ECML/PKDD 2014

EUROPEAN CONFERENCE ON MACHINE LEARNING
AND
PRINCIPLES AND PRACTICE OF KNOWLEDGE DISCOVERY IN
DATABASES

The Fifth International Workshop on Mining Ubiquitous and Social Environments

MUSE'14

September 15, 2014

Nancy, France

Editors:
Martin Atzmueller
Knowledge and Data Engineering Group, University of Kassel
Christoph Scholz
Knowledge and Data Engineering Group, University of Kassel

Table of Contents

Preface

The 5^{th} International Workshop on Mining Ubiquitous and Social Environments (MUSE 2014) was held in Nancy, France, on September 15^{th} 2014 in conjunction with the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2014).

The emergence of ubiquitous computing has started to create new environments consisting of small, heterogeneous, and distributed devices that foster the social interaction of users in several dimensions. Similarly, the upcoming social web also integrates the user interactions in social networking environments.

In typical ubiquitous settings, the mining system can be implemented inside the small devices and sometimes on central servers, for real-time applications, similar to common mining approaches. However, the characteristics of ubiquitous and social mining in general are quite different from the current mainstream data mining and machine learning. Unlike in traditional data mining scenarios, data does not emerge from a small number of (heterogeneous) data sources, but potentially from hundreds to millions of different sources. Often there is only minimal coordination and thus these sources can overlap or diverge in many possible ways. Steps into this new and exciting application area are the analysis of this new data, the adaptation of well known data mining and machine learning algorithms and finally the development of new algorithms.

Mining big data in ubiquitous and social environments is an emerging area of research focusing on advanced systems for data mining in such distributed and network-organized systems. Therefore, for this workshop, we aim to attract researchers from all over the world working in the field of data mining and machine learning with a special focus on analyzing big data in ubiquitous and social environments.

The goal of this workshop is to promote an interdisciplinary forum for researchers working in the fields of ubiquitous computing, mobile sensing, social web, Web 2.0, and social networks which are interested in utilizing data mining in a ubiquitous setting. The workshop seeks for contributions adopting state-of-the-art mining algorithms on ubiquitous social data. Papers combining aspects of the two fields are especially welcome. In short, we want to accelerate the process of identifying the power of advanced data mining operating on data collected in ubiquitous and social environments, as well as the process of advancing data mining through lessons learned in analyzing these new data.

Submissions and Sessions

This proceedings volume comprises the papers of the MUSE 2014 workshop. In total, we received seven submissions, from which we were able to accept five submissions, four full papers and one short paper, based on a rigorous reviewing process. Additionally, the scientific program also features an invited talk on structured prediction in social contexts by Ulf Brefeld (TU Darmstadt University, Germany). Based on the set of accepted papers, and the invited talk, we set up three sessions. The first session features the invited talk by Ulf Brefeld.

The second session is a joint session with the 1st International Workshop on Machine Learning for Urban Sensor Data. The work *Mining Ticketing Logs for Usage Characterization with Nonnegative Matrix Factorization* by Mickaël Poussevin, Nicolas Baskiotis, Vincent Guigue and Patrick Gallinari presents and discusses an approach for extracting activity patterns from activity logs. Next, the paper *A Latent Space Analysis of Editor Lifecycles in Wikipedia* by Xiangju Qin, Derek Greene and Padraig Cunningham presents a new latent space analysis of editor lifecycles.

The third session starts with the work *Context-Aware Location Prediction* by Roni Bar-David and Mark Last. In this paper the authors propose a new approach to predict the future location of vehicles. Next, the work *On Spatial Measures for Geotagged Social Media Contents* by Xin Wang, Tristan Gaugel and Matthias Keller presents and analyses adaptions of of geo-spatial measures that show a lower dependency on the user size. After that, the work *TweetSurf: a System for Filtering and Visualizing Tweets* by Alina Beck and Philippe Suignard introduces a new framework for the continually collection and preprocessing of tweets that mention a specific company name. Concluding the session the paper *Content-Based Sentiment and Geolocation Tagging of Informal Social Media Messages for Trend Analysis* by Gretchen Markiewicz, Saurabh Khanwalkar, Guruprasad Saikumar, Amit Srivastava and Sean Colbath presents a new approach for sentiment and geolocation tagging.

We would like to thank the invited speaker, all the authors who submitted papers and all the workshop participants. We are also grateful to members of the program committee members for their thorough work in reviewing submitted contributions with expertise and patience. A special thank is due to both the ECML PKDD Workshop Chairs and the members of ECML PKDD Organizing Committee who made this event possible.

Nancy, September 2014

Martin Atzmueller
Christoph Scholz

Workshop Organization

Workshop Chairs

Martin Atzmueller	University of Kassel - Germany
Christoph Scholz	University of Kassel - Germany

Program Committee

Albert Bifet	University of Waikato, New Zealand
Christian Bauckhage	Fraunhofer IAIS, Germany
Martin Becker	University of Wuerzburg, Germany
Ulf Brefeld	TU Darmstadt University, Germany
Ciro Cattuto	ISI Foundation, Italy
Michelangelo Ceci	University of Bari, Italy
Stephan Doerfel	University of Kassel, Germany
Joao Gama	University Porto, Portugal
Andreas Hotho	University of Wuerzburg, Germany
Jens Illig	University of Kassel, Germany
Mark Kibanov	University of Kassel, Germany
Kristian Kersting	Fraunhofer IAIS and University of Bonn, Germany
Florian Lemmerich	University of Wuerzburg, Germany
Bjoern-Elmar Macek	University of Kassel, Germany
Dunja Mladenic	Jozef Stefan Institute, Slovenia
Ion Muslea	SDL Research Labs, USA
Rasmus Pedersen	Copenhagen Business School, Denmark
Nico Piatkowski	TU Dortmund University, Germany
Gerd Stumme	University of Kassel, Germany
Ganesh Viswanathan	Amazon, USA

Invited Talk

Structured Prediction in Social Contexts

Ulf Brefeld, TU Darmstadt University, Germany

In this talk I will show that many naturally arising problems in social domains would allow for applications of dynamic programming algorithms (e.g., knapsack, etc.) if all relevant parameters were known. Unfortunately, this is hardly the case in real-world scenarios. However, the core of structured learning is often a decoding machinery that exploits principles of dynamic programming. I argue that in case labeled data is available, the unknown variables can be learned by casting the problem as a structured prediction problem. The talk revisits structured learning and dynamic programming to motivate novel application scenarios and experimental results.

