THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Université Toulouse - Jean Jaurès

Présentée et soutenue par :
HOANG Thi Bich Ngoc

le vendredi 28 septembre 2018

Titre :

Information Diffusion, Information and Knowledge Extraction From Social Networks
Diffusion d'Information, Extraction d'Information et de Connaissance sans les Réseaux Sociaux

---

École doctorale et discipline ou spécialité :
ED MITT : Image, Information, Hypermedia

Unité de recherche :
Institut de Recherche en Informatique de Toulouse UMR 5505

Directeur/trice(s) de Thèse :

Josiane MOTHE, ESPE Université de Toulouse

Jury :

Josiane MOTHE	Professeure, Université de Toulouse	Directrice
Jacques SAVOY	Professeur, Université de Neuchâte	Rapporteur
Alan SMEATON	Professeur, Dublin City University	Rapporteur
Chantal SOULE-DUPUY	Professeure, Université de Toulouse	Examinateur
Eric SANJUAN	Maître de conférence, Université d'Avignon	Examinateur
Pascal MARCHAND	Professeur, Université de Toulouse	Examinateur

Contents

List of Figures

List of Tables

List of Abbreviations

CRF Conditional Random Fields

Doc2vec Document to Vector

FP False Positive

Id Unique Identifier

LIW Location Indicative Words

NB Naive Bayes

NE Named Entities

NER Named Entity Recognition

POS Part Of Speech

RDF Resource Description Framework

RF Random Forest

SGD Stochastic Gradient Descent

SVM Support Vector Machine

Acknowledgement

First and foremost, I would like to express my sincere gratitude to my supervisor, professor Josiane Mothe. I highly appreciate all her contributions of time, dedicated help, advices, inspiration, encouragement and continuous supports to make my Ph.D experience productive and stimulating.

I also would like to thank professors Jacques Savoy and Alan Smeaton for accepting to review my thesis and for their valuable remarks.

My sincere thanks also goes to my friends and colleagues: Gia Hung Nguyen, Mahdi Washaha, Mahmoud Qudseya, Md Zia Ullah, Clement Lejeune and other lab mates for the stimulating discussions, precious supports and for all the funs we have had in the last three years.

A special mention of thanks to my best friend Giang, to Tran, Phuong, Trang, Hoai, Trinh and other friends in my hometown for their support and encouragement when I was stressful or in trouble.

Lastly, I would like to send a special thank to my parents and my brother for their unconditional love, constant inspiration and encouragement. Especially, words cannot express my gratefulness to my beloved husband, Long, for his great love, patience and tremendous support. Without him, I would not have been able to complete much of what I have done. Finally, I am thankful to my son, Tri, for giving me happiness, motivation and strength during my PhD and my life.

Abstract

The popularity of online social networks has rapidly increased over the last decade. According to Statista[1], approximated 2 billion users used social networks in January 2018 and this number is still expected to grow.While serving its primary purpose of connecting people, social networks also play a major role in successfully connecting marketers with customers, famous people with their supporters, need-help people with willing-help people. The success of online social networks mainly relies on the information the messages carry as well as the spread speed in social networks. Our research aims at modeling the message diffusion, extracting and representing information and knowledge from messages on social networks.

Our first contribution is a model to predict the information diffusion on social networks. More precisely, we predict whether a tweet is going to be diffused or not and the diffusion level. Our model is based on three types of features: user-based, time-based and content-based features. Being evaluated on various collections corresponding to dozen millions of tweets, our model significantly improves the effectiveness (F-measure) compared to the state-of-the-art, both when predicting if a tweet is going to be retweeted or not, and when predicting the level of retweet.

The second contribution of this thesis is to provide an approach to extract information from microblogs. While a message about an event is generally composed of several pieces of important information such as location, time, related entities, we focus on location which is vital for several applications, especially geo-spatial applications and applications linked to events. We proposed different combinations of various existing methods to extract locations in tweets targeting either recall-oriented or precision-oriented applications. We also defined a model to predict whether a tweet contains a location or not. We showed that the precision of location extraction tools on the tweets we predict to contain a location is significantly improved as compared to when extracted from all the tweets.

Our last contribution presents a knowledge base that better represents information from a set of tweets on events. We combined a tweet collection with other Internet resources to build a domain ontology. The knowledge

[1] https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/ (accessed February 7, 2018)

base aims at bringing users a complete picture of events referenced in the tweet collection (we considered the CLEF 2016 festival tweet collection).

Résumé long en français

Diffusion d'information, extraction d'information et de connaissance sans les réseaux sociaux

Les réseaux sociaux en ligne se sont rapidement développés au cours de la dernière décennie. Selon Statista[2], environ 2 milliards d'utilisateurs ont utilisé les réseaux sociaux en janvier 2018 et ce nombre devrait encore augmenter au cours des prochaines années. Selon une autre source[3], le service Twitter comptait en moyenne 330 millions d'utilisateurs actifs par mois avec environ 500 millions de tweets par jour en janvier 2018. En outre, Twitter a toujours été cité comme l'un des réseaux sociaux les plus populaires pour les adolescents aux États-Unis et prend de plus en plus d'importance lors des événements dans le monde entier.

Tout en servant son but premier de connecter les gens, les réseaux sociaux jouent également un rôle majeur dans le succès de connecter les spécialistes du marketing avec les clients, les gens célèbres avec leurs fans, ceux qui ont besoin d'aide et ceux qui veulent aider. Le succès des réseaux sociaux en ligne repose principalement sur l'information que les messages véhiculent ainsi que sur la vitesse de propagation dans les réseaux sociaux. Notre recherche vise à modéliser la diffusion des messages, à extraire et à représenter l'information et les connaissances des messages sur les réseaux sociaux.

La première contribution de cette thèse est d'introduire une approche pour prédire la diffusion de l'information sur les réseaux sociaux. Plus précisément, nous avons abordé deux questions de recherche:
1) Est-il possible de prédire si un message microblog (tweet) va être diffusé (retweeté) ou non?
2) Peut-on modéliser le niveau de diffusion et ainsi prédire le niveau de diffusion d'un nouveau message microblog?

Nous avons répondu à ces questions de recherche en considérant un modèle entraîné sur un sous-ensemble de tweets et en testant sur de nouveaux tweets. Nous avons étudié ce problème selon deux angles: une classification binaire (prédire si un tweet sera retweeté) et une classification multiclasse (prédire le niveau des retweets). Tout en réutilisant certaines carac-

[2] https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/

[3] https://www.omnicoreagency.com/twitter-statistics/

téristiques pour représenter les messages issues de la littérature, nous avons ajouté plusieurs nouvelles caractéristiques, que nous avons regroupées en trois catégories: basées sur l'utilisateur, basées sur le temps et basées sur le contenu. Nous avons montré que notre modèle améliore significativement la F-mesure d'environ 5% par rapport à l'état de l'art pour les deux types de prédiction lorsqu'il est évalué sur différentes collections avec un total d'environ 18 millions de tweets. De plus, nous avons également obtenu une F-mesure élevée sur les tweets de classe 1 (tweets retweetés moins de 100 fois) et de classe 2 (tweets retweetés moins de 10 000 fois) qui contiennent la majorité des tweets de chaque collection et qui étaient difficiles à prédire dans les travaux de l'état de l'art.

Certaines caractéristiques sont plus importantes que d'autres dans les modèles obtenus. Nous avons extrait les caractéristiques les plus importantes pour les deux types de prédiction et de manière cohérente à travers les jeux de données. Ces caractéristiques sont : le nombre de suiveurs, le nombre de suivis et le nombre de groupes dont l'utilisateur est membre, le nombre de favoris que l'utilisateur a réalisé dans son histoire. De plus, les fonctions temporelles que nous avons développées pour vérifier si un tweet est posté à midi, le soir, le week-end ou pendant les vacances sont également fortement corrélées avec la possibilité de retweet. Ces caractéristiques sont nouvelles par rapport à celles que l'on trouve dans la littérature.

Pour évaluer si les nouvelles caractéristiques que nous avons définies dépendent des caractéristiques existantes, nous avons également analysé les corrélations entre les caractéristiques sur trois jeux de données. Nous avons montré que la plupart des caractéristiques sont indépendantes les unes des autres. Certaines des nouvelles caractéristiques que nous avons développées sont:

- Importantes pour le modèle
- Ne sont pas corrélées aux caractéristiques existantes.

Quelques caractéristiques qui son corrélées aux caractéristiques existantes ont généralement un faible poids lorsque l'on considère leur impact pour les modèles prédictifs. De plus, les résultats présentés montrent que la combinaison des caractéristiques que nous avons définies et des caractéristiques existantes améliore significativement la performance du modèle prédictif.

Ce travail a été présenté dans un article accepté par la revie internationale "International Journal of Computational Sciences" [Hoang 2017b].

Comme une application du modèle prédictif proposé, nous avons appliqué ce modèle pour prédire la diffusion des histoires de marque sur les réseaux sociaux. Nous avons ajouté plusieurs caractéristiques supplémentaires et évalué notre modèle sur plusieurs types de collections associées à des actions de marketing : des collections d'histoires de produits ou de marques (en termes de tweets) générées par les consommateurs et des collections d'histoires de produits ou de marques générées par la société qui possède le produit ou la marque. Les résultats des expériences concordent avec nos remarques précédentes. Pour les deux types de collections, nous améliorons considérablement la F-mesure par rapport à l'état de l'art que ce soit dans le cas de la classification binaire ou de la classification multi-classe. Nous avons également classé les caractéristiques par l'ordre d'importance. Comme dans nos résultats précédents : le nombre de suivies, de suiveurs, de favoris de l'utilisateur et le nombre de groupes auxquels l'utilisateur appartient sont les caractéristiques les plus importantes pour faire retweeter un tweet sur une histoire de marque. De plus, la longueur du message, le fait qu'il contenienne un hashtag, une URL ou une image affectent également la retweetabilité. L'âge du compte et le fait qu'une personne célèbre soit mentionnée dans le contenu d'un tweet à propos d'une marque ou d'un produit le rendra également plus retweeté lorsque ce tweet est écrit par la société qui possède la marque ou le produit.

Nous pensons que nos résultats sont utiles pour les gestionnaires d'entreprise afin qu'ils comprennent mieux la diffusion d'histoires liées à leur marque et à leur produits sur les réseaux sociaux. De plus, nous avons également proposé des caractéristiques qui pourraient être utilisées pour rendre un message populaire. En se basant sur ces caractéristiques proposées, les gestionnaires peuvent former des histoires en ligne pour diffuser leurs produits ou leus marques. Ils peuvent également proposer des stratégies pour contrôler ou promouvoir les histoires générées par les clients. Notre modèle peut également être appliqué pour prédire la propagation de l'information dans d'autres domaines tels que la politique, les épidémies et les catastrophes. Nous n'avons pas évalué ces applications de notre modèle sur des collections de tweets appropriées, mais gardons cette piste de travail pour le futur.

Il y a des autres points qui pourraient être pris en considération à l'avenir. Les jeux de données que nous avons utilisés pour évaluer notre modèle prédictif ont été recueillis sur une période assez courte. Par exemple, le jeu de données de Sandy a été recueilli sur une période de trois jours, tandis que

les données de la première semaine et de la deuxième semaine ont été recueillies en une semaine. Il pourrait donc être intéressant d'analyser plus en détail l'impact du temps d'affichage du tweet sur la retweetabilité lorsque l'on considère des jeux de données recueillis sur des périodes plus longues. De plus, nous supposons également que certaines caractéristiques comme l'emplacement, les émissions de télévision mentionnées dans le contenu ou la réputation du nom d'utilisateur peuvent être plus importantes dans d'autres collections. Très peu de tweets contiennent de telles caractéristiques dans nos collections.

Pour les travaux futurs, nous aimerions mettre en œuvre certaines tâches. Tout d'abord, nous aimerions collecter des jeux de données plus importants qui incluent plusieurs tweets couvrant des caractéristiques que nous avons proposées telles que la présence d'entités nommées dans le contenu, la réputation de l'utilisateur et des temps de publication plus variés.

Par ailleurs, nous aimerions définir des caractéristiques supplémentaires pour représenter les tweets. Par exemple, nous pourrions considérer les vecteurs de type Doc-2vec [Le 2014] formé sur un jeu de données. Nous utiliserions alors ces vecteurs comme de nouvelles caractéristiques dans notre modèle. Notre hypothèse est que si les vecteurs Doc2Vec sont appris à partir des sujets, des événements et des histoires d'un grand ensemble d'information, il serait possible de déduire de "bons" vecteurs pour l'ensemble de tests et cela pourrait conduire à une amélioration de la classification.

L'analyse de sentiment d'un tweet est une des caractéristiques que nous pensions importante dans notre modèle mais cela n'a pas été confirmé dans les résultats de notre évaluation empiriqque. Une piste d'amélioration est d'appliquer des méthodes telles que celles proposée dans [Kummer 2012, Sahni 2017] pour améliorer l'efficacité de cette extraction de caractéristiques. Ces méthodes utilisent le z-score pour identifier les caractéristiques les plus saillantes appartenant aux catégories spécifiques et utilisent la subjectivité dans les tweet pour sélectionner les meilleurs tweets d'entraînement et ainsi augmenter la précision de la classification des sentiments.

Nous aimerions classifier un tweet en sujets tels que le sport, la musique, le cinéma, la mode, les nouvelles météorologiques quotidiennes ou les nouvelles technologiques avant de prédire la popularité de ce tweet. Nous pensons que les utilisateurs sont plus intéressés par certains sujets que par d'autres et que les modèles de diffusion dépendent des sujets. Enfin, une piste pourrait être d'analyser l'influence d'un suiveur qui retweete un tweet sur un de

ses amis.

Nous avons présenté ce travail dans un article qui a été accepté à la conférence internationale "International Conference of Computational Linguistics and Intelligent Text Processing" 2018 [Hoang 2018b].

Il serait plus utile de prévoir la diffusion de l'information en tenant compte de l'aspect géographique. Par exemple, les spécialistes du marketing peuvent se baser sur le niveau de diffusion de leurs histoires de marque par région pour proposer des campagnes de vente et de marketing appropriées pour chaque région. Les politiciens peuvent utiliser leur connaissance de la diffusion des nouvelles électorales par régions pour proposer des politiques pertinentes pour leurs campagnes électorales. Ainsi, l'extraction des emplacements dans les tweets joue un rôle important dans la prédiction de la diffusion de l'information par région. En outre, bien que plusieurs éléments d'information importants comme le lieu, l'heure, les entités connexes soient inclus dans un message sur un événement, l'emplacement est vital pour plusieurs applications, surtout les applications géospatiales et les applications liées aux événements [Goeuriot 2016a]. L'un des premiers éléments d'information transmis aux systèmes d'aide en cas de catastrophe est le lieu où la catastrophe s'est produite [Lingad 2013]. Un emplacement dans le texte d'un message de crise rend le message plus précieux que les autres qui ne contiennent pas un emplacement [Munro 2011]. Les utilisateurs de Twitter sont les plus susceptibles de transmettre des tweets avec des mises à jour sur l'emplacement et la situation, ce qui indique que les utilisateurs de Twitter eux-mêmes trouvent que l'emplacement est très important [Vieweg 2010].

Notre deuxième contribution dans cette thèse est de fournir une approche pour extraire efficacement la localisation dans les messages de Twitter.

Étant donné qu'il y a des applications qui nécessitent un rappel élevé (par exemple ce qui s'est produit à un endroit donné) et d'autres qui nécessitent une grande précision (par exemple sur quels endroits devrions-nous nous concentrer en premier pour un problème donné), nous avons émis l'hypothèse que la combinaison des outils d'extraction existants pourrait améliorer la précision de l'extraction des emplacements.

Nous en sommes donc arrivés à notre première question de recherche:
1) Dans quelle mesure pouvons-nous améliorer la précision et le rappel en combinant les outils existants pour extraire les mentions de lieux des microblogs?

Pour répondre à cette question, nous avons combiné différents outils, à savoir l'outil Ritter [Ritter 2011], l'environement Gate NLP[Bontcheva 2013]

et l'outil NER Stanford [Finkel 2005]. Nous avons également proposé de filtrer les emplacements extraits en utilisant DBpedia[4].

Nous avons obtenu trois résultats importants:

- La combinaison des emplacements reconnus par l'outil Ritter avec les emplacements reconnus par Stanford filtrés par DBpedia augmente la F-mesure pour l'extraction des emplacements.

- La combinaison des emplacements extraits par Ritter avec les emplacements reconnus par Gate améliore considérablement le rappel. Nous avons obtenu un taux de rappel de 82% (pour le jeu de données Ritter), ce qui est très approprié pour les applications de rappel, tandis que le meilleur outil de cette collection, Ritter, atteint 71% de rappel. Ce résultat peut s'expliquer par le fait que ces méthodes utilisent des indices différents pour extraire les emplacements des tweets.

- En utilisant DBPedia pour filtrer les emplacements que Ritter reconnaît, nous avons atteint une précision remarquable de 97% (pour le jeu de données Ritter). Ce résultat élevé a été obtenu parce que les noms de lieux imprécis et inconnus ont été écartés par le filtrage DBPedia.

Une quantité énorme de tweets sont postés chaque jour, mais très peu d'entre eux contiennent des emplacements. Par exemple, dans le jeu de données Ritter [Ritter 2011], disponible à des fins de recherche et qui a été recueilli en septembre 2010, seulement 9 % environ des tweets contiennent un emplacement. De plus, nous avons réalisé une étude préliminaire en utilisant des outils d'extraction de localisation uniquement sur les tweets qui contiennent des localisations; nous avons obtenu une précision significativement plus élevée que lors de leur implémentation sur l'ensemble des jeux de données. Nous avons donc émis l'hypothèse que nous pourrions grandement augmenter la précision si nous pouvions prédire l'emplacement des occurrences dans les tweets. Cela nous amène à notre deuxième question de recherche pour cette deuxième contribution :

2) Est-il possible de prédire si un tweet contient un emplacement ou non?

L'une des principales contributions de ce travail est une méthode permettant de prédire si un tweet contient un emplacement ou non. Nous avons défini plusieurs nouvelles fonctions pour représenter les tweets et évalué

[4] `http://dbpedia.org/snorql/`

intensivement les paramètres d'apprentissage automatique pour prédire les occurrences de localisation en variant les algorithmes d'apprentissage automatique et les paramètres utilisés. Les résultats ont montré que:

- Random Forest et Naïve Bayes sont les meilleures solutions d'apprentissage automatique pour ce problème - elles fonctionnent mieux que le Support Vector Machine (et d'autres algorithmes que nous avons essayés mais dont nous n'avaons pas rapporté les résultats car plus faibles).
- Le fait de modifier les critères d'optimisation (soit l'exactitude, soit le nombre de vrais positifs) ne modifie pas beaucoup la F-measure.
- En ce qui concerne l'extraction de localisation, nous avons amélioré la précision en nous concentrant uniquement sur les tweets dont on prévoit qu'ils contiennent une localisation.

Une autre contribution est que nous avons évalué les tweets à l'aide d'algorithmes de classification avec différents paramètres. Dans la section expérimentale, nous montrons que la précision des outils NER pour les tweets dans lesquels nous prévoyons qu'il est fait mention d'un emplacement est significativement améliorée: de 85% à 96% pour la collection Ritter et de 80% à 89% pour la collection MSM2013. Cette augmentation de la précision est significative et cruciale dans les systèmes où l'extraction de l'emplacement doit être très précise, comme les systèmes d'aide en cas de catastrophe et les systèmes de sauvetage. Nous avons montré que la prédiction de l'emplacement est une étape de prétraitement utile pour l'extraction de l'emplacement.

Notre modèle donne une prédiction exacte pour les tweets qui contiennent des mots du répertoire géographique ou qui incluent une préposition juste avant un nom propre. Nous avons également obtenu une bonne prédiction sur les tweets basés sur 'nombre de noms propres' ou 'mots spécifiant des endroits juste après ou avant le nom propre'. Toutefois, dans certains cas, la prédiction n'est pas appropriée. Puisque nous n'avons considéré que les abréviations des lieux inclus dans le répertoire toponymique de l'outil "Gate", certains tweets ne sont pas prédits avec précision s'ils mentionnent des abréviations qui ne sont pas incluses dans le répertoire toponymique telles que: "'*@2kjdream Bonjour! Nous sommes ici JPN !*" où JPN n'est pas reconnu. Nous n'avons pas non plus abordé la question de la désambiguïsation des lieux. Pour les travaux futurs, afin de résoudre ce problème, le

contexte donné par tous les mots du message devrait être pris en compte [SanJuan 2012].

Dans le cadre de travaux futurs, nous aimerions également créer des jeux de données d'entrainement pertinents pour le modèle Doc2Vec afin de déduire les caractéristiques vectorielles représentant les tweets. Des jeux de données d'entrainement appropriés permettront de surmonter les limites de notre modèle, par exemple, de mieux gérer les abréviations et la désambiguïsation. Les tweets qui contiennent des mots similaires au sujet des mêmes histoires ou événements devraient être représentés dans les vecteurs.

Nous prévoyons également d'extraire d'autres caractéristiques pour améliorer la précision de notre modèle prédictif. Certaines caractéristiques peuvent être intéressantes à considérer comme l'apparition d'un nom de l'événement dans le contenu (les gens mentionnent souvent l'emplacement avec l'événement dont ils parlent), les emplacements fréquemment vus dans les messages de l'historique d'un utilisateur et les messages de l'historique de ses amis.

Ce travail a été décrit et évalué dans deux articles acceptés par deux revues internationals: "International journal of Information Processing & Management" [Hoang 2018c] et "International Journal of Computational Linguistics and Applications [Hoang 2018a]. Ce travail a également donné lieu à des présentations et publications dans plusieurs conférences internationales et nationales et ateliers [Hoang 2017a, Hoang 2018d, Hoang 2018e].

La troisième contribution de cette thèse porte sur la construction d'une base de connaissances qui représente de façon globale et intégrée l'information provenant d'un ensemble de tweets sur des événements.

Les médias sociaux comme Twitter sont largement utilisés lors d'un événement (conférence, catastrophe, événement culturel...) pour commenter ou conseiller les acteurs liés à cet événement. Les utilisateurs des réseaux sociaux sont alors avertis par l'intermédiaire des personnes qu'ils suivent ou en cherchant des tweets en rapport avec l'événement. Cependant, étant donné le format de 140 caractères[5] d'un tweet, l'information obtenue par un seul message est souvent très partielle. Il est plus probable qu'un utilisateur ait plutôt besoin de lire un ensemble de tweets pour avoir une image claire d'un événement. Nous avons développé l'idée que l'utilisation d'un ensemble de tweets sur un événement pourrait permettre d'avoir une vue plus complète de cet événement en combinant toutes les informations partielles données

[5]Au moment de l'étude les tweets avaient une taille maximale de 140 caractères

en particulier par les tweets. La question de recherche à laquelle nous nous sommes intéressés est:

Est-il possible d'apporter à une personne une vue complète d'un événement en utilisant une base de connaissances?

Nous proposons un modèle qui représente une collection de micro-blogs sur une ontologie de domaine qui permet de mieux représenter l'information d'un ensemble de tweets sur des événements. Nous avons étudié le cas d'un festival. En combinant la collection de tweets existante sur des festivals avec d'autres ressources d'Internet, nous visons à donner une image complète du contenu de la collection qui peut donner un aperçu complet des événements référencés dans cette collection. Ce modèle peut être appliqué dans des systèmes de recommandation dans les domaines du tourisme, du transport ou du marketing. Bien que nous ayons considéré une collection de festivals, la méthode que nous proposons peut être adaptée à d'autres types d'événement.

En ce qui concerne l'ontologie du domaine, nous utilisons Wikipedia (ou plutôt DBPedia[6]) ainsi que des sites web qui fournissent des informations officielles sur la géographie, la liste des festivals et des détails connexes. Cette information est assez stable dans le temps. Ensuite, les tweets relatifs à chaque festival sont sélectionnés à l'aide de méthodes de recherche d'information. Ils sont analysés pour reconnaître et extraire les entités nommées (NE) telles que les lieux, les artistes, les noms de festivals, le temps. Ces informations extraites peuvent être utilisées pour remplir les instances des classes correspondantes dans l'ontologie.

Comme preuve de ce concept, nous avons combiné la collection de tweets de festivals [Goeuriot 2016a] avec d'autres ressources Internet pour construire une ontologie du domaine. Cette ontologie vise à donner une image complète du contenu de la collection qui peut donner une vue d'ensemble des événements du festival référencés dans cette collection.

La base de connaissances que nous avons conçue pourrait être utilisée dans des applications où les utilisateurs:

- Choisiraient un nom de festival spécifique et auraient une image de ce festival sur les tweets

[6]BDpedia structure les informations des pages de Wikipedia ; cette base peut être interrogée en utilisant SPARQL pour extraire des informations structurées

- Choisiraient un lieu et obtiendraient une liste des festivals correspondants, etc.

L'utilisateur recevrait des informations officielles provenant des sites web touristiques, accompagnées des informations les plus récentes provenant des tweets, telles que l'heure à laquelle le festival se déroule, les artistes qui se produisent et les dates auxquelles ils se produisent pour chaque festival. Les tweets liés à un festival apporteraient à l'utilisateur des nouvelles fraîches sur le trafic, la météo, l'atmosphère, les opinions et les commentaires des participants. De plus, les capacités d'inférence ontologique pourraient apporter de nouvelles connaissances à partir des données existantes.

Nous croyons qu'en utilisant une ontologie, nous avons fourni un système de base de connaissances facilement accessible. Par rapport au stockage de données dans des bases de données traditionnelles, notre approche présente plusieurs avantages. Premièrement, les données sont présentées dans un langage commun qui peut être facilement récupéré par SPARQL. Un modèle de données RDF est également plus facile à mettre à jour sans effets négatifs sur l'application et nécessite donc moins de maintenance. Deuxièmement, le mécanisme d'inférence du langage ontologique permet d'inférer facilement de nouvelles connaissances à partir de données existantes (dans la preuve de concept, nous programmons l'inférence, mais l'ontologie permet un tel processus). Enfin, en combinant plusieurs ressources telles que DBPedia, des sites web et Twitter, notre système pourrait apporter une connaissance complète et fraîche des festivals par villes dans le monde, y compris les informations officielles des sites web et les dernières nouvelles de Twitters.

Nous supposons que notre modèle de base de connaissances a un large éventail d'applications dans plusieurs domaines tels que le tourisme, le transport, le marketing et la publicité. Par exemple, dans le domaine du tourisme, cette base de connaissances peut être utilisée pour construire un système de recommandation graphique avec des résumés très informatifs sur les événements, les personnes célèbres, les activités connexes agrégées à partir de tweets.Dans le domaine du transport, un système développé sur notre modèle qui suggérerait un itinéraire ou un moyen de transport approprié pour éviter les foules, les embouteillages ou autres problèmes pourrait être bien accueilli par les voyageurs.

Pour les travaux futurs, nous aimerions évaluer notre modèle à partir d'un ensemble de données réelles et volumineuses. En outre, nous souhaitons également extraire de BDpedia des résumés courts sur les festivals ou des

techniques de réutilisation comme celle présentée dans [Ermakova 2015] pour proposer aux utilisateurs une idée de base des festivals qui les intéressent. En outre, nous prévoyons de développer notre base de connaissances pour la recommandation d'événements en fonction de l'emplacement actuel de l'utilisateur et d'autres aspects tels que son profil, son intérêt et les festivals auxquels ses amis participent.

Ce travail a été présenté à la conférence internationale 'Conference and Labs of the Evaluation Forum CLEF' 2016 [Hoang 2016a] et à une conférence nationale CORIA-RJCRI 2016 [Hoang 2016b].

Publications

The research reported in this thesis has resulted the following publications:

International journal papers

1. Thi Bich Ngoc Hoang and Josiane Mothe. Location extraction from tweets. Information Processing & Management 54.2 (2018): pp 129-144, Elsevier. Access: `https://www.sciencedirect.com/science/article/pii/S0306457317303734`

2. Thi Bich Ngoc Hoang and Josiane Mothe. Predicting information diffusion on Twitter–Analysis of predictive features. Journal of Computational Science , Elsevier (2017). Access: `https://www.sciencedirect.com/science/article/pii/S1877750317305860`

3. Thi Bich Ngoc Hoang, Véronique Moriceau and Josiane Mothe. Can we Predict Locations in Tweets? A Machine Learning Approach. International Journal of Computational Linguistics and Applications (accepted), 2018.

International conference papers and presentations

1. Thi Bich Ngoc Hoang, Josiane Mothe, Predicting the diffusion of brand's stories in social networks (regular paper). In : Computational Linguistics and Intelligent Text Processing, Hanoi, Vietnam, 18-24 March 2018, Springer LNCS.

2. Thi Bich Ngoc Hoang, Josiane Mothe, Véronique Moriceau. Predicting Locations in Tweets (poster). In : Computational Linguistics and Intelligent Text Processing, Budapest, Hungary, 17-23 April 2017.

3. Thi Bich Ngoc Hoang, Josiane Mothe. Building a Knowledge Base using Microblogs: the Case of Cultural MicroBlog Contextualization Collection (regular paper). In : Conference and Labs of the Evaluation forum (CLEF 2016), Evora, Portugal, 05-08 September 2016. Access: `http://ceur-ws.org/Vol-1609/16091226.pdf`

National conference papers and presentations

1. Thi Bich Ngoc Hoang, Josiane Mothe. Méthode d'apprentissage pour extraire les localisations dans les MicroBlog. In : EGC - Atelier Extraction et Gestion Parallèles Distribuées des Connaissances (poster), Paris, 22-26 January 2018.

2. Thi Bich Ngoc Hoang, Josiane Mothe. Extraction de Localisations dans les MicroBlogs. In : GAST - Gestion et l'Analyse de données Spatiales et Temporelles, Paris, 23 January 2018. Access: `https://gt-gast.irisa.fr/files/2018/01/ActesGAST2018-1.pdf`

3. Thi Bich Ngoc Hoang, Josiane Mothe. Building a knowledge base using MicroBlogs: the case of festivals and location-based events (regular paper). In : Rencontres Jeunes Chercheurs en Recherche d'Information (RJCRI 2016), Toulouse, 09-11 March 2016. Access: `https://www.irit.fr/publis/SIG/2016_RJCS_HM.pdf`

Chapter 1

Introduction

The online social networks has rapidly increased over the last decade. According to Statista [1], approximated 2 billion users used social networks in January 2018 and this number is still expected to grow in the next years. While serving its primary purpose of connecting people, social networks also plays a major role in successfully connecting marketers with customers, famous people with their supporters, need-help people with willing-help people. The success of online social networks mainly relies on the information the messages carry as well as the spread speed in social networks. Our research aims at modeling the message diffusion, extracting and representing information and knowledge from messages on social networks.

