temporelles et les grappes d'événements pour les accidents et les incendies en indiquant le nombre et le poids relatif de chaque taille de grappe. On note que la majorité des grappes spatio-temporelles formées (84 %) contiennent 2 ou 3 tweets par grappe mais que les grappes d'événements possèdent surtout 3 tweets (48 %).

Tableau 14- Nombre de tweets par grappe- Accidents et incendies

Tweets par grappe	Grappes spatio-temporelles		Grappes d'événements	
2	68	47 %	17	23 %
3	54	37 %	36	48 %
4	14	10 %	12	16 %
5	6	4 %	6	8 %
6	2	1 %	2	3 %
7	2	1 %	2	3 %
Total	146	100 %	75	100 %

La Figure 11 illustre la distribution géographique des grappes d'événements qui est concentrée dans la métropole de Toronto et ses environs. Pour fins de comparaison, la Figure 12 affiche la densité de la population pour le même territoire.

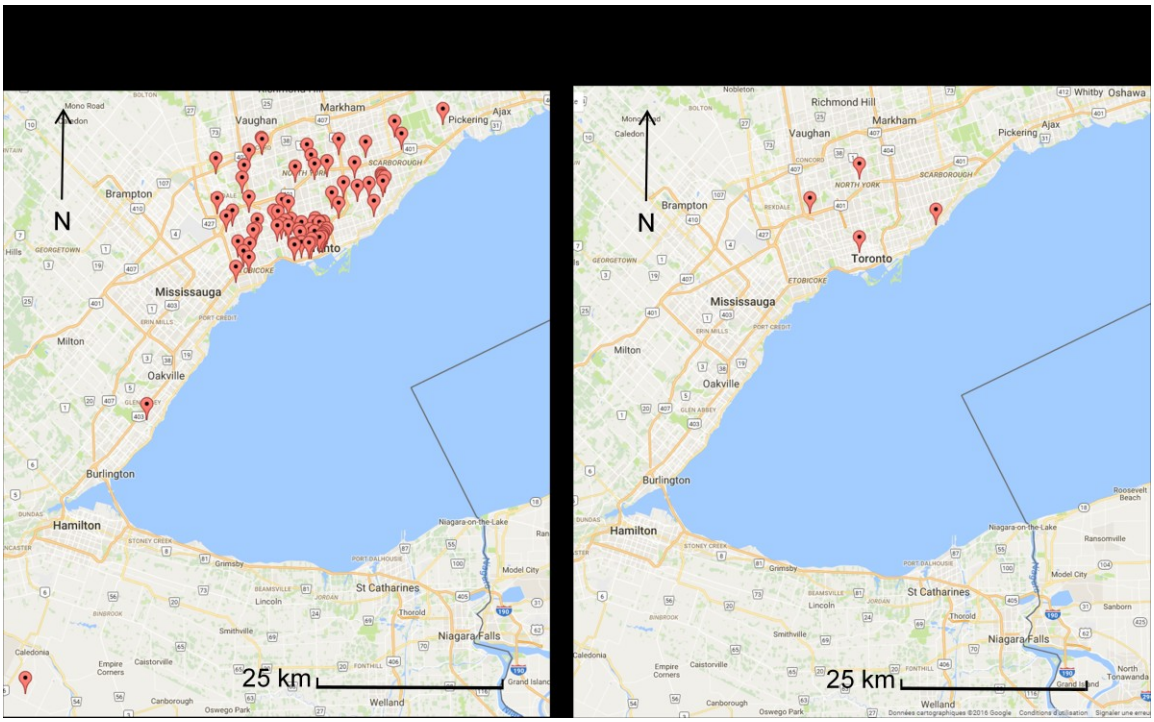
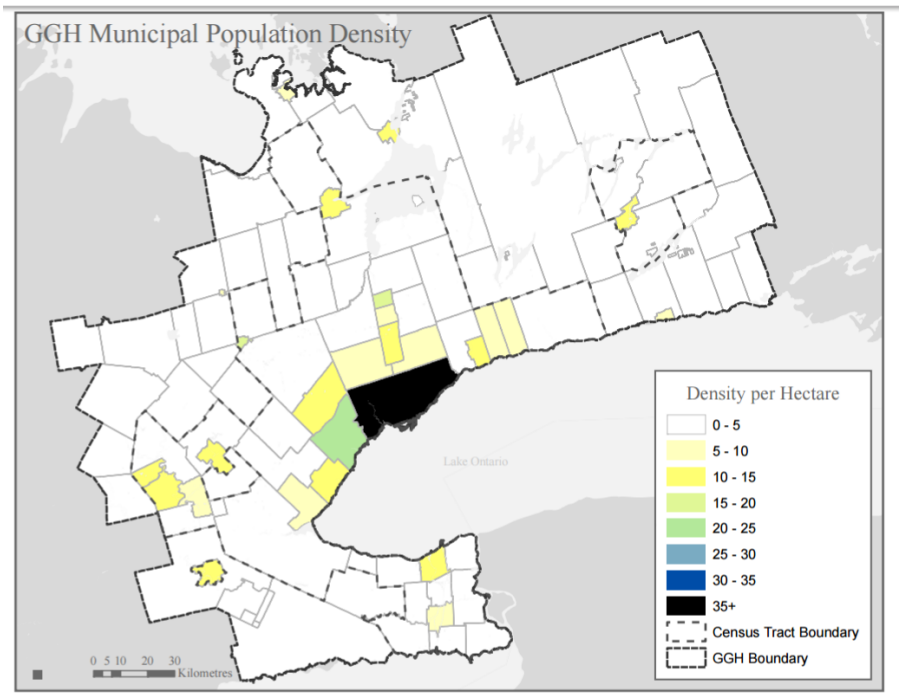


Figure 11- Cartes des grappes d'événement



Tiré de Urban Density in the greater Golden Horseshoe (2007)

Centre for Urban and Community Studies • University of Toronto

Figure 12- Densité de population de Toronto-Niagara

CHAPITRE 4 : Interprétation des résultats

En général, les résultats obtenus ont permis d'atteindre l'objectif principal du mémoire qui était de détecter et de localiser précisément un nombre suffisant d'événements de sécurité publique d'un territoire à partir du flux de Twitter avec un taux de fiabilité acceptable.

4.1 Atteinte des objectifs spécifiques

Les éléments ci-dessous montrent comment les résultats se comparent aux quatre objectifs spécifiques de l'étude.

1. Cueillir une quantité suffisante de tweets touchant la sécurité publique dans un territoire canadien afin de pouvoir appliquer un modèle informatisé de détection.

   Les 90 347 tweets cueillis sur une période d'une semaine en juillet 2016 ont été suffisants pour démontrer la validité de la méthodologie présentée dans cette étude. Notons que la région choisie possède la plus grande concentration de population au Canada, ce qui maximise l'échantillonnage. Par contre, la période du 16 au 22 juillet se situe durant les vacances estivales, ce qui pouvait représenter une diminution dans le volume hebdomadaire de tweets émis par rapport aux autres périodes de l'année.

2. Trouver la géolocalisation d'au moins 5 % des tweets cueillis à l'aide de la recherche et du géocodage des toponymes dans le texte du tweet.

   Le modèle développé est performant en ce qui concerne le géocodage. Le taux de géocodage, soit le nombre de tweets dont on a trouvé les coordonnées par rapport au total de tweets traités (5 150/90 347), est de 5,7 %, ce qui dépasse le résultat de 4,9 % et de 4,3 % obtenu par Gu *et al.* (2016) pour les tweets de Pittsburgh et de Philadelphie respectivement. Le résultat obtenu est aussi plus élevé que celui atteint par Jurgens *et al.* (2015) avec le répertoire GeoNames qui a trouvé la localisation, par nom de ville, de 3,4 % des tweets traités.

   Il est possible d'améliorer cette performance en modifiant le modèle afin de pouvoir traiter les ambiguïtés causées par les noms de rues communs trouvés dans le texte

des tweets. Rappelons que 2 000 mots communs furent supprimés du répertoire de rues de l'Ontario pour éviter ces ambiguïtés et pour simplifier le filtrage et le géocodage des tweets. De même, le géocodage des tweets signalant plus d'une intersection de rues pourrait être optimisé résultant en un taux plus élevé de tweets géocodés.

3. Obtenir un nombre suffisant de grappes pour pouvoir identifier plusieurs événements de sécurité publique ayant eu lieu sur le territoire visé.