Biography Ulf Brefeld is a professor for Knowledge Mining & Assessment at the Computer Science Department of Technische Universität Darmstadt. Simultaneously, Ulf is leading the Knowledge Mining & Assessment Group at DIPF Frankfurt. Before, he led the Recommender Systems Group at Zalando GmbH and worked at Universität Bonn, Yahoo! Research Barcelona and at machine learning groups at Technische Universität Berlin, Max Planck Institute for Computer Science in Saarbrücken, and at Humboldt-Universität zu Berlin. He received a Diploma in Computer Science in 2003 from Technische Universität Berlin and a Ph.D. (Dr. rer. nat.) in 2008 from Humboldt-Universität zu Berlin.

A Latent Space Analysis of Editor Lifecycles in Wikipedia

Xiangju Qin, Derek Greene, and Pádraig Cunningham

School of Computer Science & Informatics, University College Dublin
xiangju.qin@ucdconnect.ie,{derek.greene,padraig.cunningham}@ucd.ie

Abstract. Collaborations such as Wikipedia are a key part of the value of the modern Internet. At the same time there is concern that these collaborations are threatened by high levels of member turnover. In this paper we borrow ideas from topic analysis to editor activity on Wikipedia over time into a latent space that offers an insight into the evolving patterns of editor behavior. This latent space representation reveals a number of different categories of editor (e.g. content experts, social networkers) and we show that it does provide a signal that predicts an editor's departure from the community. We also show that long term editors gradually diversify their participation by shifting edit preference from one or two namespaces to multiple namespaces and experience relatively soft evolution in their editor profiles, while short term editors generally distribute their contribution randomly among the namespaces and experience considerably fluctuated evolution in their editor profiles.

1 Introduction

With the popularity of Web 2.0 techniques, recent years have witnessed an increasing population of online peer production communities which rely on contributions from volunteers to build software and knowledge artifacts, such as Wikipedia, OpenStreetMap and StackOverflow. The growing popularity and importance of these communities requires a better understanding and characterization of user behavior so that the communities can be better managed, new services delivered, challenges and opportunities detected. For instance, by understanding the general lifecycles that users go through and the key features that distinguish different user groups and different life stages, we can: (i) predict whether a user is likely to abandon the community; (ii) develop intelligent task routing software to recommend tasks to users within the same life-stage. Moreover, the contribution and social interaction behavior of contributors plays an essential role in shaping the health and sustainability of online platforms.

Recent studies have approached the issue of modeling user lifecycles (also termed as user profiles or user roles) in online communities from different perspectives. Such studies have so far focused on separate or a combination of user properties, such as information exchange behavior in discussion forums [5], social and/or lexical dynamics in online platforms [6, 10], and diversity of contribution

behavior in Q&A sites [7]. These studies generally employed either principle component analysis and clustering analysis to identify user profiles [5, 7] or entropy measure to track social and/or linguistic changes throughout user lifecycles [6, 10]. While previous studies provide insights into community composition, user profiles and their dynamics, they have limitations either in their definition of lifecycle periods (e.g., dividing each user's lifetime using a fixed time-slicing approach [6] or a fixed activity-slicing approach [10]) or in the expressiveness of user lifecycles in terms of the evolution of expertise and user activity for users and the communities over time. Specifically, they fail to capture a mixture of user interests over time.

On the other hand, in recent years, there has been significant advances in topic models which develop automatic text analysis models for discovering latent structures from time-varying document collections. In this paper we present a latent space analysis of user lifecycles in online communities specifically Wikipedia. Our contributions are as follows:

- We model the lifecycles of users based on their activity over time using topic modeling, thus complementing recent work (e.g. [6, 7, 10]).
- This latent space analysis reveals a number of different categories of editor (e.g. content experts, social networkers) and offers an insight into the evolving patterns of editor behavior.
- We find that long term and short term users have very different profiles as modeled by their activity in this latent representation.
- We show that the patterns of change in user activity can be used to make predictions about the user's membership in the community.

The rest of this paper is organized as follows. The next section provides a brief review of related work. In Section 3, we provide an overview of dynamic topic models and explain data collection. Next, we present results about latent space analysis of editor lifecycles in Wikipedia, followed by results about churn prediction and conclusions.

2 Related Work

Over the last decade, the study of investigating and modeling the changes in user behavior in online communities has gripped researchers. Such studies have been conducted in varied contexts, including discussion forums [5], Wikipedia [9, 13], beer rating sites [6], Q&A sites [10, 7] and other wikis. Chan *et al.* [5] presented an automated forum profiling technique to capture and analyze user interaction behavior in discussion forums, and found that forums are generally composed of eight behavior types such as popular initiators and supporters. Welser *et al.* [13] examined the edit histories and egocentric network visualizations of editors in Wikipedia and identified four key social roles: substantive experts, technical editors, vandal fighters, and social networkers. Panciera *et al.* [9] studied the contribution behaviors of long-term editors and newcomers in Wikipedia and their changes over time, and found significant difference between the two groups: long-term editors start intensely, tail off a little, then maintain a relatively high level of activity over the course of their career; new users follow the same trend

of the evolution but do much less work than long-term editors throughout their lifespans. The studies mentioned provide insights about contributor behavior at a macro level, but are limited in capturing the change of behavior at a user level.