The first contribution of this thesis is to introduce an approach to predict the diffusion of information on social networks. More precisely, we addressed two research questions: *1) Is it possible to predict whether a microblog post (tweet) is going to be diffused (retweeted) or not?* and *2) Can the level of diffusion be modeled and thus can we predict the level of the diffusion of a new microblog post?*

We answered these research questions by considering a model that we trained on a subset of tweets and test on new tweets. Our model uses three types of features: user-based, time-based and content-based features. We showed that our model significantly improves the F-measure by about 5% (statistically significant – using Student t-test, p-value < 0.05) compared to the state-of-the-art when evaluated on various collections corresponding to dozen millions of tweets. We also showed that some features we introduced are very important to predict the retweetability. This work was presented in a paper accepted by the international Journal of Computational Sciences [Hoang 2017b]. In addition, we applied this predictive model to predict the diffusion of brand stories in social networks. We added several additional features and evaluated our model on multiple 'marketing' collections. The

[1] https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/ (accessed February 7, 2018)

results showed that our approach is more effective than the state-of-the-art. We presented this work in a paper which was accepted to present in the international Conference of Computational Linguistics and Intelligent Text Processing 2018 [Hoang 2018b].

The second contribution of this thesis is to provide an approach to extract information in Twitter posts. While several pieces of important information included in a message about an event such as location, time, related entities, we focus on extracting location which is vital to several applications, especially geo-spatial applications and applications linked with events [Goeuriot 2016a]. One of the first pieces of information transmitted to disaster support systems is where the disaster has occurred [Lingad 2013] and a location within the text of a crisis message makes the message more valuable than the others that do not contain a location [Munro 2011]. Our work first answered to the following research question: *1) How much can we improve precision and recall by combining existing tools to extract the location from microblog posts?* We have proposed several combinations of different existing methods to extract locations in tweets. We showed which combinations are effective for either recall-oriented or precision-oriented applications.

Originating from the fact that there is a huge amount of messages posted daily, but only a very small proportion contains locations, we hypothesized that predicting whether a post contains a location or not, prior to extracting locations, could make the efficiency improved. Indeed, in the Ritter dataset [Ritter 2011], available for research purposes, which was collected during September 2010, only about 9% of the tweets contain a location. This leads us to our second research question for this second contribution: *2) Is it possible to predict whether a tweet contains a location or not?* To answer this question, we defined a number of features to represent tweets and use these features as location predictors. We showed that the precision of location extraction tools for the tweets that we predict to contain a location is significantly improved: 11% and 9% (statistically significant) when evaluating our model on two tweet collections. The increase of precision is meaningful and crucial in systems where the location extraction needs to be very precise such as disaster supporting systems and rescue systems.

Our approach was described and evaluated in one paper accepted by the international journal of Information Processing & Management [Hoang 2018c], one other paper accepted by International Journal of Computational Linguistics and Applications [Hoang 2018a] and presented in several international

and national conferences, workshops [Hoang 2017a, Hoang 2018d, Hoang 2018e].

The third contribution of this thesis investigated the building of a knowledge base that better represents information from a set of tweets on events. Social media are widely used during an event to collaboratively comment or advise on that event. Given the size of a tweet, the information obtained by single post is often very partial. A research question is formed as follow: *Is it possible to bring a person a complete view about an event using a knowledge base*?

We developed the idea that using a set of tweets about an event could enable having a more complete view of that event by combining all information posted. As a proof of concept, we combined the festival tweet collection [Goeuriot 2016a] with other Internet resources to build a domain ontology. This ontology aims at bringing a complete picture of the collection content that can make a complete view of festival events referenced in this collection. This work was presented in an international conference CLEF 2016 [Hoang 2016a] and in a national conference RJCRI 2016 [Hoang 2016b].

To develop these three main contributions of our work, this thesis is organized into 5 chapters. The content of each chapter is described as follows:

Chapter 1 is this introduction in which the research questions and main contributions have also been presented.

Chapter 2 presents our model of predicting the information diffusion on social networks. Firstly, we describe the features that represents tweets. Afterward, we detail the experiments and evaluation of our model on various collections. We also present the application of our model on predicting the diffusion of brand stories on social networks.

Chapter 3 introduces an approach for extracting locations from tweets. We first present results when combining several named entities extraction tools to extract locations from tweets, targeting either precision-oriented or recall-oriented results. Subsequently, a model for predicting whether a tweet contains a location or not is proposed. The results of location extraction on predicted tweets are detailed.

Chapter 4 proposes a model to represent the collection of microblogs into a knowledge base. The domain ontology and the way to populate this ontology are presented. Finally, we describe how the knowledge base could be used to provide a complete view of an even.

Chapter 5 concludes this thesis, discusses main contributions of our work and outlines some future work.

Chapter 2

Information Diffusion on Social Networks

Summary

Abstract.

Information propagation on online social networks focuses much attention in various domains such as varied as politics, disasters, or marketing. Modeling information diffusion in such growing communication media is crucial in order both to understand information propagation and to better control it. Our work aims at predicting whether a tweet is going to be forwarded or not. Moreover, we aim at predicting how much it is going to be diffused. Our model is based on three types of features: user-based, time-based and content-based. Evaluating our model on vaious collections corresponding to about 18 millions of tweets, we show that our model significantly improves the F-measure by about 5% compared to the state-of-the-art (statistically significant – using Student t-test, p-value < 0.05). Some features from the literature are confirmed to be important such as the number of followers and followees of a user. We also show that some features we introduced are very important to predict retweetability such as the number of groups that a user is a member of, the posting time of a tweet. In the last part of this chapter, we apply our model to predict the diffusion of brand stories on social networks and show that the results are consistent with previous findings.

2.1 Introduction

On-line social networks are more and more popular as information channels. For example, Statista[1] reports 2.2 billion monthly active FaceBook users in the fourth quarter of 2017. In another source[2], the monthly active Twitter users has been dramatically increased from 2010 to 2017 (see Figure 2.1). The Twitter service averaged at 330 million monthly active users with about 500 million tweets per day in the third quarter of 2017. In addition, Twitter

[1] https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/

[2] https://www.statista.com/topics/737/twitter/

has consistently been named as one of the most popular social networks for teenagers in the United States and is becoming increasingly prominent during events over the world.

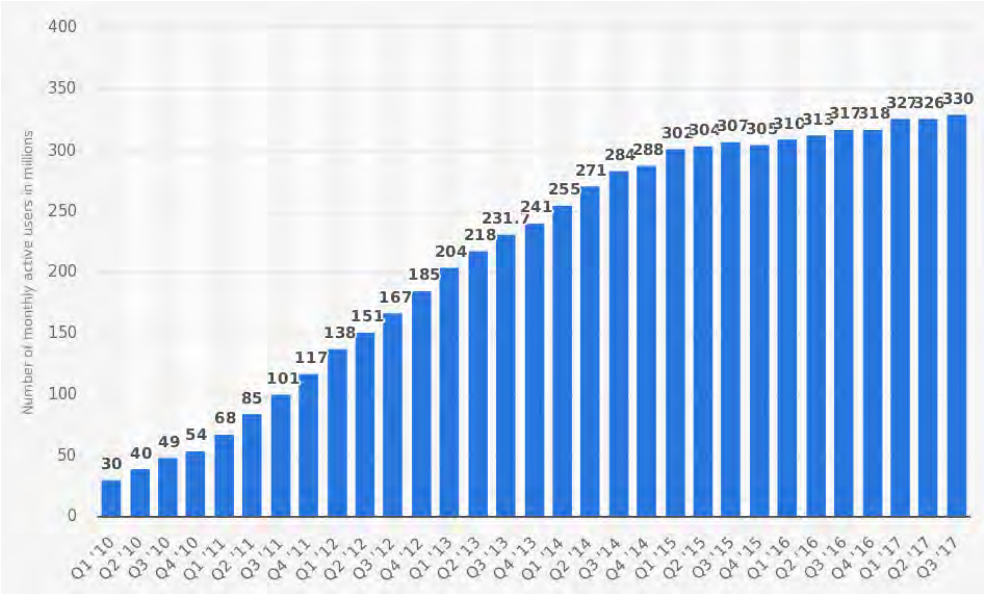

Figure 2.1: The number of monthly active Twitter users worldwide from the 1st quarter 2010 to the 3rd quarter 2017.
Source: https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/

While serving its primary purpose of connecting people, social networks also play a major role in successfully connecting marketers with customers, famous people with their supporters, help-needed people with help-willing people and information-sharing people with information-searching people. Many people and organizations use Twitter as a way to share and spread their messages. As shown in Figure 2.2, Barack get 179,000 retweets for his words about new year while supporters of Selena forward her movie advertisement post 16,000 times. In Houston devastation, Penn State was successfully in asking 1.2 millions of people to retweet the post to help victims of devastation in Houston.

Modeling information diffusion in such growing communication media is crucial in order both to understand information propagation and to better control it. Indeed, some studies have investigated the impact of social media in the recent elections both in US or in France, focusing mostly on fake news

Figure 2.2: The retweet number of some tweet examples.

and their propagation on social media. The authors in [Allcott 2017] have collected 115 pro-Trump fake stories shared on Facebook for a total of 30 millions times while 41 pro-Clinton fake stories were shared a total of 7.6 million times. Since a high percent of voters use social media (35% of people 18 to 29 years old, according to Pew Research Center[3]), the hug number of share make fake stories successfully reach voters.

This chapter provides an approach to predict the diffusion of messages on social networks, specifically on Twitter. More precisely, we studied two related questions: (1) Is it possible to predict whether a post (a tweet) is going to be propagated (or re-tweeted)? and (2) Can the level of propagation be modeled and thus can we predict the level of propagation of a new post?

We answer these research questions by considering a model that we train on a subset of tweets and test on new tweets. Our model is based on three types of features: user-based, time-based and content-based. While some features are reused from previous work in the domain of tweet diffusion [Suh 2010], we also introduce new features and evaluate the added value

[3] http://www.journalism.org/2016/02/04/the-2016-presidential-campaign-a-news-event-thats-hard-to-miss/

of these new features for both predicting whether a tweet is going to be retweeted or not and predicting the level of the propagation.

In the later part of this chapter, we apply our model to a specific area - Marketing. The emergence and growing of social media allows one consumer or company to communicate with thousands or millions other consumers. The consumer-generated stories or company-generated stories about a brand or a product can be widely propagated and as a consequence, can have a big impact on the marketplace and indirectly affect the success of the brand. Therefore, modeling the brand stories diffusion on social media is crucial for business managers in order both to understand the brand stories propagation and to better control it.

The remainder of this chapter is organized as follows: Section 2.2 presents the related work. Section 2.3 describes the model, features and the evaluation of the predictive model for predicting the information diffusion on Twitter. Section 2.4 present results of applying the proposed model to predict the brand stories on social networks. Section 2.5 is the conclusions and discussions.

2.2 Related work

Information diffusion have attracted a number of researchers' attention in recent years. Several pieces of work have made efforts to study the prediction of information propagation on social networks.

Suh *et al.* [Suh 2010] identified a number of features that may correlate with the number of retweets of a given tweet. They evaluated the correlation considering a large-scale analysis on 74 million tweets. They showed that numbers of followers, numbers of followees, and ages of the account have a very strong relationship with the retweet number. The larger the number of the followers and followees of the sender is, the more likely his tweets get retweeted is. In addition, tweets posted by "senior users", who registered more than 300 days before writing, get a higher number of retweets than the average. On the contrary, the presence of hashtag or URL in a tweet does not highly correlates with the number of retweets. The authors reported that 20.8% of retweets only contain hashtags while 28.4% of retweets contain URL. They also found that the number of past tweets has little or no relationship with the average number of daily tweets or with the retweet rate; the number of tweets that are favorited by users seem not to impact

the retweetability since only 8.7% of retweets are written by authors with more than 100 favorited items [Suh 2010]. In our work, we considered all the features proposed by Suh *et al.* including the presence of hashtags and URL in the tweet content, the number of followers, followees, number of tweets that the user has liked in his timeline, total of past tweets and ages of the user's account [Suh 2010]. We also added several new features including user-based, time-based, and content-based features.

Kwa *et al.* [Kwak 2010] studied the relationship between the number of followers of a user and the number of retweets for his posts on a collection of 106 million tweets. The authors constructed retweet trees and examined tree temporal and spatial characteristics. They showed that people only retweets from a small number of people and only a subset of a user's followers actually retweet. In addition, users with less than 1,000 followers tend to have the same average number of retweets for their posts. Similarly, Remy *et al.* [Remy 2013] studied the correlation between the number of users' followers and the capacity to spread their messages. They implemented their method on a Twitter dataset centered on the Japanese Earthquake and Tsunami in March 2011. Surprisingly, they showed that the impact of users with a lot of followers is not statistically greater than users with a few followers . In our model, we also took into account the relationship between the number of followers of a user and the retweetability of his or her tweets.

Hong *et al.* [Hong 2011] addressed the problem of predicting the future retweet number of a given tweet. They formulated the task into binary classification and multi-class classification. For binary classification, class-0 represents for tweets that are not retweeted while class-1 includes tweets that are retweeted. For multi-class classification, the authors suggested 4 classes: class-0 (not retweet), class-1 (retweets less than 100), class-2 (retweets less than 10,000), and class-3 (retweets more than 10,000). They used logistic regression as a classifier considering the message content, meta data and structural properties of the users' social graph features. However, in their paper, Hong *et al.* did not describe the features they used explicitly. They achieved 0.60 F-measure for binary classification (recall 0.44 and precision 0.99). With regard to multi-class classification, Hong *et al.* achieved good accuracy only for the smallest and largest categories: class-0 and class-3 but very low accuracy for the two other classes: 0.15 on class-1 and 0.43 on class-2 [Hong 2011].

Our idea of classifying tweets into classes is similar to Hong's. In the

evaluation section of our work (Section 2.3.5), we show that using Random forest as a machine learning algorithm and several new features we introduced, recall and F-measure can be improved for binary classification. We also improve the F-measure for class-1 and class-2 which are supposed to be more challenging classes since most of the tweets are in these two classes.

Hu *et al.* [Hu 2016] proposed an approach for predicting the short-term popularity of viral topics based on time series forecasting. They used historical popularity data of a given topic and showed that the popularity is relatively changeable for burst topics and past popularity have an impact on future popularity for non-burst topics. Xiong *et al.* [Xiong 2012] characterized information propagation on Twitter by considering the topic of the tweet. They proposed a propagation model with four possible states: susceptible, contacted, infected and refractory. People who read a message but have not decided to forward it are in the contacted state. They may become infected or refractory, and these two states are stable. They supposed that users select the topic that they are most interested in and then retweet. The more topics a user participates in, the less the user will turn attention to a new topic. The authors also supposed the inhibition between topics is important to user's decision. As a result, by using more than 20,000 tweets to train the model, they found that individual decision mainly depends on the topic itself. In our work, we did not consider the topic of the tweet but instead we added several content features which users may use to enhance the tweet content such as checking if the tweet contains location name, company name, TV show, picture or video.

Other work related to the diffusion of information on social networks can be found in [Ren 2016, Zhang 2013, Yang 2010]. Yang *et al.* [Yang 2010] studied the retweet process on social network. They first performed an analysis on a Twitter dataset. They found that almost 25.5% of the tweets posted by users are actually retweeted from their friends' posts. Then, they proposed a semi-supervised framework on a factor graph model to predict Twitter user's retweeting behaviors. The features of the users' history preferences, messages content and information of the trace were considered but are not explicitly described in their paper. In the experiments, the authors reported F-measure of 0.33 on the prediction, outperforming the L1-regularized logistic regression method. However their method did not outperformed the Support Vector Machine baseline in terms of recall. In a similar study, Zhang *et al.* [Zhang 2013] addressed the problem of how users' behaviors are in-

fluenced by friends in their ego network. They first tested whether the influence locality exists in the microblog network and whether it significantly influences user's retweet behavior. They found that the fraction of active users (retweeted a message) with two active neighbors (followees who have retweeted the same message) is about double compared to the fraction of active users with only one active neighbors. They also showed that, although the probability a user retweets a message is positively correlated with the number of active neighbors, it is negatively correlated with the number of connected circles that are formed by those neighbors. We did not consider the influence of followers' retweeting behavior on their friends in our work since the datasets we used do not contain any information of users' followers (except number of followers); but this could be an interesting feature to improve our model in the future.

In our work, we re-used some main features that previous research has shown to be good indicators for retweetability. We also suggest several new features that use to evaluate for the task of predicting retweets.

2.3 Predicting information diffusion on microblogs

In this section, we present the model, features and evaluation of the model for predicting information diffusion on Twitter.

The model in itself is based on machine learning; with this respect it is similar to Hong's, which used machine learning techniques to predict the popularity of messages as measured by the number of future retweets [Hong 2011] (see Section 2.2). Using machine learning implies that (1) each tweet is represented by a set of features (2) a training set is used in order to learn the model before the model is used on the test set or new tweets.

The process of our predictive model is described in the Figure 2.3.

2.3.1 Tweet representation

We hypothesized that both the tweet content and the user who writes it have an impact on tweet diffusion. To decide on possible useful features to represent tweets, we manually analyzed about 500 tweets from the Sandy collection [Tamine 2016]. The idea was to detect clues that could be useful to predicted retweet or/and the retweet rate. We also relied on large scale

Figure 2.3: The process of our predictive model

.

analytics of factors affecting retweetability [Suh 2010] to enrich the tweet representation.

Finally, in our model, tweets are represented by user-based, time-based and content-based. There are a total of 29 features. The features along with their short description are presented in Table 2.1.

Shu *et al.* mentioned that some features highly correlate with retweet rate such as the number of followers, number of followees, age of the user's account while other features have slight impact only on this rate such as the presence of URL and hashtag. Moreover, the total number of past tweets and the number of tweets that are favorited by the user seem to have little or no relationship with the retweet number [Suh 2010]. We reused all these features in our model. Those features are marked with a^+ in Table 2.1 and in the rest of this chapter. The other features are features that we defined and correspond to one main contribution of the work reported in this chapter.

2.3.1.1 User-based features

We hypothesized that a person who highly interacts with other people will in turn receive corresponding attention. Thus we took into account the interaction between the user who sends the tweet and social networks. We first reused the features that are related to the retweet number mentioned in [Suh 2010]:

- *Total_of_tweets*$^+$*:* the total tweets that the user has posted in his timeline in the past.
- *No_of_followers*$^+$*:* the number of followers this user currently has.
- *No_of_followees*$^+$*:* the number of other users that this user is following.

Table 2.1: Features used to predict retweet rate of a given tweet. Features with a$^+$ correspond to Suh *et al.* features [Suh 2010] while the other features correspond to one important contribution of this work.

	Features	Description	Data Type
User-based	1. Total_of_tweets$^+$	The total of past tweets that the user has posted in the time line	#Numeric
	2. No_of_followers$^+$	The number of followers this user currently has	#Numeric
	3. No_of_followees$^+$	The number of other users that this user is following	#Numeric
	4. Age_of_account$^+$	The number of days since the user account has been created	#Numeric
	5. No_of_favourite$^+$	The number of tweets the user has liked in the timeline	#Numeric
	6. No_groups_user_belongs	The number of public groups that the user is a member of	#Numeric
	7. Aver_favou_per_day	The average of likes that the user has made per day	#Numeric
	8. Aver_tweets_per_day	The average of tweets that the user has posted per day	#Numeric
	9. User_name_len	The length of the user's name	#Numeric
Time-based	10. Is_posted_at_hol	The tweet is created on public holiday	Boolean
	11. Is_posted_at_noon	The tweet is created from 11.am-13.pm	Boolean
	12. Is_posted_at_eve	The tweet is created from 6.pm-9.pm	Boolean
	13. Is_posted_at_wee	The tweet is created at weekend	Boolean

Continue on the next page

Table 2.1 Features used to predict retweet rate of a given tweet. Continued from previous page

	Features	Description	Data Type
Content-based	14. Contain_location	The tweet contains a location name	Boolean
	15. Contain_org	The tweet contains an organization name	Boolean
	16. Contain_tvshow	The tweet contains a television show name	Boolean
	17. Sentiment_level	The tweet is classified into sentiment levels	{positive, negative, objective}
	18. Contain_video	The tweet contains a video	Boolean
	19. Contain_picture	The tweet contains a picture	Boolean
	20. Contain_upper	The tweet contains upper words	Boolean
	21. Contain_number	The tweet contains number	Boolean
	22. Contain_excl	The tweet contains an exclamation mark	Boolean
	23. Contain_rt_term	The tweet contains 'RT' term	Boolean
	24. Con_user_mentioned	The tweet mentions a user name	Boolean
	25. Contain_rt_sugges	The tweet contains one of the retweet suggestion term:Pls RT, please retweet, RT for..	Boolean
	26. Contain_URL$^+$	The tweet contains an URL	Boolean

Continue on the next page

Table 2.1 Features used to predict retweet rate of a given tweet. Continued from previous page

	Features	Description	Data Type
Content-based	27. Contain_hashtag^{+}	The tweet contains a hashtag	Boolean
	28. Opt_length	The length of the content is between 70 to 100 characters	Boolean
	29. Len_of_text	The length of the content	#Numeric

- *Age_of_account*$^+$*:* the number of days since the user account has been created until the day the tweet was collected.
- *No_of_favourite*$^+$*:* the total number of tweets the user has liked in the timeline.

In addition, we added several new features:
- *No_groups_user_belongs:* the number of public groups or communities that the user is a member of.
- *Aver_favou_per_day:* Average number of likes that the user likes per day. This features is calculated by dividing *No_of_favourite* by *Age_of_account.*
- *Aver_tweets_per_day:* Average number of tweets that the user writes per day. This features is calculated by dividing *Total_of_tweets* by *Age_of_account.*
- *User_name_len:* the length of the user's name.

All the features from this category are numeric values. These features are extracted and calculated from the fields a tweet is composed of when collected using Twitter API[4].

2.3.1.2 Time-based features

We hypothesized that a majority of retweets are written shortly after the tweet is posted and thus that a tweet posted in 'free hours' is more likely to receive more retweets. The time-base features that consider the time the tweet is generated, include:
- *Is_posted_at_hol:* we checked if the tweet is posted during holidays using the Holiday python library (`https://pypi.python.org/pypi/holidays`). We first considered the public holiday of the user's location during the time of collecting the datasets (as available in subsection 2.3.4). If the user does not mention any location in her or his profile, we checked the tweet posting time with holidays of all 23 countries which is included in the Holiday python library such as United States, United Kingdom, Spain, Germany and others.
- *Is_posted_at_noon*: we checked whether the tweet is posted at noon from 11 a.m to 1p.m or not.
- *Is_posted_at_eve:* we checked whether the tweet is posted in the early evening from 5 p.m to 9 p.m or not.
- *Is_posted_at_wee:* we checked whether the tweet is posted at the weekend or not.

[4] `https://developer.twitter.com/en/docs/api-reference-index`

Each of these checks corresponds to a boolean feature in the tweet representation.

2.3.1.3 Content-based features

We added several new content-based features considering the content of the message such as Named Entities (NE), sentiment level, media attachment, content enhancement, content size and others.

Named entity: A tweet that mentions a specific location name makes it more attractive [Lingad 2013] and may lead to retweetability. For example, the tweet: *"Tonight's moonrise over the #statueofliberty in New York City."* got 1.200 retweets. Also, a TV show or a business company included in a tweet makes it more popular. 4,600 people have retweeted the post: *"Here's a look at our #PrimeDay sneak peek of #TheGrandTour Season 2"*. We used Ritter's Named Entity Recognition (NER) tool [Ritter 2011] to check if the tweet contains a location name (*Contain_location*), an organization name (*Contain_org*) or a TV show reference (*Contain_tvshow*). We supposed that information about well-known named entities included in the tweet will get much attention and will be shared more. The *Contain_location*, *Contain_org* and *Contain_tvshow* features are boolean values.

We distinguished between sentiment level, media attachment, content enhancement, and content size.

Sentiment level: We hypothesized that in special events such as epidemics or promotion campaigns, extremely positive or negative tweets are normally used to express hot and updated news and these tweets are more prone to be retweeted.

For example, the tweet about the death toll from a hurricane in Haiti *"The death toll in Haiti from Hurricane Matthew is 339. That's what environmental racism looks like. #BlackLivesMatter"* got more attention as 1,500 retweets were posted in a short time. Another tweet about the winner of Golden globe awards in 2017: *"Congratulations to Three Billboards Outside Ebbing, Missouri (@3Billboards) - Best Motion Picture - Drama - #GoldenGlobes* has been retweeted 1,900 times.

We thus defined a new feature to capture the sentiment of tweets that we called Sentiment_level. We used a "scikit-learn" machine learning library[5] to classify tweets into positive, negative or neutral sentiment. We

[5] `http://scikit-learn.org/stable/`

trained the model on the training dataset including 6,030 annotated sentiment tweets provided by Semval-2013 international workshop on Semantic Evaluation, Sentiment analysis on Twitter task[6] [Hltcoe 2013] and on 10,600 shorten annotated sentiment movie reviews[7] [Pang 2004]. The first dataset was annotated by the Mechanical Turker who first marked all the subjective words/phrases in the sentence and then indicated the overall polarity of the sentence which is positive, negative or objective. The sentiment of movie reviews in the second dataset is determined based on the star rating accompanied. For example, with a five-star system (or compatible number systems): three-and-a-half stars and up are considered positive, two stars and below are considered negative while with a letter grade system: B or above is considered positive, C- or below is considered negative. From our experiments, Stochastic Gradient Descent (SGD) classifier gives the best accuracy on the training set among classifiers, thus we used the SGD classifier to extract sentiment features in the three collections of tweets described in subsection 2.3.4. We kept three possible values for this sentiment feature: positive, negative or objective.

Media attachment: Twitter users often attach media sources to make their tweets more lively and more attractive. A picture attached in a message *"When you're finally home alone and u could be yourself"* probably contributed this tweet to get 2,231 retweets. We therefor defined features related to attached items. More specifically, we checked if the tweet contains a picture *(Contain_picture)* or a video *(Contain_video)*. These two features are Boolean values.

Content enhancement: We took into account some features that can enhance retweetability such as the fact the tweet contains an upper word (*Contain_upper*), a number (*Contain_number*), an exclamation mark (*Contain_excl*), a 'RT'[8] term (*Contain_rt_term*) or mentions a user name (*Con_user_mentioned*). These features were defined as Boolean values.

We also considered some retweet suggestion terms which are effective in asking people to retweet (*Contain_rt_suggest*). For example the tweet *"For every retweet this gets, Pedigree will donate one bowl of dog food to dogs in need! #tweetforbowls"* got 788,844 retweets. The other tweet: *"With the current devastation in Houston, we are pledging $0.15 for every RT this gets! Please*

[6] https://www.cs.york.ac.uk/semeval-2013/task2/index.html

[7] https://pythonprogramming.net/new-data-set-training-nltk-tutorial/

[8] On Twitter, people often use 'RT' to stand for retweet

forward this along to help out those in need! has been widely spread since the number of retweet reached 1,161,494. We checked if a tweet includes the following retweet suggestion terms: 'please retweet', 'pls rt','retweet if','rt if','retweet to','rt to',' rt!','retweet for','rt for', 'retweet', 'please forward'. This feature is a Boolean value.

Besides, we reapplied two boolean features from [Suh 2010] which check if the tweet contains a URL (*Contain_URL*$^+$) or a hashtag (*Contain_hashtags*$^+$).

Content size: We considered the length of the tweet content which is limited to 140 characters (*Len_of_text*). We suppose that the ideal length of a message should be in between 70 and 100 characters so that there is room for people to put comments in addition to the content that they want to retweet (*Opt_length*). These two features are Boolean.

2.3.2 Processing time

The feature extraction process was implemented on the Osirim-IRIT platform[9] with 1 CPU 1.6 Ghz, and 64 GB of RAM.

For each dataset, we extracted the features from the tweets that are not retweeted and from unique tweets which are retweeted. Since a tweet may be retweeted several times, it can be stored repeatedly in the datasets. We thus only considered the original tweet one time with the latest 'number of retweets'. It took one week to extract features for the FirstWeek dataset and one week for the SecondWeek dataset but just few days for the Sandy dataset because of fewer number of tweets as presented in the Table 2.2.

2.3.3 Machine learning model

We cast the problem in two types of classification: i) binary classification to predict whether a tweet is going to be retweeted or not, and ii) multi-class classification to predict the level of retweet, like Hong Hong *et al.* [Hong 2011] did, using several classes corresponding to several levels of retweet.

There are several commonly used machine learning algorithms that could have been used for our purpose. We used different machine learning algorithms such as Naive Bayes (NB), Support Vector Machine (SVM) and Ran-

[9]IRIT, UMR5505 CNRS, France

dom Forest (RF) implemented on Java Weka library[10]. For SVM, there are two types of algorithm: kernel SVM and linear SVM. While kernel SVM works fast on small datasets, it took several days on large scale datasets and not applicable in our case. We thus choose a linear support vector classification Liblinear library[11] implemented on Weka to apply support vector classification.

For each collection, we used 10-fold cross validation. We also formed an experiment that implements transfer learning: we trained the model on one collection and tested it on a different collection.

Among these classifiers, NB and SVM gave very low results which are even smaller than the baseline while RF consistently achieved the best results. We thus only detailed the results of RF in the next session.

2.3.4 Data and evaluation framework

We conducted experiments and evaluated our model on three datasets which were collected from Twitter APIs: Sandy, FirstWeek and SecondWeek datasets.

The first dataset has initially been used by Tamine *et al.* [Tamine 2016] collected from 29th October 2012 to 31st October 2012 using the 3 keywords "sandy", "hurricane" and "storm" while the second and the third datasets were 1 percent of tweets collected during the first week and second week of January 2017 by IRIT, France [12] within a spam detection project [Washha 2016].

Table 2.2: The number of tweets and their distribution on the Sandy, FirstWeek and SecondWeek datasets used to evaluate our predictive model.