   À partir des 90 347 tweets cueillis, 172 grappes spatio-temporelles ont été identifiées automatiquement par le modèle, ce qui représente un ratio de formation de grappes de 0,2 % (172/90 347). Ceci se compare favorablement au résultat obtenu par Arcaini *et al.* (2016) pour la formation de grappes spatio-temporelles à partir d'un nombre de tweets d'un même ordre de grandeur. Avec l'aide de leur algorithme DBSCAN modifié, ils ont traité 60 630 tweets cueillis avec les mots-clés '*Soccer World Cup 2014*', et détecté de 5 à 150 grappes en faisant varier la distance exigée entre les tweets d'une grappe et en utilisant des paramètres constants de 2 tweets minimum par grappe et de 10 minutes de délai maximum entre les tweets d'une grappe. Leurs résultats représentent un ratio de formation de grappes variant de 0,008 à 0,25 %.

4. Réaliser un taux de fiabilité de 50% pour les grappes détectées par le modèle.

   Le taux de fiabilité global de 45,9 % obtenu par le modèle se doit d'être réparti selon le type d'incident pour une interprétation plus exacte. Ainsi, le taux de 51,4 % réalisé pour les accidents et incendies dépasse l'objectif de 50 %; le taux de 15,4 % des incidents criminels est bien en deçà de l'objectif, mais le faible nombre de grappes identifiées pour ce type (26) et la taille moyenne de celles-ci (2,1 tweets par grappe) n'étaient pas favorables à l'obtention d'un taux de fiabilité élevé. Rappelons qu'une grappe devait contenir au moins deux tweets ayant un sujet commun pour être considérée comme un événement. La Figure 8 indique que le taux de conversion de grappes contenant deux tweets est faible par rapport aux plus grosses grappes, soit de 26,4 % (18/68).

La définition du taux de fiabilité est très similaire à celle du taux de pureté compilé par Shakira et Abdolreza (2014). Ce taux de pureté, variant de 65 à 75 %, est plus élevé que celui obtenu dans cette étude, car la probabilité de retrouver deux tweets ou plus ayant le même sujet était beaucoup plus élevée. Les deux raisons pour cette situation sont : a) la requête utilisée pour l'API de Twitter contenait uniquement les mots '*End%of%World'* (événement spécifié) et, b) la méthode employée dépendait entièrement des mots du texte pour former les grappes sans tenir compte de l'aspect spatio-temporel.

Les résultats obtenus se rapprochent aussi de ceux de Cheng et Wicks (2014) qui ont aussi utilisé une méthode spatio-temporelle pour détecter les événements. Ils ont déclaré avoir trouvé 87 grappes dont 30 étaient significatives pour un taux de fiabilité de 34 % (30/87) en agrégeant par journée les 183 731 tweets cueillis dans la région de Londres. Ils ont obtenu un taux supérieur (69 %) en agrégeant par heure au lieu de journée avec 48 grappes trouvées et 33 grappes significatives.

En résumé, le modèle conçu pour le projet peut détecter les événements à partir du flux de Twitter et fait la preuve que la proximité en temps et en distance de plusieurs tweets communs (i.e. ceux dont le message traite le même sujet), améliore la fiabilité de la détection d'événements.

4.2 Autres constats

Les autres constats quant à l'interprétation des résultats sont énumérés selon l'ordre établi dans la méthodologie de l'étude.

4.2.1 Données

Selon Bruns (2012), les usagers qui génèrent le plus de tweets représentent 1 % de tous les usagers Twitter. Ceci se compare favorablement au taux de 2,3 % des usagers qui génèrent plus de 30 % des tweets cueillis avec les mots-clés, tel qu'indiqué au Tableau 6.

4.2.2 Filtrage

Une des prémisses de l'étude pour la rétention d'un tweet est que le message contienne au moins un nom de rue. Cette contrainte garantit qu'une localisation, même approximative avec un seul nom de rue, permettra au service de police municipal de se rendre et d'intervenir s'il y a lieu.

Étant donné que la majorité des tweets cueillis ne sont pas pertinents et qu'une partie importante des tweets décrivant un événement contiennent un toponyme (Vieweg *et al.*, 2010 et Gelernter et Balaji, 2013), nous croyons que le filtrage par nom de rue ou par toponyme est une façon efficace de parcourir le flux Twitter pour en tirer des messages pertinents et actionnables.

4.2.3 Géocodage

L'API de Google a offert une performance de géocodage intéressante avec un taux de rejet de 21 % sur l'ensemble des requêtes soumises, ce qui est très bas en considérant que très peu de tweets avaient des adresses exactes et que la plupart étaient géocodés à partir des intersections.

Étant donné que la majorité des tweets (91 %) ont été géocodés à partir d'intersections de rues, tel qu'illustré dans le Tableau 10, il appert que la plupart des usagers signalant un événement tendent à le localiser avec une intersection, ce qui facilite de beaucoup les opérations de géocodage.

Les modes de filtrage et de géocodage utilisés pour l'étude ont résulté en une couverture adéquate du territoire à l'étude. La distribution géographique des tweets géocodés de la Figure 9 est assez conforme à la densité de population de la région illustrée à la Figure 12; cela confirme la représentativité des données cueillies pour l'étude.

La concentration des tweets géocodés illustrée au Tableau 11 avec une moyenne de 88 % des tweets provenant de services publics n'est guère surprenante étant donnée qu'aucun événement d'envergure ne s'est produit durant la période de cueillette. En effet, une recherche Internet effectuée dans les médias locaux de nouvelles n'a révélé aucun événement majeur relatif aux accidents, aux incendies ou aux crimes dans la grande

région de Toronto-Niagara durant la période de cueillette du 16 au 23 juillet 2016.

Nous croyons que si un événement d'envergure tel un attentat ou un cataclysme s'était produit durant la période de cueillette, celui-ci aurait été immédiatement repéré par le modèle à partir de tweets en provenance autant d'usagers publics que d'individus.

La concentration de 61 % des tweets d'accident durant la journée est normale à cause de la grande intensité de la circulation routière à cette période.

4.2.4 Formation de grappes

Les paramètres de délai (30 minutes) et de distance (3 km) utilisés pour la formation des grappes ont permis de générer suffisamment de grappes pour atteindre les objectifs de cette étude. En agrandissant le délai et/ou la distance, plus de grappes sont nécessairement formées mais le nombre de grappes d'événements demeure sensiblement le même.

De même, la méthode de recherche de grappes existantes répondant aux paramètres pour un tweet existant avant de former une nouvelle grappe a diminué le total de grappes générées et en même temps augmenté sensiblement le nombre de tweets par grappe, élevant du même coup la probabilité de détecter un événement à partir d'au moins deux tweets communs.

Le Tableau 14 affichant le nombre de tweets par grappe pour les accidents et incendies indique peu de grappes formées de 4 tweets ou plus : cela démontre que les incidents ayant eu lieu durant la période de l'étude reflétaient une certaine normalité sur le réseau routier.

4.2.5 Détection d'événement

Le nombre de tweets d'une grappe d'événements révèle son importance relative telle qu'illustré dans l'exemple de grappe du Tableau 12 contenant sept (7) tweets qui se rapportent à un incendie ayant lieu sur la rue Elmsdale. Ainsi, plus la taille des grappes formées est importante, plus la probabilité de détecter un événement devient grande.

CHAPITRE 5 : Discussion des résultats et conclusion

5.1 Discussion

Les résultats obtenus par cette étude permettent d'appliquer une méthode efficace de détection d'événements à partir du flux Twitter au niveau d'un territoire spécifique desservi par un service de police municipal. L'efficacité de la méthode réside dans l'utilisation de grappes spatio-temporelles à partir de tweets géocodés avec les toponymes cités dans leur texte et cueillis pour un même territoire.

La détection d'événements effectuée à partir de l'analyse linguistique du texte des tweets (NLP) et de la formation de grappes à partir de scores rattachés aux mots, domine la recherche actuelle. Cette méthode ne convient pas aux besoins des services de police locaux étant donné que la localisation précise des grappes n'est pas un enjeu pour les chercheurs.

Par contre, la détection d'événements à partir de grappes spatio-temporelles a été étudiée par quelques scientifiques surtout avec l'algorithme DBSCAN appliqué à un corpus de tweets auto-localisés obtenu avec une requête d'événement spécifié ou d'un polygone de capture très étendu. Malheureusement, le nombre de tweets auto-localisés cueillis pour un territoire réduit tel que celui d'un service de police n'est pas suffisant pour appliquer cette méthode, étant donné qu'on ne peut prélever plus de 0,5 % des tweets (proportion maximale de tweets auto-localisés selon Ajao et Liu (2015)) d'un échantillon de 1 % du flux Twitter résultant en 0,05 % du flux total.

Du côté des méthodes de géocodage des tweets, plusieurs articles sont recensés qui appliquent la recherche de toponymes contenus dans le texte des tweets obtenus surtout par une requête d'événement spécifié auprès de l'API de Twitter. La plupart des articles utilisent des répertoires toponymiques internationaux pour géocoder les tweets au niveau des villes ou des pays. Quelques auteurs s'efforcent d'identifier les coordonnées de tweets avec un répertoire de noms de rue ou de place locale et de vérifier la précision de l'opération à partir des tweets comportant aussi une auto-localisation. Nous n'avons pas trouvé d'article mettant en relief l'importance des intersections de rues dans la précision du géocodage.