Danescu-Niculescu-Mizil *et al.* [6] examined the linguistic changes of online users in two beer-rating communities by modeling their term usages, and found that users begin with an innovative learning phase by adopting their language to the community but then transit into a conservative phase in which they stop changing the language. Rowe [10] modeled how the social dynamics and lexical dynamics of users changed over time in online platforms relative to their past behavior and the community-level behavior, mined the lifecycle trajectories of users and then used these trajectories for churn prediction. Based on the diversity, motivation and expertise of contributor behaviors in five Q&A sites, Furtado *et al.* [7] examined and characterized contributor profiles using hierarchical clustering algorithm and K-means algorithm, and found that the five sites have very similar distributions of contributor profiles. They further identified common profile transitions by a longitudinal study of contributor profiles in one site and found that although users change profiles with some frequency, the site composition is mostly stable over time. The aforementioned works provide useful insights into community composition, user profiles and their dynamics, but they are limited either in their definition of lifecycle periods (e.g., dividing each user's lifetime using a fixed time-slicing approach [6] or a fixed activity-slicing approach [10]) or in the expressiveness of user profiles in terms of the evolution of expertise and user activity for users and the communities over time [7]. Specifically, they fail to capture a mixture of user interests over time.

In recent years, there has been an increasing interest in developing automatic text analysis models for discovering latent structures from time-varying document collections. Blei and Lafferty [2] presented a dynamic topic model (DTM) which utilizes state space models to link the word distribution and popularity of topic over time. Wang and McCallum [11] proposed the topics over time model which employs a beta distribution to capture the evolution of topic popularity over timestamps. Ahmed and Xing [1] introduced the infinite dynamics topic models (iDTM) which can adapt the number of topics, the word distributions of topics, and the topics' popularity over time. Based on topic models, Ahmed and Xing (2011) proposed a time-varying user model (TVUM) which models the evolution of topical interests of a user while allowing for user-specific topical interests and global topics to evolve over time. Topic modeling plays a significant role in improving the ways users search, discover and organize web content by automatically discovering latent semantic themes from a large and otherwise unstructured collection of documents. Moreover, topic modeling algorithms can be adapted to many types of data, such as image datasets, genetic data and history user activity in computational advertising. However, to our knowledge, there exists no attempt to understand how users develop throughout their lifecycles in online communities from the perspective of topic modeling.

This paper complements the previous works by characterizing the evolution of user activity in online communities using dynamic topic modeling, examines

the patterns of change in users as modeled by their activity in the latent representation and demonstrates the utility of such patterns in predicting churners.

3 Model Editor Lifecycles

In this section, we provide a brief overview about dynamic topic models that we will use to model the lifecycle of Wikipedia users and introduce how we collect the data for evaluation.

3.1 Dynamic Topic Model

The primary goal of this study is to apply topic models on the evolving user activity collections in order to identify the common work archetypes[1] and to track the evolution of common work archetypes and user lifecycles in online communities. For this purpose, we employ an LDA based dynamic topic model proposed by Blei and Lafferty [2], in which the word distribution and popularity of topics are linked across time slices using state space models. First, we review the generative process of the LDA model [3], in which each document is represented as a random mixture of latent topics and each topic is characterized by a multinomial distribution over words, denoted by Multi(β). The process to generate a document d in LDA proceeds as follows:

1. Draw topic proportions θ_d from a Dirichlet prior: $\theta_d|\alpha \sim Dir(\alpha)$.
2. For each word
   (a) Draw a topic assignment from θ_d: $z_{di}|\theta_d \sim Mult(\theta_d)$.
   (b) Draw a word w_{di}: $w_{di}|z_{di}, \beta \sim Mult(\beta_{z_{di}})$.

Where α is a vector with components $\alpha_i > 0$; θ_d represents a topic-mixing vector for document d that samples from a Dirichlet prior (i.e. Dir(α)), each component (i.e. z_{di}) of θ_d defines how likely topic i will appear in d; $\beta_{z_{di}}$ represents a topic-specific word distribution for topic z_{di}.

LDA is not applicable to sequential models for time-varying document collections for its inherent features and defects: (1) Dirichlet distributions are used to model uncertainty about the distributions over words and (2) the document-specific topic proportions θ are drawn from a Dirichlet distribution. To remedy the first defect, Blei and Lafferty [2] chained the multinomial distribution of each topic $\beta_{t,k}$ in a state space model that evolves with Gaussian distributions, denoted as follows:

$$\beta_{t,k}|\beta_{t-1,k} \sim N(\beta_{t-1,k}, \sigma^2 I) \tag{1}$$

To amend the second defect, the same authors employed a logistic normal with mean α to capture uncertainty over proportions and used the following dynamic model to chain the sequential structure between models over time slices [2]:

$$\alpha_t|\alpha_{t-1} \sim N(\alpha_{t-1}, \sigma^2 I) \tag{2}$$

[1] Common work archetypes refer to the types of contribution that users make in online platforms, e.g., answering questions in Q&A sites, editing main pages in Wikipedia.

More details on the generative process of a dynamic topic model for a sequential corpus can be found in [2]. Note that documents generated using the DTM will have a mixture of topics. In Wikipedia, this indicates that a user has diverse edit interests and edits multiple namespaces in a specific time period.

3.2 Data Collection

In Wikipedia, the pages are subdivided into 'namespaces'[2] which represent general categories of pages based on their function. For instance, the article (or main) namespace is the most common namespace and is used to organize encyclopedia articles. Users can make edits to any namespace based on their edit preference. The amount of edits across all the namespaces can be considered as work archetypes. A namespace can be considered as a 'term' in the vector space for document collections, the number of edits to that namespace is analogous to word frequency. A user's edit activity across different namespaces in a time period can be regarded as a 'document'. One motivation of this study is to identify and characterize the patterns of change in user edit activity over time in Wikipedia. For this purpose, we parsed the August 2013 dump of English Wikipedia[3], collected the edit activity of all registered users, then aggregated the edit activity of each user on a quarterly basis (measured in 3-month period). In this way, we obtained a time-varying dataset consisting of the quarterly editing activity of all users from the inception of Wikipedia till August 5th, 2013. The statistics about the complete dataset are as follows: 51 quarters; 28 namespaces (or features); 5,749,590 unique registered users; 10,336,278 quarterly edit observations for all users[4]. An example of our dataset is as follows:

Table 1: A simple example of the dataset

Uname	quarter	articles	article talk	wikipedia	wikipedia talk	cat temp port pages	user	user talk
User A	1	650	233	2	299	33	0	81

Figure 1 plots the statistics about the number of active quarters for all registered users. We observe that an overwhelming number of users (i.e. 4,468,352 out of 5,749,590) stayed active for only one quarter; the figures for 2, 3 and 4 quarters are 600,417, 219,455 and 117,698, respectively. One obvious trend is that the number of users who stayed active for longer time periods is becoming smaller and smaller, indicating that Wikipedia experiences high levels of member withdrawal. To avoid a bias towards behaviors most dominate in communities with larger user bases, following Furtado *et al.* [7], we randomly selected 20% of users who stayed active for only one quarter, included these users and those who were active for at least two quarters into our dataset, generating a time-varying dataset with 2,162,978 unique users and 6,661,973 observations.