	Sandy	**FirstWeek**	**SecondWeek**
#of tweets	2,119,854	8,009,112	8,171,080
#of non-retweeted tweets	1,156,223	4,025,157	4,058,066
#of (unique) retweeted-tweets	204,232	2,017,979	2,080,962

Each tweet in these datasets is composed of pieces of information regarding a tweet such as the Unique Identifier (Id), the content of the tweet, the

[10] `http://weka.sourceforge.net/doc.stable/`
[11] `https://github.com/bwaldvogel/liblinear-java`
[12] IRIT, URM CNRS 5505 Université de Toulouse, France

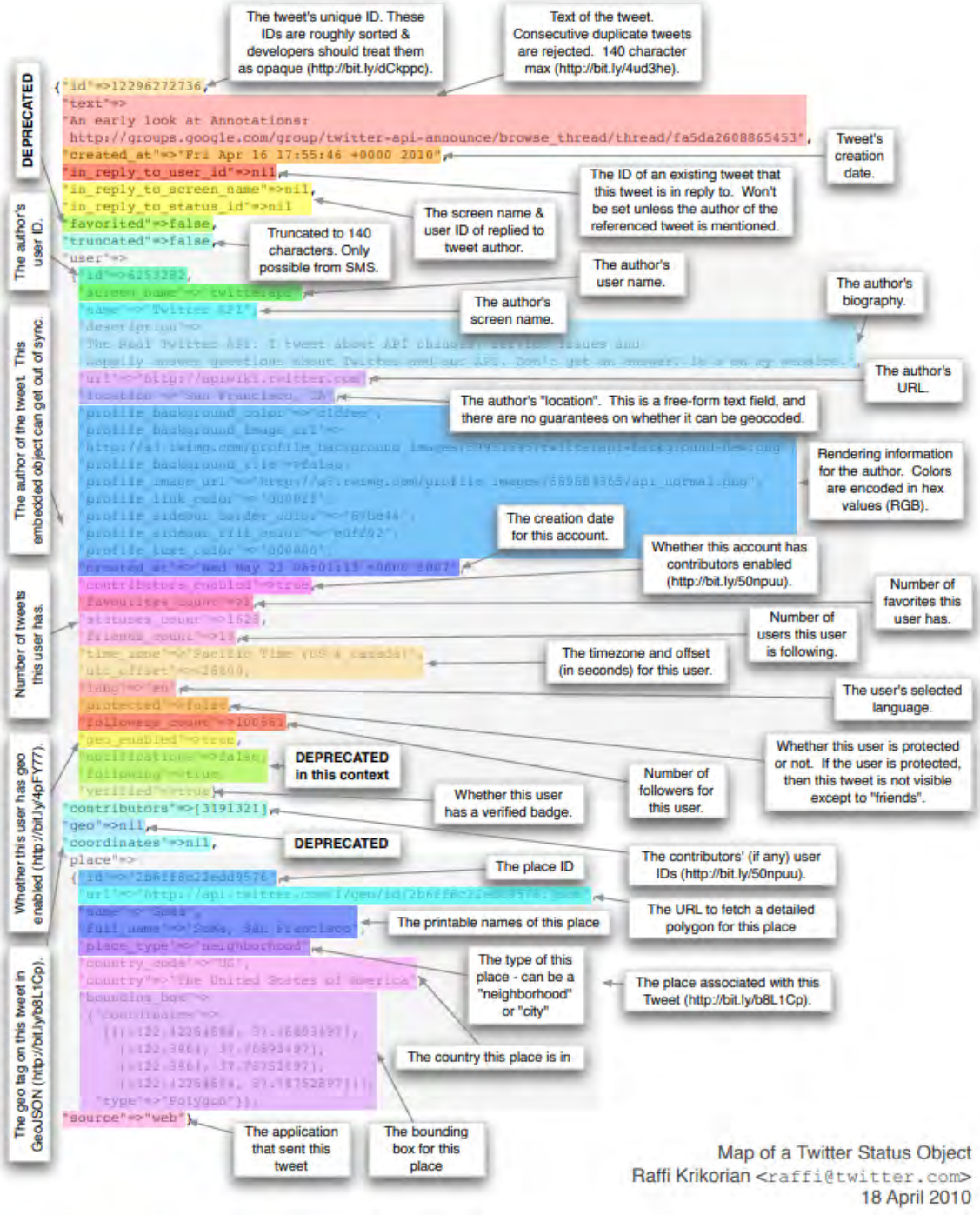

Figure 2.4: The map a Twitter status object.
Source: https://www.scribd.com/doc/30146338/map-of-a-tweet

time this tweet was created, the author of this tweet and others. Figure 2.4 presents map of a tweet object when collected from Twitter API. We used the value of the 'retweet_count' field which specifies the numbers of times a tweet has been retweeted to classify tweets in the predictive model (Section 2.3.5).

Table 2.2 reports the number of tweets and their distribution in the three datasets.

Baseline. The baseline model we report in this section uses all Suh's features [Suh 2010] and the Random Forest classifier which achieves the highest results among NB, SVM and RF. We compared it with the model that considers all the features we presented in Table 2.1 including the ones we defined in this work.

Table 2.3: Classes distribution of Sandy, FirstWeek and SecondWeek datasets used for muti-class classification. Class-0 corresponds to tweets that are not retweeted at all; class-1: tweets that are retweeted less than 100 times; class-2: tweets that are retweeted less than 10,000 times; class-3: tweets that are retweeted more than 10,000 times.

	Sandy	**FirstWeek**	**SecondWeek**
Class-0	1,156,223	4,025,157	4,058,066
Class-1	202,397	1,675,859	1,727,666
Class-2	1,832	327,381	339,328
Class-3	3	14,739	13,905

2.3.5 Experiments and results

2.3.5.1 Binary classification

To predict if a given tweet will be retweeted or not, we classified tweets into two classes: class-0 corresponds to tweets that are not retweeted while class-1 corresponds to tweets that are retweeted. Since there is a huge difference between the number of tweets from class-0 and tweets from class-1, we balanced these numbers during the classification process. There are several ways to deal with imbalanced data such as resampling the dataset, gener-

ating synthetic samples or penalizing models[13]. We chose to divide each dataset into several sub-sets. The tweets from class-1 are all kept whatever the sub-set is while the tweets from class-0 are divided into sub-sets so that the number of tweets from class-0 is approximately equal to the number of tweets from class-1 for each sub-set. More specifically, the sub-sets are built as follows:

- **Sandy dataset**. The tweets from class-0 were divided into five parts. Each sub-set included the entire tweets from class-1 (204,232 tweets) and one part class-0 tweets (about 231,245 tweets). We had thus five sub-sets for which we consider the average results when reporting them in Table 2.4.

- **FirstWeek dataset**. The tweets from class-0 was divided into two parts. Each sub-set included the whole tweets from class-1 (2,017,979 tweets) and one part class-0 tweets (about 2,012,579 tweets). We had thus two sub-sets for which we consider the average results when reporting them in Table 2.4.

- **SecondWeek dataset**. Similar to the FirstWeek dataset, the tweets from class-0 was divided into two parts. Each sub-set included the whole tweets from class-1 (2,080,962 tweets) and one part class-0 tweets (2,029,033 tweets). As in the previous case, we had two sub-sets for which we consider the average results when reporting them in Table 2.4.

Table 2.4 reports the F-measure of the binary classification (a tweet is predicted to be retweeted or not) on the Sandy, FirstWeek and SecondWeek datasets. * indicates statistically significant differences by Student's t-test with p-value smaller than 0.05. For each dataset, we report the average of F-measure over the sub-sets.

As it can be seen in the Table 2.4, our method significantly improves the F-measure of the binary classification on average and on every class compared to the baseline for all datasets.

On average, we achieve the F-measure of 0.704 for the Sandy dataset while this number is 0.654 for the baseline; it corresponds to an improvement

[13] `http://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/`

Table 2.4: F-measure of the binary classification using Random Forest on three datasets. * indicates statistically significant differences when using Student's t-test (p-value < 0.05).

	Class	Baseline	Our Method (RF)
Sandy	Class-0	0.692	0.734
	Class-1	0.614	0.668
	Av.	0.654	**0.704***
FirstWeek	Class-0	0.790	0.827
	Class-1	0.767	0.810
	Av.	0.776	**0.819***
SecondWeek	Class-0	0.790	0.818
	Class-1	0.773	0.804
	Av.	0.781	**0.811***
Training on FirstWeek, testing on SecondWeek	Class-0	0.860	0.873
	Class-1	0.672	0.708
	Av.	0.796	**0.817***

of 5%. For the FirstWeek dataset, the F-measure is improved from 0.776 to 0.819 which corresponds to an improvement of 4,3% while this improvement is 3% (from 0.781 to 0.811) for the SecondWeek dataset. When training the model on the FirstWeek dataset and testing on the SecondWeek dataset, we obtained the F-measure of 0.817 compared to 0.796 for the baseline, which corresponds to 2,1% of improvement. All of these improvements are statistically significant.

Interestingly, our model improves the F-measure on class-1 more than on class-0 when compared to the baseline even the number of tweets in class-1 is smaller than the number of tweets in class-0. For the Sandy dataset, the F-measure on class-1 is increased by 0.054 (from 0.614 to 0.668) while it is increased by 0.042 (from 0.692 to 0.734) on class-0 compared to the baseline. When the model is trained on the FirstWeek and tested on the SecondWeek dataset, the F-measure is improved by 0.036 (from 0.672 to 0.708) on class-1 but just by 0.013 (from 0.860 to 0.873) on class-0.

2.3.5.2 Multi-class classification

To predict the volume of retweets that a particular message will receive in the future, we divided the messages into four different classes like Hong *et al.* did [Hong 2011]: class-0 corresponds to tweets that are not retweeted at all, class-1 represents tweets that are retweeted less than 100 times, class-2 represents tweets that are retweeted less than 10,000 times, and finally class-3 represents tweets that are retweeted more than 10,000 times.

Table 2.5: F-measure of the multi-class classification using Random Forest on the three datasets. * indicates statistically significant differences when using Student's t-test (p-value <0.05).

	Classes	**Baseline**	**Our Method (RF)**
Sandy	Class-0	0.690	0.736
	Class-1	0.599	0.656
	Class-2	0.529	0.548
	Class-3	0.812	0.926
	Aver.	0.647	**0.698***
FirstWeek	Class-0	0.786	0.823
	Class-1	0.643	0.694
	Class-2	0.729	0.742
	Class-3	0.571	0.570
	Aver.	0.721	**0.760***
SecondWeek	Class-0	0.786	0.815
	Class-1	0.647	0.740
	Class-2	0.726	0.741
	Class-3	0.568	0.564
	Aver.	0.721	**0.755***
Training on FirstWeek, testing on SecondWeek	Class-0	0.856	0.868
	Class-1	0.513	0.545
	Class-2	0.588	0.651
	Class-3	0.449	0.547
	Aver.	0.734	**0.758***

Table 2.3 presents the class distribution of the Sandy, FirstWeek and SecondWeek collections.

As can be seen in Table 2.3, the number of tweets in classes are very imbalanced. To solve this problem we combined two steps:

- **Step 1** Generating synthetic samples by randomly sampling attributes from instances of class-2 and class-3 using Synthetic Minority Oversampling Technique (SMOTE). This algorithm selects some similar instances (using a distance measure) and perturbs an instance, one attribute at a time by a random amount within the difference to the neighboring instances [Chawla 2002]. We configured SMOTE implemented on java Weka library to oversample class-2 and class-3 as follow: setNearestNeighbors = 5 and setPercentage = 100. As a result, the number of tweets from class-2 and class-3 were doubled.
- **Step 2** We divided each dataset into numbers of sub-sets like for binary classification. The tweets from class-1, class-2 (after SMOTE) and class-3 (after SMOTE) were kept the same for all sub-sets while the tweets from class-0 were divided into sub-sets so that the number of tweets from class-0 was approximately equal to the number of tweets in class-1.

As a result, we dealt with datasets as follow:

- **Sandy dataset**. The class-0 tweets were divided into five parts. Each sub-set included the whole tweets from class-1, whole tweets from class-2 (after SMOTE) and whole tweets from class-3 (after SMOTE) with a total of 206,067 tweets and one part class-0 tweets including about 231,245 tweets. We had thus five sub-sets.
- **FirstWeek**. The class-0 was divided into two parts. Each sub-set included the whole tweets from class-1, whole tweets from class-2 (after SMOTE) and whole tweets from class-3 (after SMOTE) with a total of 2,360,099 tweets and one part class-0 tweets including about 2,012,579 tweets. We had thus two sub-sets.
- **SecondWeek**. Like we did with the FirstWeek dataset, the class-0 was divided into two parts. Each sub-set included the whole tweets from

class-1, whole tweets from class-2 (after SMOTE) and whole tweets from class-3 (after SMOTE) with a total of 2,434,132 tweets and one part class-0 tweets including about 2,029,033 tweets. As in the previous case, we had two sub-sets.

When reporting the results, we averaged the performance over the sub-sets for a given collection.

These divisions do not completely guarantee the exact balance among classes, but reduce the importance of the majority class(es).

Table 2.5 presents the results of multi-class classification on three datasets in terms of averaged F-measure over sub-sets. * indicates statistically significant differences by Student's t-test with p-value smaller than 0.05.

Similarly to the binary classification, our method significantly improves the F-measure of the multi-class classification on average and on every class compared to the baseline for all three datasets.

On average, comparing to the baseline, we improve the F-measure by 0.051 for the Sandy dataset (from 0.647 to 0.698), about 0.04 both for the FirstWeek (from 0.721 to 0.760) and SecondWeek (from 0.721 to 0.755) datasets and 0.024 when training the model on the FirstWeek and testing on the SecondWeek datasets (from 0.734 to 0.758). All these improvements are significantly different from the baseline.

Whatever the class of all three datasets is, our method improves the F-measure compared to the baseline but with different performances. We achieved high F-measure on class-0, class-1 and class-2 (from 0.694 to 0.823 – see Table 2.5, column 4, line 6-7) but lower F-measure on class-3 (from 0.564 to 0.570 - see Table 2.5, column 4, line 9, 15) for the FirstWeek and SecondWeek datasets. This may be caused by the huge difference of the number of tweets in each class. The number of tweets in class-1 is about five time the number of tweets in class-2 and more than one hundred times the number of tweets in class-3 .

Compared to the FirstWeek and the SecondWeek datasets, we achieved lower F-measure for the Sandy dataset. The F-measures on class-0, class-1 and class-2 are 0.736, 0.656 and 0.548 respectively. However, we got very high F-measure on class-3 as it is 0.926. Since the number of tweets on class-3 is extremely small compared to thousand or hundreds of thousand in other classes, the similarity between the tweets from class-3 may have lead to the high performance of the classification for this class.

To conclude, our predictive model highly improves the F-measure compared to the baseline (statistically significant) both when predicting whether a tweet is going to be retweeted and when predicting the level of retweet. We improved the F-measure about 5% compared to the baseline when evaluating our model on three collections with a total of 18 millions tweets. Moreover, we achieved high F-measure on class-1 and class-2 which contain the majority of tweets in each collection and which were hard to predict in the state-of-the-art.

2.3.6 Most important features.

Our predictive model uses 29 features of which we have proposed 22 features in this work as a contribution. Some of these features are more useful than others to predict retweet numbers. We evaluated the importance of each feature by measuring the so-called Infogain attribute evaluator using Ranker search method in Weka. This tool calculates the relative weight of each feature in the model. The results are presented in the next sections.

2.3.6.1 Binary classification

The best five features when classifying tweets in binary classes are as follows (numbers in brackets corresponds to the weight; the higher the value is, the more important the feature is for the model) :

- **Sandy dataset:** No_of_followers$^+$ (0.118), No_groups_user_belongs (0.100), Is_posted_at_eve (0.077), Is_posted_at_noon (0.044), No_of_followees$^+$ (0.033)
- **FirstWeek dataset:** No_of_followers$^+$ (0.227), No_groups_user_belongs (0.113), Is_posted_at_hol (0.072), No_of_followees$^+$ (0.047), No_of_favourite$^+$ (0.041)
- **SecondWeek dataset:** No_of_followers$^+$ (0.237), No_groups_user_belongs (0.130), No_of_followees$^+$ (0.051), No_of_favourite$^+$ (0.043), Contain_picture (0.041).

We found that two features we reapply from Suh *et al.* (number of followers and followees) are consistently in the top five features. This result

matches with their finding that the number of followers and followees have a very strong relationship with the retweetability. On the contrary, the number of tweets that the user has liked in his timeline was found to have very little impact on the retweet number by Suh *et al.* [Suh 2010] while it is one of the best five features on our Firstweek and Secondweek datasets.

One important result is that one of the new features we defined, the number of groups or communities that the user is a member of (No_groups_user_belongs), is the second best features over the three datasets. The results also show our time-based features play an important role in predicting whether the tweet will be retweeted or not. The retweetability of a given tweet on two over three collections is affected by the time posting features: in the evening (Is_posted_at_eve) and at noon (Is_posted_at_noon) or during holiday (Is_posted_at_hol).

The *Contain_picture* is the most important content-based feature in the top five features of the SecondWeek dataset while this feature is the sixth best in the FistWeek dataset and sixteenth best in the Sandy dataset. The low rank of Contain_picture in the Sandy dataset may be caused by the very small number of tweets containing pictures.

Apart from the above features, the next important features on three datasets with different weight are: Aver_tweets_per_day, Total_of_tweets$^+$, Len_of_text, Aver_favour_per_day, Contain_hashtag $^+$, User_name_len, Contain_URL$^+$, Sentiment_level, Con_user_mentioned and Contain_rt_suggestion.

2.3.6.2 Multi-class classification

Similarly to binary classification, two features from the literature No_of_followers$^+$, No_of_followees$^+$ and one of features that we defined (No_groups_user_belongs) are consistently in the best five features.

More precisely, the best five features when classifying tweets in multi-class classification are as follow:

- **Sandy dataset:** No_of_followers$^+$ (0.141), No_groups_user_belongs (0.119), Is_posted_at_eve (0.077), Is_posted_at_noon (0.045) , No_of_followees$^+$ (0.038)

- **FirstWeek dataset:** No_of_followers$^+$ (0.329) , No_groups_user_belongs (0.228), Len_of_text (0.213), No_of_followees$^+$ (0.131), Age_of_account$^+$ (0.115)

- **SecondWeek dataset:** No_of_followers$^+$ (0.372), No_groups_user_belongs (0.331), Len_of_text (0.262), No_of_followees$^+$ (0.150), Age_of_account$^+$ (0.125)

While the number of tweets that the user has liked in his timeline (No_of_favourite) is very important for binary classification, it is not so important in multi-class classification. Instead, the tweet length (Len_of_text) is significant for multi-class classification while it was not for binary classification. Indeed it is the third best feature in both the FirstWeek and the SecondWeek datasets. Our result for the Age_of_account feature matches with Suh's finding when they showed that it has a significant relationship with retweet rate. In both the FirstWeek and SecondWeek datasets, Age_of_account is the fifth best feature with the weights 0.115 for the FirstWeek dataset and 0.125 for the SecondWeek dataset.

When considering the Sandy dataset, the order of the best five features in multi-class classification is the same as in binary classification, although the weights are little higher for all the features. The top five features in multi-class classification for the FirstWeek and the SecondWeek datasets are similar; but relatively different from those for binary classification. The Is_posted_at_hol, Contain_picture and No_of_favourite $^+$ features are significant in binary classification but not in multi-class classification.

Apart from the above features, the next important features on the three datasets are: Aver_tweets_per_day, Aver_favour_per_day, Total_of_tweets$^+$, Contain_picture, No_of_favourite$^+$, Contain_hashtag $^+$, User_name_len, Contain_URL$^+$, Sentiment_level and Con_user_mentioned.

2.3.7 Correlations between features

To evaluate if the new features we defined are dependent from existing features and independent from each others, we calculated the correlations between features. We applied the Principle Component evaluator using Ranker search method implemented on Weka. We obtained a correlation matrix which measures the degree of association between features for each dataset. We also used R programing language to visualize the correlations.

Figure 2.5, 2.6, 2.7 presents the correlation matrices between features for the Sandy, FirstWeek and SecondWeek datasets. The higher the correlations are, the larger and bolder the circles are.

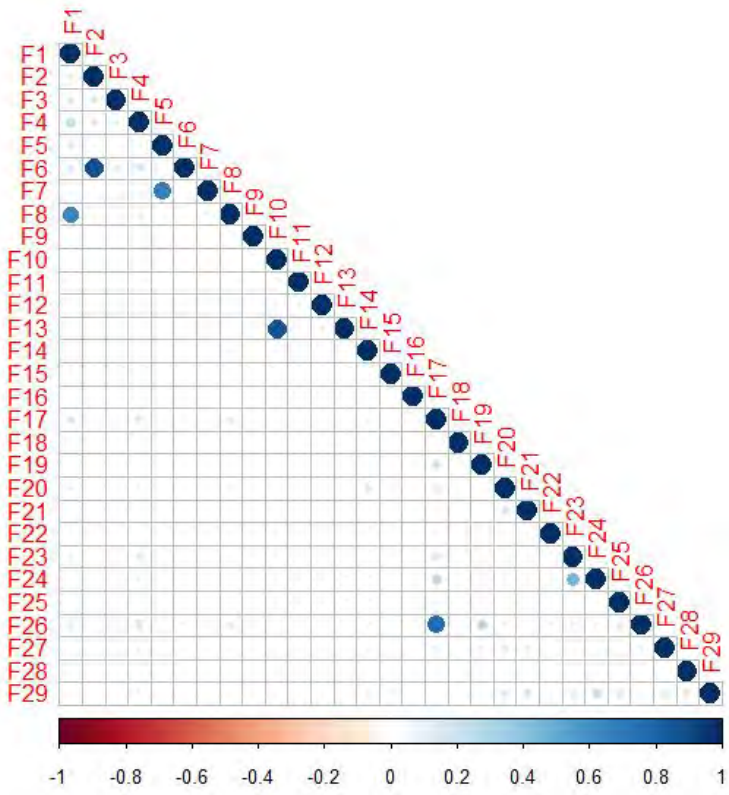

Figure 2.5: The correlation between features in the Sandy dataset. The large and bold circles represent high correlations. The features are in the same order as in Table 2.1

The first important point is that there are a few correlations that are significant. As it can be seen in Figure 2.5, and this holds also for the two other datasets, most of the features are independent from each others. Indeed, most of the correlation values are between -0.2 to 0.2 for the three datasets. The highest correlations in each dataset are as follow:

- **Sandy dataset:** *No_groups_user_belongs correlates with No_of_followers$^+$ (0.86)*; Is_posted_at_week correlates with Is_posted_at_hol (0.86); Sentiment_level correlates with Contain_URL$^+$ (0.75); Aver_favou_per_day correlates with No_of_favourite$^+$ (0.68); Aver_tweets_per_day correlates with Total_of_tweets$^+$ (0.65);

- **FirstWeek dataset:** *No_groups_user_belongs correlates with No_of_followers$^+$ (0.74)*; Sentiment_level correlates with Con_user_mentioned

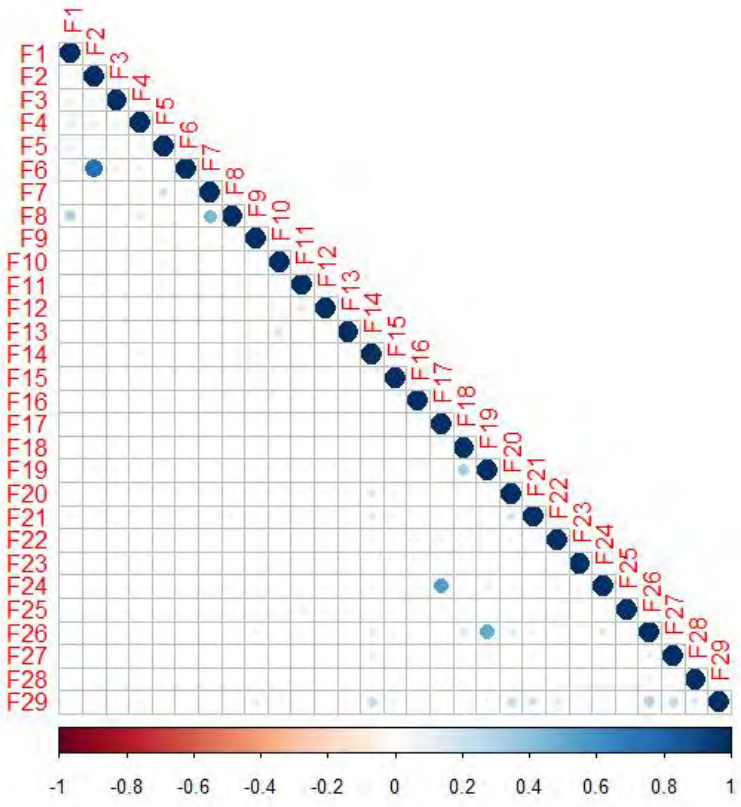

Figure 2.6: The correlation between features in the FirstWeek dataset. The large and bold circles represent high correlations. The features are in the same order as in Table 2.1

(0.53) ; Contain_picture correlates with Contain_URL$^+$ (0.5) ; Aver_favou_per_day correlates with Aver_tweets_per_day (0.45);

- **SecondWeek dataset:** *No_groups_user_belongs correlates with No_of_followers$^+$ (0.84)*; Sentiment_level correlates with Con_user_mentioned (0.52) ; Contain_picture correlates with Contain_URL$^+$ (0.49); Is_post_at_week correlates with Is_posted_at_hol (-0.33).

The correlations for the FirstWeek and the SecondWeek datasets are very similar to each other but slightly different from the Sandy dataset. The only significant correlation that exists across the three datasets is between No_groups_user_belongs (a feature that we defined) and No_of_followers$^+$ (a feature from the literature).

Apart from this, the other significant correlations are between existing

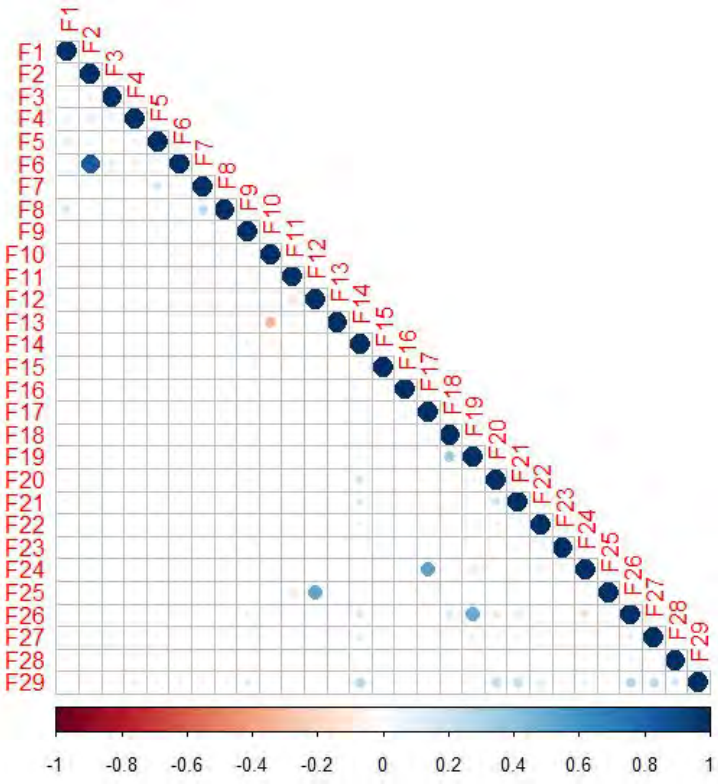

Figure 2.7: The correlation between features in the SecondWeek dataset. The large and bold circles represent high correlations. The features are in the same order as in Table 2.1

features and some features that we defined but that are little weighted in the predictive model and thus which are not important for the model. For example, in the Sandy dataset, Sentiment_level (which correlates with Contain_URL$^+$) got 0.0009 importance weight while the weight of the Aver_favou_per_day feature (correlates with No_of_favourite$^+$) is 0.003. In addition, Aver_tweets_per_day which correlates with Total_of_tweets$^+$ is also a weak feature in our model.

To conclude, there is very few meaningful correlations between the features in the three datasets; most of the correlation values are in between -0.2 and $+0.2$. The correlations that are statistically significant between the features we defined in this work and the features from the literature are not important for the predictive model (low weights). The features we developed in this work and which are important for the predictive models (main features) do not correlate with existing features from the literature. This

is the case for Is_posted_at_noon, Is_posted_at_eve, Is_posted_at_hol, and Len_of_text. Moreover, the results presented in Section 2.3.5 show that the combination of our features and existing features significantly improves the performance of the predictive information- diffusion model.

2.4 Predicting the diffusion of brand stories on microblogs

The popularity of on line social networks has rapidly increased over the past few years. While serving its primary purpose of connecting people, social networks also plays a major role in successfully connecting marketers with customers. According to Twitter Stats for Businesses[14], 65.8% of U.S. companies are now using Twitter for marketing purposes. As in the same source, 47% of people who follow a brand on Twitter are more likely to visit that company's website. During discussions among consumers on social networks, stories about products or brands are formed and spread thanks to the retweet functionality. By repeating the message, all user's followers are able to read the message, thus helping the message to be broadcasted and reach a large amount of people.

Recently, there have been a few studies focused on social networks in marketing. Researchers showed that using social networks opens several new opportunities for businesses to market their products.

According to Assaad and Gotta, the established communities around products and services help businesses to build the brand loyalty, trust and to facilitate the viral marketing through self-emergent customer testimonials. Social networking can also help businesses to find new customers and to build brand intelligence as well as markets. In addition, the interactive contact between stakeholders can be created and that enable businesses to get feedback directly from their customers [Assaad 2011, Mike Gotta 2006]. In another study, Mangole *et al.* hypothesized that, since the social media allows one person to communicate with other thousands or millions people about products or brands, the impact of customer-to-customer communications increased in the marketplace. Therefore, managers should start brand stories or discussions to be followed by customers or contribute to existing discus-

[14] https://expandedramblings.com/index.php/march-2013-by-the-numbers-a-few-amazing-twitter-stats/

sions in a way that serves the business and performance goal [Mangold 2009].

Similarly, Gensler *et al.* supposed that social media significantly affects the brand management because of its dynamic, ubiquitous and regular interaction. Consumers are becoming pivotal authors of brand stories. Such stories can create advertisements that are more effective than usual advertisements created by company-generated stories. Thus, businesses may want to stimulate and promote positive stories to spread information on their brand [Gensler 2013].

In a review of existing work in network-based marketing on social media [Rogers 2012], the authors supposed that network structure, themes and user profiles significantly impact on the diffusion and adoption of marketing post in Facebook. In addition, the life cycle of a viral content includes four different stages: introduction, growth, maturity and decline. These phases are important to be understood for people so that they can know when they no longer benefit from the viral process or whether additional investment should be included to delay a potential decline. In [Hennessy 2016], Hennessy *et al.* proposed a method to develop a profile for social media personality based on "official" resources from a person's website, his tweets and data from his followers. The authors also suggested a method that helps businesses to determine which social media personalities would be a good fit for their marketing campaign.