Ainsi, la présente recherche a combiné : a) l'analyse linguistique pour déceler les toponymes dans le texte des tweets cueillis, b) une méthode de géocodage appuyée sur l'identification des intersections à partir des toponymes et, c) la détection d'événements à partir de grappes spatio-temporelles contenant au moins deux tweets ayant le même sujet, ce qui permet de corroborer les rapports d'événement.

Avec les résultats de cette recherche, on peut donc avoir accès à un plus grand échantillon du flux Twitter pour un territoire restreint et pouvoir déceler et corroborer les événements de sécurité publique et leur localisation à partir du texte des tweets au lieu de dépendre de l'auto-localisation de l'usager.

5.2 Conclusion

Ce projet de recherches a dû faire face à certaines limites telles que :

a. La vitesse et les contraintes du flux Twitter : durant la période de cueillette, le modèle n'a pas pu maintenir une connexion constante avec l'API *Streaming* de Twitter ce qui nous a obligé à effectuer une requête par jour en continu avec quelques interruptions totalisant une dizaine d'heures sur l'ensemble de la période de collecte de sept jours.
b. Territoire choisi pour l'étude : afin de maximiser le nombre de tweets cueillis, nous avons choisi la région la plus populeuse du Canada; l'application du modèle à d'autres régions du Canada pourrait résulter en un échantillon trop petit pour la détection efficace d'événements sur un territoire donné.
c. Texte seulement : la méthode de détection d'événement utilise uniquement le texte des tweets comme source de données. De nombreux tweets sont accompagnés d'adresse url et de photos/vidéos qui pourraient être utiles pour en savoir davantage sur un événement donné.
d. Traitement des données en langue anglaise : le modèle devrait être ajusté pour tenir compte du traitement en langue française pour son utilisation au Québec.
e. Modèle hors ligne et traitements en lots : pour simplifier le processus, le modèle opère hors ligne et traite les données en lots à chaque étape. Cette approche ne reflète pas les réalités opérationnelles rencontrées dans le domaine policier.

f. Pas d'événement majeur durant la période de cueillette : le modèle n'a malheureusement pu être testé lors d'une crise majeure, ce qui aurait augmenté considérablement l'apport des individus par rapport aux services publics. Ceci explique la concentration élevée de tweets par usager dans les grappes d'événements.
g. Détection non automatique : nous avons utilisé la lecture à vue des grappes pour identifier les grappes contenant deux tweets ou plus ayant le même sujet. Une automatisation de l'identification des grappes d'événements deviendrait nécessaire pour la mise en place d'un modèle opérationnel pour les services de police municipaux.

Les principales faiblesses de la recherche présentée dans cette étude sont les suivantes :

a. Le modèle ignore les tweets ne contenant pas de noms de rue; ceux-ci pourraient être importants dans la détection d'un événement de sécurité publique.
b. Le filtrage initial des tweets avec seulement les noms de rue ignore les toponymes et les ambiguïtés dans les noms de rue causées par les noms communs et par les fautes d'orthographe des usagers.
c. La période de collecte se trouve en milieuhttp://owlshead.com/ d'été ce qui pourrait diminuer la représentativité de l'échantillon utilisé par rapport à un échantillon colligé lors d'une période plus 'normale' où les gens ne sont pas en vacances.
d. Les grappes d'événements peuvent contenir plusieurs tweets provenant du même usager, ce qui diminue la précision de corroboration.
e. L'imposition par le modèle de la contrainte de formation de grappes pour détecter les événements écarte la possibilité qu'un événement soit détecté à partir de certains tweets individuels non en grappe.

Plusieurs éléments pourraient faire l'objet de recherches futures afin d'améliorer le modèle :

Cueillette simultanée

Plusieurs sessions de cueillette pourraient être exécutées simultanément, par exemple par polygone de capture et par mots clés de façon à maximiser le nombre de tweets cueillis sur un territoire.

Géocodage

Le taux de géocodage pourrait être amélioré avec un traitement permettant de distinguer les mots communs utilisés comme nom de rue au lieu d'ignorer ceux-ci. De même, la précision du géocodage pourrait être augmentée en développant davantage la modélisation pour les tweets contenant une (1) et trois (3) rues et plus.

Automatisation de la détection d'événement

Pour ce projet, l'identification des tweets communs parmi les grappes créées s'est effectuée par lecture à vue; la possibilité d'automatiser leur identification et ainsi en dégager directement les grappes d'événements bonifierait le système.

Tweets géocodés non attribués à une grappe

Plusieurs tweets pertinents mais ne contenant pas de nom de rue ont été ignorés par le modèle. Une analyse linguistique plus poussée du contenu des messages pourrait dégager des éléments d'intérêt pour les analystes des services de police.

Exécution du modèle en ligne

Pour les besoins du projet actuel, le traitement des tweets a été effectué hors ligne. Dans un cadre opérationnel, le modèle devrait pouvoir fonctionner en ligne de façon continue pour détecter les grappes d'événement.

Incorporer les photos et les vidéos

Tel que suggéré par de nombreux gestionnaires de crise lors du sondage de MacEachren *et al* .(2011), l'ajout de photos et de vidéos lors de l'affichage des messages contribuerait à bonifier le contenu informatif des tweets liés à la détection d'événement.

Étendre le modèle à d'autres médias sociaux

Plusieurs autres médias sociaux offrent un API pour cueillir des messages à partir de mots-clés ou d'autres paramètres. En particulier, les réseaux Flickr et Instagram donnent accès à une bonne proportion des messages publics circulant sur leur plateforme.

Quant aux retombées de ce projet, étant donné qu'il existe déjà un certain intérêt pour l'application de la recherche touchant la vigie des réseaux sociaux par les services policiers du Québec et du Canada, il est probable et souhaité que les résultats de ce projet seront divulgués dans le milieu et que, si ceux-ci sont probants, que le modèle soit à son tour appliqué au Québec. Éventuellement, le modèle pourrait même être utilisé comme une centrale d'alertes pour l'ensemble des corps policiers avec un système intégré

d’envoi d’alertes par courriel. Le modèle pourrait aussi être appliqué à d’autres secteurs tels que la santé publique et la sécurité civile.

Références

Abel, F., Hauff, C., Houben, G., Stronkman, R. et Tao, K. (2012) Semantics + Filtering + Search = Twitcident Exploring Information in Social Web Streams. Proceedings of the 23rd ACM International Conference on Hypertext and Social Media, 2012.

Ajao, O. et Liu,W. (2015) A Survey of Location Inference Techniques on Twitter. Journal of Information Science Vol. 41- 6, p. 855-864.

Arcaini, P., Bordogna, G., Ienco, D. et Sterlacchini, S. (2016) User-Driven Geo-Temporal Density-Based Exploration of Periodic and Not Periodic Events Reported in Social Networks. Information Sciences, 340–341, p. 122–143.

Atefeh, F. et Khreich, W. (2015) A Survey of Techniques for Event Detection in Twitter. Computational Intelligence, Vol. 31, no. 1

Bendler, J., Ratku A., et Neumann, D. (2014) Crime Mapping through Geo-Spatial Social Media Activity. Thirty Fifth International Conference on Information Systems, p. 1–16.

Bruns, A. (2012) Quantitative Approaches to Comparing Communication Patterns on Twitter. p. 160–185.

Burbary, K. (2016) A Wiki of Social Media Monitoring Solutions.
http://wiki.kenburbary.com/

Cheng, T. et Wicks, T. (2014) Event Detection Using Twitter : A Spatio-Temporal Approach. PLoS One, Vol. 9, no. 6 .

Chung-hong, L. (2012) Expert Systems with Applications Mining Spatio-Temporal Information on Microblogging Streams Using a Density-Based Online Clustering Method. Expert Systems With Applications, Vol. 39, no. 10, p. 9623–9641.

Edwards M., Awais R. et Rayson, P. (2015) A Systematic Survey of Online Data Mining Technology Intended for Law Enforcement. ACM Computing Surveys, Vol. 48, Article 15.

Forum Research Inc. (2015) Forum Poll ™, Janvier 2015.
http://poll.forumresearch.com/post/213/facebook-leads-in-penetration-linkedin-shows-most-growth/
http://poll.forumresearch.com/data/Federal%20Social%20Media%20News%20Release%20(2015%2001%2006)%20Forum%20Research.pdf

Gelernter, J. et Balaji, S. (2013) An Algorithm for Local Geoparsing of Microtext. Geoinformatica, Vol. 17, p. 635–667.

Gerber, M.S. (2014) Predicting Crime Using Twitter and Kernel Density Estimation. Decision Support Systems, Vol. 61, no. 1, p. 115–125.