[2] http://en.wikipedia.org/wiki/Wikipedia:Namespace

[3] http://dumps.wikimedia.org/enwiki/20130805/

[4] The first quarter for each user is the same, i.e., the one where Wikipedia was founded. For a specific quarter, a user had observation only if that user had edit activity.

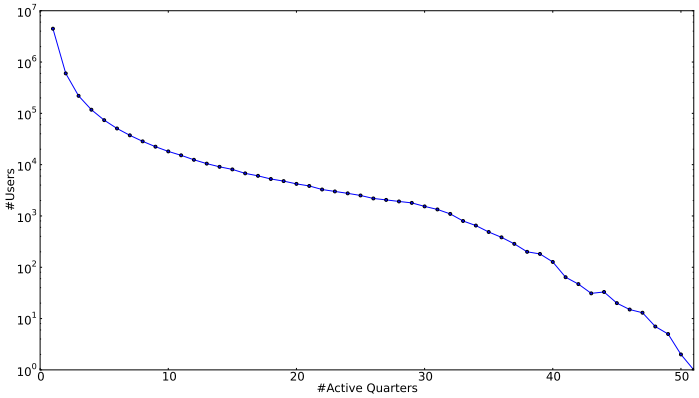


Fig. 1: Statistics about the lifespans of all registered users in Wikipedia. X-axis represents the number of active quarters, y-axis represents the number of users who stayed active for a specific number of quarters, and is ploted in log10 scale.

4 Editor Lifecycles as Released by Shift in Participation

In this section, we present the analysis of editor lifecycles from two perspectives: (1) community-level evolution; and (2) user-level evolution for group of editors. From the analysis, we identify some basic features that are useful for predicting how long will a user stay active in an online community (Section 5).

4.1 Community level change in lifecycle

Online communities experience dynamic evolution in terms of a constantly changing user base and the addition of new functionalities to maintain vitality. For instance, online platforms like Wikipedia generally experience high levels of member withdrawal, with 60% of registered users staying only a day. Wikipedia introduced social networking elements to MediaWiki in order to attract and retain user participation[5], which is believed to increase user edit activity. As such, to understand what shape these communities, it is essential to take into account both dimensions of evolution. The dynamic topic models can accommodate both aspects of evolution, which provides valuable insights about the development of community and its users across time. To analyze the evolution of time-varying user edit activity, we ran the DTM software[6] released by Blei

[5] http://strategy.wikimedia.org/wiki/Attracting_and_retaining_participants
[6] Available at: `http://www.cs.princeton.edu/~blei/topicmodeling.html`

and Lafferty [2] with default hyperparameters and different numbers of topics $k \in [5,7,8,9,10,15]$[7].

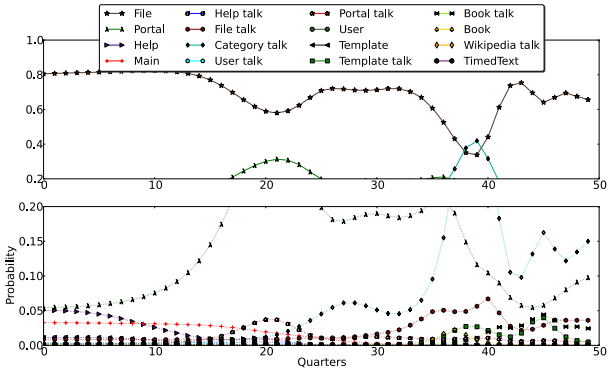
Table 2: Summary of common user roles. Dominant features are in bold font.

Id	Name	Edit Namespaces (Sequence indicates the importance)
1	Social Networkers	***user talk***, *main, Wikipedia talk*
2	Content Proposal; conversial coordinator	***article talk***, ***main***, *user talk, Wikipedia talk*
3	Content Experts	***main***
4	All-round contributors	***file***, ***portal***, ***category talk***, *main, portal talk, book talk, template talk*
5	Admin related roles I	***user***, *main, article talk, user talk*
6	Technical experts	***category***, ***template***, *main, template talk, category talk*
7	Admin related roles II	***Wikipedia***, ***Wikipedia talk***, *main, user talk, article talk, user*

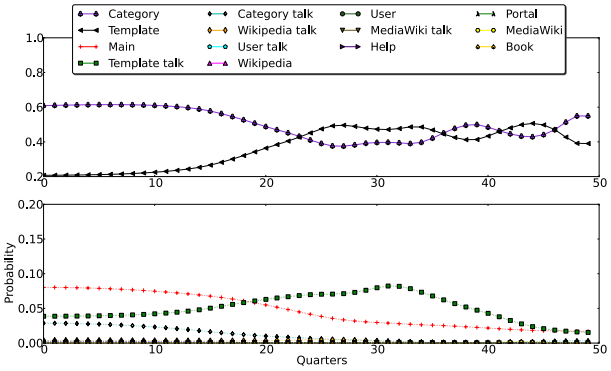
Summary of user roles. Table 2 presents a summary of the common user roles identified by DTM [2]. It is obvious from Table 2 that, each user role is defined by different combinations of namespaces. For instance, user role "Social Networkers" is dominated by *user talk* namespace, then *main* namespace and *Wikipedia talk* namespace, indicating that users who are assigned to this group spend most of their time in Wikipedia interacting with other users; user role "All-round contributors" is dominated by *file, portal, category talk, main* and other namespaces, as suggested by the large amount of namespaces, this group of users contribute to a diversity of namespaces and hence the name for this user role; user role "Admin related roles I" and "Admin related roles II" mainly relate to maintenance, management and organization aspects of Wikipedia. Note that different user roles correspond to different common work archetypes.