Yu *et al.* analyzed the characteristics that contribute to the attractiveness of a social marketing messages in terms of the number of "likes". They considered the content and media type of the post and evaluate the method on a Facebook collection regarding to restaurants. They found that restaurants use some common marketing strategies to promote their product such as using unique public images, introducing new dishes and running advertisement campaigns like contests. Besides, the messages in the form of "status" or "photo" are more popular than message in the form of "link" or "video", probably because of the extra effort to click or play the link/video [Yu 2011]. These findings are partly as similar as what Sabate *et al.* concluded in their work. The authors showed that images and videos included in a message increase the number of "likes". In addition, using images and posting in a proper time are significantly impact on the number of comments, whereas the use of links may decrease this metric. These results are released from their conceptual model which reflects the influence of the content and the time frame on the attractiveness of a branded message by using several linear

regressions on 164 Facebook posts [Sabate 2014].

Our work aims at helping business managers to predict the diffusion of a given brand story on social networks as well as which features make a message popular. This work also helps managers to understand and better control the propagation of stories related to their brand or products. In addition, the managers can create a discussion or join/contribute to the discussion in order to be consistent with business's missions and goals. Also, they can propose solutions to control or promote the brand stories on social networks.

More precisely, we study two related research questions: (1) Is it possible to predict whether a tweet about a brand story is going to be spread i.e. re-tweeted? and (2) Can the level of diffusion be modeled and thus can we predict the level of diffusion of a new tweet that is advertised a specific product?

We reapplied our model which is presented in Section 2.3 plus some new features. We show that, we significantly improves by about 4% F-measure compared to the state of art methods for predicting retweetability of a tweet when evaluating our model on tweet collections about a brand stories generated by consumers and by the owner of the brand.

2.4.1 Tweet representation

We hypothesized that both the tweet content and the user who generates it have impacts on tweet diffusion. In this section, we reused all 29 features including user-based, time-based and content-based features presented in Section 2.3.1 (see the short description of these 29 features in Table 2.1). In addition, we added three additional features that we considered to be important in making tweet about a product or brand more popular:

- *User_is_verified:* indicates whether the user's account is verified or not. An account is verified if it is an account of public interest in the areas of music, acting, fashion...The verified Twitter accounts are mostly of companies or famous people in entertainment area such as music, fashion or movie. For example, A tweet from Chanel official account has been shared 7,700 times *"The story of the #CHANELSpringSummer 2018 show. #PFW."*. We hypothesize that stories about a product/brand written by a verified user are easily forward by a large number of their fans.

- *User_is_well_known:* indicates whether the user is well-known or not.

We supposed that tweets created by well-known people get more attention from audiences and thus are more likely to be retweeted. Indeed, a tweet from Tim Cook - a CEO of Apple - about the Pokemon application got 3,000 retweets in a short time: *"You never know who you'll run into on the Apple campus! The power of ARKit is coming to @PokemonGoApp today — taking its AR to a new level, including more interactivity between Pokémon and Trainers."*. We considered a user as well-known if his or her name appears in in DBPedia [15]. We used an end point framework (http://dbpedia.org/snorql/) to check the existence of the user name in DBpedia.

- *Contain_famous_person.* We hypothesized that tweets about a product or brand containing a well-known name in its content will make it more attractive and will be shared more. A tweet about a Gucci custom mention Harry Styles made it being retweeted 4,800 times: *"Performing at NY's Radio City Music Hall, @Harry_Styles wore a #Gucci custom metallic floral silk jacquard Monaco suit."*. We used Ritter's named entity extraction tool [Ritter 2011] to check whether the tweet contains a person name in the tweet content and then checked if this name is introduced as a person on DBpedia as previously.

All these additional features are boolean values.

2.4.2 Machine learning model

We used different machine learning algorithms such as Naive Bayes (NB), Support Vector Machine (SVM) and Random Forest (RF) implemented on Java Weka library[16]. We report RF results only since they correspond to the best results we obtained, both for the baseline and for our model. For each collection, we used 10-fold cross validation.

2.4.3 Data and evaluation framework

We conducted experiments and evaluated our model on two types of collections: 1) collections of tweets about a brand stories generated by consumers and 2) collections about a brand stories generated by the company who owns the brand. This section presents the datasets and experiments on

[15] BDpedia structures the information from Wikipedia pages; it can be queried using SPARQL to extract structured information locally stored in DBpedia or through an endpoint framework.

[16] `http://weka.sourceforge.net/doc.stable/`

the first type of collection generated by consumers (namely IPhone, Galaxy and Gucci) while the datasets and experiments on collections generated by the company are presented in section 2.4.5

The IPhone, Galaxy and Gucci datasets were extracted from 1 percent of tweets dataset collected by IRIT, France[17] from 21 September 2015 to 31 May 2017 using three corresponding keywords 'iphone', 'galaxys' and 'gucci'.

Each tweet in these datasets is composed of several pieces of information regarding a twitter status as presented in the Figure 2.4. We used the value of the 'retweet_count' field which specifies the numbers of times a tweet has been retweeted to classify tweets (Section 2.4.4).

Table 2.6 reports the number of tweets and their distribution in the three datasets.

Table 2.6: The number of tweets and their distribution for the IPhone, Galaxy and Gucci datasets used to evaluate our predictive model.

	IPhone	**Galaxy**	**Gucci**
# of tweets	2,188,923	174,909	242,956
# of non-retweeted tweets	1,483,705	134,443	74,543
# of (unique) retweeted tweets	312,003	19,391	51,805

Table 2.7: Classes distribution of three datasets used for multi-class classification. Class-0 corresponds to tweets that are not retweeted at all, class-1: tweets that are retweeted less than 100 times, class-2: tweets that are retweeted less than 10,000 times, and class-3: tweets that are retweeted more than 10,000 times.

	IPhone	**Galaxy**	**Gucci**
Class-0	1,483,705	134,43	74,543
Class-1	271,147	17,446	41,752
Class-2	37,355	1,915	9,968
Class-3	501	30	85

[17]IRIT, URM CNRS 5505 Université de Toulouse, France

Baseline. The baseline model we report uses RF on all Suh's features [Suh 2010]. We compare it with the model that considers all the features we presented in Section 2.4.1 (our features plus Suh's features).

2.4.4 Experiments and results

2.4.4.1 Binary classification

As we did in Section 2.3.5.1, to predict whether a given tweet about a product or brand will be retweeted or not, we classified tweets into two classes: class-0: tweets that are not retweeted and class-1: tweets that are retweeted. Since there is a huge difference between the number of tweets in the two classes (see Table 2.6), we balanced these numbers during the classification process using the same type of process as in Section 2.3.5.

Table 2.8: F-measure of the binary classification using different machine learning models on the IPhone, Galaxy and Gucci datasets. * indicates statistically significant differences by Student's t-test (p-value < 0.05) compared to the baseline.

	Class	Baseline	Our Method (RF)
IPhone	Class-0	0.824	0.853
	Class-1	0.820	0.851
	Av.	0.822	**0.852***
Galaxy	Class-0	0.864	0.879
	Class-1	0.857	0.873
	Av.	0.861	**0.876**
Gucci	Class-0	0.788	0.825
	Class-1	0.779	0.817
	Av.	0.783	**0.821**

For the IPhone and Galaxy datasets, we divided each dataset into several sub-sets. The tweets from class-1 are all kept for all sub-sets while the tweets from class-0 are divided into sub-sets so that the number of tweets from class-0 is as approximately same as the number of tweets from class-1. For

the Gucci dataset, since the number of tweets from class-0 are about one and a half as many as the number of tweets from class-1, we generated synthetic samples in class-1 50%. More specifically, the balance of classes are dealt as follows:

- **IPhone dataset**. The tweets from class-0 were divided into five parts. Each sub-set included the entire class-1 tweets (312,003 tweets) and one part class-0 tweets (about 296,741 tweets). We had thus five sub-sets.
- **Galaxy dataset**. The tweets from class-0 was divided into seven parts. Each sub-set included the whole class-1 tweets (19,391 tweets) and one part class-0 tweets (about 19,206 tweets). We had thus seven sub-sets.
- **Gucci dataset**. We generated synthetic samples by randomly sampling attributes from instances from class-1 using Synthetic Minority Over-sampling Technique (SMOTE) on Weka. The configure for SMOTE are setNearestNeighbors = 5 and setPercentage = 50. As a result, the tweets from class-1 are one and a haft the number of original: 77,707 tweets from class-1 and 74,543 tweets from class-0.

Table 2.8 reports the F-measure of the binary classification (a tweet is predicted to be retweeted or not) on the IPhone, Galaxy and Gucci datasets. For the IPhone and Galaxy datasets, we report the average of F-measure over the sub-sets.

As it can be seen in the Table 2.8, the trend is similar to the results presented in Section 2.3. Our method highly improves the F-measure of the binary classification on average and on every class compared to the baseline for all datasets.

On average, we achieve the F-measure of 0.852 for the IPhone dataset while this number is 0.822 for the baseline; it corresponds to an improvement of 3%, statistically significant. For the Galaxy datasets, the F-measure is improved from 0.861 to 0.876 which corresponds to an improvement of 1.5%. While the F-measure achieves 0.821, it increases by 3.8% compared to the baseline on Gucci dataset.

For all the three datasets, both our model and the baseline achieve higher performance on class-0 (tweets are not retweeted) than class-1 (tweets are retweeted) although the number of tweets in class-0 is smaller than the number of tweets in class-1. However, our method improves the results on class-1

more than on class-0 for the three datasets. The F-measure is improved by 3.1% on class-1 and by 2.9% on class-0 compared to the baseline for the IPhone dataset. For galaxy dataset, our method increases the F-measure from 0.857 to 0.873 (it corresponds to 1.6% increase) on class-1 and from 0.864 to 0.879 (1.5%) on class-0 compared to the baseline. We improve the F-measure by 3.8% on class-1 and 3.7% on class-0 compared to the baseline on the Gucci dataset.

We evaluated the importance of 32 features (we defined 25 features and reused 7 features from the literature) by applying the Infogain attribute evaluator using Ranker search method in Weka. The results are generally consistent with our finding in the previous Section 2.3.6. Since we used more features than we did in Section 2.3.5, we report here the best seven features when classifying tweets in binary classes ass follows (numbers in brackets corresponds to the weight; the higher the value is, the more important the feature is for the model):

- **IPhone dataset:** No_of_followers$^+$ (0.298), No_of_favourite$^+$ (0.116), No_of_followees$^+$ (0.093), Aver_favour_per_day (0.091), No_groups_user_belongs (0.084), Aver_tweets_per_day (0.066), Age_of_account$^+$ (0.062).
- **Galaxy dataset:** No_of_followers$^+$ (0.342), No_of_favourite$^+$ (0.219), Aver_tweets_per_day (0.185), Aver_favour_per_day (0.179), Age_of_account$^+$ (0.146), No_of_followees$^+$ (0.128), No_groups_user_belongs (0.121).
- **Gucci dataset:** No_of_followers$^+$ (0.242), No_groups_user-_belongs (0.168), Len_of_text (0.168), User_name_len (0.137), Aver_tweets_per_day (0.112), No_of_favourite$^+$ (0.108), Aver_favour_per_day (0.089).

Consistently with the results in the previous Section 2.3.6, we found that one feature we reapply from Suh *et al.* (namely No_of_followers$^+$) is consistently the best features on the three datasets. This result matches with their finding. Besides, the number of followees (No_of_followees$^+$) and age of account (Age_of_account$^+$), which are considered to be important in affecting to retweet rate by Suh, are also important features for the IPhone and the Galaxy datasets. The number of tweets that the user posted in the past (Total_of_tweets$^+$) has not much impact on retweetability on both Suh's findings and on ours. However, the number of tweets that the user has favourited

in his timeline was found to have very little impact on the retweet number by Suh *et al.* [Suh 2010] while it is one of the best seven features on our three datasets.

One important result is that some of the new features we defined, the number of groups or communities that the user belongs to (No_groups_user_belongs), average tweets (Aver_tweets_per_day) and average likes that the user makes per day (Aver_favour_per_day) are in the best seven features whatever the dataset we consider.

The best features for the IPhone dataset are similar to those for the Galaxy dataset with different weights. The situation is a little different for the Gucci dataset. The length of text (Len_of_text) and user name (User_name_len) are important in the Gucci dataset but not in the two other datasets. The reason might be that the length of messages and the length of the users' name are various in this dataset and those features are important for the diffusion of the messages while the values of those features little vary in the two other datasets.

Apart from the above features, the next important features on three datasets with different weights are as follow: User_is_verified, Total_of_tweets$^+$, Contain_hashtag$^+$, Contain_video, Contain_picture, Contain_upper and the Sentiment_level.

2.4.4.2 Multi-class classification

We predict the popularity of a tweet that is to say the volume of retweets that a given tweet about a brand/product will receive in the future. As we did in Section 2.3.5.2, we classified tweets into four different classes: class-0 (tweets that are not retweeted); class-1 (tweets that are retweeted less than 100 times; class-2 (tweets that are retweeted less than 10,000 times and class-3 (tweets that are retweeted more than 10,000 times).

Table 2.7 presents the class distribution of the IPhone, Galaxy and Gucci datasets. Similarly to the case of binary classification, the number of tweets in classes are very imbalanced (see Table 2.7). We dealt with this problem using the same type of process as in Section 2.3.5:

For the IPhone and Gucci datasets, we first divided each dataset into several sub-sets like we did for the binary classification. The tweets from class-1, class-2 and class-3 were all kept for all sub-sets while the tweets from class-0 were divided into sub-sets so that the number of tweets from class-0 is ap-

Table 2.9: F-measure of the multi-class classification using Random Forest on the three datasets. * indicates statistically significant differences by Student's t-test compared to the baseline.

	Class	Baseline	Our Method (RF)
IPhone	Class-0	0.821	0.849
	Class-1	0.719	0.761
	Class-2	0.588	0.640
	Class-3	0.130	0.114
	Av.	0.749	**0.787***
Galaxy	Class-0	0.861	0.878
	Class-1	0.772	0.800
	Class-2	0.582	0.613
	Class-3	0.115	0.184
	Av.	0.796	**0.818**
Gucci	Class-0	0.785	0.821
	Class-1	0.645	0.687
	Class-2	0.617	0.628
	Class-3	0.021	0.056
	Av.	0.707	**0.743***

proximately equal to the number of tweets from class-1. Then, we SMOTE tweets from class-2 and class-3 100% (setNearestNeighbors = 5 and setPercentage = 100).

For the Gucci dataset, since the number of tweets in class-0 are about one and half the number of tweets from class-1, we SMOTE the tweets from class-1 50% with (setNearestNeighbors = 5 and setPercentage = 50) and SMOTE the tweets from class-2 and tweets from class-3 100% (setNearestNeighbors = 5 and setPercentage = 100)

As a result, three datasets are processed as follow:

- **IPhone dataset**. The class-0 tweets were divided into five parts. Each sub-set included the whole tweets from class-1, whole tweets from class-2 (after SMOTE) and whole tweets from class-3 (after SMOTE) with a total of 346,859 tweets and one part class-0 tweets including

about 296,741 tweets. This process results in five sub-sets.

- **Galaxy dataset**. The class-0 was divided into seven parts. Each subset included the whole tweets from class-1, whole tweets from class-2 (after SMOTE) and whole tweets from class-3 (after SMOTE) with a total of 21,336 tweets and one part class-0 tweets including about 19,206 tweets. This process results in five sub-sets.

- **Gucci dataset**. The class-0 is kept as original. We formed a new set including whole tweets from class-1 (SMOTE 50%), whole tweets from class-2 (SMOTE 100%) and class-3 (SMOTE 100%) with a total of 82,734 tweets and all tweets from class-1 (74,543 tweets).

These divisions do not completely guarantee the exact balance among classes, but reduce the importance of the majority class(es).

Table 2.9 reports the results of multi-class classification on the three datasets in terms of averaged F-measure over the sub-sets.

Similarly to the binary classification, RF improves the F-measure of the multi-class classification on average and on every class compared to the baseline for all three datasets.

On average, comparing to the baseline, our method improves the F-measure by 3.8%, statistically significant, for the IPhone dataset (from 0.749 to 0.787), 2.2% for the Galaxy dataset (from 0.796 to 0.818) and 3.6%, statistically significant, for the Gucci dataset (from 0.707 to 0.743).

On every class of all the three datasets, our method improves the F-measure compared to the baseline but with different effectiveness. We achieve high F-measure on class-0, class-1 and class-2 (from 0.613 to 0.878) but lower F-measure on class-3 (0.056 to 0.184) for the three datasets. This may be caused by the very huge difference of the number of tweets per class. In the three datasets, the number of tweets in class-1 is about from four to seven times the number of tweets in class-2 and more than about five hundred times the number of tweets in class-3.

We also analyzed the most important features in the obtained model. Similarly to the binary classification, two features from the literature No_of_followers$^+$, No_of_favourite$^+$ and one of features that we defined (No_groups_user_belongs) are consistently in the best seven features.

More precisely, the best seven features when classifying tweets in multi-class classification are as follow:

- **IPhone dataset:** No_of_followers$^+$ (0.3414), Len_of_text (0.217), No_groups_user_belongs (0.199), No_of_favourite$^+$ (0.1504), User_name_len (0.1503), Aver_favour_per_day (0.142), No_of_followees$^+$ (0.137)

- **Galaxy dataset:** No_of_followers$^+$ (0.396), No_of_favourite$^+$ (0.256), Aver_favour_per_day (0.218), Aver_tweets_per_day (0.204), Age_of_account$^+$ (0.162), No_of_followees$^+$ (0.149), No_groups_user_belongs (0.148)

- **Gucci dataset:** No_of_followers$^+$ (0.316), No_groups_user-_belongs (0.215), Len_of_text (0.210), User_name_len (0.160), No_of_favourite$^+$ (0.125), Aver_favour_per_day (0.121), No_-of_followees$^+$ (0.113)

The number of followers (No_of_followees$^+$), which has strong relationship with retweetability in Suh's finding, is confirmed again since it is one of the best features in multi-class classification over the three datasets.

When considering the Galaxy dataset, the order of the best seven features in multi-class classification is the same as in binary classification. The top seven features in multi-class classification for the IPhone and the Gucci datasets are similar; but relatively different from those for binary classification.

Apart from the above features, the next important features on these three datasets are similar to those in the case of binary classification. These features are: User_is_verified, Total_of_tweets$^+$, Contain_hashtag$^+$, Contain_video, Contain_picture, Contain_upper and Sentiment_level.

2.4.5 Further experiments on datasets collected from official account of companies

In Section 2.4.4, we evaluated our model on three tweet collections about brand stories generated by consumers on social networks. In this section, we completed the set of experiments by considering tweets directly collected from the official Twitter accounts to see if the diffusion of stories written by the company is different from the diffusion from stories written by consumers.

There are three datasets of tweets that were collected from official accounts as follows:

- **@Samsung dataset**: is collected from the @SamsungMobileUS account using the keyword "galaxy".
- **@Chanel dataset**: is collected from the @CHANEL account using the keyword "chanel".
- **@Gucci dataset**: is collected from the @Gucci account using the keyword "gucci".

These datasets were collected from 21 September 2015 to 9 October 2017. The tweets and their distribution are presented in Table 2.10 and Table 2.11.

Table 2.10: The number of tweets and their distribution on three datasets.

	@Samsung	**@Gucci**	**@Chanel**
# of tweets	19,231	2,611	432
# of non-retweeted tweets	14,311	0	0
# of (unique) retweeted tweets	4,920	2,611	432

Table 2.11: Classes distribution of three datasets used for muti-class classification. Class-0 corresponds to tweets that are not retweeted at all; class-1: tweets that are retweeted less than 100 times; class-2: tweets that are retweeted less than 10,000 times; class-3: tweets that are retweeted more than 10,000 times.

	@Samsung	**@Gucci**	**@Chanel**
Class-0	14,311	0	0
Class-1	4,625	1,593	2
Class-2	295	1,017	423
Class-3	0	1	8

We formed experiences for binary classification on the @Samsung dataset and for multi-class classification on over three datasets. The imbalance data between classes are dealt as previously to make the data more balanced using the SMOTE technique (see Section 2.3.5 and 2.4.4)

Table 2.12: F-measure of the binary classification using Random Forest on the @Samsung dataset.

	@Samsung		
	Class-0	Class-1	Average
Baseline	*0.820*	*0.789*	*0.804*
Our model (RF)	0.848	0.834	**0.841***

Table 2.13: F-measure of the multi-class classification using Random Forest on the three datasets. * indicates statistically significant differences by Student's t-test.

	Classes	**Baseline**	**Our model (RF)**
@Samsung	Class-0	0.848	0.847
	Class-1	0.774	0.793
	Class-2	0.513	0.731
	Class-3	–	–
	Av.	0.794	**0.816***
@Gucci	Class-0	–	–
	Class-1	0.708	0.737
	Class-2	*0.665*	0.704
	Class-3	0	0
	Av.	0.688	**0.720**
@Chanel	Class-0	–	–
	Class-1	0	0
	Class-2	0.937	0.979
	Class-3	0	0.364
	Av.	*0.929*	**0.948**

The results for binary classification and multi-class classification are presented in Table 2.12 and Table 2.13 respectively. As can be seen from these tables, the results are consistent with the results that we showed in subsection 2.4.4. Our method improves the F-measure compared to the baseline on both types of classifications. We increase the F-measure by 3.7% on the @Samsung dataset for binary classification and about 2.2% on the @Sam-

sung, 3.2% on the @Gucci, 1.7 % on the @Chanel datasets for the multi-class classification. The number of tweets in class-3 are very few thus the F-measures on this class are low for all three datasets. However, we get high results on class-1 (except for the Chanel dataset because only two tweets belong to this class) and class-2 in which most of tweets in three collections belong to.

We also evaluated the importance of features for three datasets by applying the same method as we did in Section 2.3.6. For the binary classification, the important features are consistent with those we got for the datasets in the subsection 2.4.4.1: No_of_followers$^+$, Age_of_account$^+$, Aver_favou_per_day, Aver_tweets_per_day, No_of_followeees$^+$, No_groups_user_belongs and No_of_favourite$^+$.

For the multi-class classification, the features that are important in binary classification, are also important in this type of classification for the @Samsung and @Gucci datasets. Interestingly, some of our content-based features are most important features in the @Chanel dataset: Contain hashtag, Contain URL, Contain famous person, Contain Picture and Contain Video. In this dataset, all tweets are retweeted and almost all tweets are retweeted in high volumes (the rate of tweets in class-2 and class-3 are highest over three datasets). We would thus suggest to the business managers to combine the user-based features, time-based features and content-based features to increase the retweetability of a message on social networks.

To summary, when evaluating our model on the brand story datasets formed by the official companies, results are consistent with the ones when using the brand story datasets formed by consumers. In both cases, we highly improve the F-measure compared to the baseline. We also found that the diffusion of a brand story highly correlates with several features such as the number of follower, followees of the user who creates the story, number of groups that the user is a member of, the number of likes that the user made in his timeline, average tweets written per day, average likes per day. In addition, the hashtag, pictures, videos attached in the content also make the high retweetability. Notably, when the story is written by the official account of the company who creates the product, the popular of story is also affected by some other features such as age of account and the content contains famous person.

2.5 Discussions and conclusions

In this chapter, we addressed the problem of predicting the popularity of a given message on social networks. We casted this problem into binary classification (predict whether a tweet is going to be retweeted) and multi-class classification (predict the level of retweets). While reusing some features from literature, we added several new features including user-based, time-based and content-based features. We showed that, our model significantly improves the F-measure compared to the state of art (statistically significant) for both types of prediction when evaluating our model on various collections with total of about 18 millions tweets. In addition, we also achieved high F-measure on class-1 (tweets that are retweeted less than 100 times) and class-2 (tweets that are retweeted less than 10,000 times) which contain the majority of tweets in each collection and which were hard to predict in the state-of-the-art.

There are some features that are more important than others. We showed that the number of followers, followees, and the number of groups that the user is a member of, number of likes that the user has made in his timeline are the most important features for both types of prediction and consistently across the datasets. In addition, the time-based features we developed to check if a tweet is posted at noon, in the evening, at weekend or during holiday also strongly correlate with the retweetability. These two new features do not correlate with features from the literature.

Indeed, we also analyzed the correlations between features in the three datasets. Most of features are independent from each others. Some of our new features are 1) important to the model 2) do not correlate with existing features. The few features of ours that correlate with existing features have generally low weights when analyzing their impact for the predictive models. In addition, the results presented in Section 2.3.5, 2.4.4 and 2.4.5 show that the combination of the features we defined and existing features significantly improves the performance of the predictive model.

The second contribution of this chapter is the application of the proposed predictive model to predict the diffusion of brand stories on Twitter with some new additional features. We evaluated our model on two types of 'marketing' collection: collections of product/brand stories (in term of tweets) written by the consumers and written by the companies who own the brands/products. The results of experiments are consistent with our previ-

ous findings. For both types of collections, we highly improve the F-measure compared to the state-of-the-art for both binary classification and multi-class classification. We also ranked the features in the order of importance. As in our previous results: the number of followers, followees, favourites of the user and the number of groups that the user belongs to, are the most important features in making a tweet about a brand story to be retweeted. In addition, length of message, containing hashtag, URL and picture also affect on the retweetability. The age of account and famous person mentioned in the content of a tweet about a brand/product will make it more retweeted when this tweet is written by the company who owns the brand/product.

We believe that, our finding will help business managers to understand and predict the diffusion of stories related to their brand/products on social network. In addition, we also proposed features that could be used to make a message being popular. Based on these proposed features, managers can form stories on-line to broadcast their brands/products as well as propose strategies to control or promote customer-generated stories. Our model can also be applied to predict the propagation of information in other areas such as politics, epidemic, and disaster. We did not evaluate this by considering new tweets but keep this for future work.

There are several other points that could be considered in the future. The datasets we used (in Section 2.3) to evaluate our predictive model were collected during a rather short time. For example, the Sandy dataset was collected during a three days period while the Firstweek and Secondweek were collected in one week. Thus, it could be interesting to analyze further the impact of tweet posting time on retweetability when considering datasets collected in longer periods of time. In addition, we also suppose that some features such as the location, TV shows mentioned in the content or the reputation of the user name may be more important in other collections. A very few tweets contain such features in our collections.

For future work, we would like to implement some tasks. Firstly, we would like to collect larger datasets which include several tweets covering features that we proposed such as containing named entities in the content, the reputation of the user and more varied posting time.

Furthermore, we would like to defined additional features to represent tweets. For example, we could consider the Document to Vector (Doc2vec) [Le 2014] trained on one dataset to infer vectors for tweets for the other set. We would use these vectors as features. Our hypothesis is that if the

Doc2Vec is learned from topics, events and stories from a large training set, it would infer 'good' vectors for the testing set and lead to the improvement of classification.

One of features that we think it may be important in our model but it has not confirmed by the results is checking the sentiment level of a tweet. We thus could apply methods such the one proposed in [Kummer 2012, Sahni 2017] to improve the effectiveness of this feature extraction; this may lead to the improvement of the model effectiveness.

Finally, we would like to classify a tweet into topics such as sport, music, movie, fashion, daily weather news or technology news before predicting the popularity of this tweet. We believe that users are more interested in some topics than others. Finally, a track could be to analyze the influence when a follower retweets a tweet on one of his friends.

Chapter 3

Location Extraction from Microblogs

Summary

Abstract.

Five hundred million tweets are posted daily, making Twitter a major social media platform to broadcast events in several areas. These events are represented by three main dimensions: time, location and entity-related information. This work focuses on recognizing location in tweets which is an essential dimension for several applications especially for tweet-based geo-spatial applications, either when helping rescue operations during a disaster or when used for contextual recommendations. While the first type of application needs high recall, the second is more precision-oriented. This chapter studies the recall/precision trade-off, combining different methods to extract locations in tweets. In the context of short posts, applying tools that have been developed for natural language is not sufficient given the nature of tweets which are generally too short to be linguistically correct. Also bearing in mind the high number of posts that need to be handled, we hypothesized that predicting whether a post contains a location or not could make the location extractors more focused and thus more effective. We thus introduced a model to predict whether a tweet contains a location or not and show that location prediction is a useful pre-processing step for location extraction. When applying named entity recognition tools on the tweets we predicted as containing a location, the precision is significantly improved, from 85% to 96% for the Ritter collection and from 80% to 89% for the MSM2013 collection.

3.1 Introduction

The power of social networking is demonstrated by the huge number of worldwide social network users. According to Statista [1], this number is 2.46 billion in 2017. Twitter, which enables users to create short, 140-character messages, is one of the leading social networks. The extensive use, speed and coverage of Twitter makes it a major source for detecting new events and gathering social information on events [Weng 2011].

[1] https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/

As set out in Message Understanding Conference (MUC) campaigns [2], events have several dimensions that are equally important and require specific attention. The main dimensions are as follows:

- Location information which indicates where the event takes place;
- Temporal information which indicates when the event takes place;
- Entity-related information which indicates what the event is about or who the participants are.

The work presented in this chapter focuses on the location dimension. More specifically, it focuses on location extraction from tweets, which is vital for many applications, specifically for geo-spatial applications as well as applications linked with events [Goeuriot 2016a]. One of the first pieces of information transmitted to disaster support systems is where the disaster has occurred [Lingad 2013]. A location within the text of a crisis message makes the message more valuable than messages that do not contain any location [Munro 2011]. In addition, Twitter users are most likely to pass on tweets with location and situational updates, indicating that Twitter users themselves find location to be very important [Vieweg 2010].

Recognizing locations (a part of named entity recognition) in formal texts such as news and long documents has attracted many researchers. However, very little work has been successfully carried out on microblogs. The Stanford named entity recognizer (NER) [3] [Finkel 2005] achieves an 89% F-measure[4] for entity names on newswire, but only 49% for microblog texts [Bontcheva 2013]. Similarly, the Gate NLP framework[5] [Bontcheva 2013] achieves a 77% F-measure for long texts but only 60% for short texts. The Ritter named entity recognition [6][Ritter 2011], which is considered to be the state-of-the art, only achieves a 75% F-measure for Twitter.

As mentioned in [Bontcheva 2013], each tool has its strengths and limitations. While the Gate NLP framework achieves high recall (83%) and low

[2] http://www.itl.nist.gov/iaui/894.02/related_projects/tipster/muc.htm/. This conference were organized to encourage the developement of new and better methods of information extraction.

[3] http://nlp.stanford.edu/software/CRF-NER.shtml

[4] F-measure is approximately the average (harmonic mean) of the precision and recall

[5] https://gate.ac.uk/family/developer.html

[6] https://github.com/aritter/twitter_nlp

precision (47%), the Stanford NER achieves the opposite (recall 32%, precision 59%) for the development part of the Ritter dataset [Bontcheva 2013].