Gouvernement de l'Ontario (2015) Réseau routier de l'Ontario en 2010 : tronçon avec adresse Ministère des richesses naturelles et des forêts de l'Ontario.
https://www.ontario.ca/fr/donnees/reseau-routier-de-lontario-troncon-avec-adresse

Gu, Y., Zhen, S. et Chen, F. (2016) From Twitter to Detector : Real-Time Traffic Incident Detection Using Social Media Data. Transportation Research, Part C 2016 p. 321-342.

IACP- International Association of Chiefs of Police (2015), 2015 Social Media Survey.

Jurgens, D., Finnethy, T., Mccorriston, J., Tian, X. et Ruths, D. (2015) Geolocation Prediction in Twitter Using Social Networks : A Critical Analysis and Review of Current Practice. Proceedings of the 9th International AAAI Conference on Weblogs and Social Media (ICWSM) 2015, Association for the Advancement of Artificial Intelligence.

LexisNexis Risk Solutions (2014) Survey of Law Enforcement Personel and Their Use of Social Media.
http://www.lexisnexis.com/risk/downloads/whitepaper/2014-social-media-use-in-law-enforcement.pdf

MacEachren, A., Jaiswal, A., Robinson, A., Pezanowski, S., Savelyev, A. et Mitra, P. (2011) SensePlace2: GeoTwitter analytics support for situational awareness. Proceedings of the Visual Analytics Science and Technology (VAST), IEEE Conference, 2011.

Maxwell, D., Raue, S., Azzopardi, L., Johnson, C. et Oates,S. (2012) Crisees: Real-time monitoring of social media streams to support crisis management. Advances in Information Retrieval, 2012.

Morstatter, F., Pfeffer, J., Liu, H. et Carley, K .M. (2013) Is the Sample Good Enough ? Comparing Data from Twitter's Streaming API with Twitter's Firehose. Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media, p. 400–408.

Motherboard (11 octobre 2016) Facebook, Instagram, Twitter block tool for cops to surveil you on social media.
http://motherboard.vice.com/read/facebook-instagram-and-twitter-block-geofeedia

Rogstadius, J., Kostakos, V., Laredo, J. et Vukovic, M. (2011) A real-time social media aggregation tool: Reflections from five large-scale events. Proceedings of the European Conference on Computer-Supported Cooperative Work (ECSCW), 2011.

RT.com (11 octobre 2016) Twitter drops Geofeedia over claims it helped police spy on protesters.
https://www.rt.com/usa/362422-twitter-geofeedia-protest-spying/

Sankaranarayanan, J., Teitler, B., Samet, H., Lieberman, M.D. et Sperling, J. (2009) TwitterStand : News in Tweets. Proceedings ACM GIS 2009, Association for Computing Machinery.

Shakira, K. et Abdolreza, A. (2014) Cluster-Discovery of Twitter Messages for Event Detection and Trending. Journal of Computational Science, Vol. 6, p. 47–57.

Statistique Canada (2013) Tableaux thématiques du recensement du Canada de 2011 (âge et sexe de la population).

Studies, J., Reuter, C., Ludwig, T., Kaufhold, M.A. et Spielhofer, T. (2016) Emergency Services ' Attitudes towards Social Media : A Quantitative and Qualitative Survey across Europe. Journal of Human Computer Studies, Volume 95, p. 96–111.

Sykora, M., Jackson, T.W., O'Brien, A. et Elayan, S. (2013) EMOTIVE Ontology: Extracting fine-grained emotions from terse, informal messages. IADIS Intelligent Systems and Agents Conference, 2013.

TalkEnglish.com (2016) Top 2000 Vocabulary Words. British National Corpus (BNC top 3000) and Corpus of Contemporary American English (COCA) top 5 000. http://www.talkenglish.com/vocabulary/top-2000-vocabulary.aspx

TheNextWeb.com (2015) Twitter to cut off firehose resellers as it brings data access fully in-house. http://thenextweb.com/dd/2015/04/11/twitter-cuts-off-firehose-resellers-as-it-brings-data-access-fully-in-house/#gref

Tzelepis, C., Zhigang M., Vasileios M., Ionescu, B. et Kompatsiaris, I. (2016) Event-Based Media Processing and Analysis : A Survey of the Literature. IMAVIS 53 p. 3–19.

Vieweg, S., Hughes, A.L., Starbird, K. et Palen, L.(2010) Microblogging During Two Natural Hazards Events : What Twitter May Contribute to Situational Awareness. CHI Atlanta 2010 conference, ACM.

Wang, X., Gerber, M. et Brown, D.E. (2012) Automatic Crime Prediction Using Events Extracted from Twitter Posts. Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 7227 LNCS.

Annexe 1- Mots clés utilisés pour la cueillette de données

La liste de mots clés utilisés inclut des lemmes.

Accidents et incendies	lane, collis, vehicl, car, truck, tractor, semi, semi-trailer, trailer, block, eb, wb, sb, nb, collector, eastbound, northbound, southbound, westbound, express, hwy, highway, ramp, disabl, shoulder, debri, traffic, spill, injur, construct,tire, wheel, roadway, stall, accid, closur, crash, roadwork, fire
Incidents criminels	armed, arson, assailant, assault, attack, attempt, blast, bomb, brawl, break-in, bullet, burglar, detonat, die, explosion, explosive, fatal, felony, firebomb, gang, gun, gunshot, homicid, hostage, injur, kidnap, kill, knife, larceny, pistol, rape, rifle, riot, robber, shoot, shot, slash, stab, streetgang, suicid, theft, threat, thwart, vandal, victim, violen, weapon, witness

Annexe 2- Encodage du modèle en langage Python

Annexe 2A- Cueillette de données avec mots clés et module TWEEPY

```
# -*- coding: utf-8 -*-
__author__ = 'donald'
# IMPORTATION DES LIBRAIRIES ET MODULES
import tweepy
import sys
from tweepy import Stream
from tweepy import OAuthHandler
from tweepy.streaming import StreamListener
import time
import datetime


# PARAMÈTRES D'ACCÈS AU FLUX TWITTER
ckey='VOTRE CODE1'
csecret='VOTRE CODE2'
atoken='VOTRE CODE3'
asecret='VOTRE CODE4'

# TRAITEMENT ET SAUVEGARDE DES TWEETS
class listener(StreamListener):
   def on_data(self, data):
      try:
         if data:

                  # RECHERCHE DE MOTS DE LOCALISATION

               if data.find('Ontario')!=-1 OR data.find('Toronto')!=-1 :

                   tweet=data.split(',"text":"')[1].split('","source":')[0]
                      # ON NE TRAITE PAS LES RETWEETS (« RT ») : SAUVEGARDE DES
                      CHAMPS DANS UNE CHAÎNE
                   if tweet[:2]!='RT':
                            TimeStamp=time.time()

                           ReadableTime=datetime.datetime.fromtimestamp(TimeStamp).strftime('%Y-
                   %m-%d %H:%M:%S')

                   user_screen_name=data.split(',"screen_name":')[1].split(',"location":')[0]

                   followers=data.split(',"followers_count":')[1].split(',"friends_count":')[0]
                                    location=data.split(',"location":')[1].split(',"url":')[0]
```

```
                    saveThis=str(ReadableTime)+'='+tweet+'='+user_screen_name+'='+followers+'='+loc
                    ation
                    # OUVERTURE ET SAUVEGARDE DES CHAÎNES DANS UN FICHIER
                    saveFile=open('Tweets_July_24_8.csv','a')
                    saveFile.write(saveThis)#Message)
                    saveFile.write('\n')
                    saveFile.close()
                    return True
            except BaseException,e:
                print 'failed ondata,',str(e)
                time.sleep(5)
            except TweepError as e:
                if 'Failed to send request:' in e.reason:
                    print "Time out error caught."
                    time.sleep(180)

            return True

        def on_error(self, status):
            print status

        def on_timeout(self):
            print >> sys.stderr, 'Timeout...'
            return True

# INSTRUCTIONS D'ACCÈS AU FLUX TWITTER VIA TWEEPY

auth= OAuthHandler(ckey,csecret)
auth.set_access_token(atoken,asecret)
twitterStream=Stream(auth,listener())


# LISTE DES MOTS CLÉS
key_words=

"armed"+","+"arson"+","+"assailant"+","+"assault"+","+"attack"+","+"attempt"+","+\
"blast"+","+"bomb"+","+"brawl"+"breakin"+","+"bullet"+","+"burglar"+","+"detonat"+","+\"die"+","+"ex
plosion"+","+"explosive"+","+"fatal"+","+\
"felony"+","+"fire"+","+"firebomb"+","+"gang"+","+"gun"+","+"gunshot"+","+"homicid"+","+
"hostage"+","+"injur"+","+"kidnap"+","+"kill"+","+"knife"+","+"larceny"+","+\
"pistol"+","+"rape"+","+"rifle"+","+"riot"+","+"robber"+","+\
"shoot"+","+"shot"+","+"slash"+","+"stab"+","+"streetgang"+","+"suicid"+","+\
"theft"+","+"threat"+","+"thwart"+","+"vandal"+","+"victim"+","+"violen"+","+"weapon"+","+"witness"+
","+\
"lane"+","+"collis"+","+"vehicl"+","+"car"+","+"truck"+","+"tractor"+","+"semi"+","+"semi-trailer"+","+\
"trailer"+","+"block"+","+"eb"+","+"wb"+","+"sb"+","+"nb"+","+"collector"+","+"eastbound"+","+"north
bound"+","+"southbound"+","+"westbound"+","+"express"+","+"hwy"+","+"highway"+","+"ramp"+","+"
```

```
disabl"+","+"shoulder"+","+"debri"+","+"traffic"+","+"spill"+","+"injur"+","+"construct"+","+"tire"+","+"
wheel"+","+"roadway"+","+"stall"+","+"accid"+","+"closur"+","+"crash"+","+"roadwork"

twitterStream.filter(track=[key_words],languages=["en"])
stream.filter(track=[t], stall_warnings=True)
```