Community-level change in user roles. As users stay long enough in Wikipedia, they tend to shift their participation by changing their edits to other namespaces. Moreover, the addition of new functionalities also encourages shift in user participation. Figure 2 visualizes the evolution of two user roles at an aggregate level, from which we make several observations. First, in Figure 2(a), the probabilities fluctuate over time, with namespaces emerged in some timestamp. For example, *category talk* namespace gradually emerges as a feature from the 21st quarter, reaches its peak at the 38th quarter, followed by a decrease afterwards. The emergence of namespaces may well be corresponding to new elements being added to Wikipedia, which promotes shift in user participation. Second, by contrast, in Figure 2(b), the probabilities experience a more soft evolution, with *category* and *template* namespaces dominated the user role profile. Last, corresponding to the emergence of namespace, some namespaces appear as features in profiles with low probabilities in the early lifespan, but disappear from the profiles in the late lifespan. For instance, in Figure 2(b), *category talk* namespace appears as a feature in the first 20 quarters with a probability of 0.03, but disappears from this quarter onwards.

[7] We analyzed the runs with different k, found that the run with 7 topics provides more interpretable and expressiveness results in terms of interpretation and overlapping between different topics. The results are reported in Table 2 and Figure 2.



(a) User role "All-round contributors"



(b) User role "Technical experts"

Fig. 2: Example of the evolution of the common user roles. X-axis corresponds to quarters, y-axis indicates the probability of a namespace appearing in a user role profile. Each curve represents the evolution of the probability of a namespace appearing in the profile. Note that the scales are different in two parts of y-axis.

The other user roles evolve in similar ways as those in Figure 2, but are omitted due to space limitation. The evolution of user roles captures the overall changes in editor profiles due to the addition of new functionalities and shifts in user participation over time, but it provides little insight about what is the general trajectory of a user's participation as the user transitions from being a newcomer to being an established member of the community. We explore this question in the next section.

4.2 User Lifecycle

We now move on to examining how group of users evolve throughout their life-cycle periods. Understanding how individual users develop over time in online

communities makes it possible to develop techniques for important tasks such as churn prediction, as we will show through experiments in Section 5.

Clustering of Wikipedia editor profiles. Different from previous studies which analyze user lifecycle on a medium size dataset (with about 33,000 users [6, 7, 10]), our analysis is based on a large dataset with approximately half a million users[8]. Different users are more likely to follow a slightly or totally different trajectories in their lifecycle. As a result, it is very unlikely for us to employ one single model to perfectly fit the development of lifecycle for all users, as Rowe [10] chose linear regression model to characterize the development of user properties. On the other hand, Non-negative Matrix Factorization (NMF) clustering is an efficient clustering approach for large scale dataset. In this analysis, we employ the NMF approach to cluster Wikipedia editor profiles, and identify the general trajectories for group of editors.

To perform NMF clustering on the dataset, we restructured the editor profiles as follows: for each editor, we combined all her quarterly profile assignments obtained by DTM into one record, with each dimension representing the probability that the editor of a specific quarter was assigned to a user role. If an editor had no edit activity in a quarter and thus had no profile assignment, we used missing data (i.e. 0 values) to represent the corresponding dimensions.

We clustered the Wikipedia editor profile data as follows. Firstly, we constructed a 455233×350 profile-feature matrix, where each row corresponds to a different editor profile. Rows were subsequently L2 normalized to ensure that each profile vector had unit length. To cluster this matrix efficiently, we use the fast alternating least squares variant of NMF introduced by Lin [8]. To produce deterministic results and avoid a poor local minimum, we used the Non-negative Double Singular Value Decomposition (NNDSVD) strategy [4] to choose initial factors for NMF. We ran this process for different numbers of clusters $K \in [4, 10]$, and inspected both the resulting coefficient matrix (*i.e.* the profile clusters) and basis vector matrix (*i.e.* the feature clusters) for each value of k. Finally, to produce disjoint clusters, where each profile is assigned to a single cluster, we discretized the coefficient matrices. Table 3 presents a summary of NMF clustering on the editor profile data[9].

In Table 3, column 'Active quarter statistics' represents the minimum, maximum, median or average number of quarters that the group of editors stay active in Wikipedia; column 'Dominant user roles' indicates which user roles are dominant in the profiles for each cluster as suggested by NMF clustering. Note that user role 3 is one of the dominant features in 8 clusters, suggesting that user role 3 is a common and important role in Wikipedia. Column '#admins' represents the number of administrators in the clusters: Cluster 3 contains the largest number of administrators. We observe from Table 3 that Cluster 9 is the

[8] For the clustering and churn analysis, we only considered editors who were active for at least 4 quarters, resulting in 455233 editors. The data for the 51st quarter (01/07–30/09, 2013) was incomplete and excluded from the analysis.

[9] We applied NMF clustering on the dataset with $K \in [4, 10]$, and found that the clustering with K=10 provides a more reasonable grouping of editors.

Table 3: Summary for NMF clustering of editor profiles (N=455233)

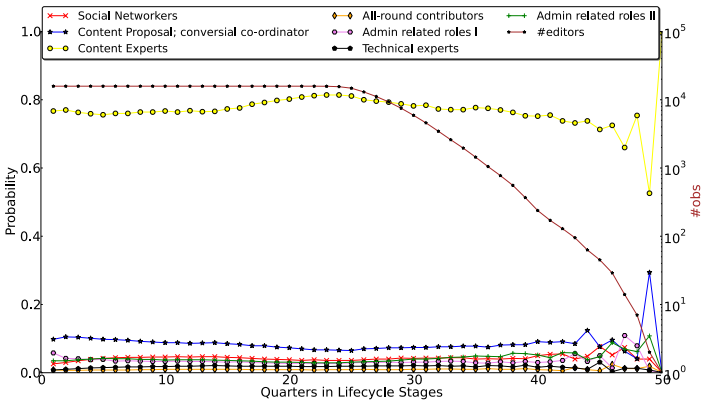
cluster_id	#editors	Fraction	Active quarter statistics				Dominant features		#admins
			Min	Max	Median	Avg	Quarters	User roles	
1	9673	0.021	7	20	9	8.633	1–8	3, 2, 5	12
2	34348	0.075	13	42	19	19.279	8–17	3	317
3	15812	0.035	20	50	28	28.992	17–26	3	1086
4	10096	0.022	6	32	7	7.222	4–7	3, 5, 2, 1, 7	10
5	20889	0.046	4	49	7	8.76	1–10	2	82
6	5268	0.012	5	25	5	6.081	2, 4, 5	1–7	7
7	54575	0.120	9	42	12	12.286	6–13	3, 2	131
8	8511	0.019	4	27	4	4.47	1–3	1–5, 7	1
9	277118	0.609	4	42	5	5.764	1–4	1–7	349
10	18943	0.042	4	39	6	7.052	1–8	5, 1	78