Since there are applications that need high recall (e.g. what has happened in a given location) and others that need high precision (e.g. which locations should we concentrate on first for a given problem) we hypothesized that combining existing location extraction tools could improve the accuracy of location extraction. We thus derived our first research question:

RQ1: *How much can we improve precision and recall by combining existing tools to extract the location from microblog posts?*

To answer this question, we combined various tools, namely, the Ritter tool [Ritter 2011], the Gate NLP framework (Gate)[Bontcheva 2013] and the Stanford NER [Finkel 2005]. We also proposed to filter the extracted locations using DBpedia[7]. We used DBpedia as follows: the locations extracted by previous tools are only considered as locations if DBpedia considers them as locations (taking account of the DBpedia endpoint framework). We therefore targeted either recall-oriented or precision-oriented applications.

By associating locations that both Ritter and Gate recognize, we achieved 82% recall (for the Ritter dataset) which is very appropriate for recall-oriented applications while the best single tool on this collection, Ritter, achieves 71% recall. This result can be explained by the fact that these methods use different clues to extract locations from tweets. On the other hand, when using DBPedia to filter out locations that Ritter recognizes, we reached a remarkable precision of 97% (for the Ritter dataset). This high result was obtained because imprecise recognized location names were discarded.

As mentioned earlier, social networks and microblogs are widely used media of communication. As a result, a huge number of posts and tweets are posted daily, but only a very small proportion contains locations [Sloan 2015, Ritter 2011, Cano Basave 2013]. For instance, in the Ritter dataset [Ritter 2011], which was collected during September 2010, only about 9% of the tweets contain a location. It is thus time consuming to try to extract locations from texts where no location occurs. If we could filter out tweets that do not contain locations, *prior* to extracting locations, then efficiency would be improved.

[7] `http://dbpedia.org/snorql/` BDpedia structures the information from Wikipedia pages; it can be queried using SPARQL to extract structured information locally stored in DBpedia or through an endpoint framework

This leads us to our second research question:

RQ2: *Is it possible to predict whether a tweet contains a location or not?*

We conducted a preliminary study by using location extraction tools only on tweets that contain locations; we achieved significantly higher accuracy than when implementing them on the entire datasets. This first result shows that if we could predict the fact that the text contains a location, it would be easier to extract this location.

One main contribution of this work is that we defined a number of new tweet features and used them as location predictors. Another contribution is that we evaluated the tweets using machine learning classifier algorithms with various parameters. In the experimental section, we show that the precision of NER tools for the tweets we predict to contain a location is significantly improved: from 85% to 96% for the Ritter collection and from 80% to 89% for the MSM2013 collection. This increase in precision is meaningful and crucial in systems where the location extraction needs to be very precise such as disaster supporting systems and rescues systems.

The rest of the chapter is organized as follows: Section 3.2 presents the related work; Section 3.3 details the location extraction method we promote and its evaluation. In Section 3.4, we explain our original method to predict location occurrence in tweets and show its usefulness and effectiveness. Finally, Section 4.5 is the discussions and conclusion.

3.2 Related work

With the rising popularity of social media, many studies proposed different ways to extract information from this resource. Previous similar studies can be grouped into two categories: location extraction and location prediction.

3.2.1 Location extraction

A piece of text related to a certain location includes information about that location. This information is either explicitly mentioned or inferred from the content. Identifying location names in a text is part of NER. In information extraction, it is a critical task for recognizing which parts of a text are mentioned as entity names.

Several NER systems address the problem of extracting a location specified in documents [Roberts 2008, Kazama 2008, Finkel 2005, Bontcheva 2013, Etzioni 2005]; however they do not perform well on informal texts. The reason is probably because text parsers use features such as word type, capitalized letters and aggregated context, which are often not exact in noisy, unstructured, short microblogs [Huang 2015].

Previous studies on location identification rely mainly on: 1) searching and comparing the text for entity names in a gazetteer, and/or 2) using text structure and context. The former method is simple but limits the extraction to a predefined list of names, whereas the latter is able to recognize names even if they are not on the list [Huang 2015].

Stanford NER is a very popular NER system. It applies a machine learning-based method and is distributed with Conditional Random Fields (CRF) models to detect named entities in English newswire text. Finkel *et al.* [Finkel 2005] used simulated annealing in place of Viterbi coding in sequence models to enhance an existing CRF-based system with long-distance dependency models. The authors outperform the NER on long documents but do not perform well on microblogs as they achieve 89% for newswire but only 49% for tweets in the development of Ritter dataset [Bontcheva 2013].

Agarwal *et al.* [Agarwal 2012] introduced an approach that combines the Stanford NER tool and a concept-based vocabulary to extract location information from tweets. To filter out noisy terms from extracted location phrases, they used a Naive Bayes classifier with the following features: the Part Of Speech (POS) tags of the word itself, three words before this word, and three words after this word. To disambiguate place names, the authors extracted longitude and latitude information from a combination of an inverted index search on World Gazetteer data, and a search using Google Maps API.

Kazama *et al.* [Kazama 2008] introduced a method that uses large-scale clustering of dependency relations between verbs and multi-word nouns to build a gazetteer for detecting named entities in Japanese texts. They argue that, since the dependency relations capture the semantics on multi-words, their cluster dictionary is a good gazetteer for NER. In addition, they also combined the cluster gazetteers with a gazetteer extracted from Wikipedia to improve accuracy. Krishnan *et al.* presented a two-stage method to deal with non-local dependencies in NER [Krishnan 2006] for long documents using CRF. Their first CRF-based NER system used local features to make

predictions while the second CRF was trained using both local information and features extracted from the output of the first CRF. This helped them build a rich set of features to model non-local dependencies and conduct the inference efficiently since the inference time is merely one of two sequential CRF. As a result, their method yielded a 12.6% relative error reduction on the F-Measure, which is higher than the state of the art Stanford NER at 9.3%. Li et al. extracted locations mentioned by Singapore users in their tweets. They built a location gazetteer by exploiting the crowdsourcing knowledge embedded in the tweets associated with Foursquare check-ins. This inventory includes formal names and abbreviations commonly used to mention users' points of interest. When applying a linear-chain CRF model that accounts for lexical, grammatical, and geographical features derived from the tweets and the gazetteer, the F-measure for location recognition is about 8% higher than the Stanford NER [Li 2014]. Ji et al. [Ji 2016] reapplied the method from [Li 2014] to address location recognition, which was a subtask in their work. This task is a sequential token tagging task applied according to the BILOU scheme in [Ratinov 2009]. As a result, they improved the F-measure by about 0.05% compared to [Li 2014].

Also applying CRF, but in a more complex way, Liu *et al.* [Liu 2011] combined a K-Nearest Neighbors (KNN) classifier with a linear CRF model under a semi-supervised learning framework to find named entities in tweets. They first used a KNN classifier to conduct word level classification, which exploits the similar, recently labeled tweets. These re-labeled results, together with other conventional features, were then fed into the CRF model to capture fine-grained information from a single tweet and from 30 gazetteers which cover common names, countries, locations and temporal expressions. By combining global evidence from KNN and the gazetteer with local contextual information, the researchers' approach was successful in dealing with the unavailability of training data.

Li *et al.* [Li 2012], in a different approach compared to previous studies, collectively identified named entities from a batch of tweets using an unsupervised method. Rather than relying on local linguistics features, they aggregated information garnered from the World Wide Web to construct local and global contexts for tweets. Firstly, they exploited the global context retrieved from Wikipedia and the Web N-Gram collection to segment microblogs. Each tweet segment was then considered as a candidate named entity. Next, they built a random model to exploit the gregarious property

in the local context collected from the Twitter stream. The named entity is the highest ranked segment. In another study, Ozdikis *et al.* [Ozdikis 2016] determined the location of an event based on GPS geotags, tweet content and user profiles. They first separated these features and then combined them into a single solution using combination rules from Dempster–Shafer theory. On average, the city-level error distance was 107,9 km.

Recently, some approaches have been successful in detecting locations in tweets. Bontcheva *et al.* customized their NER systems for newswire, adapting the Gate NLP framework [Bontcheva 2013] for tweets. They also adapted and retrained a Stanford tagger [Toutanova 2003] for tweet collections. They used gazetteers of personal names, cities and a list of unambiguous company and website names frequently mentioned in the training data. As a result, they increased the F-measure from 60% to 80%, but mainly with respect to Person, Organization and Time, rather than Location.

Ritter *et al.* [Ritter 2011] addressed the problem of NER for microblogs by using chunking to rebuild the NLP pipeline, beginning with POS tagging. They applied a probabilistic model, LabelledLDA to exploit an open-domain database (Freebase) as a source of distant supervision. Their experiments showed that their approach outperformed the existing NER tools on tweets for the location entity type with a 77% F-measure in finding location names in their own dataset, namely the Ritter dataset. While the Gate NLP framework achieves high recall, Stanford NER and Ritter are more efficient in terms of precision [Bontcheva 2013]. In this work, we introduce a method that combines these tools to target either recall-oriented or precision-oriented applications. We also propose to filter the extracted locations using DBpedia to increase the precision of the tools.

3.2.2 Prediction of locations

Location prediction in tweets has been little studied. Recent work addressing this problem has followed two directions: content-based and non-content-based. The first approach analyses the textual content while the second uses the information provided in user profiles, geo-tagged tweets and social network information.

Wing *et al.* analyzed raw text to predict documents geo-location in terms of latitude and longitude coordinates [Wing 2011]. They applied several supervised methods and used a geodesic grid as a discrete representation of

the Earth's surface. Geo-tagged documents were presented in a corresponding cell. New documents were geo-located to the most similar cell based on Kullaback-Leibler divergence [Zhai 2001]. Their prediction is impressive for Wikipedia articles with a median error of just 11.8 kilometers; however, they do not perform well on tweets as the median error is 479 km.

Lee *et al.* [Lee 2010] developed a geo-social event detection system by monitoring posts from Twitter users. They predicted the occurrence of events based on geographical regularities, which includes the three following indicators: the number of tweets, crowds and moving users, inferred from the usual behavior patterns of crowds with geo-tag tweets. They compared these regularities with the estimated regularities to show the unusual events organized in the monitored geographical area. The sudden increase of tweets in a region and the increase of Twitter users in a short period of time are two important clues in their approach.

More recently, Ikawa *et al.* predicted the location where a message is generated by using its textual content. They derived associations between each location and its relevant keywords from past messages during the training and inferred where a new message comes from by comparing the similarities between the keywords in the training with the ones in the new message. They trained their model using two methods: for each user and for every user. They concluded that the training method for each user is more efficient in terms of recall and precision than the training method for every user [Ikawa 2012]. Bo *et al.* predicted the geo-location of a message or a user based on the aggregation of tweets from that user. They identified Location Indicative Words (LIW) that implicitly or explicitly encode an association with a particular location. They first detected LIW via feature selection and then established whether the reduced feature set boosts geo-location accuracy. Their results decreased the mean and median of the prediction error distance by 45km and 209 km respectively [Bo 2012] .

In [Backstrom 2010], the authors proposed an approach to predict the location of a user based on the user's friends. They modeled the relation between geographical distance and friendship and calculated the probability of a user being located at a specific place. The place with the maximum probability is estimated as the user location. As a result, they were able to estimate the location of 69% of users with 16 or more located friends to within 25 miles. Mahmud *et al.* inferred the home location of Twitter users by extracting features from a user's tweets content and their tweeting behavior. They

combined statistical and heuristic classifiers to predict locations and used a geography gazetteer to recognize location named entities [Mahmud 2014]. By using a user's profile and multiple map APIs, Kulshrestha *et al.* addressed the problem of finding a user's location at the country level. They compared the location information obtained from multiple map APIs to reduce inference errors. Their approach was able to infer the location of 24% of users with 95% accuracy; however, it is not effective in cases where users input incorrect information in the location field or leave it empty. Following this line of thought, Chandra *et al.* [Chandra 2011] proposed a method of estimating the location of Twitter users, based purely on the content of the users' tweets along with the content of related-reply tweets. They assumed that terms included in a user's tweets can be assigned as terms related to his or her town/city. Thus, they made use of a probabilistic framework that considers a distribution of terms found in the tweets from a specific dialogue, including reply tweets, initiated by the user. They also estimated the top K probable towns/cities for a given user and achieved the highest accuracy at 59% with K=5, and an error distance of 300 miles.

Related studies focus on predicting the location of the users or where the text was generated, but not on predicting the occurrence of locations in the tweet themselves. On the contrary, our study examines this prediction. The goal is to extract the smallest number of tweets that are most likely to contain locations. If we are able to correctly predict the tweets in which a location is mentioned, we hypothesize that the precision and efficiency of NER tools can be improved since a very small proportion of tweets contain a location in their content.

In this work, we rely on existing tools for location extraction and propose a method which predicts whether a tweet contains a location or not.

3.3 Combining location extraction methods

Named entities recognition (NER) in formal texts, like news and documents, has attracted many researchers. Location recognition is a part of the NER process in which locations are names of politically or geographically defined places such as regions, countries, cities, provinces, rivers and mountains. Locations also contain man-made infrastructures such as airports, seaports, highways, streets and factories.

For Twitter, some approaches have been proposed and have been successful for location identification such as the Ritter tool [Ritter 2011], the Gate NLP framework (Gate) [Bontcheva 2013] and the Stanford NER [Finkel 2005].

In this section, we focus on research question 1 ("How much can we improve precision and recall by combining existing tools?"). We propose an approach to identify location names in tweets by combining these three tools and filtering out locations after extraction by DBPedia [8].

We first *obtained* the locations identified by each of the three tools. Then, we *merged* the extracted location names and finally we evaluated the accuracy and precision.

Table 3.1: Some features of the Ritter and MSM2013 datasets used to evaluate our location extraction and prediction models.

	Ritter's dataset	**MSM2013 dataset**
# of tweets	2,394	2,815
# of tweets containing a location (TCL)	213 (8.8%)	496 (17.6%)
# of tweets without location (TNL)	2,181	2,319

To *filter* the locations, we checked their existence on a DBpedia endpoint framework which takes into account the official name, abbreviation, postcode and nickname for the location and rejects location candidates not listed on DBpedia.

The results for recall, precision and F-measure are shown in Table 3.2. We used the Student's t-test, with the entire dataset processed by the Ritter location extraction tool as the baseline (first row of Table 3.2).

We conducted experiments and evaluated our method for two public collections: Ritter [Ritter 2011] and MSM2013 [Cano Basave 2013], both are reference collections in the domain. The first collection was initially used by Ritter *et al.* [Ritter 2011] while the second was the training dataset from Making Sense of Microposts 2013 (MSM2013). These two datasets are provided along with manual annotations on locations. Table 3.1 shows the number of tweets along with their distribution (according to whether they mention a location or not) in the two datasets.

[8] `http://dbpedia.org/snorql/`

Table 3.2: Effectiveness when combining extraction models: Ritter, Gate, Stanford, and filtering with DBPedia. Recall - R(%), Precision - P(%), F-measure - F(%) for the Ritter and MSM2013 datasets. A statistically significant value is indicated by a star (*) when compared to the baseline using Student's t-test (p-value < 0.05).

	Ritter dataset			**MSM2013 dataset**		
	R(%)	**P(%)**	**F(%)**	**R(%)**	**P(%)**	**F(%)**
Ritter (baseline)	71	82	77	61	80	69
Stanford	51	63	56	65	78	70
Gate	59	55	57	69	69	69
Ritter+DBp	45	**97***	62	48	**88***	62
Ritter+Gate	**82***	56	66	**78***	64	71
Ritter+Stanford	80*	64	72	**78***	72	**75***
Ritter+Gate+DBp	78*	71	74	74*	77	**75***
Ritter +Stanford+DBp	77*	79	**78**	72*	79	**75***

As presented in Table 3.2, the combination of the Ritter location extraction tool and the Stanford NER filtered by DBpedia gives the best F-measure, although it is only one percent higher than the baseline for the Ritter dataset. The F-measure for the MSM2013 dataset has considerably increased with this combination (from 69% to 75%). The locations recognized by Ritter along with the locations identified by the Gate filtered by DBpedia (third row in Table 3.2) gives the second highest F-measure for the Ritter dataset at 74% while the locations found by Ritter and Stanford (fourth row in Table 3.2) reach the F-measure of 72%. These two combinations give the best results; an F-measure of 75% for the SM2013 dataset.

Recall-Precision trade-off is well known. However, we significantly improve recall in some cases and precision in others, which can be useful when either recall-oriented or precision-oriented applications are targeted.

Recall-oriented applications. The combination of Ritter and Gate gives the best recall, significantly increasing from 71% to 82% for the Ritter dataset while Ritter plus Stanford gives the second highest recall at 80% for the same dataset. The trend is similar for the MSM2013 dataset: the combination of Ritter with either Stanford or Gate gives the best recall at 78%; 27.9% (in relative percentage) higher than the baseline. As expected, precision is decreased

in both combinations. Ritter combined with Stanford achieves a precision of 64% and 72% for the Ritter and MSM2013 datasets respectively while Ritter combined with Gate achieves 56% precision for Ritter dataset and 64% precision for the MSM2013 dataset. Overall, the F-measure remains steadily, even increasing in the case of MSM2013 dataset. These combinations can be applied in recall-oriented applications such as Festival Recommender Systems, Entertainment Recommender Systems and Travel Recommender Systems.

Precision-orientated applications. Following our intuitive first idea to improve precision, we filtered out extracted locations by using DBpedia. When locations identified by Ritter are filtered by DBpedia, as expected, precision is greatly increased from 82% to 97% and from 80% to 88% for the Ritter and MSM2013 datasets respectively (see the fourth row of Table 3.2). However this improvement takes place to the detriment of recall: only 45% for the Ritter dataset and 48% for the MSM2013 dataset. This combination can be applied to precision-oriented applications in which the precision is meaningful and essential, such as disaster support systems and rescue systems.

With regard to our first research question, we can conclude that combining Ritter and Gate is most appropriate in recall-oriented applications since this combination significantly increases the recall from 71% to 82% for the Ritter and from 61% to 78% for the MSM2013 datasets. This may arise because these methods use different clues to extract locations in tweets. On the other hand, when precision is urgently required for precision-oriented applications, the most effective method is filtering out locations recognized by Ritter: precision increases by 18.29% (in relative percentage) for the Ritter dataset (see the fourth row in Table 3.2) and 10% (in relative percentage) for the MSM2013 dataset.

As a good recall-precision trade-off, associating locations extracted by Ritter and Stanford filtered out by DBpedia is successful since it increases the F-measure from 77% to 78% and from 80% to 88% for the Ritter and MSM2013 datasets respectively.

3.4 Location prediction

In this section, we focus on the second research question: "Is it possible to predict whether a tweet contains a location or not?". We also examine if this prediction is useful for location extraction accuracy. We first conducted a preliminary analysis to study the usefulness of location occurrence pre-

diction by only applying prediction to tweets containing location and show that this is conclusive. We then proposed a model to predict the location occurrence in tweets and show the effectiveness of this model.

3.4.1 Location extraction on tweets containing locations

As a preliminary study, we conducted the same experiments as in Section 3.3 only for tweets containing locations. The objective was to see if it is more effective to extract locations from these tweets than from entire dataset. The results in terms of recall, precision and F-measure are reported in Table 3.3. Overall, recall is unchanged but precision is greatly improved compared to the location extraction from the entire dataset. This leads to an increase in the F-measure as well. As a baseline, Ritter tool leads to a sizeable increase in the F-measure, from 77% to 83% and from 69% to 74% for the Ritter and MSM2013 datasets respectively.

Table 3.3: Effectiveness of combining location extraction tools on Recall - R(%), Precision - P(%), F-measure - F(%) in tweets containing locations from the Ritter and MSM2013 datasets. A statistically significant value is indicated by a star (*) when compared to the baseline.

	Ritter dataset			**MSM2013 dataset**		
	R(%)	**P(%)**	**F(%)**	**R(%)**	**P(%)**	**F(%)**
Ritter (baseline)	71	98	83	61	93	74
Stanford	51	84	63	65	91	76
Gate	59	88	70	69	90	78
Ritter+DBp	45	**99**	62	48	**96***	64
Ritter+Gate	**82***	87	84	**78***	87	83*
Ritter+Stanford	80*	87	84	78*	89	**83***
Ritter+Gate+DBp	78*	95	85	74*	91	82*
Ritter +Stanford+DBp	77*	95	**85**	72*	93	81*

The various combinations share the same general trend. When using DBPedia to filter named entities extracted by Ritter (the fourth row of Table 3.3), we achieved the highest precision, 99% for the Ritter dataset and 96% for the MSM2013 dataset. The F-measure is highest (85%) when combining Ritter with Stanford filtered by DBpedia for the Ritter dataset; the highest

F-measure for the MSM2013 dataset (83%) is also reached when combining Ritter with either Stanford or Gate.

From these results, it is obvious that using location extraction tools only on the tweets that contain locations, considerably improves precision, leading to an increase in the F-measure. In addition, as in several papers and available research datasets [Sloan 2015, Ritter 2011, Cano Basave 2013] a huge amount of tweets are posted daily but very small proportion of tweet contains locations . Therefore if we could exactly predict tweets that contain locations, unnecessary tweets could be filtered out. This would save time and resources, and hopefully improve precision, which is essential and meaningful in precision-oriented applications such as disaster support systems and rescue systems. This is why we have developed a model to predict whether a tweet contains a location or not; this model is presented in detail in the next sub-section.

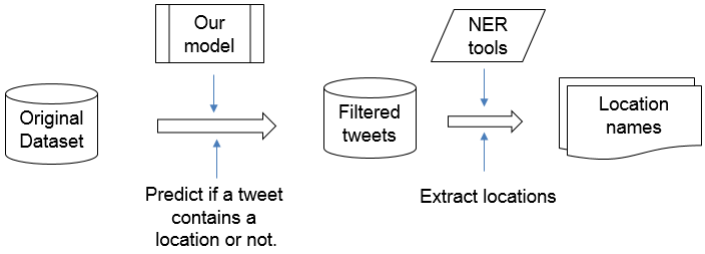

Figure 3.1: The location extraction process.

Figure 3.1 describes our work of the rest of this chapter. From the original dataset, we filter the tweets that contain location by our model. Next, we implement the NER tools on those predicted tweets to see if our model improves the efficiency of the location extraction.

3.4.2 Predictive model for locations in tweets

In this section, we propose a model to predict whether a tweet contains a location or not. Figure 3.2 shows some examples of tweet that contain a location. The objective of our model (for this example) is to give the result: the first two tweets (the upper of the figure) contain a location while the third tweet does not.

Figure 3.2: Examples of tweets containing a location in the content.

3.4.2.1 Tweet features

Predicting whether a tweet contains a location name or not is not an easy task since tweets are usually written in a pseudo-natural language and may not correspond to grammatically correct sentences.

We manually analyzed some tweets from the festival tweet collection used in CLEF 2015 [Goeuriot 2016b] to detect clues that could be used to predict whether a location occurs in a tweet or not. We also relied on the related work regarding the prepositions that introduce a location.

Table 3.4 presents the features we propose along with some examples that support our choices. They are just examples, and some counter examples may exist, but we will revisit this aspect in the evaluation section.

Geography gazetteer. This feature checks if a tweet contains at least one word appearing in a geography gazetteer. We chose the Gate NLP framework's gazetteer which includes a list of countries, cities, regions and states with their abbreviations; it is available online for open access and performs well in microblogs [Bontcheva 2013]. For example, the tweet "Today I got a promotion at work , and tomorrow I 'm going home to Wisconsin for a few days." contains the 'Wisconsin' term included in Gate geography gazetteer.

As there is usually a preposition before a place name, we propose two features based on prepositions:

Prep. We define a binary feature to capture the presence of prepositions of place and movement[9] (*at, in, on, from, to, toward, towards*). A preposition often appears in the content about a location. For example, the tweet *"Feeling really good after great week in our London office."* contains the preposition 'in' when telling a story about an office in London.

[9]http://grammar.ccc.commnet.edu/grammar/prepositions.htm

Prep+PP. This feature checks if a tweet includes a preposition just before a proper noun (PP). We used Ritter POS to part of speech the tweet and check if the tweet contains a preposition right before a proper noun. For example, the tweet *"- RT @RMBWilliams : Here in Gainesville!"* contain the preposition 'in' right before the location named 'Gainesville'.

Place+PP. This feature checks the presence of a specific word which often appear just after or just before a proper noun of place. We use the following words: *town, city, state, region, department and country*. The tweet: *"The football fever : Ohio head coach Frank Solich says Ohio state knows they have a special team and season underway."* specify the 'state' when mentioning 'Ohio'.

Time. We assume that a text about a specific place often includes a time expression. The time expressions checked included the words: *today, tomorrow, weekend, tonight*, the days of a week, and months. For example, *"Come check out Costa Lounge tonight."*

DefArt+PP. The definite article "the" is used before country names such as *the Czech Republic, the United Arab Emirates* and *the United States* or before rivers, oceans, seas and mountain names. Thus, we define a binary feature that checks the presence of the following string type: "the"+PP. For example *"Beautiful day! Nice to get away from just before proper noun the Florida heat"*

Htag. Hashtag is one of the most ubiquitous aspects of Twitter. It is used to categorize tweets into topics. For events such as festivals or conferences, hashtag which specify the location of the events is widely used. This binary feature checks whether the tweet contains a hashtag or not.

PP, Adj, Verb. We count the numbers of proper nouns, adjectives and verbs in a tweet recognized by the Ritter POS. We use these features in a predictive model that is derived using a training/testing framework.

The features "PP", "Adj", "Verb" are integers while the others are Yes/No values.

We used the Ritter tool [Ritter 2011], which is a state-of-the-art POS in microblogs, to tag POS, and Python programing language to extract the features. The feature extraction processes took a few hours for each data collection on a computer with a i7-core processor and 16GB of RAM.

As reported later in Section 3.4.3.1, some features of the predictive model are more important than others and results may depend on optimized criteria (Section 3.4.3.2). Overall, we show that location extraction is more effective when applied to predicted tweets (Section 3.4.4).

Additionally, we evaluated our model using the Doc2vec model to infer vector features to represent tweets; however these features do not give good results for the prediction. The feature extraction as well as the results are detailed in sub-section 3.4.5.

3.4.2.2 Learning models and evaluation framework

We used the same collections as in the Section 3.3 to evaluate our model: the Ritter dataset [Ritter 2011] and MSM2013 dataset [Cano Basave 2013]. These two datasets are previously described in Table 3.1.

We tried different machine learning classifiers: the Naive Bayes (NB), Support Vector Machine (SVM) and Random Forest (RF) using 10-fold cross validation. For SVM, we used an algorithm which implements John Platt's sequential minimal optimization algorithm for training a support vector classifier. This algorithm is called 'SMO' (Sequential Minimal Optimization) in Weka. In the rest of this chapter we used the term 'SMO' when mentioning the algorithm implementing the SVM.

When training the model, it is possible to optimize various criteria. We consider that either accuracy or true positive should be optimized.

Machine learning algorithms also have some parameters. The so called "manual threshold" is a parameter for NB and RF classifiers and affects the prediction results. It corresponds to the statistically significant point which affects the output probability of the classifier. In our experiments, we varied the threshold in $(0.05, 0.20, 0.50, 0.75)$. On the other hand, SMO has an internal parameter called epsilon. This parameter is for the round-off error. We varied epsilon in $(0.05, 0.20, 0.50, 0.75)$.

Baseline. We converted the content of tweets into word vectors classified by SMO (default setting) and considered it as the baseline.

All the classification processes were implemented on Weka graphical user interface [Hall 2009]. Some classifiers took longer than others, but all of them took a few minutes on a computer with a i7-core processor and 16GB of RAM.

Table 3.4: Features used to predict location occurrence in a tweet and examples of corresponding tweets.

Name	Description	Examples
1. Geography gazetteer	Contains a word appearing in Gate geography gazetteer	- Today I got a promotion at work , and tomorrow I 'm going home to **Wisconsin** for a few days.
2. Prep+PP	Contains a preposition just before proper nouns	- RT @RMBWilliams : Here **in Gainesville**! - Greek Festival **at St Johns** before ASPEN!
3. PP	Number of proper noun	going **to** alderwood :). # PP: 1
4.Prep	Contains one of the 7 prepositions of place and movement [10]: *at, in, on, from, to, toward, towards*	- Feeling really good after great week **in** our London offices - @Strigy got mine **in** bbt aintree today
5. Place+PP	Contains a word specifying place (*town, city, state, region, country*) just before or after proper noun	- The football fever : Ohio head coach Frank Solich says Ohio **state** knows they have a special team and season underway
6. Time	Contains a time expression (*today, tomorrow, weekend, tonight...*)	- Headed to da gump **today** alabama here I come - Come check out Costa Lounge **tonight**!
7. DefArt+PP	Contains a definite article just before proper noun	- Beautiful day! Nice to get away from **the Florida** heat
8. Htag	Contains a hashtag	**#Brazil**
9. Adj	Number of adjectives	- **Bad** time for leicester fans. # Adj:1
10. Verb	Number of verbs	- Willingham **took** a turn. # Verb: 2

3.4.3 Experiments and results

3.4.3.1 Most important features for training

Our predictive model used 10 features, which were not all equally useful. We evaluated the importance of attributes by measuring the information gained with respect to the class. By setting the Infogain attribute evaluator and the Ranker search method in Weka, we obtained the most important features, including the weight, as follows:

- **Ritter's dataset**: Geography gazetteer (0.145), Prep+PP (0.108), PP (0.0776), Pre+Place (0.02), Place+PP (0.002)
- **MSM2013 dataset**: Geography gazetteer (0.190), Prep+PP (0.093), Pre+Place (0.028), PP (0.023), DefArt+PP (0.005)

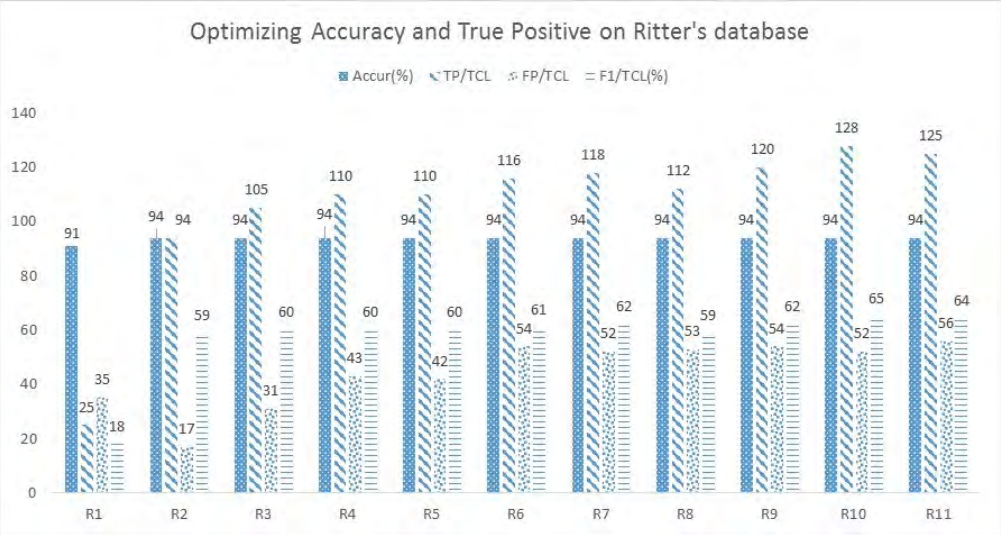

Figure 3.3: Accuracy (%), true positive (TP), false positive (FP), and F-measure (%) for TCL (tweets containing a location) when optimizing *accuracy* and *TP* obtained by a RandomForest threshold of 0.5 for the Ritter dataset with different numbers of features representing tweets.