Annexe 2B- Filtrage des tweets cueillis et géocodage

```
# -*- coding: utf-8 -*-
__author__ = 'donald'

from __future__ import division
import nltk
import string
import csv,sys
from collections import Counter
from nltk.corpus import stopwords
from nltk.stem.porter import *
from nltk.util import ngrams
from urllib2 import Request, urlopen, URLError
import json
import time
import datetime
from datetime import datetime
from datetime import timedelta
import math
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.basemap import Basemap


# DÉCLARATION DES VARIABLES
cum_scored_words=int()
cum_traffic=int()
cum_crime=int()
cum_detects_places=int()
cum_detects_2words=int()
US_tweets=int()
cum_tweets_scored=int()
cum_1_street_found=int()
cum_2_streets_found=int()
cum_3_streets_found=int()
cum_4_streets_plus_found=int()
```

```
cum_geocoded=int()
cum_outarea=int()
cum_tweets_read=int()
geocoded_lats_traffic=[]
geocoded_lons_traffic=[]
geocoded_lats_crime=[]
geocoded_lons_crime=[]
cum_score_low=int()
cum_score_mid=int()
cum_score_high=int()
cum_geocode_calls=int()
scored_wordlist=str()
cum_scored_words=int()
score=float()
traffic_list_all=[]
crime_list_all=[]
traffic_list_geo=[]
crime_list_geo=[]
coordinates=str()
traffic_clusters=[]
crime_clusters=[]


# DÉCLARATION DE L'OUTIL DE LEMMATISATION (PORTER)
stemmer = PorterStemmer()

# FONCTIONS DE LEMMATISATION
def get_tokens(file):
  with open(file, 'r') as mytweets:
   text = mytweets.read()

        # TRANSFORMER LE TEXTE EN MINUSCULES
   lowers = text.lower()
   # SUPPRIMER LA PONCTUATION
   no_punctuation = lowers.translate(None, string.punctuation)
   tokens = nltk.word_tokenize(no_punctuation)
   return tokens

def stem_tokens(tokens, stemmer):
```

```
    stemmed = []
    for item in tokens:
        try:
            stemmed.append(stemmer.stem(item))
        except BaseException,e:
            print 'failed ondata,',str(e)
    return stemmed

def stem_tweet(self):
    lowers = self.lower()
    # SUPPRIMER LA PONCTUATION
    no_punctuation = lowers.translate(None, string.punctuation)
    tokens_tweet = nltk.word_tokenize(no_punctuation)
    # SUPPRIMER MOTS QUI NE SONT PAS DES VERBES, NOMS OU ADJECTIFS (stopwords)
    filtered = [w for w in tokens_tweet if not w in stopwords.words('english')]
    return stem_tokens(filtered, stemmer)


# FONCTION POUR TROUVER LAT/LONG AVEC L'API DE GOOGLE MAPS
def geocode(value1,value2,mode):
    global cum_geocoded
    global cum_outarea
    global cum_geocode_calls
    cum_geocode_calls+=1
    if mode=='intersection':
        request = Request('https://maps.googleapis.com/maps/api/geocode/json?address='+

                          value1+'+and+'+value2+',+ON&key=’VOTRE CODE’’)
    if mode=='address' or mode=='route':
        request = Request('https://maps.googleapis.com/maps/api/geocode/json?address='+

                          value1+'+'+value2+',+ON&key=’VOTRE CODE’’)
    try:
        response = urlopen(request)
        end_result = response.read()
        dict=json.loads(end_result)

            # VÉRIFIER SI LIMITE QUOTIDIENNE DE GÉOCODAGE ATTEINTE
        if dict.get("status")=='OVER_QUERY_LIMIT':
```

```
            print 'GOOGLE API OVER DAILY LIMIT!!'
            print ""
            print 'Total tweets read: ',cum_tweets_read
            print 'Total tweets scored: ',cum_tweets_scored
            print 'Geocode calls to API: ',cum_geocode_calls
            print "tweets geocoded: ",cum_geocoded
            print 'Traffic/fire tweets: ',cum_traffic
            print 'Crime tweets: ',cum_crime
            print "tweets with coordinates out of range: ",cum_outarea

              # VÉRIFIER SI RÉPONSE À LA REQUÊTE ET SAUVEGARDE DES RÉSULTATS
        if dict.get("status")!='ZERO_RESULTS':
            coords=dict.get("results")[0].get("geometry").get("location")
            type_location=dict.get("results")[0].get("types")
            lat_lon=coords.get('lat'),coords.get('lng')
            # VÉRIFIER SI LES COORDONNÉES SONT SITUÉES DANS LE TERRITOIRE VISÉ DE
TORONTO-NIAGARA
            if 42.763297<=coords.get('lat')<=44.109850 and -80.521871<=coords.get('lng')<=-79.120585:
              # AJOUT À LA LISTE DES COORDONNÉES POUR LA CARTOGRAPHIE SELON LE TYPE
D'INCIDENT
              if type==1:
                geocoded_lats_traffic.append(coords.get('lat'))
                geocoded_lons_traffic.append(coords.get('lng'))
              if type==2:
                geocoded_lats_crime.append(coords.get('lat'))
                geocoded_lons_crime.append(coords.get('lng'))
              cum_geocoded+=1
              return lat_lon
            else:
              cum_outarea+=1
      except URLError, e:
        print 'Got an error code:', e
        return None


# FONCTION DE TRAITEMENT DES NOMS DE RUES COMPOSÉS
def list_2words(self):
        # UTILISATION DE BIGRAMMES POUR TROUVER LES NOMS DE RUES COMPOSÉS
```

```
        bigrams=list(ngrams(self,2))
        new_bigrammed_tweet=[]
        for grams in bigrams:
            new_bigrammed_tweet.append((' '.join([w for w in grams])).strip())

        # DÉTECTION DES RUES AYANT DEUX MOTS
        tw1=set(new_bigrammed_tweet)
        kw1=set(streetslist)
        if tw1.intersection(kw1)<>set([]):
            last_tweet_content=stemmed_tweet
            # SUPPRIMER MOTS INDIVIDUELS ET REMPLACER AVEC NOMS COMPOSÉS DE RUES
            for i in range(0,len(tw1.intersection(kw1))):
                element=list (tw1.intersection(kw1))[i]
                two_words=element.split(" ")
                # VÉRIFIER QUE LES MOTS FIGURENT DANS LA LISTE À JOUR AVANT LA SUPPRESSION
                if (two_words[0])in new_list:
                    # MÉMORISER L'INDICE POUR INSÉRER DANS LE NOM COMPOSÉ DES RUES
                    # DANS LE CAS DE RECHERCHE POUR UNE ADRESSE (TWEETS À UNE RUE
SEULEMENT)
                    pos=new_list.index(two_words[0])
                    new_list.remove(two_words[0])
                if (two_words[1])in new_list:
                    new_list.remove(two_words[1])
                new_list.insert(pos,element)


# TRAITEMENT LINGUISTIQUE ET SCORES DE MOTS CLÉS

def keywords_scoring(tweet_text,keyword_list):
    global scored_wordlist
    global cum_scored_words
    global score
    scored_wordlist=""
    cum_scored_words=0
    score=0
    text_wordcount=tweet_text.split()
    stemmed_tweet1 = stem_tweet(tweet_text)
    tw=set(stemmed_tweet1)
```

```
    kw=set(keyword_list)
    if tw.intersection(kw)<>set([]):
        for item in tw:
            if item in keyword_list:
                scored_wordlist=scored_wordlist+item+", "
                cum_scored_words+=1
        # VÉRIFIER PERTINENCE DU TWEET (NOMBRE DE MOTS 'SCORED' VS. TOTAL DE MOTS)
        if len(text_wordcount)!=0:
            score=round(cum_scored_words/len(text_wordcount),3)
        return cum_scored_words,score,scored_wordlist
    else:
        return None
#DÉFINITION DU TYPE D'INCIDENT ('1' ÉTANT 'ACCIDENT-INCENDIE' ET '2' 'CRIME')
def find_type(result1,result2):
    if result1!=None and result2!=None:
        # compare number of scored words
        if result1[0]>result2[0]:
            return 1
        else:
            if result1[0]==result2[0]:
                # compare relevance score
                if result1[1]>result2[1]:
                    return 1
                else:
                    return 2
            else:
                return 2
    else:
        if result1!=None:
            return 1
        else:
            if result2!=None:
                return 2
# TROUVER LA GÉOLOCALISATION D'UN TWEET
def find_location(tw2,kw2):
    global cum_1_street_found
    global cum_2_streets_found
```

```
    global cum_3_streets_found
    global cum_4_streets_plus_found

# SI UNE(1) RUE DÉTECTÉE
    if len(tw2.intersection(kw2))==1:
        cum_1_street_found+=1
        strt_name=list(tw2.intersection(kw2))[0]
        position=new_list.index(strt_name)
        strt_name=list(tw2.intersection(kw2))[0].replace(" ","_")
        # VÉRIFIER S'IL Y A UNE ADRESSE PRÉCÉDENT LE NOM DE RUE
        if new_list[position-1].isdigit()==True:
            coordinates=geocode(new_list[position-1],strt_name,'address')
            if coordinates!=None:
                return True,'1',coordinates
            else:
                return False,'1',0
        # VÉRIFIER S'IL Y A UN DESCRIPTIF SUIVANT LE NOM DE RUE
        descriptor_list=['street','st','avenue','ave','blvd','bl','road','rd']
        if position<=len(new_list)-2:
            value=new_list[position+1]
            if new_list[position+1].lower() in descriptor_list:
                coordinates=geocode(strt_name,new_list[position+1],'route')
                if coordinates!=None:
                    return True,'199',coordinates
                else:
                    return False,'199',0
        else:
            return False,'199',0
# SI DEUX (2) RUES DÉTECTÉES
    if len(tw2.intersection(kw2))==2:
        cum_2_streets_found+=1
        #replace space with '_' for Google API
        strt_a=list(tw2.intersection(kw2))[0].replace(" ","_")
        strt_b=list(tw2.intersection(kw2))[1].replace(" ","_")
        coordinates= geocode(strt_a,strt_b,'intersection')
        if coordinates!=None:
            return True,'2',coordinates
        else:
```

```
            return False,'2',0
    # SI TROIS (3) RUES DÉTECTÉES
        if len(tw2.intersection(kw2))==3:
            cum_3_streets_found+=1
            strt_a=list(tw2.intersection(kw2))[0].replace(" ","_")
            strt_b=list(tw2.intersection(kw2))[1].replace(" ","_")
            strt_c=list(tw2.intersection(kw2))[2].replace(" ","_")
            coordinates=geocode(strt_a,strt_b,'intersection')
            if coordinates!=None:
                return True,'3',coordinates
            else:
                coordinates=geocode(strt_b,strt_c,'intersection')
                if coordinates!=None:
                    return True,'3',coordinates
                else:
                    coordinates=geocode(strt_a,strt_c,'intersection')
                    if coordinates!=None:
                        return True,'3',coordinates
                    else:
                        return False,'3',0
    # SI PLUS DE TROIS (3) RUES DÉTECTÉES
        if len(tw2.intersection(kw2))>3:
            cum_4_streets_plus_found+=1
            return False,'4+',0


    # FONCTION DE CALCUL DE DISTANCE ENTRE TWEET EN COURS ET GRAPPES OU AUTRES
    TWEETS
    def calculate_distance(lat1, lon1, lat2, lon2):
            # Calcul de la distance entre 2 points (lat,long) en présumant que la terre est sphérique
            # Utilise la loi sphérique des cosinus (http://en.wikipedia.org/wiki/Spherical_law_of_cosines):
            R = 6371  # rayon de la terre en km
            if ((lat1 == lat2) and (lon1 == lon2)):
                return 0
            try:
                delta = lon2 - lon1
                a = math.radians(lat1)
                b = math.radians(lat2)
```

```
        C = math.radians(delta)
        x = math.sin(a) * math.sin(b) + math.cos(a) * math.cos(b) * math.cos(C)
        distance = math.acos(x)*R # En radians
        return distance;
    except:
        return 0


scored_wordlist=str()
wordlist=str()

#****************Liste de mots clés pour détection*********************
traffic_keywords=['lane','collis','vehicl','car','truck','tractor','semi','semi-
trailer','trailer','block','eb','wb','sb','nb',

'collector','eastbound','northbound','southbound','westbound','express','hwy','highway','ramp','fire'
,
            'disabl','shoulder','debri','traffic','spill','injur','construct',
            'tire','wheel','roadway','stall','accid','closur','crash','roadwork']
#EXCLUS: 'rd','road','avenu','street','st','blvd','ave','west','east','north','south'

crime_keywords=['911','alarm','alert','ambul','ambush','arm','armor','arrest','arsen','arson',
            'assail','assassin','assault','attack','attempt','beat','blast','bomb','brawl','break','breakin',
            'bullet','burglar','cocain','cop','crime','danger','dead','demonstr','detain',
            'detect','detonat','disturb','doa','drug','enforc','explos','fatal',
            'feloni','firebomb','gang','gun','gunshot','harass','hit','hitandrun','homicid',
            'hooligan','hostag','incid','injur','kidnap','kill','knife','larceni','lethal',

'meth','missingperson','murder','param','pistol','polic','protest','quarrel','rape','rifl','riot','rob',
            'sabotag','shoot','shot','slash','stab','steal','streetgang','suicid','suspect',
            'terror','theft','threat','thwart','vandal','victim','violen','weapon']


#****************LISTE DE NOMS D'USAGERS EXCLUS**************************
exclude_userlist=[]#'tofire','tpscalls']#'tofire','tofireS','tofireE','tofireW','tofireN"OPP']

#***************PRÉPARER LISTE DES NOMS DE RUES************************
```

```
streetslist=[]
with open('C:\Users\donald\Documents\Corpus\Ontario_Streets.csv', 'rb') as csvfile:
    # lecture du fichier csv avec délimiteur 'virgule' et caractère de chaîne 'guillemet'
    tweet_reader = csv.reader(csvfile, delimiter=';', quotechar='"')
    # parcours des lignes créées par la fonction csv.reader
    for row in tweet_reader:
        if row[0]<>"":
            placename=row[0]
            lowers = placename.lower()
            streetslist.append(placename)




 #********** TRAITEMENT DU TWEET EN COURS **********

with open('C:\Users\donald\Documents\Corpus\Tweets_July_21_12 EVALUATION.csv', 'rb') as
csvfile:
    tweet_reader = csv.reader(csvfile, delimiter=';', quotechar='"')
    # REMISE À ZÉRO DU COMPTEUR DE LIGNES
    nbr_rows=0
    for row in tweet_reader:
            cum_tweets_read+=1
            # VÉRIFIER SI LIMITE ATTEINTE POUR L'API DE GOOGLE
            if cum_geocode_calls==2499:
                print 'Reached Google Geocoding limit...'
                break
           # EXCLURE USAGERS DÉSIGNÉS
            if row[3] in exclude_userlist:
                continue
            new_list=[]
            # VÉRIFIER SI LE MESSAGE EXISTE
            if row[1]<>"":
                text1=row[1]
                nbr_rows=nbr_rows+1
                if nbr_rows>50000:
                    break
            else:
```

```
            continue
        # IGNORER LES 'RT' (retweetS)
        if text1[:3]=='#RT':
            continue
        # IGNORER SI TWEET CONTIENT UN url
        if text1.find('http')!=-1:
            continue
        # IGNORER SI DOUBLON
        find=False
        for sublist in traffic_list_all:
            if sublist[5]==text1:
                find=True
        for sublist in crime_list_all:
            if sublist[5]==text1:
                find=True
        if find==True:
            continue
        # VÉRIFIER POUR TYPE D'INCIDENT
        result1=keywords_scoring(text1,traffic_keywords)
        result2=keywords_scoring(text1,crime_keywords)
        type=find_type(result1,result2)
        if type==1:
            scored_words=result1[0]
            relevance=result1[1]
            wordlist=result1[2]
            # AJOUTER TWEET À CHAQUE LISTE POUR ANALYSE STATISTIQUE

traffic_list_all.append((nbr_rows,'=',row[0],'=',row[2],'=',row[3],'=',row[4],'=',text1,'=',relevance,'=',scor
ed_words,'=',wordlist))
        if type==2:
            scored_words=result2[0]
            relevance=result2[1]
            wordlist=result2[2]
            #add tweet to general crime listing for stats analysis

crime_list_all.append((nbr_rows,'=',row[0],'=',row[2],'=',row[3],'=',row[4],'=',text1,'=',relevance,'=',scor
ed_words,'=',wordlist))
```

```
        if type!=None:
            cum_tweets_scored+=1
            # replace specific characters to track 2 word street or place names
            text=text1.replace('@',' intersect1 ').replace(' and ',' intersect_and ').replace(' at ',' intersect_at ').replace('b\/w',' between ').replace('/',' intersect/ ').replace(' X ',' intersectX ')
            tokens_tweet1 = nltk.word_tokenize(text)
            no_stop_words=[w for w in tokens_tweet1 if not w in stopwords.words('english')]
            new_list=no_stop_words
            # VÉRIFIER POUR NOMS DE RUE COMPOSÉS: SI TROUVÉ, MODIFIER DANS NOUVELLE LISTE
            list_2words(new_list)
            # IDENTIFIER LES NOMS DE RUE DANS LE TWEET
            if new_list<>[]:
               tw2=set(new_list)
            else:
               continue
            kw2=set(streetslist)
            if tw2.intersection(kw2)<>set([]):
               cum_detects_places+=1
               location=find_location(tw2,kw2)
               if location<>None:
                  if location[0]==True:
                     if score<=0.050:
                        cum_score_low+=1
                     if 0.050<score<=0.080:
                        cum_score_mid+=1
                     if 0.080<score:
                        cum_score_high+=1
                     if type==1:
                        cum_traffic+=1
                        # add tweet to list

traffic_list_geo.append((row[0],'=',nbr_rows,'=',row[2],'=',row[3],'=',row[4],'=',text1,'=',relevance,'=',scored_words,'=',wordlist,'=',location[1],'=',list(tw2.intersection(kw2)),'=',location[2]))

                     if type==2:
```

```
                    cum_crime+=1
                # AJOUTER LE TWEET À LA LISTE

crime_list_geo.append((row[0],'=',nbr_rows,'=',row[2],'=',row[3],'=',row[4],'=',text1,'=',relevance,'=',sco
red_words,'=',wordlist,'=',location[1],'=',list(tw2.intersection(kw2)),'=',location[2]))

        stemmed_tweet=[]
        cum_scored_words=0
        scored_wordlist=""
print ""
print 'Total tweets read: ',cum_tweets_read
print 'Total tweets scored: ',cum_tweets_scored
print "tweets geocoded: ",cum_geocoded
print 'Traffic/fire tweets: ',cum_traffic
print 'Crime tweets: ',cum_crime
print "tweets with coordinates out of range: ",cum_outarea
print ""
print "tweets detected with streets:",cum_detects_places
print 'Tweets with 1 street: ',cum_1_street_found
print 'Tweets with 2 streets: ',cum_2_streets_found
print 'Tweets with 3 streets: ',cum_3_streets_found
print 'Tweets with 4 streets * more: ',cum_4_streets_plus_found
print ""
print 'score <=0,05: ',cum_score_low
print '0.05<score <=0,08: ',cum_score_mid
print 'score >0,08: ',cum_score_high
print 'total scores: ',cum_score_low+cum_score_mid+cum_score_high


# SAUVEGARDER LES APPELS À L'API DE GOOGLE DANS UN FICHIER
saveFile=open('Google_geocoding.csv','a')
TimeStamp=time.time()
ReadableTime=datetime.fromtimestamp(TimeStamp).strftime('%Y-%m-%d %H:%M:%S')
saveThis=str()
saveThis=ReadableTime+", "+'Geocode calls: '+str(cum_geocode_calls)
saveFile.write(saveThis)#Message)
saveFile.write('\n')
```

```
saveFile.close()

# SAUVEGARDER LA LISTE DES TWEETS GÉOCODÉS DANS UN FICHIER
Add_Time=datetime.fromtimestamp(TimeStamp).strftime('%Y-%m-%d %H')
saveFile=open('Tweets_traffic_geo_'+Add_Time+'.csv','a')
for item in traffic_list_geo:
    saveFile.write(str(item))
    saveFile.write('\n')
saveFile.close()

saveFile=open('Tweets_crime_geo_'+Add_Time+'.csv','a')
for item in crime_list_geo:
    saveFile.write(str(item))
    saveFile.write('\n')
saveFile.close()

# SAUVEGARDER LA LISTE DES TWEETS DÉTECTÉS DANS UN FICHIER

Add_Time=datetime.fromtimestamp(TimeStamp).strftime('%Y-%m-%d %H')
saveFile=open('Tweets_traffic_all_'+Add_Time+'.csv','a')
for item in traffic_list_all:
    saveFile.write(str(item))
    saveFile.write('\n')
saveFile.close()

saveFile=open('Tweets_crime_all_'+Add_Time+'.csv','a')
for item in crime_list_all:
    saveFile.write(str(item))
    saveFile.write('\n')
saveFile.close()
```