largest cluster with about 61% of editors: its average number of active quarters is about 6, the editors in this cluster mainly active in quarter 1–4, suggesting that a large proportion of editors in this cluster stayed active in Wikipedia for only 4 quarters. Cluster 6 is the smallest cluster with only 1.2% of editors: its dominant quarters is 1–6, indicating that this group of editors also stayed active for relatively short periods of time compared with other clusters. The editors in Cluster 2 and 3 stayed active for very long periods of time, with their average number of active quarters being 18 and 28, respectively. As we will show next, the NMF clustering provides a reasonable good clustering of editor profiles.

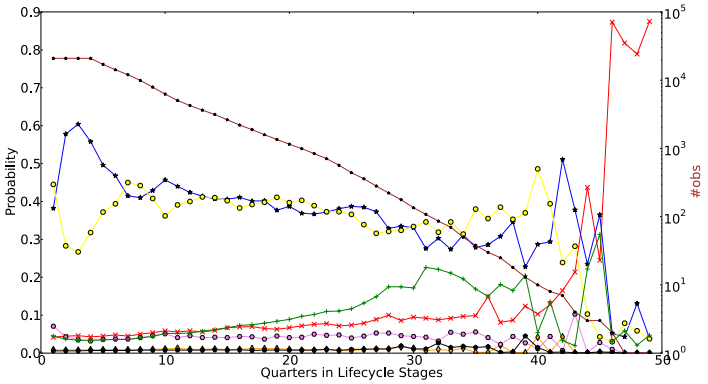
Group-level change in editor profiles. We now rely on the clustering of editors to analyze the general development of editor profiles on a group basis. Following previous studies [6, 10] in which the general evolution of user lifecycle was visualized against user life-stage, Figure 3 plots the average assignment of each editor profile to user roles at different life-stages for 3 clusters[10]. Inspecting the evolution of profile assignments over the editors' life-stages in Figure 3, we make the following observations for two group of editors:

- Based on the trend of evolution in the plots and the statistics in Table 3, we can further divide the editor profiles into 2 groups: long term editors with their profile assignments experienced relatively soft evolution (Cluster 2, 3, 5, 7, 9, 10), short term editors with their profile assignments experienced considerably fluctuated evolution (Cluster 1, 4, 6, 8). On average, group of long term editors stay active in Wikipedia for much longer time periods than group of short term editors.
- Long term editors: in the early stage of lifespans, they generally focused their contribution in one or two namespaces (e.g., *main* namespace in Cluster 2, 3, 7, 9; *main* and *article talk* namespaces in Cluster 5; *main* and *user* namespaces in Cluster 10); in the late stage of lifespans, this group of editors gradually diversified their participation by shifting edit preference from

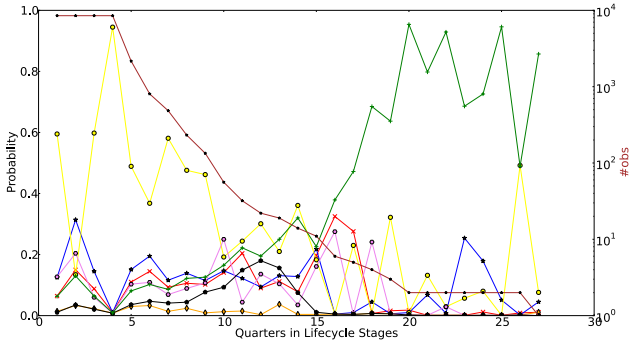
[10] The plots for Cluster 1, 4, 6 are very similar to that of Cluster 8; the trend of the plots for Cluster 2, 7, 9 is similar to that of Cluster 3; the plot for Cluster 10 is very similar to that of Cluster 5; these plots are omitted due to space limitation.



(a) Cluster 3



(b) Cluster 5



(c) Cluster 8

Fig. 3: Lifecycle on a group basis: x-axis represents the relative quarter in editors' lifestage, the left y-axis represents the the probability of a user role being assigned to editor profile, the right y-axis represents the number of editor profiles observes in a quarter. Different curves represent the evolution of profile assignments.

dominant namespaces to multiple namespaces; we also observe from Figure 3(a) that editors in Cluster 3 mainly edited the *main* namespace over the course of their career.
- Short term editors: their profile assignments experienced more fluctuations across their lifecycle periods, indicating these users did not develop long term edit interest in one namespace and generally distributed their contribution randomly among the namespaces.

To summarize, we find that: long term editors generally have long term edit preference in one or more namespances throughout their career and diversify their participation in the late stage of their lifespans; by contrast, short term editors generally do not develop long term edit interest and tend to distribute edits among the namespaces, thus experience more fluctuation in their profile assignments. The observations suggest that the fluctuation in profile assignments may be an indicator that users are likely to leave the community. Next, we generate features corresponding to these findings for churn prediction task.

5 Application of Editor Lifecycles to Churn Prediction

We have so far focused on understanding the lifecycle trajectories of editors based on their exhibited behavior. We now move on to exploring how predictive are the features generated from patterns of change in editor profile in identifying whether a user will abandon a community. Churners present a great challenge for community management and maintenance as the leaving of established members can have a detrimental effect on the community in terms of creating communication gap, knowledge gap or other gaps.