To evaluate how the results are improved after adding new features, we systematically combined features listed in Table 3.4 and ran additional experiments. For each run, we added one more feature (ordered as in Table 3.4). We started our experiments by running R1 including the first feature (Geography gazetteer) only. R2 consists of the first two features (Geography

gazetteer and Prep+PP) while R3 contains the first three features (Geography gazetteer, Prep+PP and PP). The same rule was applied until all 10 features are included in the experiment which is R10. R11 was formed after removing features that decreased the results for runs from R1 to R10. R11 will be detailed later in this section.

In Figure 3.3 we present the results for accuracy (%), number of TP, FP and F-measure (%) when optimizing accuracy and true positive for the Ritter dataset (threshold 0.5) for all runs from R1 to R11 as described above. Logically, the best results are obtained at R10 which combines 10 features together.

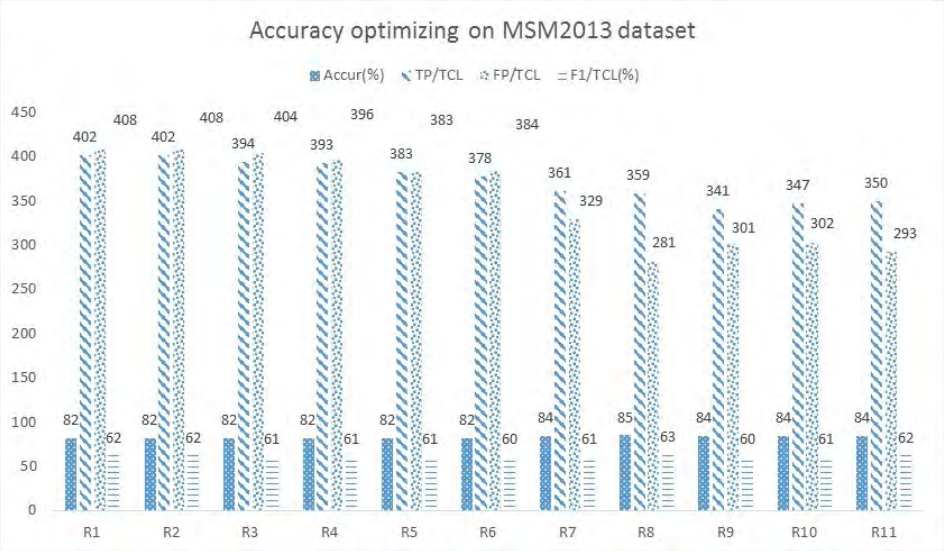

Figure 3.4: Accuracy (%), true positive (TP), false positive (FP), and F-measure (%) for TCL when optimizing *accuracy* obtained by a RandomForest threshold of 0.75 for the MSM2013 dataset with different numbers of features representing tweets.

When comparing the results for each run from R1 to R10 in Figure 3.3, we can see that the F-measure tends to increase as we add new features. There is one exception: the F-measure for the R8 run decreases compared to the R7 run. Thus, we formed the R11 run including all features except the eighth feature - "Hashtag" (see the ordered list in Table 3.4). However, the result for R11 is not higher than that for R10. We may suppose that the "Hashtag" might decrease the result for R8, but it may improve the result if combined with the ninth and tenth features, we therefore kept ten features.

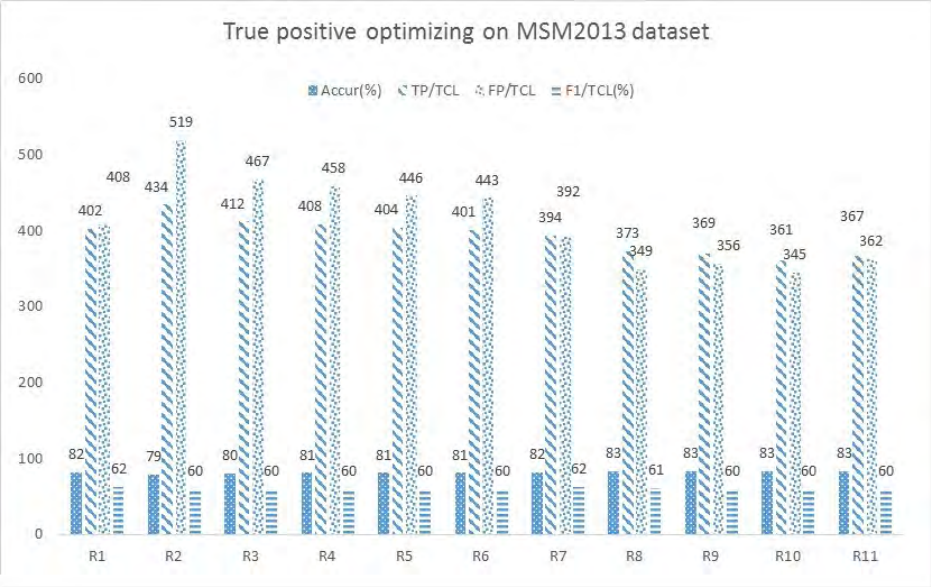

Figure 3.5: Accuracy (%), true positive (TP), false positive (FP), and F-measure (%) for TCL when optimizing *true positive* obtained by a Randomforest threshold of 0.2 for the MSM2013 dataset with different numbers of features representing tweets.

Figure 3.4 and Figure 3.5 present the results for accuracy (%), number of TP, FP and F-measure (%) for the R1 to R11 runs when optimizing accuracy and the true positive for the MSM2013 dataset respectively. Accuracy increases as we add new features to the model, while the F-measure remains stable. The highest result when optimizing accuracy is obtained by applying a RF threshold of 0.75 while the highest result when optimizing true positive is obtained by applying a RF threshold of 0.2. From these two figures, we can see that some features have a reverse effect: these features increase the accuracy but decrease the true positive, for example, the R8 run is better than the R7 run when optimizing accuracy but lower when optimizing the TP.

From the results above, we combined all 10 features for our later experiments.

3.4.3.2 Optimized criteria

Table 3.5 presents the results for the various machine learning models.

Table 3.5: Accuracy (Acc - %), true positive (TP), false positive (FP), and F-measure (F-%) for TCL when optimizing either *accuracy* or *true positives* - 10-fold cross validation when using Naive Bayes (NB) and Random Forest (RF) for both collections. The number next to the ML algorithm indicates the threshold used. The number next to TP is the percentage of TP obtained out of the TCL while the number next to FP is the percentage of FP obtained out of TNL.

		Ritter dataset				**MSM2013 dataset**			
Optimize	ML (parameter)	Acc (%)	TP ($\frac{TP}{TCL}$%)	FP ($\frac{FP}{TNL}$%)	F (%)	Acc (%)	TP ($\frac{TP}{TCL}$%)	FP ($\frac{FP}{TNL}$%)	F (%)
Baseline	SMO (1e^{-}12)	92	36(17)	8(0.4)	28	87	184(37)	50(2.2)	50
Acc	SMO (1e^{-}12)	94	99 (47)	21 (1.0)	60	88	226 (46)	61 (3.0)	58
Acc	NB (0.75)	90	153 (72)	177 (8.0)	56	82	357 (72)	375 (16)	58
Acc	RF (0.75)	92	152 (71)	133 (6.0)	61	**84**	**347 (70)**	**302 (13)**	**61**
Acc	NB (0.5)	92	129 (61)	96 (4.0)	59	89	236 (48)	107 (5.0)	56
Acc	RF (0.5)	**94**	**128 (60)**	**52 (2.0)**	**65**	87	263 (53)	130 (6.0)	59
TP	SMO (1e^{-}12)	94	99 (47)	21 (1.0)	59	88	22 (4.0)	61 (3.0)	58
TP	SMO (0.05)	93	133 (62)	97 (4.0)	60	86	267 (54)	160 (7.0)	50
TP	SMO(0.2)	92	137 (64)	124 (6.0)	58	82	327 (66)	350 (15)	56
TP	SMO(0.5)	86	132 (62)	253 (12)	44	76	325 (66)	509 (22)	49
TP	SMO(0.75)	91	0 (0.0)	0 (0.0)	0.0	82	0.0 (0.0)	0.0 (0.0)	0.0
TP	NB (0.05)	*86*	*190 (89)*	*319 (15)*	*53*	*74*	*450 (91)*	*685 (30)*	*55*
TP	NB (0.2)	89	160 (75)	203 (9.0)	56	80	400 (81)	472 (20)	59
TP	NB (0.5)	92	129 (61)	96 (4.0)	59	87	236 (48)	107 (5.0)	56
TP	NB (0.75)	93	119 (56)	69 (3.0)	59	87	183 (37)	40 (2.0)	51
TP	RF(0.05)	84	181 (85)	341 (16)	49	70	428 (86)	781 (34)	50
TP	RF(0.20)	91	158 (74)	164 (8.0)	59	**83**	**361 (73)**	**345 (15)**	**60**
TP	RF(0.5)	**94**	**128 (60)**	**52 (2.0)**	**65**	87	263 (53)	130 (6.0)	59
TP	RF(0.75)	94	84 (39)	20 (1.0)	53	87	188 (38)	49 (2.0)	51

The rows in the first part of the table report the results when accuracy is optimized, while the second part reports the results when the number of TP is optimized. The second column reports the results for the Ritter dataset while the third column reports the results for the MSM2013 dataset. The rows in bold highlight the best F-measure while the rows in italic highlight the highest true positive score obtained.

The best F-measure (65%) for the Ritter dataset is obtained using a RF with threshold of 0.5 (second row, Ritter column in Table 3.5). Prediction accuracy is 94% with 128 TP for 213 tweets containing a location - TCL (60%), 52 False Positive (FP) over 2.181 tweets not containing a location (TNL) (2%) when optimizing accuracy. When optimizing TP, the same configuration achieves the best results in terms of the F-measure.

This configuration is second best only when applied to the MSM2013 dataset (F-measure 59%). For this dataset, the highest F-measure when optimizing accuracy is obtained by a RF threshold of 0.75 (61% F-measure). When optimizing TP the best threshold for RF is 0.2 (F-measure 60%). Interestingly, NB with a threshold of 0.05 achieves an impressive TP for both collections although the number of FP increases. We obtain 190TP/213TCL (89%) and 319FP/2181TNL (15%) for the Ritter collection compared to 450TP/496TCL (91%) and 685FP/2319TNL (30%) for the MSM2013 collection.

Together with RF, SMO gives the highest accuracy (94%) but RF does not give the best F-measure (for TCL) or TP relative to RF and NB, which are presented in Table 3.5.

For the Ritter dataset, accuracy is from 84% to 94%; it is a little lower for the MSM2013 dataset but still higher than 80% in most cases. When calculating accuracy, both the predicted TCL and TNL are considered, although we are more interested in the correct prediction for TCL. This is why Table 3.5 also reports the results for TCL: true positive, false positive and the F-measure.

Optimizing the TP criteria rather than accuracy leads to different TP results although the F-measure does not change much apart from the RF model.

To sum up our findings, applying RF with a threshold of 0.5 gives the best F-measure at 65% for the Ritter dataset when optimizing both accuracy and TP, this configuration achieves the second best F-measure for the MSM2013 dataset, which is 2% lower than the best F-measure when optimizing accuracy (using a RF threshold of 0.75) and 1% lower than the best F-measure when optimizing TP (using a RF threshold of 0.2).

3.4.4 Location extraction for predicted tweets

We showed in sub-sections 3.4.2 and 3.4.3 that it is possible to train a model to predict if a tweet contains a location. Table 3.7 presents the results we obtained when extracting locations from those predicted tweets. We report the results both on predicted TCL and the results when the test sets are used (the details of test sets are explained below). We used three draws and report the average numbers. The number in brackets is the best result over the three draws.

Table 3.6: Description of data used for training and testing.

	Ritter's dataset	**MSM2013 dataset**
Training	142 TCL, 1420 TNL	331 TCL, 1655 TLN
Testing	71 TCL, 761 TNL	165 TCL, 664 TNL

Table 3.7: Effectiveness of the Ritter algorithm for the Ritter and MSM2013 data collections in terms of Recall, Precision, F-measure, considering the entire testing set as described in Table 3.6 and the tweets we predict as containing a location. A statistically significant value is indicated by a star (*) when compared to the baseline. The number in brackets is the best result over the three draws.

		Ritter dataset			**MSM2013 dataset**		
		R(%)	P(%)	F(%)	R(%)	P(%)	F(%)
Baseline	Entire testing set	69	85	75	60	80	69
Accuracy	TCL predicted by RF (0.5)	45(51)	**96*(98)**	61(66)	37(40)	**89*(92)**	52(55)
Accuracy	TCL predicted by RF (0.75)	53(58)	**92*(96)**	67(68)	46(48)	**86*(88)**	60(61)
TP	TCL predicted by RF (0.2)	56(63)	**91*(96)**	69(71)	49(51)	**87*(88)**	63(64)
TP	TCL predicted by RF (0.5)	45(51)	**96*(98)**	61(66)	37(40)	**89*(92)**	52(55)
TP	TCL predicted by NB (0.05)	64(69)	**88(93)**	74(75)	58(61)	**82(85)**	68(70)

Statistical significance is marked by a *. We used the Student's t-test (p-value < 0.05) considering the entire testing data set treated by the Ritter location extraction tool as the baseline (first row Table 3.7). When several draws were used, the individual significance of each draw was calculated and a * means that the difference with the baseline is statistically significant for

the three draws. The training and testing sets were built from the Ritter and MSM2013 collections following the unbalanced nature of the dataset; 2/3 of TCL are used for training and 1/3 for testing. Exact numbers are provided in Table 3.6.

As in Table 3.7, precision significantly increases for both Ritter and MSM-2013 collections from 85% to 96% and from 80% to 89% respectively; although recall decreases due to the errors caused by filtering tweets with BDpedia; specifically because abbreviations of locations are usually not mentioned in this resource.

A high precision is important in precision-oriented applications. In addition, by running NER tools only on the tweets that are predicted to contain a location, we can save time and resource compared to running these tools on the complete original collections.

3.4.5 Applying Doc2Vec to location prediction

In addition to the features of our model mentioned in Table 3.4, we tried to build other vector features using the Doc2Vec model [Le 2014]. We hypothesized that tweets about a given location will somehow relate to each other. For instance, consider the following two tweets: *"Vietnam, what a cool country to visit!!!"* and *"Valras, that was cool"* . Intuitively, these two tweets do not seem to "relate" to each other, but since they share some words in sentence structure and Vietnam is obviously a location, we can inferthat Valras is also a location.

Following that idea, we tried to represent tweets as vectors and used these vectors as features to classify tweets according to whether they contain a location or not. Tweets which have similar vectors should be in the same class. We used the document vector (Doc2Vec) model, which is "an unsupervised framework that learns continuous distributed vector representations for pieces of texts"[Le 2014] trained on different large datasets to infer vector for tweets in the two collections we used previously: Ritter and MSM2013. These vectors are used in turn as features for the classification model as presented in Section 3.4.2, with the same classifier algorithms and parameters. We chose this model because Doc2Vec is considered as an efficient tool to compute vectors representing documents and has recently been applied in various research areas. We believe that if we used a sufficiently large and appropriate training dataset which covers information on locations around

the world, we could infer appropriate vector representations that could lead to better location prediction.

We respectively trained the Doc2vec model on three different datasets as follows:

- English Wikipedia dataset [Lau 2016] which is dump dated 2015-12-01 including approximately 35 million documents.
- English tweets (Iso language code "en") of CLEF festival dataset [Goeuriot 2016b] which is collected from June to September 2015, including 9,073,707 tweets.
- English tweets of 1 percent tweets collection which was collected from September 2015 to October 2016, composed of 21,634,176 tweets.

When trained on the above three datasets, the Doc2Vec model is configured using the following hyper-parameter values: the dimensionality of feature vectors size=300, the initial learning rate alpha=0.025, the number of core machine used for this process workers=6, takes into consideration the words with total frequency at least min_count=3. The other parameters are set as default.

We respectively ran location prediction experiments using the features described below. The other settings (algorithms and parameters) are the same as in Section 3.4.2.

- **Run 1**. The features are vectors inferred from the Doc2Vec model trained on the English Wikipedia collection, mean, max, min and standard deviation of these vectors. The results for location prediction are reported in Table 3.8.
- **Run 2**. The features are vectors inferred from the Doc2Vec model trained on the English Wikipedia collection, mean, max, min and standard deviation of these vectors, plus 10 features mentioned in Table 3.4. The results of location prediction are reported in Table 3.9.
- **Run 3**. The features are vectors inferred from the Doc2Vec model trained on the CLEF festival collection, mean, max, min and standard deviation of these vectors. The results of location prediction are reported in Table 3.10.

- **Run 4**. The features are vectors inferred from the Doc2Vec model trained on the CLEF festival collection, mean, max, min and standard deviation of these vectors, plus 10 features mentioned in Table 3.4. The results of location prediction are reported in Table 3.11.

- **Run 5**. The features are vectors inferred from Doc2Vec model trained on the 1 percent tweet collection, mean, max, min and standard deviation of these vectors. The results of location prediction are reported in Table 3.12.

- **Run 6**. The features are vectors inferred from the Doc2Vec model trained on the the 1 percent tweet collection, mean, max, min and standard deviation of these vectors, plus 10 features mentioned in Table 3.4. The results of location prediction are reported in Table 3.13.

Our intuition when applying a model to represent tweets as vectors and predict location occurrence in tweets based on the similarity of vectors has not been confirmed by the results. We achieved lower F-measure in almost configurations in all runs compared to the results presented in Section 3.4.3.2 (see Table 3.5), except for the increased F-measure 62% and 67% (compared to 60% and 58%) when applying SMO (epsilon 1e-12, both accuracy and true positive optimizing) for the Ritter and MSM2013 data collection respectively (see the first and sixth rows in Table 3.9) using vectors inferred from the Doc2Vec model trained on the English Wikipedia collection combined with 10 features mentioned in Table 3.4. We also achieved the highest F-measure 67% when applying this configuration to the MSM2013 dataset using vectors inferred from the Doc2Vec model trained on the CLEF festival collection combined with the 10 features mentioned in Table 3.4 (see the first and sixth rows in Table 3.11). We suppose that the main reason for the prediction failure is the quality of the datasets used for training the Doc2Vec model. Although, the English Wikipedia collection covers information related to locations around the world, it includes documents and structured texts written in formal language. Thus, when applied to noisy, short, unstructured texts such as tweets, the inferred vectors are not exact. Besides, the 1-percent tweets collection is randomly collected from Twitter which might contain very little information related to locations while the CLEF festival collection is more about events than locations and may not be large enough.

Table 3.8: Accuracy (Acc - %), true positive (TP), false positive (FP), and the F-measure (F%) for TCL when optimizing either *accuracy* or *true positives* - 10-fold cross validation when using features: vectors inferred from the Doc2Vec model trained on the English Wikipedia collection, mean, max, min and standard deviation of these inferred vectors.

		Ritter dataset				**MSM2013 dataset**			
Optimize	ML (parameter)	Acc (%)	TP ($\frac{TP}{TCL}$%)	FP ($\frac{FP}{TNL}$%)	F (%)	Acc (%)	TP ($\frac{TP}{TCL}$%)	FP ($\frac{FP}{TNL}$%)	F (%)
Acc	SMO (1e^{-}12)	91	68	61	40	86	218	106	53
Acc	NB (0.75)	77	133	479	32	77	317	473	49
Acc	RF (0.75)	92	44	32	30	82	305	328	54
Acc	NB (0.5)	78	128	449	32	78	292	425	48
Acc	RF (0.5)	91	0	0	0	83	29	3	11
TP	SMO (1e^{-}12)	94	68	61	40	86	218	106	53
TP	SMO (0.05)	86	97	211	37	77	312	462	49
TP	SMO(0.2)	79	119	408	32	76	311	479	48
TP	SMO(0.5)	83	43	240	17	70	132	467	24
TP	SMO(0.75)	91	0	0	0	82	0	0	0
TP	NB (0.05)	74	143	548	32	74	342	583	48
TP	NB (0.2)	76	136	490	32	76	324	497	49
TP	NB (0.5)	78	128	449	32	77	292	425	48
TP	NB (0.75)	79	121	415	32	79	271	373	48
TP	RF(0.05)	43	200	1348	23	25	487	2096	31
TP	RF(0.20)	89	71	114	36	75	375	591	51
TP	RF(0.5)	91	0	0	0	83	29	3	11
TP	RF(0.75)	91	0	0	0	82	0	0	0

Table 3.9: Accuracy (Acc - %), true positive (TP), false positive (FP), and the F-measure (F -%) for TCL when optimizing either *accuracy* or *true positives* - 10-fold cross validation when using features: vectors inferred from the Doc2Vec model trained on the English Wikipedia collection, mean, max, min and standard deviation of these inferred vectors plus 10 features mentioned in Table 3.4.

		Ritter dataset				**MSM2013 dataset**			
Optimize	ML (parameter)	Acc (%)	TP ($\frac{TP}{TCL}$%)	FP ($\frac{FP}{TNL}$%)	F (%)	Acc (%)	TP ($\frac{TP}{TCL}$%)	FP ($\frac{FP}{TNL}$%)	F (%)
Acc	SMO (1e^{-}12)	**93**	**126**	**70**	**62**	**89**	**314**	**126**	**67**
Acc	NB (0.75)	79	144	429	37	79	336	418	54
Acc	RF (0.75)	93	105	58	56	87	367	239	67
Acc	NB (0.5)	80	140	401	37	80	319	374	54
Acc	RF (0.5)	91	0	0	0	85	90	3	31
TP	SMO (1e^{-}12)	**93**	**126**	**70**	**62**	**89**	**314**	**126**	**67**
TP	SMO (0.05)	92	149	127	61	84	365	323	62
TP	SMO(0.2)	91	156	165	58	84	359	309	62
TP	SMO(0.5)	89	29	87	18	79	54	137	16
TP	SMO(0.75)	91	0	0	0	82	0	0	0
TP	NB (0.05)	77	158	487	37	77	357	510	52
TP	NB (0.2)	78	146	439	37	79	338	432	53
TP	NB (0.5)	80	140	401	37	80	319	374	54
TP	NB (0.75)	81	133	372	37	81	298	348	60
TP	RF(0.05)	53	211	1127	27	29	493	1987	33
TP	RF(0.20)	92	138	119	59	79	406	493	58
TP	RF(0.5)	91	0	0	0	85	90	3	31
TP	RF(0.75)	91	0	0	0	82	0	0	0

Table 3.10: Accuracy (Acc - %), true positive (TP), false positive (FP), and the F-measure (F-%) for TCL when optimizing either *accuracy* or *true positives* - 10-fold cross validation when using features: vectors inferred from the Doc2Vec model trained on the CLEF festival collection, mean, max, min and standard deviation of these inferred vectors.

		Ritter dataset				**MSM2013 dataset**			
Optimize	ML (parameter)	Acc (%)	TP ($\frac{TP}{TCL}$%)	FP ($\frac{FP}{TNL}$%)	F (%)	Acc (%)	TP ($\frac{TP}{TCL}$%)	FP ($\frac{FP}{TNL}$%)	F (%)
Acc	SMO (1e^{-}12)	91	10	22	8.0	84	106	65	32
Acc	NB (0.75)	73	86	523	21	71	234	552	37
Acc	RF (0.75)	91	4	8	4.0	78	214	340	41
Acc	NB (0.5)	76	75	434	21	74	197	444	35
Acc	RF (0.5)	91	0	0	0	82	2	2	1
TP	SMO (1e^{-}12)	91	10	22	8.0	84	106	65	32
TP	SMO (0.05)	81	86	338	27	79	204	291	41
TP	SMO(0.2)	77	84	418	24	67	286	719	38
TP	SMO(0.5)	87	18	123	10	67	185	614	29
TP	SMO(0.75)	91	0	0	0	82	0	0	0
TP	NB (0.05)	67	102	687	20	66	283	748	37
TP	NB (0.2)	72	86	543	20	70	244	588	37
TP	NB (0.5)	76	75	434	21	74	197	444	35
TP	NB (0.75)	79	67	361	21	75	163	366	32
TP	RF(0.05)	27	185	1709	18	20	492	2248	30
TP	RF(0.20)	90	17	54	12	69	316	695	42
TP	RF(0.5)	91	0	0	0	82	2	2	1
TP	RF(0.75)	91	0	0	0	82	0	0	0

Table 3.11: Accuracy (Acc - %), true positive (TP), false positive (FP), and the F-measure (F-%) for TCL when optimizing either *accuracy* or *true positives* - 10-fold cross validation when using features: vectors inferred from the Doc2Vec model trained on the CLEF festival collection, mean, max, min and standard deviation of these inferred vectors plus 10 features mentioned in Table 3.4.

		Ritter dataset				**MSM2013 dataset**			
Optimize	ML (parameter)	Acc (%)	TP ($\frac{TP}{TCL}$%)	FP ($\frac{FP}{TNL}$%)	F (%)	Acc (%)	TP ($\frac{TP}{TCL}$%)	FP ($\frac{FP}{TNL}$%)	F (%)
Acc	SMO (1e^{-}12)	93	124	72	61	**89**	**307**	**109**	**67**
Acc	NB (0.75)	80	121	398	57	77	280	449	46
Acc	RF (0.75)	94	84	31	51	86	346	254	63
Acc	NB (0.5)	82	111	333	33	78	249	362	45
Acc	RF (0.5)	91	0	0	0	83	15	1	6
TP	SMO (1e^{-}12)	93	124	72	61	**89**	**307**	**109**	**67**
TP	SMO (0.05)	89	151	208	53	85	323	243	61
TP	SMO(0.2)	90	135	169	52	84	358	300	62
TP	SMO(0.5)	91	15	19	12	74	91	336	20
TP	SMO(0.75)	91	0	0	0	82	0	0	0
TP	NB (0.05)	75	137	524	31	72	321	620	45
TP	NB (0.2)	79	124	418	33	76	285	472	46
TP	NB (0.5)	82	111	333	34	79	249	362	45
TP	NB (0.75)	84	94	275	32	80	213	277	43
TP	RF(0.05)	45	206	1305	24	24	493	2139	32
TP	RF(0.20)	92	122	105	56	77	399	546	55
TP	RF(0.5)	91	0	0	0	83	15	1	6
TP	RF(0.75)	91	0	0	0	82	0	0	0

Table 3.12: Accuracy (Acc - %), true positive (TP), false positive (FP), and the F-measure (F-%) for TCL when optimizing either *accuracy* or *true positives* - 10-fold cross validation when using features: vectors inferred from the Doc2Vec model trained on the 1Ptweets dataset, mean, max, min and standard deviation of these inferred vectors.

		Ritter dataset				**MSM2013 dataset**			
Optimize	ML (parameter)	Acc (%)	TP ($\frac{TP}{TCL}$%)	FP ($\frac{FP}{TNL}$%)	F (%)	Acc (%)	TP ($\frac{TP}{TCL}$%)	FP ($\frac{FP}{TNL}$%)	F (%)
Acc	SMO (1e^{-}12)	91	20	33	15	83	107	79	31
Acc	NB (0.75)	68	92	649	19	67	223	659	32
Acc	RF (0.75)	91	6	9	5.3	76	197	390	36
Acc	NB (0.5)	69	85	605	19	69	196	567	31
Acc	RF (0.5)	91	0	0	0	82	0	1	0
TP	SMO (1e^{-}12)	91	20	23	15	69	196	567	31
TP	SMO (0.05)	78	97	401	27	74	238	473	39
TP	SMO(0.2)	73	95	518	23	64	283	781	36
TP	SMO(0.5)	85	29	171	14	66	161	613	25
TP	SMO(0.75)	91	0	0	0	82	0	0	0
TP	NB (0.05)	64	110	761	20	61	275	866	34
TP	NB (0.2)	67	95	663	20	66	229	629	32
TP	NB (0.5)	69	85	605	19	69	196	567	31
TP	NB (0.75)	71	77	547	18	71	163	477	29
TP	RF(0.05)	33	197	1590	20	22	490	2201	30
TP	RF(0.20)	89	26	77	17	65	303	770	39
TP	RF(0.5)	91	0	0	0	82	0	1	0
TP	RF(0.75)	91	0	0	0	82	0	0	0

Table 3.13: Accuracy (Acc - %), true positive (TP), false positive (FP), and the F-measure (F-%) for TCL when optimizing either *accuracy* or *true positives* - 10-fold cross validation when using features: vectors inferred from the Doc2Vec model trained on the 1Ptweets, mean, max, min and standard deviation of these inferred vectors plus 10 features mentioned in Table 3.4.

		Ritter dataset				**MSM2013 dataset**			
Optimize	ML (parameter)	Acc (%)	TP ($\frac{TP}{TCL}$%)	FP ($\frac{FP}{TNL}$%)	F (%)	Acc (%)	TP ($\frac{TP}{TCL}$%)	FP ($\frac{FP}{TNL}$%)	F (%)
Acc	SMO (1e^{-}12)	93	118	78	58	87	261	125	59
Acc	NB (0.75)	74	123	532	28	72	260	560	40
Acc	RF (0.75)	93	85	33	51	84	333	276	60
Acc	NB (0.5)	75	110	488	27	74	237	470	39
Acc	RF (0.5)	91	1	0	0	83	13	1	5
TP	SMO (1e^{-}12)	93	118	78	58	87	261	125	59
TP	SMO (0.05)	89	144	185	53	84	335	287	60
TP	SMO(0.2)	89	149	190	54	83	307	288	56
TP	SMO(0.5)	91	15	19	12	76	62	229	16
TP	SMO(0.75)	91	0	0	0	82	0	0	0
TP	NB (0.05)	70	132	630	27	67	314	701	42
TP	NB (0.2)	73	124	552	28	71	269	581	40
TP	NB (0.5)	75	110	488	27	74	237	470	39
TP	NB (0.75)	77	104	444	27	75	208	413	37
TP	RF(0.05)	44	207	1322	24	25	492	2101	32
TP	RF(0.20)	92	130	103	58	77	414	568	56
TP	RF(0.5)	92	1	0	0	83	13	1	5
TP	RF(0.75)	91	0	0	0	82	0	0	0

Although we have not been successful when using inferred vectors from the Doc2Vec model trained on different data collections, we believe that we could achieve better results if we had a "good" enough training dataset for the Doc2Vec model covering information related to locations around the world; but this question will have to be left for a future work.