Annexe 2C- Formation des grappes

```
# -*- coding: utf-8 -*-
from __future__ import division

__author__ = 'donald'


import nltk
import string
import csv,sys
from collections import Counter
from nltk.corpus import stopwords
from nltk.stem.porter import *
from nltk.util import ngrams
from urllib2 import Request, urlopen, URLError
import json
import time
import datetime
from datetime import datetime
from datetime import timedelta
import math
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.basemap import Basemap


# DÉCLARATION DE VARIABLES

geotweet_list=[]
cluster_list=[]
coordinates=str()
cluster_matches=0
total_clusters=0
tweets_read=0
duplicates_existing=0
```

```
duplicates_new=0


# FONCTION DU CALCUL DE LA DISTANCE
def calculate_distance(lat1, lon1, lat2, lon2):
        # Calcul de la distance entre 2 points (lat,long) en présumant que la terre est sphérique
       # Utilise la loi sphérique des cosinus (http://en.wikipedia.org/wiki/Spherical_law_of_cosines):
        R = 6371  # rayon de la terre en km
        if ((lat1 == lat2) and (lon1 == lon2)):
            return 0
        try:
            delta = lon2 - lon1
            a = math.radians(lat1)
            b = math.radians(lat2)
            C = math.radians(delta)
            x = math.sin(a) * math.sin(b) + math.cos(a) * math.cos(b) * math.cos(C)
            distance = math.acos(x)*R # en radians
            return distance;
        except:
            return 0


# FONCTION DE CONVERSION DATE-HEURE EN SECONDES
def time_to_secs(self):
    time_tuple=time.strptime(self, "%Y-%m-%d %H:%M")
    time_sec=int(time.mktime(time_tuple))
    return time_sec


# FONCTION DE CONVERSION D'UNE CHAÎNE LATITUDE EN COORDONNÉES GÉO

def convert_lat(self):
    myString=str(self)
    startString='('
    endString=','
    lat=float(myString[myString.find(startString)+len(startString):myString.find(endString)])
    return lat


# FONCTION DE CONVERSION D'UNE CHAÎNE LONGITUDE EN COORDONNÉES GÉO
def convert_long(self):
```

```
    myString=str(self)
    startString=','
    endString='))'
    long=float(myString[myString.find(startString)+len(startString):myString.find(endString)])
    return long

# FONCTION IDENFIANT LES MOTS COMMUNS ENTRE LES GRAPPES (À FINALISER PLUS TARD)
def intersect(*d):
    sets = iter(map(set, d))
    result = sets.next()
    for s in sets:
        result = result.intersection(s)
    return result




# ********************** TRAITEMENT DES TWEETS GÉOCODÉS POUR LA FORMATION DE
GRAPPES *************

# PARAMÈTRES DE FORMATION DE GRAPPES (FIXÉS À 30 MINUTES ET À 3 KILOMÈTRES)
Tot_Minutes=int(raw_input("Enter time delay minutes: "))
Rayon_km=float(raw_input("Enter cluster radius in km.: "))
time_delay=Tot_Minutes*60 #En secondes

# LECTURE DES TWEETS GÉOCODÉS
with open('C:\Users\donald\Documents\Corpus\Tweets_traffic_geo_Evaluation.csv', 'rb') as csvfile:
    tweet_reader = csv.reader(csvfile, delimiter=';', quotechar='"')
    # CRÉER UNE LISTE POUR COMPARER LES TWEETS GÉOCODÉS
    for row in tweet_reader:
        geotweet_list.append(row)
    # remettre le compteur de lignes à zéro
    nbr_rows=0



# PARCOURS DE LA LISTE
```

```
for element in geotweet_list:
    if int(element[1])==234:
        cg=8
    next_geotweet=False
    tweets_read+=1
    curr_tweet_secs=time_to_secs(element[0])
    # COORDONNÉES DU TWEET EN COURS DE TRAITEMENT
    lat_a=convert_lat(element[11])
    long_a=convert_long(element[11])
    # COMPARER LE TWEET D'ABORD À LA LISTE COURANTE DES GRAPPES (SI AUCUN RÉSULTAT, COMPARER À LA LISTE COURANTE DE TWEETS)
    for clust in cluster_list:
        # DÉMARRER UN CUMUL
        cum_time=curr_tweet_secs
        cum_lat=lat_a
        cum_long=long_a
        # VÉRIFIER SI LE TWEET EST DÉJÀ DANS LA GRAPPE
        if element[1] in clust[3]:
            continue
        # IGNORER LES MESSAGES IDENTIQUES (DOUBLONS)
        if element[5] in clust[6]:
            duplicates_existing+=1
            next_geotweet=True
            break

            # APPLIQUER LE PARAMÈTRE DE DÉLAI ET, SI OK, CELUI DE LA DISTANCE
        if curr_tweet_secs-time_delay<=clust[1]<=curr_tweet_secs:
            lat_b=convert_lat(clust[2])
            long_b=convert_long(clust[2])
            Current_Distance=calculate_distance(lat_a,long_a,lat_b,long_b)
            if 0<Current_Distance<Rayon_km:
                # RECALCULER L'HEURE MOYENNE ET LE CENTRE DE GRAVITÉ DE LA GRAPPE CORRESPONDANTE
                cluster_matches+=1
                tup_1=eval(clust[4])
                for item in tup_1:
                    cum_time=cum_time+int(item)
                tup_2=eval(clust[5])
```

```
            for item in tup_2:
                cum_lat=cum_lat+float(item[0])
                cum_long=cum_long+float(item[1])
            if clust[0]>0:
                tot_tweets=clust[0]+1
                avg_time=cum_time/tot_tweets
                avg_lat=round(cum_lat/tot_tweets,4)
                avg_long=round(cum_long/tot_tweets,4)
            # CONVERSION DE TUPLE VERS LISTE POUR MODIFICATION
            clust[0]=clust[0]+1
            clust[1]=int(avg_time)
            clust[2]="("+str(avg_lat)+','+str(avg_long)+")"
            clust[3]=clust[3]+','+str(element[1])
            clust[4]=clust[4]+','+str(curr_tweet_secs)
            clust[5]=clust[5]+", ("+str(lat_a)+','+str(long_a)+")"
            clust[6]=clust[6]+element[5]
            next_geotweet=True
            break
    if next_geotweet==False:
        # AUCUNE GRAPPE CORRESPONDANTE TROUVÉE DONC: COMPARAISON DU TWEET EN
COURS AVEC AUTRES TWEETS DANS LA LISTE 'GEOLIST'