Definition of churn prediction. Following the work by Danescu-Niculescu-Mizil *et al.* [6], we define churn prediction task as predicting whether an editor is among the 'departed' or the 'staying' class. Considering that our dataset spans for about 13 years (i.e. 50 quarters), and that studies about churn prediction generally follow the diagram of predicting the churn status of users in the prediction period based on user exhibited behavior in the observation period (e.g. [12, 6]), we employed sliding-window based method for churn prediction. Specifically, we make predictions based on features generated from editor profile assignments in a w=4 quarters sliding window. An editor is in the 'departed' class if she leaved the community before active for less than m=1 quarter after the sliding window, lets denote the interval [w, w+m] as the departed range. Similarly, an editor is in the 'staying' class if she stayed active in the community long enough for a relatively large $n \geq 3$ quarters after the sliding-window; we term the interval $[w+n, +\infty]$ as the staying range.

Features used for the task. Our features are generated based on the findings reported in the previous section. For simplicity, we assume the w quarters included in the i-th sliding-window being i=$[j, \cdots, j+w-1]$ ($j \in [1, 50]$), and denote the Probability Of Activity Profile of an editor in quarter j being assigned to the k-th user role as $\text{POAP}_{i,j,k}$. We use the following features to characterize the patterns of change in editor profile assignments:

- *First active quarter*: the quarter in which an editor began active in Wikipedia. The timestamp a user joined the community may affect her decision about whether to stay for longer as research suggested that users joined later in Wikipedia may face a more severe situation in terms of contribution being accepted by the community and being excluded by established members.
- *Cumulative active quarters*: the total number of quarters an editor had been active in the community till the last quarter in the sliding window.
- *Fraction of active quarters in lifespan*: the proportion of quarters a user stayed active till the sliding window. For instance, if an editor joined in quarter 10, and stayed active for 8 quarters till the current sliding window (16–19), then the figure is calculated as: 8/(19-10+1) = 0.80.
- *Fraction of active quarters in sliding window*: the fraction of quarters a user stayed active in current sliding window.
- *Diversity of edit activity*: denotes the average entropy of $\text{POAP}_{i,j,k}$ for each quarter j in window i. This measure captures the extent to which an editor diversified her edit towards multiple namespaces.
- *mean(POAP$_{i,j,k}$)*: denotes the average of $\text{POAP}_{i,j,k}$ for each user role k in window i. This measure captures whether an editor focused on one or more namespace in window i.
- $\Delta POAP_{i,j,k}$: denotes the change in $\text{POAP}_{i,j,k}$ between the quarter $j-1$ and j (for $j \in [2, 50]$), and is measured by $\Delta\text{POAP}_{i,j,k}$=($\text{POAP}_{i,j,k}$ − $\text{POAP}_{i,j-1,k} + \delta$)/($\text{POAP}_{i,j-1,k} + \delta$), where δ is a small positive real number (i.e. 0.001) to avoid the case when $\text{POAP}_{i,j-1,k}$ is 0. This measure also captures the fluctuation of $\text{POAP}_{i,j,k}$ for each user role k in window i.

For each editor, the first three features are global-level features which may be updated with the sliding window, the remaining four features are window-level features and are recalculated within each sliding window. The intuition behind the last four features is to approximate the evolution of editor lifecycle we sought to characterize in previous section. The dataset is of the following form: D=(x_i, y_i), where y_i denotes the churn status of the editor, $y_i \in$ {Churner, Non-churner}; x_i denotes the feature vector for the editor.

Experimental setup. The prediction task is a binary classification problem, we use the RandomForest algorithm[11] for the purpose. To avoid bias in the results due to highly imbalance of class distribution, for each sliding window, we randomly sampled the editors in order to generate a dataset with a desired class ratio (churners:non-churners) being 1:2. Each time we slide the window by one quarter. The results reported next are averaged over 10-fold cross-validation.

Performance of sliding-window based churn prediction. Figure 4 plots the performance of churn prediction. We observe that in the first 10 windows, the number of editors observes in each window is relatively small (less than 1000), which results in a fluctuation in all performance measures; from the 10th window onwards, the number of editors grows steadily and then maintains at a level of about 65,000 after the 20th window, the performance measures become

[11] We also experimented with other algorithms (e.g. Logistic regression and SVM) and obtained similar performance.

relatively stable: with FP rate being **0.31**, ROC area being **0.80**, other measures (i.e. TP rate, Precision, Recall, F-measure) being **0.75**.

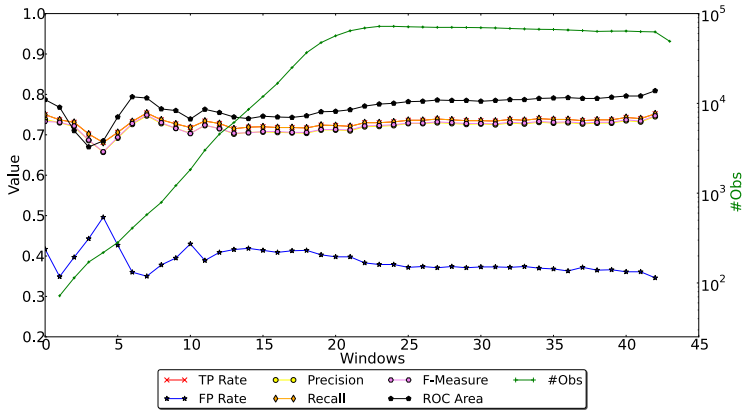


Fig. 4: Performance for sliding-window based churn prediction. X-axis represents the sliding window, the left y-axis represents the performance, the right y-axis represents the number of editor profiles observes in a window.

Note when Danescu-Niculescu-Mizil *et al.* [6] performed churn prediction on two beer rating communities (i.e. BeerAdvocate and RateBeer) based on user's linguistic change features, they obtained the best performance of Precision being 0.77, Recall being 46.9, F-measure being 0.56. Rowe [10] evaluated churn prediction task on three online platforms (i.e. Facebook, SAP and Server Fault) using user's social and lexical change related features, and obtained the best performance of Precision@K being 0.791 and AUC being 0.617. Comparison of the figures indicates that our study achieves at least comparable overall and average performance with the other two studies for churn prediction in online communities. This observation suggests that in online communities, the sudden change in user activity can be an important signal that the user is likely to abandon the community.