3.5 Conclusions and discussions

Location is one of the most important dimensions when considering an event represented by tweets. A location within the content of a crisis message makes the message more valuable [Munro 2011] and Twitter users are most likely to pass on tweets with location and situational updates [Vieweg 2010].

In this chapter, we have proposed an approach for location extraction and a model to predict the location occurrence in tweets. Our approach for location extraction is first based on the combination of existing location extraction methods and significantly improves performance when we target either recall or precision-oriented applications. We show that:

(1) Combining locations recognized by the Ritter tool with locations recognized by Stanford filtered by DBpedia increases the F-measure for location extraction.

(2) Combining the locations extracted by Ritter with locations recognized by Gate considerably improves recall while using DBpedia to filter out location entities recognized by Ritter remarkably increases precision.

A vast amount of tweets are posted daily however very little proportion of them contains locations. In addition, running location extraction tools only on the tweets that contain locations significantly improves the results. We hypothesized that we could greatly increase the precision if we could predict the location occurrences in tweets. We thus introduced a method to predict whether a tweet contains a location or not. We defined several new features to represent tweets and intensively evaluated machine learning settings to predict location occurrences by varying the machine learning algorithms and parameters used. The results showed that:

(3) Random Forest and Naïve Bayes are the best machine learning solutions for this problem - they perform better than Support Vector Machine (and other algorithms we tried but did not report).

(4) Changing the criteria to optimize (accuracy or true positive) does not change the F-measure much while it has an impact on true positive and false

positive.

(5) When considering location extraction, we improved precision by focusing only on the tweets that are predicted as containing a location.

Our model gives an exact prediction for tweets that contain words from the geography gazetteer or include a preposition just before a proper noun. We also obtained a good prediction on tweets based on 'number of proper nouns' or 'words specifying places just after or before proper noun'. However, we have some cases where prediction is not appropriate. Since we only considered the abbreviations of locations included in the Gate framework's gazetteer, some tweets are not predicted accuratly if they mention abbreviations not included in the gazetteer such as: *"@2kjdream Good morning ! We are here JPN!"* where JPN is not recognized. We also have not dealt with location disambiguation. We believe that for future work and in order to solve this problem, the context given by all the words in the message should be considered [SanJuan 2012].

Besides, our attempts to improve the results using word embedding representations for tweets were not successful; we believe this might be due to the non-appropriate training collections available to date.

In this chapter and previous chapter, we applied several machine learning algorithms in our model and select the best algorithm. The selection of suitable machine learning algorithms could also be assisted by methods as the ones proposed in [Raynaut 2015, Aligon 2017].

In future work, we would also like to build relevant training datasets for the Doc2Vec to infer vector features representing tweets. We think that appropriate training datasets will overcome the limitations of our model i.e. abbreviations and disambiguation. Tweets that contain similar words about the same stories or events should be about the same locations.

We also plan to extract more features to improve the accuracy of our predictive model. Some features could be interesting to consider such as the occurrence of an even name in the content (people often mention the location along with the event they mention about), the frequently-seen locations in a user's history posts and his friend's history posts.

Finally, while this work has focused on locations, we would also like to define predictive models for other dimensions of an even such as time and entity-related information (e.g.person).

Chapter 4

Building a Knowledge Base using Microblogs

Summary

Abstract.

Social media like Twitter are widely used during an event (conference, catastrophe, cultural events ...) to collaboratively comment or advise on that event. Social network users are then notified through the people they follow or by seeking tweets related to the event. However, given the size of a tweet, the information obtained by a single post is often very partial. We developed the idea that using a set of tweets about an event could enable having a more complete view of that event by combining all information posted. In this chapter, we propose a model to represent the collection of microblogs into a knowledge base. Considering the set of tweets on festival events, we define a domain ontology and show how to populate this ontology based not only on the tweet collection but also on external data. We detail how the knowledge base could be used to provide a complete view of an event.

4.1 Introduction

Twitter is one of the leading worldwide social networks based on active users [1]. It enables users to send short 140-character messages and to follow posts from other users. Live-tweeting events such as conferences or cultural events is very popular and is basically a community that engages online while sharing topical conversations and thoughts on current experiences [Nagarajan 2010]. During an event, some Twitter users will discuss, comment, or advise on this event while their followers will be notified. Alternatively, it is possible for a Twitter user to search for tweets related to some content using the Search API [2].

However, given the 140-character size of a tweet, the information obtained by a single tweet is often very partial. It is more likely that a user rather needs to read a set of tweets to get a clear picture of an event.

For example, the three following tweets, all related to Cannes cinema festival 2015, provide different and complementary pieces of information:

[1] http://www.statista.com/topics/1164/social-networks/

[2] https://developer.twitter.com/en/docs/tweets/search/overview/standard

Ouverture de la route des Golden Globes avec Carol de Todd Haynes, Le fils de Saul et Mustang! A suivre! #Cannes2015 pic.twitter.com/YKd43HORmk
Vincent Lindon & Gaspar Noé, guests of honour at #VentanaSur Festival de Cannes Film Week from 30/11/15 to 6/12/15! pic.twitter.com/slPVKflt24
Irina Shayk, somptueuse, lors du tapis rouge du 19 mai 2015 à Cannes, pinterest.com/pin/4530340437. . .

The first tweet is about the film *Carol* directed by *Todd Haynes* to be presented at the *Cannes 2015* festival. While the second tweet provides the date of a related event in Buenos Aires (*VentanaSur*) along with two actors who were there; it is an add for the Buenos Aires festival. Finally the third tweet gives the information about a specific date at festival de Cannes 2015 where the model *Irina Shayk* showed up.

When considering these three individual tweets, it is obvious that some users will lack of context to understand them individually. However, some pieces of information from various tweets could help understanding a given tweet. For example, given the second tweet, if the user does not know the VentanaSur festival, he may mismatch festival de Cannes and VentanaSur festival. When considering both the second and the third tweets, he will find that festival of Cannes is in May and not at the end of the year, which was not obvious when considering the second tweet only. Each tweet taken individually provides partial information; but the sum of them could give a better picture of the information or of an event. If all pieces of information from the tweet set could be used to enrich a knowledge base, it would then be possible to understand better each tweet individually by contextualizing it using additional knowledge.

Moreover, some parts of the knowledge could rely on existing resources such as geographical hierarchies or domain knowledge rather than on tweets only. For example, understanding the second tweet would be easier if the user knew the entity types “Vincent Lindon” and “Gaspar Noé” belong to (V. Lindon is a player and G. Noé a director) and that “VentanaSur” is a “Festival”.

In the previous chapter, we introduce an approach of extraction locations in tweets. Together with other dimensions such as temporal information, entity-related information, location information in an even-based tweet

bring complete view to audiences.

In this chapter, we propose a model that represents a collection of microblogs on a domain ontology that allows better represent information from a set of tweets on events. By combining the (festival) tweet collection with other Internet resources, we aim at bringing a complete picture of the collection content that can make a complete view of (festival) events referenced in this collection. This model can be applied in recommender systems in the areas of tourism, transportation or marketing. While we considered a festival collection, the method we suggest could be adapted to any types of events.

Regarding the domain ontology, we use Wikipedia (or rather DBPedia[3]) as well as websites which provide official pieces of information about geography, list of festivals and related details. This information is quite stable in time. Next, the tweets related to each festival are selected using information retrieval methods. They are analyzed to recognize and extract named entities (NE) such as locations, artists, festival names, time. This extracted information can be used to populate instances of the corresponding classes in the ontology.

The knowledge base we designed then could be used in applications where the users (1) would choose a specific festival name and have a picture of that festival through the tweets (2) would choose a location and would get a list of corresponding festivals, etc. The user would be provided with official information from the tourist websites accompanied with the most fresh information from tweets such as the time when the festival is celebrated, artists perform and when they perform for each festival. Tweets related to a festival would bring the user fresh news about traffic, weather, atmosphere, opinions and feedback from attendees. Moreover, ontology inferences capabilities could bring new knowledge from existing data. For example, from the three tweets mentioned above, our ontology could help a user infering from *"Ouverture de la route des Golden Globes avec Carol de Todd Haynes #Cannes2015"* and *"Irina Shayk, somptueuse, lors du tapis rouge du 19 mai 2015 à Cannes"* that the film *Carol* was presented in May 2015 at the Cannes festival.

Currently, there are various ways to represent knowledge, but we believe that ontology (e.g. OWL-based) is an appropriate and efficient solution because of the following reasons. Firstly, it makes our system an easily ac-

[3]BDpedia structures the information from Wikipedia pages; it can be queried using SPARQL to extract structured information

cessible knowledge base. The ontology-based knowledge represents data in a common language platform which can be shared and retrieved by Resource Description Framework (RDF) query language. Moreover, it allows inferring new knowledge from existing data that makes users understand more about incomplete data in tweets. Finally, it could provide complete and updated information about festivals by combining Internet resources and the tweet collection.

This chapter aims at proposing a prototype which focuses on the domain representation and on the ontology population. We also mention some ways this knowledge base could be used in some applications.

The rest of the chapter is organized as follows: Section 4.2 presents the previous studies related to our work. Section 4.3 details the model we suggest to represent the festival domain. Section 4.4 explains how the knowledge base is populated. Finally, section 4.5 concludes this paper, discusses about applications and future work.

4.2 Related work

Due to the rising popularity of social media, many studies propose ways to extract information from this resource. Prior works related to ours are grouped into three categories: ontology-based information extraction, event detection, and location estimation in microblogs.

4.2.1 Ontology-based information extraction

In recent years, a number of papers have addressed the ontology-based information extraction. Narayan *et al.* [Narayan 2010] suggested an approach to populate an ontology with the events retrieved from Twitter. Data is parsed and mined for various features such as name, date, time, location, type and URL that are later used to populate the ontology. The authors used the existing ontology from [Hobbs 2004] to identify *time* and use Alexandria Digital Library Gazetteer (1999) to recognize Location and Name. Using these methods, they are not able to detect NE when it is not explicitly mentioned in a tweet content. Our work also aims at identifying named entities in tweets (artist, time, location, festival...) but we combined different techniques such

as using Stanford Named Entities Recognition (NER) [4], mining Twitter users' profile and inferring information from festivals that tweets related to.

To detect festivals in tweets, we matched tweet content with a list of festivals accompanied by some properties such as festival names, twitter accounts, twitter hashtags and keywords that we extracted from DBpedia or could extract from tourism websites. In turn, when mining tweets, we used terms which are often used in the tweet content such as the twitter account and hashtag for name recognition.

Kontopoulos *et al.* [Kontopoulos 2013] presented a method for sentiment analysis of tweets based on an ontology. They used a domain ontology for providing more elaborate sentiment scores related to notions included in a tweet. They first identified the topic discussed in tweets and then gave each tweet the sentiment score for each distinct aspect relevant to the topic. Another study is from [Nebhi 2011], the authors proposed an ontology-based information extraction for recognizing and semantically disambiguating named entities in tweets. They solved the problem of entity disambiguation by using syntactical context and Linked Data as Freebase. However they did not perform well in their experiments. In a study [Iwanaga 2011], the authors introduced a method for populating an existing earthquake evacuation ontology with information extracted from tweets in order to provide the most suitable evacuation center based on the earthquake victims' behaviors in the real time. They first extracted evacuation-related information from tweets such as evaluation center names, products offered at the centers and the timestamp of each tweet. Then, by using the Web, they appended more additional information such as the center address (through Google maps), the center's latitude and longitude (through Geocoding) and Japanses-to-English translation of all above information.

4.2.2 Event detection

In the area of event detection, Weng *et al.* introduced a method of detecting events by analyzing tweets. They first built signals for individual words by applying wavelet analysis on the frequency-based raw signals of the words and then filtered out trivial words based on their corresponding auto correlation signal. The remaining words are in turn clustered to form events using a technique of modularity-based graph partitioning [Weng 2011].

[4] `http://stanfordnlp.github.io/CoreNLP/`

Similarly, Zhao *et al.* [Zhao 2007] addressed the problem of even detection from social text streams by combining text-based clustering, temporal segmentation and information flow-based graph analysis. They defined an event as a piece of information flow among a group of social actors on a specific topic in a specific period of time. By evaluating the model on a collection of email and a political blog dataset, they show that their method outperformed the content-based method.

Besides, by aggregating information across multiple messages, Benson *et al.* [Benson 2011] presented a structured graphical model to detect entertainment events. Their model analyzed individual messages, clustered them according to event and induces a canonical value for each event property simultaneously. As a result, they get a set of canonical records, the values of which are consistent with aligned messages. They showed that their method is able to induce event records from tweets. Sakaki *et al.* [Sakaki 2010] proposed a model to detect earthquakes occurrence in the real time and send a warning to people before the earthquake actually happens in a specific place. They first devised a classifier of tweets based on some features such as keywords, number of words and their context. They then produced a probabilistic spatio-temporal model for the target event. They achieved good performance when 96% of earthquakes of Japan Meteorological Agency seismic intensity scale 3 or more are detected.

Using a different approach, Quack *et al.* [Quack 2008] detected local events by analyzing community photo collections using of geospatial tiles. The retrieved photos are clustered into potential entities. These resulting clusters are then analyzed and classified into objects and events which are labeled with an automatically created and verified link to Wikipedia. Lee *et al.* [Lee 2010] and Watanabe *et al.* [Watanabe 2011] analyzed the geographical distribution of geo-tagged microblogs to detect events . Lee *et al.* first established the usual status of crowd tweets in geographical region and then mapped these tweets into relevant locations on a map. They focused on the sudden increase of tweets in a place and the increasing number of Twitter users in a place in a short time. From the time-ordered geo-tagged tweets, they can trace the movement histories of crowds and grasped the overall degree of activities of local crowds [Lee 2010]. Watanabe *et al.* detected local events by first identifying groups of tweets (describing the same theme) generated within a short time in small geographic area. Then, for each group, they extracted co-occurring terms to identify the group's theme and deter-

mine if the theme is about an event or not. They did not achieved a high resul when only 25.5% of detected local events are accurate [Watanabe 2011].

4.2.3 Location extraction

The previous work related to location extraction is presented in Section 3.2. Here we just briefly list some related work.

A location is either explicitly mentioned or should be inferred from content. Named entity recognition (NER) systems have addressed the problem of retrieving location specified on formal documents [Roberts 2008, Kazama 2008, Finkel 2005, Bontcheva 2013, Etzioni 2005]; however they do not perform very well on informal texts. The possible reason may be the text parsers use some features such as word type, capitalized letters and aggregated context, which are often not exact in noisy, unstructured, short microblogs [Huang 2015].

The literature proposes some methods to improve this limitation. Liu *et al.* [Liu 2011] combined a K-Nearest Neighbors classifier with a linear Conditional Random Fields model under a semi-supervised learning framework to tackle the lack of information in microblogs, while Krishnan *et al.* [Krishnan 2006] proposed a two-stage approach to handle non-local dependencies in NER. By aggregating information garnered from the World Wide Web to build local and global contexts from tweets, Li *et al.* [Li 2012] targeted the error-prone and short nature challenges. Another location estimation approach is relying on analyzing geo-location by content analysis either with terms in gazetteer [Fink 2009], with probabilistic model [Cheng 2010], or users' networking [Chandra 2011].

In our approach presented in this chapter, we solved the problem of identifying locations in a tweet by combining three techniques: 1) using Stanford NER; 2) inferring from the location of the event that this tweet relates and 3) extracting user hometown. These three techniques complement each others in the location detection process. In the cases location is not detected by Stanford NER, we used an inference technique which considers the event that a tweet is related to. In addition, if a tweet does not contain any information that can help to identify location by the first two methods, we mined the profile of the tweet's author to extract his hometown. We considered this hometown as the event location following conclusions in [Lee 2012] where the authors found that 50% users post most of their tweets in their home residence.

In the next sections, we present the knowledge base we promote as well as the way we populate it. We also present some preliminary results based on the CLEF 2016 festival tweet set [Goeuriot 2016a].

4.3 Knowledge base model: the geographical-festival ontology

Events have several dimensions, the main ones are:

- Location information which indicates *where* the event takes place;
- Temporal information that indicates *when* the event takes place;
- Entity-related information which indicates what the even *is about*.

In the case of festival-related events, we can have a more specific representation. Figure 4.1 depicts the model of the knowledge base that represents the events associated to festivals.

The geographical-festival ontology we build includes four sub-parts: the first part (top part of the Figure 4.1 - Location) represents the locations of the events, the second part (bottom part of the same Figure - Performance) represents the performance information related to each even while the third part (Festival) concerns the festival in general. Finally, Tweet class includes the tweets related to festivals or locations. The classes and relationships between them are presented in the Figure 4.1. We make this splitting in four parts mainly to ease the description of the ontology. We describe in more details each part in the next paragraphs; the way each part of the ontology is populated is presented in Section 4.4.

- The first part (top part of the Figure 4.1 - *Location*) represents the locations of the events. The location part of the ontology is a hierarchy. Countries over the world are constituted in different ways, for example the United-States is divided in States, then in counties or county-equivalents, then in towns, while France is divided into regions, departments, then towns. Towns can in turns be divided in arrondissements. Considering the domain we are interested in, the town level looks appropriated as the deeper level.

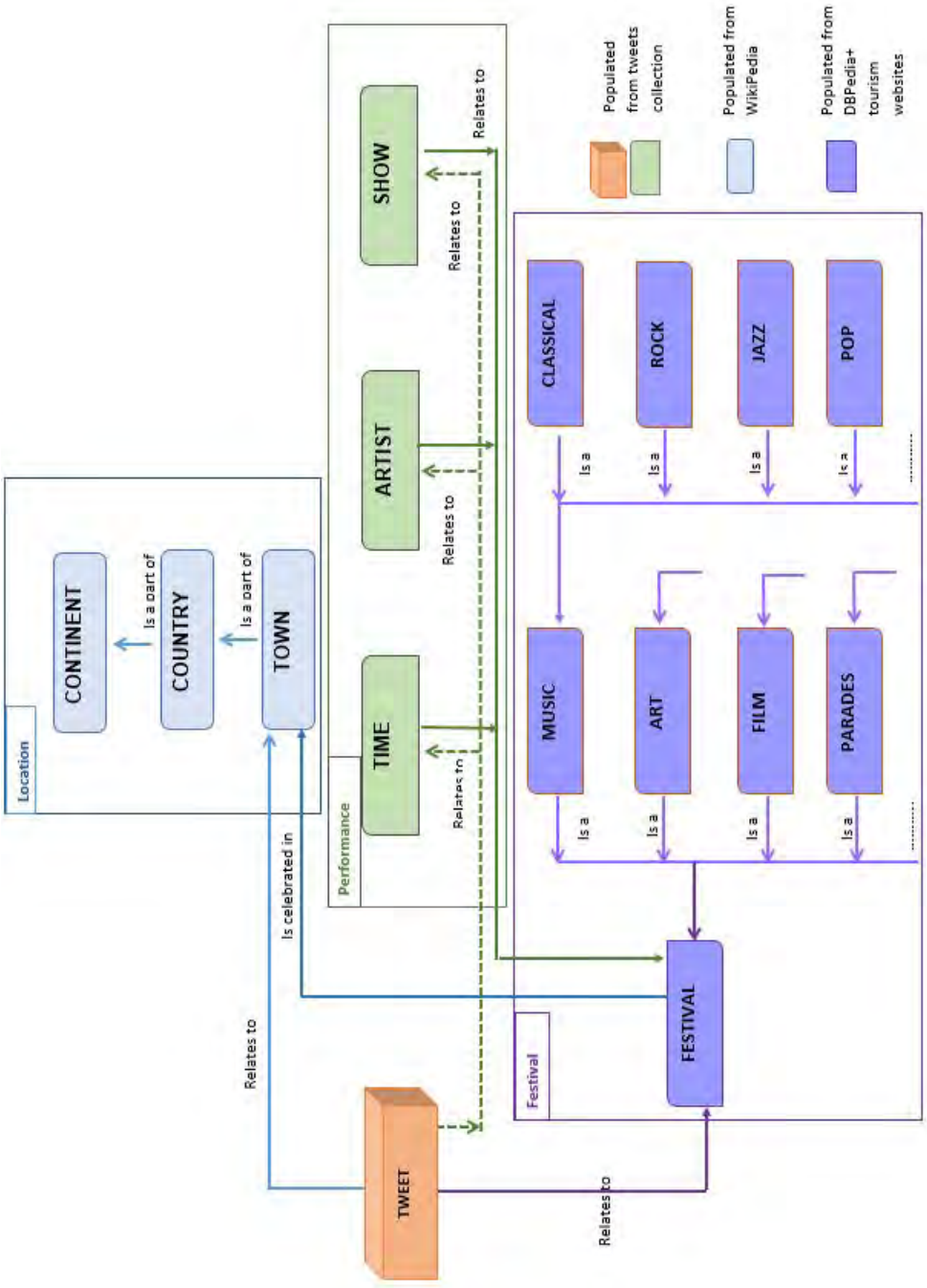

Figure 4.1: Model to represent events - the case of the Festival ontology

We thus simplify the hierarchy so that it works for any part of the world. We finally kept a three-levels hierarchy: *Town*, *Country*, *Continent*, related by Is-part-of relationships.

- The second part (*Performance*) presents performance information related to each event; it gathers information related to each festival with three classes: *Time*, *Artist*, and *Show*.

- The third part of the ontology concerns the Festivals in general. Festivals can be classified into a set of categories that can be hierarchical. For instance, the *Music* class consists of *Classical*, *Rock*, *Jazz*, *Pop*... We use a set of categories to contribute to the *Festival* part of our ontology including a number of classes such as *Music*, *Art*, *Film*, *Parades*, which are types of festivals. This hierarchy of categories is proposed by DBPedia. It might not be complete but it is appropriate to start with and it can be completed later on, considering tweets contents.

- Lastly, the *Tweet* class contains tweets which relate either to a specific festival or a location. Tweets that cannot be related to either a festival or a location are not stored and considered as useless. One tweet might be about entities from the Performance part of the ontology such as *Time*, *Artist*, *Show* ...or contain fresh information of a festival or a location such as traffic, weather, stories and feedback of attendees. We do not store this type of information in various classes but keep the tweets that can be associated to either a location, or a festival (or both) to be able to retrieve fresh information on atmosphere and twitters' comments.

4.4 Populating the domain ontology

In this section, we first provide the general principles of the knowledge base population then we detail the various steps of the ontology population.

4.4.1 Principles

The domain ontology is populated considering complementary resources. We use both a flow of tweets that match the information need *festival* and which can be seen as our main resource for fresh (and possibly subjective)

information, and external resources such as DBPedia or tourism websites that contain more stable information even if they can be frequently up-dated (specifically considering festivals to come).

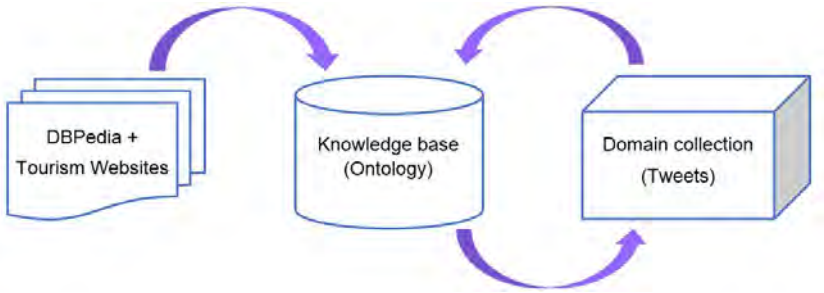

Figure 4.2: The process of populating the knowledge base. The arrows show how a resource is used. DBPedia and tourism websites are used to populate the ontology; the ontology is used to help information extraction from the tweet collection and the additional extracted information is used to populate the ontology.

Figure 4.2 depicts the overall principle of the ontology population: Web and DBPedia resources are used to first populate the skeleton of the ontology. DBPedia provides general information about existing locations, festival categories and even most of well-known festivals in the world; official festival and tourism websites provide more specific information about some festivals (for example for the Jazz festival in Marciac, the official festival website can be analyzed) and some hubs such as the Syndicats d'initiative websites can also provide some additional links to other festivals.

Then the ontology and the tweet collection are used in a process that combines pieces of information: from the ontology, we know festivals and locations that help analyzing the tweets which in turns can be used to extract new information to populate the ontology. For instance, from the ontology, it is possible to know that in *Cannes*, there is a event named *Cannes film festival*. Then, *Cannes film festival* is used to detect all tweets related to this event. These tweets, in turn, are used to extract time, artists, and shows to populate the ontology.

The ontology population using DBPedia and official websites resources can be seen as resources for background ontology population while the tweet collection is a resource for providing complementary views about the events.

To begin with, we chose Protege [5] to build the ontology that implements the knowledge base. We created the ontology structure as described in Figure 4.1 including classes such as Continent, Country, Town, Tweet, Festival.... The Location and Festival parts are to be created by data extracted from resources such as DBPedia and official websites. Then tweets related to each festival can be identified and populate the Tweet class; the relationships with *Location* and *Festival* are established in the knowledge base. In addition, information from those tweets such as *Time*, *Artist* and *Show* are extracted to populate the *Performance* part of the ontology when possible. The process will be finalized by applying inference mechanism to get new information from existing data.

In the next sections, we explain in details the populating process accompanied by preliminary results. We run the main steps of our approach on 500 tweets about Cannes and Lyon extracted from the CLEF 2016 festival collection [Goeuriot 2016a] . This collection contains 38,686,650 tweets about festivals in the world and was collected from May to October 2015.

4.4.2 Location population

The location part of the ontology is populated using the results presented by Ngo *et al.* in [Ngo 2012]. They extract the geographic data from Wikipedia which provides the list of locations for each countries. For example, for France it includes communes (overseas departments included) with a population over 20,000. The data is structured using 3 levels: “commune”, “departement”, and “region”. We used the country and town (“commune”) of their data to populate the ontology. There are $3,885$ instances of locations for France. Concretely, we only keep a few in our first prototype since Protege is limited in the number of instances it can handle without using a database. An alternative solution for geographic data could have been to use other geographic resources such as GeoName [6] or GEOnet Names Server [7], but Wikipedia provides accurate and reliable information on this topic and was enough for our Proof-of-Concept application.

[5] `http://protege.stanford.edu/` Protégé is an open-source platform for building knowledge-based ontologies.

[6] `http://www.geonames.org/`

[7] `http://geonames.nga.mil/gns/html/`

4.4.3 Festival population

The Festival part of the ontology is populated using the list of festivals provided by DBPedia [8] . Although the information from these resources changes, the update rate is not necessarily very high to keep the ontology accurate. This structured information can be extracted using SPARQL on locally stored DBpedia or through endpoint framework [9]. In our work, for the first implementation, we query information from DBPedia using the endpoint framework.

In addition, other information related to a festival could also be retrieved from DBPedia such as the festival location and official website. In turn, it would then be possible to collect the corresponding Twitter account, hashtags (from twitter page) and keywords about the festivals and consider them as additional properties to detect festivals in tweets as presented in Section 4.4.4. We keep the automation of this process for later and handle now this task manually for a few festivals for Proof-of-Concept.

4.4.4 Relationship between tweets, festivals and locations

We associate tweets related to specific festivals or locations. We compare the list of festivals and properties resulting from the *Festival* population (section 4.4.3) with the tweet contents in order to identify all tweets related to each festival. The priority is set for festival names, twitter accounts, hashtags and keywords respectively.

When considering the sub-collection of 500 tweets, we detected 137 festivals from 137 tweets including 70 festivals detected by names, 61 festivals detected by hasgtags and 6 festivals detected by Twitter account.

To recognize locations in tweets, we combined Stanford NER with other techniques such as inferring from festival location and mining the Twitter user's profile.

We used Stanford NER to recognize locations that are explicitly mentioned in tweet contents. Since numerous twitters specify locations in their text right after a hashtag (#) Stanford NER does not extract it. For this reason, we removed all hashtags in texts before using Stanford NER. In the case

[8] `http://dbpedia.org/page/Lists\_of\_festivals`:The root page provides festivals by categories of all countries in the world

[9] `http://dbpedia.org/snorql/`

locations are not specified in a tweet, we inferred the location from the festival that this tweet relate to. Finally, if a tweet does not contain any text about location or festival, we mined the Twitter user's profile to extract the home residence.

We set a priority for the three location extraction techniques: Stanford NER, inference mechanism and profile mining. In case a location in a tweet is recognized by more than one method, we chose the most suitable one (detected by the highest priority technique).

Using the 500 tweets, we detected 487 locations from 409 tweets including: 1) 313 locations identified by Stanford NER in 225 tweets, 2) 137 locations for 137 tweets based on the festivals 3) 245 locations recognized by Twitter users' profile. More sophisticated techniques to extract location from a tweet have been presented in Chapter 3 that could be also applied here.

4.4.5 Performance population

From tweets that can be related to festivals or locations (see Section 4.4.4), we use Stanford NER to extract entities such as time, artists, shows... In the 500 tweet collection, we detected 131 artists from 103 tweets, 99 time points from 99 tweets. These instances and relationships and the corresponding tweets are stored in the ontology.

4.4.6 Inferring new knowledge

In ontologies, the inference mechanism is used to infer the relationships between instances in the case they are not directly set up from previous steps. Back to an example mentioned in the introduction part, a user can extract that festival of Cannes is in May even if the time is not mentioned in the first and second tweets. It is inferred from the third tweet. In our approach, we inferred 137 locations for 137 tweets based on the festivals that these tweets related to, 30 relationships between Festival and Artist, 19 relationships between Artist and Time classes, 55 relationships between Festivals and Time.

4.5 Conclusions and discussions

In this chapter, we have introduced an approach for building a knowledge base which brings a complete view of festivals. We used Twitter festival

collection combined with other external resources.

Our model considers festivals organized in a specific location and related information such as time, artists or shows. By combining the festival tweet collection with DBPedia and official websites resources, we help building a more complete picture of festivals occurring in the data collection.

For this purpose, we defined a festival ontology. As a background task, the population of the location and festival parts is based on resources such as DBPedia and official websites. In addition, tweets related to specific festivals or locations are retrieved and analyzed to extract related data.