        cum_tweet_list=[]
        cluster_repeat=False
        cum_time=0
        cum_lat=0
        cum_long=0
        tweet_IDs=""
        tweet_times=""
        tweet_coords=""
        tweet_messages=""
        for item in geotweet_list:
            use_next_item=False
            # IGNORER SI LE MÊME IDENTIFIANT DE TWEET
            if item[1]==element[1]:
                continue
            # IGNORER SI INDICE DE LISTE EST ÉGAL OU MOINDRE AFIN D'ÉVITER LA RÉPÉTITION
```

```
DE TWEETS DÉJA EN GRAPPE
        if geotweet_list.index(item)>=geotweet_list.index(element):
            break
        # IGNORER SI CONTENU DU MESSAGE EST LE MÊME  (DOUBLON)
        # 1: COMME ÉLÉMENT
        if item[5]==element[5]:
            duplicates_new+=1
            continue
        # 2: COMME ITEM DANS LA LISTE CUMULATIVE DE TWEETS
        if int(element[1])==240 and int(item[1])==234:
             cg=8
        for part in cum_tweet_list:
            ab=part[3]
            if part[3]==item[5]:
                duplicates_new+=1
                use_next_item=True
                break

        # FORMATION DES GRAPPES
        if use_next_item==False:
            a=curr_tweet_secs
            b=time_delay
            c=item[0]
            curr_list_item_secs=time_to_secs(item[0])
            if curr_tweet_secs-time_delay<=curr_list_item_secs<=curr_tweet_secs:
                lat_b=convert_lat(item[11])
                long_b=convert_long(item[11])
                Current_Distance=calculate_distance(lat_a,long_a,lat_b,long_b)
                if 0<Current_Distance<Rayon_km:
                    if cluster_repeat==False:
                        cum_tweet_list.append((element[1],time_to_secs(element[0]),element[11],element[5]))
                        cluster_repeat=True
                    cum_tweet_list.append((item[1],curr_list_item_secs,item[11],item[5]))


    # CALCUL DE L'HEURE MOYENNE ET DU CENTRE DE GRAVITÉ DE LA NOUVELLE GRAPPE
    for tweet in cum_tweet_list:
        if cum_tweet_list.index(tweet)==0:
```

```
            tweet_IDs=tweet[0]
            cum_time=tweet[1]
            cum_lat=convert_lat(tweet[2])
            cum_long=convert_long(tweet[2])
            tweet_times=str(tweet[1])
            tweet_coords=str(tweet[2])
            tweet_messages=tweet[3]
        else:
            tweet_IDs=tweet_IDs+','+tweet[0]
            cum_time=cum_time+tweet[1]
            cum_lat=cum_lat+convert_lat(tweet[2])
            cum_long=cum_long+convert_long(tweet[2])
            tweet_times=tweet_times+','+str(tweet[1])
            tweet_coords=tweet_coords+','+str(tweet[2])
            tweet_messages=tweet_messages+tweet[3]
    if len(cum_tweet_list)!=0:
        avg_secs=int(cum_time/len(cum_tweet_list))
        avg_lat=round(cum_lat/len(cum_tweet_list),4)
        avg_long=round(cum_long/len(cum_tweet_list),4)
     # AJOUTER LA NOUVELLE GRAPPE À LA LISTE(mean time,c.gravity coord.,tweet IDs,tweet times,
tweet coords, tweet messages)
        total_clusters+=1

clust_listform=[len(cum_tweet_list),avg_secs,(avg_lat,avg_long),tweet_IDs,tweet_times,tweet_coords,twe
et_messages]
        cluster_list.append(clust_listform)

print cluster_list
print 'tweets read: ',tweets_read
print 'cluster matches: ',cluster_matches
print 'duplicates new/existing: ',duplicates_new,"/",duplicates_existing
print 'total clusters: ',total_clusters

clust_count=0

# SAUVEGARDE DES GRAPPES SELON LE TYPE D'INCIDENT ('Clusters_crime_words_users.csv'
pour incidents criminels)
saveFile=open('Clusters_traffic_words_users.csv','a')
```

```
for item in cluster_list:
    lat_a=convert_lat(item[2])
    long_a=convert_long(item[2])
    list_users=[]
    list_words=[]
    list_message=""
    contents=""
    freq_list_users=[]
    freq_list_words=[]
    common_users=0
    common_words=0
    cum_dist_cg=float()
    cum_time_delta=0
    tup_3=eval(item[3])
    for tweet in geotweet_list:
        if int(tweet[1]) in tup_3:
            tweet_secs=sec_b=time.mktime(time.strptime(tweet[0], "%Y-%m-%d %H:%M"))
            cum_time_delta=cum_time_delta+(((item[1]-tweet_secs)/60)**2)
            # IDENTIFIER ET REGROUPER USAGERS AVEC SUFFIXE 'tofire'
            user_name=tweet[2]
            if 'tofire' in tweet[2]:
                user_name='tofire'
            list_users.append(user_name)
            list_words.append(tweet[8])
            list_message=list_message+'../..'+tweet[0]+'..'+tweet[1]+'..'+tweet[2]+'..'+tweet[5]
            lat_b=convert_lat(tweet[11])
            long_b=convert_long(tweet[11])
            cum_dist_cg=cum_dist_cg+calculate_distance(lat_a,long_a,lat_b,long_b)
    # USAGERS COMMUNS (min:2)
    freq_list_users=Counter(list_users).most_common()
    for user in freq_list_users:
        if user[1]>1:
            common_users=common_users+user[1]
    # MOTS 'SCORED' COMMUNS (min:2)
    freq_list_words=Counter(list_words).most_common()
    for word in freq_list_words:
        if word[1]>1:
```

```
            common_words=common_words+word[1]
        time_hrs=datetime.fromtimestamp(item[1])
        if int(item[0])!=0:
            avg_dist_cg=round(cum_dist_cg/float(item[0]),2)
            stddev_time=round(((cum_time_delta/int(item[0]))**0.5),1)
        contents=str(Tot_Minutes)+'='+str(Rayon_km)+'='+str(item[0])+'='+str(time_hrs)+
'='+str(stddev_time)+'='+str(item[2])+'='+str(avg_dist_cg)+'='+
','.join(list_users)+'='+str(common_users)+'='+','.join(list_words)+
'='+str(common_words)+'='+list_message


        saveFile.write(contents)
        saveFile.write('\n')
    saveFile.close()
```