We believe that by employing ontology technology, we provided an easily accessible knowledge base system. Comparing to storing data in traditional databases, our approach has several pros. Firstly, data is presented in a common language platform which can be much easily retrieved by SPARQL. A RDF data model is also easier to be updated without adverse effects to the application, thus it requires less maintenance. Secondly, the inference mechanism of ontology language allows inferring new knowledge from existing data easily (in the proof-of-concept we program the inference, but ontology allows such a process). Lastly, by combining several resources such as DBPedia, websites and Twitter, our system could bring a complete and fresh knowledge about festivals by cities in the world including official information from websites and the latest stories from Twitters.

To recognize named entities in the festival collection, we combined Standford NER with inferring techniques and mining user's profiles. Applying Standford NER [Finkel 2005] on microblogs might not be optimal; some methods have been developed on the specific case of tweets such as [Ritter 2011] [Bontcheva 2013] that have been tested in Chapter 3. However, we used this method [Finkel 2005] for initial experiments.

For future work, we could extract short summaries of festivals from BDpedia or official websites to propose users a basic idea of the festivals. In addition, we could develop our knowledge base for event recommendation based on user's current location and other aspects such as his profile, interest and festivals his friends participate to.

We suppose that the knowledge base model we built have a broad range of applications in several domains such as tourism, transportation, marketing and advertisement.

In the field of tourism, using our knowledge base to build a graphical recommender system with highly informative summaries about events, famous

people, related activities aggregated from tweets would be valuable. Tourists do not have to spend time to search and process information for their need. Moreover, latest news, opinions and feedback are more likely to appear in tweets rather than in official websites.

Besides, festivals could be perfect places for companies to market their brand. They can communicate with thousands of participants and engage participants through targeted campaigns. Knowing the type of festivals, type of participants as well as the artists, shows, dates, companies could propose and implement effective advertisement campaigns for their products.

Chapter 5

Conclusions

Online social networks have been very popular over the last years. While serving its primary purpose of connecting people, social networks also play a major role in successfully connecting marketers with customers, famous people with their supporters, need-help people with willing-help people. The success of on-line social networks mainly relies on the information the messages carry as well as the spread speed in social networks. Our research aims at modeling the message diffusion, extracting and representing information and knowledge from messages in social networks.

The first contribution we made is an approach to predict the diffusion of information on social networks. We casted this problem into binary classification to predict whether a tweet is going to be retweeted and multi-class classification to predict the level of retweet. Our model uses three types of features: user-based, time-based and content-based features including some features we reused from literature (7 features) and several new features we defined (25 features). By evaluating the model on various collections corresponding to about 18 millions of tweets, we showed that our model significantly improves the F-measure on average compared to the state-of-the-art (statistically significant) for both types of prediction. In addition, we also achieved high F-measure on class-1 (tweets that are retweeted less than 100 times) and class-2 (retweeted less than 10,000 times) which contain the majority of tweets in each collection and are thus hard to predict. In state-of-the-art, proposed methods do not perform well on these two classes.

We also evaluated the importance of each feature by measuring the so-called Inforgain attribute evaluator using Rank search method. The results showed that the number of followers, followees, and the number of groups that the user belongs to, number of likes that the user has made in his timeline are the most important features for both types of prediction and consistently across the datasets. In addition, the time-based features we developed to check if a tweet is posted at noon, in the evening, at weekend or during holiday also strongly correlate with the retweetability.

To evaluate if the new features we defined are dependent from existing features, we also analyzed the correlations between features in the three datasets. Features which are important for the model are independent from each others. In addition, the results from experiments showed that the combination of the features we defined and existing features significantly improves the performance of the predictive model.

As a concrete application of the proposed predicted model, we applied this model to predict the diffusion of brand stories on Twitter. When evaluating the model on two types of collections: collections of brand stories (in terms of tweets) written by consumers and written by the company who creates the brand, we showed that the results of F-measure, the feature importance and the feature correlation are consistent with previous findings. One more finding is that in an 'advertising' tweet (from official account of the company who owns the brand/product), the age of account and famous person names mentioned in the content make this tweet get more retweets.

We believe that our model can help business managers to understand and to predict the diffusion of stories related to their brand/products on social networks. We also suggested several features that help businesses managers to form a popular tweet. Our model can also be applied to predict the propagation of information in other areas such as politics, epidemic, and disaster.

Predicting the information diffusion would be more useful if the information diffusion is predicted by regions. For example, marketers may base on the diffusion level of their brand stories by regions to offer appropriate sale and marketing campaigns for each area. The politicians may use knowledge of the election news diffusion by regions to propose relevant policies for their election campaigns. Thus extracting locations in tweets plays an important role in predicting the information diffusion by regions. In addition, since a location in within the content of tweets make the tweet more valuable and attractive [Munro 2011, Vieweg 2010], extracting locations in tweets has several applications. Our second contribution is a method to effectively extract locations in tweets. We first proposed several combinations of existing methods to extract locations in tweets namely Ritter, Gate and Stanford tools. We showed that these combinations are effective for either recall-oriented or precision-oriented applications: (1) Combining locations recognized by the Ritter tool with locations recognized by Stanford filtered by DBpedia increases the F-measure for location extraction. (2) Combining the locations extracted by Ritter with locations recognized by Gate consid-

erably improves recall while using DBpedia to filter out location entities recognized by Ritter remarkably increases precision.

As shown in previous work [Sloan 2015, Ritter 2011, Cano Basave 2013], a huge number of tweets are posted daily but very little proportion of tweets contains location. The extraction of location in all the tweets would be time and resource consuming. In addition, by experiments, we showed that running location extraction tools only on the tweets that contain locations significantly improves the results. We hypothesized that we could highly increase the precision if we could predict the location occurrence in tweets. We thus proposed a model to predict whether a tweet contains a location or not. By implementing location extraction tools on only tweets that we predicted as containing a location, we significantly improve the precision which is very important in several applications, especially geo-spatial applications and applications linked with events. We showed that location prediction is a useful pre-processing step for location extraction.

We supposed that applying this model for extracting location features in the predictive model presented in Chapter 2 would made that model more accurate. We leave this consideration for future work.

Besides strengths, some limitations remain in our model. Since we only considered the abbreviations of locations included in the Gate framework's gazetteer, we miss-predict some cases. We also have not dealt with location disambiguation. In future work, in order to solve this problem, we should consider the context given by all words in the message. While our attempts to improve the results using word embedding representations for tweets were not successful; we believe this might be due to the non-appropriate training collections available to date and thus can think of other experiments to complete this track.

Recognizing a location in messages helps to select all the tweets in a collection about a specific location and that help to get several pieces of information surrounding a place. Our third contribution is to provide a model to build a knowledge base that brings a user a complete view about festival events by locations using a tweet collection combined with other Internet resources.

We first defined a festival ontology. The web and DBpedia resources are used to first populated the skeleton of the ontology including the locations and well-known festival events. Then from the ontology, we know festivals and locations that help analyzing the tweets which in turns can be used to

extract new information such as time, artists and show to populate the ontology.

We believe that by employing ontology technology we provide an easily accessible knowledge base system. Comparing to storing data in traditional databases, our approach has several advantages. Firstly, data is presented in a common language platform which can be much easily retrieved by SPARQL. A RDF data model is also easier to updated without adverse effects to the application, thus it requires less maintenance. In addition, the inference mechanism of ontology language allows inferring new knowledge from existing data easily. Lastly, by combining several resources such as DBPedia, websites and tweets, our model could bring a complete and fresh view about festivals by locations, including official information from websites and the updated stories from twitters.

We suppose that our knowledge base model have a broad range of applications in several domains such as tourism, transportation, marketing and advertisement. For example, in the field of tourism, this knowledge base can be used to build a graphical recommender system with highly informative summaries about events, famous people, related activities aggregated from tweets would be useful. In the transportation area, a system developed on our model that would suggest a suitable route or transportation mean to avoid crowds, traffic jams or other problems could be welcomed by travels.

For future work, we first would like to collect larger datasets which include several tweets covering features that we proposed such as containing named entities in the content and the posting time is varied to predict the information diffusion. In addition, we also would like to classify a tweet into topics such as sport, music, fashion, daily weather news or technology news before predicting the diffusion of this tweet. We believe that people are more interested in some topics than in others. Finally, a trac that could be considered is the influence when a follower retweets a tweet on his friends.

For the location extraction model, we would like to build relevant training datasets for the Doc2Vec to infer vector features representing tweets. Moreover, we plan to add more features to improve the accuracy of our predictive model such as the occurrence of an even name in the content (people often mention the location along with the event they mention about), the frequently-seen locations in a user's history posts and his friend's history posts.

We also want to extract short summaries about festivals from BDpedia

or reuse techniques such the one presented in [Ermakova 2015] to propose users a basic idea of the festivals they are interested in for the model in Chapter 4. Besides, we plan to develop our knowledge base for event recommendation based on user's current location and other aspects such as his profile, interest and festivals that his friends participate to.

Bibliography

[Agarwal 2012] Puneet Agarwal, Rajgopal Vaithiyanathan, Saurabh Sharma et Gautam Shroff. *Catching the long-tail: extracting local news events from Twitter.* In Sixth international AAAI conference on weblogs and social media, 2012.

[Aligon 2017] Julien Aligon, William Raynaut, Philippe Roussille, Chantal Soulé-Dupuy et Nathalie Valles-Parlangeau. *Towards a meta-analysis-based user assistant for analysis processes.* In International conference on computer science and information technology (CSIT 2017), 2017.

[Allcott 2017] Hunt Allcott et Matthew Gentzkow. *Social media and fake news in the 2016 election.* Rapport technique, National bureau of economic research, 2017.

[Assaad 2011] Waad Assaad et Jorge Marx Gomez. *Social network in marketing (social media marketing) opportunities and risks.* International journal of managing public sector information and communication technologies, vol. 2, no. 1, page 13, 2011.

[Backstrom 2010] Lars Backstrom, Eric Sun et Cameron Marlow. *Find me if you can: improving geographical prediction with social and spatial proximity.* In Proceedings of the 19th international conference on world wide web, pages 61–70. ACM, 2010.

[Benson 2011] Edward Benson, Aria Haghighi et Regina Barzilay. *Event discovery in social media feeds.* In Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies-volume 1, pages 389–398. Association for computational linguistics, 2011.

[Bo 2012] Han Bo, Paul Cook et Timothy Baldwin. *Geolocation prediction in social media data by finding location indicative words.* In COLING 2012, 24th International conference on computational linguistics, Mumbai, India, pages 1045–1062, 2012.

[Bontcheva 2013] Kalina Bontcheva, Leon Derczynski, Adam Funk, Mark A Greenwood, Diana Maynard et Niraj Aswani. *TwitIE: An open-source information extraction pipeline for microblog text.* In Recent advances in natural language processing, RANLP , Hissar, Bulgaria., pages 83–90, 9-11/9/2013.

[Cano Basave 2013] Amparo Elizabeth Cano Basave, Andrea Varga, Matthew Rowe, Milan Stankovic et Aba-Sah Dadzie. *Making sense of microposts (# msm2013) concept extraction challenge.* 2013.

[Chandra 2011] Swarup Chandra, Latifur Khan et Fahad Bin Muhaya. *Estimating twitter user location using social interactions–a content based approach.* In Privacy, security, risk and trust (PASSAT) and 2011 IEEE third international conference on social computing (SocialCom), pages 838–843. IEEE, 2011.

[Chawla 2002] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall et W Philip Kegelmeyer. *SMOTE: synthetic minority over-sampling technique.* Journal of artificial intelligence research, vol. 16, pages 321–357, 2002.

[Cheng 2010] Zhiyuan Cheng, James Caverlee et Kyumin Lee. *You are where you tweet: a content-based approach to geo-locating twitter users.* In Proceedings of the 19th ACM international conference on information and knowledge management. ACM, 2010.

[Ermakova 2015] Liana Ermakova. *A method for short message contextualization: experiments at CLEF/INEX.* In International conference of the cross-language evaluation forum for european languages, pages 352–363. Springer, 2015.

[Etzioni 2005] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld et Alexander Yates. *Unsupervised named-entity extraction from the web: An experimental study.* Artificial intelligence, vol. 165, no. 1, pages 91–134, 2005.

[Fink 2009] Clayton Fink, Christine D Piatko, James Mayfield, Tim Finin et Justin Martineau. *Geolocating blogs from their textual content.* AAAI

Spring symposium: social semantic web: Where Web 2.0 Meets Web 3.0, 2009.

[Finkel 2005] Jenny Rose Finkel, Trond Grenager et Christopher Manning. *Incorporating non-local information into information extraction systems by gibbs sampling*. In Proceedings of the 43rd annual meeting on association for computational linguistics, pages 363–370. Association for Computational Linguistics, 2005.

[Gensler 2013] Sonja Gensler, Franziska Völckner, Yuping Liu-Thompkins et Caroline Wiertz. *Managing brands in the social media environment*. Journal of interactive marketing, vol. 27, no. 4, pages 242–256, 2013.

[Goeuriot 2016a] Lorraine Goeuriot, Josiane Mothe, Philippe Mulhem, Fionn Murtagh et Eric SanJuan. *Overview of the CLEF 2016 cultural micro-blog contextualization workshop*. In International conference of the cross-language evaluation forum for european languages, pages 371–378. Springer, 2016.

[Goeuriot 2016b] Lorraine Goeuriot, Josiane Mothe, Philippe Mulhem, Fionn Murtagh et Eric SanJuan. *Overview of the CLEF 2016 cultural micro-blog contextualization workshop*. In International conference of the cross-language evaluation forum for European languages, pages 371–378. Springer, 2016.

[Hall 2009] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann et Ian H Witten. *The WEKA data mining software: an update*. ACM SIGKDD explorations newsletter, vol. 11, no. 1, pages 10–18, 2009.

[Hennessy 2016] Ciarán Hennessy et Alan F Smeaton. *Profiling, assessing and matching personalities active in social media*. In Irish conference on artificial intelligence and cognitive Science , 20-21 Sept 2016, UCD, Dublin, 2016.

[Hltcoe 2013] J Hltcoe. *Semeval-2013 task 2: Sentiment analysis in Twitter*. Atlanta, Georgia, USA, vol. 312, 2013.

[Hoang 2016a] Thi Bich Ngoc Hoang et Josiane Mothe. *Building a knowledge base using microblogs: the case of cultural microblog contextualization*

collection). In Conference and labs of the evaluation forum (CLEF 2016), Evora, Portugal, 05/09/16-08/09/16, 2016.

[Hoang 2016b] Thi Bich Ngoc Hoang et Josiane Mothe. *Building a knowledge base using microBlogs: the case of festivals and location-based events*. In Rencontres jeunes chercheurs en recherche d'information (CORIA-RJCRI 2016), Toulouse, France, 9-11/3/2016, pages pp–295, 2016.

[Hoang 2017a] Thi Bich Ngoc Hoang, Véronique Moriceau et Josiane Mothe. *Location extraction from tweets (poster)*. In Computational linguistics and intelligent Text Processing, Budapest, Hungary, 17-23 April, 2017.

[Hoang 2017b] Thi Bich Ngoc Hoang et Josiane Mothe. *Predicting information diffusion on Twitter–analysis of predictive features*. Journal of Computational Science, DOI: 10.1016/j.jocs.2017.10.010, 2017.

[Hoang 2018a] Thi Bich Ngoc Hoang, Veronique Moriceau Moriceau et Josiane Mothe. *Can we Predict locations in tweets? a Machine learning approach (accepted)*. International journal of computational linguistics and applications, 2018.

[Hoang 2018b] Thi Bich Ngoc Hoang et Josiane Mothe. *Predicting the diffusion of brand stories in social network (regular paper)*. In Computational linguistics and intelligent text processing, Hanoi, Vietnam. Springer LNCS, 18-24 March, 2018.

[Hoang 2018c] Thi Bich Ngoc Hoang et Josiane Mothe. *Location extraction from tweets*. Information processing & management, vol. 54, no. 2, pages 129–144, 2018.

[Hoang 2018d] Thi Bich Ngoc Hoang et Josiane Mothe. *Méthode d'apprentissage pour extraire les localisations dans les MicroBlog*. In EGC - Atelier extraction et gestion parallèles distribuées des connaissances, Paris, 22-26 January, 2018.

[Hoang 2018e] Thi Bich Ngoc Hoang et Josiane Mothe. *Extraction de localisations dans les microBlogs*. In Gestion et l'Analyse de données Spatiales et Temporelles, Paris, 23 January, 2018.

[Hobbs 2004] Jerry R Hobbs, Pan et Feng. *An ontology of time for the semantic web*. ACM transactions on Asian language information processing, vol. 3, no. 1, pages 66–85, 2004.

[Hong 2011] Liangjie Hong, Ovidiu Dan et Brian D Davison. *Predicting popular messages in twitter*. In Proceedings of the 20th international conference companion on World wide web, pages 57–58. ACM, 2011.

[Hu 2016] Ying Hu, Changjun Hu, Shushen Fu, Peng Shi et Bowen Ning. *Predicting the popularity of viral topics based on time series forecasting*. Neurocomputing, vol. 210, pages 55–65, 2016.

[Huang 2015] Yan Huang, Zhi Liu et Phuc Nguyen. *Location-based event search in social texts*. In Computing, networking and communications (ICNC), 2015 international conference on, pages 668–672. IEEE, 2015.

[Ikawa 2012] Yohei Ikawa, Miki Enoki et Michiaki Tatsubori. *Location inference using microblog messages*. In Proceedings of the 21st international conference on world Wide Web, pages 687–690. ACM, 2012.

[Iwanaga 2011] Isabel Shizu Miyamae Iwanaga, The-Minh Nguyen, Takahiro Kawamura, Hiroyuki Nakagawa, Yasuyuki Tahara et Akihiko Ohsuga. *Building an earthquake evacuation ontology from twitter*. In Granular computing (GrC), 2011 IEEE international conference on, pages 306–311. IEEE, 2011.

[Ji 2016] Zongcheng Ji, Aixin Sun, Gao Cong et Jialong Han. *Joint recognition and linking of fine-grained locations from tweets*. In Proceedings of the 25th international conference on world wide web, pages 1271–1281. International world wide web conferences steering committee, 2016.

[Kazama 2008] Jun'ichi Kazama et Kentaro Torisawa. *Inducing gazetteers for named entity recognition by large-scale clustering of dependency relations*. In Proceedings annual meeting of the association of computational linguistics, pages 407–415, June 2008.

[Kontopoulos 2013] Efstratios Kontopoulos, Christos Berberidis, Theologos Dergiades et Nick Bassiliades. *Ontology-based sentiment analysis of*

twitter posts. Expert systems with applications, vol. 40, no. 10, pages 4065–4074, 2013.

[Krishnan 2006] Vijay Krishnan et Christopher D Manning. *An effective two-stage model for exploiting non-local dependencies in named entity recognition.* In Proceedings of the 21st international conference on computational linguistics and the 44th annual meeting of the association for computational linguistics, pages 1121–1128. Association for computational linguistics, 2006.

[Kummer 2012] Olena Kummer, Jacques Savoy et Rue Emile Argand. *Feature selection in sentiment analysis.* In CORIA, Bordeaux, France 2012. Citeseer, 2012.

[Kwak 2010] Haewoon Kwak, Changhyun Lee, Hosung Park et Sue Moon. *What is Twitter, a social network or a news media?* In Proceedings of the 19th international conference on World wide web, pages 591–600. ACM, 2010.

[Lau 2016] Jey Han Lau et Timothy Baldwin. *An empirical evaluation of doc2vec with practical insights into document embedding generation.* In Proceedings of the 1st Workshop on representation learning for NLP, 2016.

[Le 2014] Quoc V Le et Tomas Mikolov. *Distributed representations of sentences and documents.* In ICML, volume 14, pages 1188–1196, 2014.

[Lee 2010] Ryong Lee et Kazutoshi Sumiya. *Measuring geographical regularities of crowd behaviors for Twitter-based geo-social event detection.* In Proceedings of the 2nd ACM SIGSPATIAL international workshop on location based social networks, pages 1–10. ACM, 2010.

[Lee 2012] Bumsuk Lee, Hwang et Byung-Yeon. *A Study of the correlation between the spatial attributes on Twitter.* In Data engineering workshops (ICDEW), 2012 IEEE 28th international conference on, pages 337–340. IEEE, 2012.

[Li 2012] Chenliang Li, Jianshu Weng, Qi He, Yuxia Yao, Anwitaman Datta, Aixin Sun et Bu-Sung Lee. *Twiner: named entity recognition in targeted twitter stream.* In Proceedings of the 35th international ACM

SIGIR conference on research and development in information retrieval, pages 721–730. ACM, 2012.

[Li 2014] Chenliang Li et Aixin Sun. *Fine-grained location extraction from tweets with temporal awareness.* In Proceedings of the 37th international ACM SIGIR conference on research & development in information retrieval, pages 43–52. ACM, 2014.

[Lingad 2013] John Lingad, Sarvnaz Karimi et Jie Yin. *Location extraction from disaster-related microblogs.* In Proceedings of the 22nd international conference on world wide web, pages 1017–1020. ACM, 2013.

[Liu 2011] Xiaohua Liu, Shaodian Zhang, Furu Wei et Ming Zhou. *Recognizing named entities in tweets.* In Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies-volume 1, pages 359–367. Association for computational linguistics, 2011.

[Mahmud 2014] Jalal Mahmud, Jeffrey Nichols et Clemens Drews. *Home location identification of twitter users.* ACM transactions on intelligent systems and technology (TIST), vol. 5, no. 3, page 47, 2014.

[Mangold 2009] W Glynn Mangold et David J Faulds. *Social media: The new hybrid element of the promotion mix.* Business horizons, vol. 52, no. 4, pages 357–365, 2009.

[Mike Gotta 2006] Peter O'Kelly Mike Gotta. *Trends in social software.* Collaboration and content strategies in-depth research Overview, 2006.

[Munro 2011] Robert Munro. *Subword and spatiotemporal models for identifying actionable information in Haitian Kreyol.* In Proceedings of the fifteenth conference on computational natural language learning, pages 68–77. Association for Computational Linguistics, 2011.

[Nagarajan 2010] Meenakshi Nagarajan, Hemant Purohit et Amit P Sheth. *A Qualitative examination of topical tweet and retweet practices.* ICWSM, vol. 2, no. 010, 2010.

[Narayan 2010] Shashi Narayan, Srdjan Prodanovic, Mohammad Fazleh Elahi et Zoë Bogart. *Population and Enrichment of Event Ontology*

using Twitter. In Proceedings of the workshop on semantic personalized information management (SPIM) in conjunction with the 7th international conference on language resources and evaluation (LREC), Malta 2010, 2010.

[Nebhi 2011] Kamel Nebhi. *Ontology-based information extraction from Twitter*. In Proceedings of the Workshop on information extraction and entity analytics on social media data - COLING 2012. Mumbai (India), pages 17–22, 2011.

[Ngo 2012] Quoc-Hung Ngo, Son Doan et Werner Winiwarter. *Using Wikipedia for extracting hierarchy and building geo-ontology*. International journal of Web information systems, 2012.

[Ozdikis 2016] Ozer Ozdikis, Halit Oğuztüzün et Pinar Karagoz. *Evidential estimation of event locations in microblogs using the Dempster–Shafer theory*. Information processing & management, vol. 52, no. 6, pages 1227–1246, 2016.

[Pang 2004] Bo Pang et Lillian Lee. *A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts*. In Proceedings of the ACL, 2004.

[Quack 2008] Till Quack, Bastian Leibe et Luc Van Gool. *World-scale mining of objects and events from community photo collections*. In Proceedings of the 2008 international conference on Content-based image and video retrieval, pages 47–56. ACM, 2008.

[Ratinov 2009] Lev Ratinov et Dan Roth. *Design challenges and misconceptions in named entity recognition*. In Proceedings of the thirteenth conference on computational natural language learning, pages 147–155. Association for computational linguistics, 2009.

[Raynaut 2015] William Raynaut, Chantal Soulé-Dupuy, Nathalie Vallés-Parlangeau, Cédric Dray et Philippe Valet. *Characterization of learning instances for evolutionary meta-learning*. In European conference on machine learning and principles and practice of knowledge discovery in databases (ECML-PKDD 2015), pages pp–198, 2015.

[Remy 2013] Cazabet Remy, Nargis Pervin, Fujio Toriumi et Hideaki Takeda. *Information diffusion on twitter: everyone has its chance, but all chances are not equal.* In Signal-image technology & Internet-based systems (SITIS), 2013 international conference on, pages 483–490. IEEE, 2013.

[Ren 2016] Xiaoxuan Ren et Yan Zhang. *Predicting information diffusion in social networks with users' social Rroles and topic interests.* In Information retrieval technology, pages 349–355. Springer, 2016.

[Ritter 2011] Alan Ritter, Sam Clark, Oren Etzioni*et al. Named entity recognition in tweets: an experimental study.* In Proceedings of the conference on empirical methods in natural language processing, pages 1524–1534. Association for Computational Linguistics, 2011.

[Roberts 2008] Angus Roberts, Robert J Gaizauskas, Mark Hepple et Yikun Guo. *Combining terminology resources and statistical methods for entity recognition: an evaluation.* In Proceedings of the international conference on language resources and evaluation, Marrakech, Morocco, 26/5 - 1/6, 2008.

[Rogers 2012] Mark Rogers, Clovis Chapman et Vasileios Giotsas. *Measuring the diffusion of marketing messages across a social network.* Journal of direct, data and digital marketing practice, vol. 14, no. 2, pages 97–130, 2012.

[Sabate 2014] Ferran Sabate, Jasmina Berbegal-Mirabent, Antonio Cañabate et Philipp R Lebherz. *Factors influencing popularity of branded content in Facebook fan pages.* European management journal, vol. 32, no. 6, pages 1001–1011, 2014.

[Sahni 2017] Tapan Sahni, Chinmay Chandak, Naveen Reddy Chedeti et Manish Singh. *Efficient Twitter sentiment classification using subjective distant supervision.* In Communication systems and networks (COMSNETS), 2017 9th international conference on, pages 548–553. IEEE, 2017.

[Sakaki 2010] Takeshi Sakaki, Makoto Okazaki et Yutaka Matsuo. *Earthquake shakes Twitter users: real-time event detection by social sensors.*

In Proceedings of the 19th international conference on World wide web, pages 851–860. ACM, 2010.

[SanJuan 2012] Eric SanJuan, Véronique Moriceau, Xavier Tannier, Patrice Bellot et Josiane Mothe. *Overview of the INEX 2012 tweet contextualization track*. In Conference on multilingual and multimodal information access evaluation (CLEF 2012), Rome, Italie, 17/09/12-20/09/12, pages 148–160, 2012.

[Sloan 2015] Luke Sloan et Jeffrey Morgan. *Who tweets with their location? Understanding the relationship between demographic characteristics and the use of geoservices and geotagging on Twitter*. PloS one, vol. 10, no. 11, page e0142209, 2015.

[Suh 2010] Bongwon Suh, Lichan Hong, Peter Pirolli et Ed H Chi. *Want to be retweeted? large scale analytics on factors impacting retweet in twitter network*. In Social computing (socialcom), 2010 IEEE second international conference on, pages 177–184. IEEE, 2010.

[Tamine 2016] Lynda Tamine, Laure Soulier, Lamjed Ben Jabeur, Frederic Amblard, Chihab Hanachi, Gilles Hubert et Camille Roth. *Social media-based collaborative information access: analysis of online crisis-related twitter conversations*. In Proceedings of the 27th ACM conference on hypertext and social media, pages 159–168. ACM, 2016.

[Toutanova 2003] Kristina Toutanova, Dan Klein, Christopher D Manning et Yoram Singer. *Feature-rich part-of-speech tagging with a cyclic dependency network*. In Proceedings of the 2003 conference of the North American chapter of the association for computational linguistics on human language technology-volume 1, pages 173–180. Association for computational linguistics, 2003.

[Vieweg 2010] Sarah Vieweg, Amanda L Hughes, Kate Starbird et Leysia Palen. *Microblogging during two natural hazards events: what twitter may contribute to situational awareness*. In Proceedings of the SIGCHI conference on human factors in computing systems, pages 1079–1088. ACM, 2010.

[Washha 2016] Mahdi Washha, Aziz Qaroush et Florence Sedes. *Leveraging time for spammers detection on Twitter*. In Proceedings of the 8th in-

ternational conference on management of digital ecoSystems, pages 109–116. ACM, 2016.

[Watanabe 2011] Kazufumi Watanabe, Masanao Ochi, Makoto Okabe et Rikio Onai. *Jasmine: a real-time local-event detection system based on geolocation information propagated to microblogs.* In Proceedings of the 20th ACM international conference on information and knowledge management, pages 2541–2544. ACM, 2011.

[Weng 2011] Jianshu Weng et Bu-Sung Lee. *Event detection in twitter.* In Proceedings of the fifth international AAAI conference on weblogs and social media, volume 11, pages 401–408, 2011.

[Wing 2011] Benjamin P Wing et Jason Baldridge. *Simple supervised document geolocation with geodesic grids.* In Proceedings of the 49th annual meeting of the association for computational minguistics: human language technologies-volume 1, pages 955–964. Association for computational linguistics, 2011.

[Xiong 2012] Fei Xiong, Yun Liu, Zhen-jiang Zhang, Jiang Zhu et Ying Zhang. *An information diffusion model based on retweeting mechanism for online social media.* Physics letters A, vol. 376, no. 30, pages 2103–2108, 2012.

[Yang 2010] Zi Yang, Jingyi Guo, Keke Cai, Jie Tang, Juanzi Li, Li Zhang et Zhong Su. *Understanding retweeting behaviors in social networks.* In Proceedings of the 19th ACM international conference on information and knowledge management, pages 1633–1636. ACM, 2010.

[Yu 2011] Bei Yu, Miao Chen et Linchi Kwok. *Toward predicting popularity of social marketing messages.* In International conference on social computing, behavioral-cultural modeling, and prediction, pages 317–324. Springer, 2011.

[Zhai 2001] Chengxiang Zhai et John Lafferty. *Model-based feedback in the language modeling approach to information retrieval.* In Proceedings of the tenth international conference on information and knowledge management, pages 403–410. ACM, 2001.

[Zhang 2013] Jing Zhang, Biao Liu, Jie Tang, Ting Chen et Juanzi Li. *Social influence locality for modeling Retweeting Behaviors.* In IJCAI, volume 13, pages 2761–2767, 2013.

[Zhao 2007] Qiankun Zhao, Prasenjit Mitra et Bi Chen. *Temporal and information flow based event detection from social text streams.* AAAI, vol. 7, pages 1501–1506, 2007.