Cumulative gains for churn prediction. The lift factors are widely used by researchers to evaluate the performance of churn-prediction models (e.g. [12]). The lift factors achieved by our model are shown in Figure 5. In lift chart, the diagonal line represents a baseline which randomly selects a subset of editors as potential churners, i.e., it selects s% of the editors that will contain s% of the true churners, resulting in a lift factor of 1. In Figure 5, on average, our model was capable of identifying 10% of editors that contained 21.2% of true churners (i.e. a lift factor of 2.12), 20% of editors that contained 39.3% of true churners (i.e. a lift factor of 1.97), and 30% of editors that contained 54.7% of true churners (i.e. a lift factor of 1.82). Evidently, our model achieved higher lift factors than the baseline. Thus if the objective of the lift analysis is to identify a

small subset of likely churners for an intervention that might persuade them not to churn, then this analysis suggests that our model can identify a set of 10% of users where the probability of churning is more than twice the baseline figure.

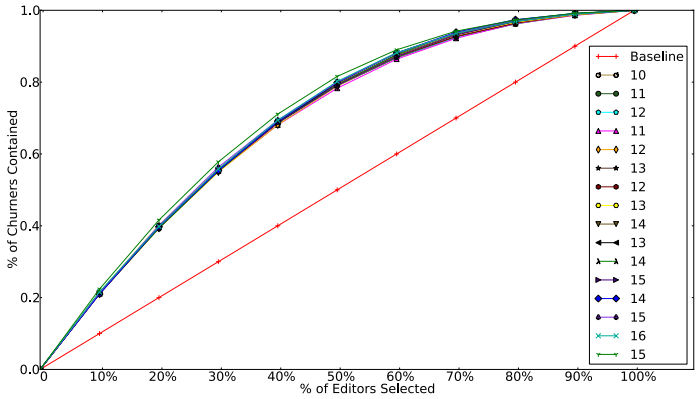


Fig. 5: Lift chart obtained by the proposed churn-prediction model. Different curves represent the lift curves for different sliding windows.

Feature analysis. To better understand the contribution of each feature set in the context of all the other features, we performed an ablation study. Specifically, for each set of features, we ran a classifier to contain all of the features except the target set. We then compared the performance of the resulting classifier to that of classifier with complete set of features. We find that the classifier with $\Delta\text{POAP}_{i,j,k}$ excluded results in the most decrease in performance, indicating that the sudden change in user behavior can be an important signal that the user is likely to abandon the community. We also observe that none of the features can by themselves achieve the performance reported in Figure 4, which suggests that the features we generated complement each other in predicting churners in online communities.

6 Conclusion

In this paper we have presented a novel latent space analysis of editor lifecycles in online communities based on user exhibited behavior. The analysis reveals a number of different categories of editor (e.g. content experts, social networkers) and provides a means to track the evolution of editor behavior over time. We also perform a non-negative matrix factorization clustering of editor profiles in the latent space representation and find that long term and short term users generally have very different profiles and evolve differently in their lifespans.

We show that understanding patterns of change in user behavior can be of practical importance for community management and maintenance, in that the features inspired by our latent space analysis can differentiate churners from

non-churners with reasonable performance. This work opens a few interesting questions for future research. Firstly, social networking plays a foundational role in online communities for knowledge and resource sharing and member retention, future work should accommodate social dynamics in the model. Secondly, the topic model we used in this paper did not account for the evolution of activity in user-level, it will be very helpful to develop topic models that accommodate the evolution in the user and community level. Moreover, online platforms are very similar in terms of allowing multiple dimensions of user activities, the approach presented in this paper can be generalized to other platforms very easily.

Acknowledgements. This work is supported by Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289 (Insight Centre). Xiangju Qin is funded by UCD-CSC Scholarship Scheme 2011.

References

1. Ahmed, A., Xing, E.P.: Timeline: A Dynamic Hierarchical Dirichlet Process Model for Recovering Birth/Death and Evolution of Topics in Text Stream. In: Proceedings of UAI 2010. pp. 20–29. AAAI (2010)
2. Blei, D.M., Lafferty, J.D.: Dynamic Topic Models. In: Proceedings of ICML'06. pp. 113–120. ACM (2006)
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
4. Boutsidis, C., Gallopoulos, E.: SVD based initialization: A head start for non-negative matrix factorization. Pattern Recognition (2008)
5. Chan, J., Hayes, C., Daly, E.M.: Decomposing Discussion Forums using User Roles. In: Proceedings of ICWSM 2010. pp. 215–218 (2010)
6. Danescu-Niculescu-Mizil, C., West, R., Jurafsky, D., Leskovec, J., Potts, C.: No Country for Old Members: User Lifecycle and Linguistic Change in Online Communities. In: Proceedings of WWW'13. pp. 307–318. Rio de Janeiro, Brazil (2013)
7. Furtado, A., Andrade, N., Oliveira, N., Brasileiro, F.: Contributor Profiles, their Dynamics, and their Importance in Five Q&A Sites. In: Proceedings of CSCW'13. pp. 1237–1252. ACM, San Antonio, Texas, USA (2013)
8. Lin, C.: Projected gradient methods for non-negative matrix factorization. Neural Computation 19(10), 2756–2779 (2007)
9. Panciera, K., Halfaker, A., Terveen, L.: Wikipedians are born, not made: A study of power editors on wikipedia. In: Proceedings of GROUP'09. pp. 51–60. ACM (2009)
10. Rowe, M.: Mining User Lifecycles from Online Community Platforms and their Application to Churn Prediction. In: Proceedings of ICDM 2013. pp. 1–10. IEEE, Dallas, Texas, USA (2013)
11. Wang, X., McCallum, A.: Topics over Time: A non-Markov Continuous-time Model of Topical Trends. In: Proceedings of KDD'06. pp. 424–433. ACM (2006)
12. Weia, C.P., Chiub, I.T.: Turning telecommunications call details to churn prediction: a data mining approach. Expert Systems with Applications 23(2), 103–112 (2002)
13. Welser, H.T., Cosley, D., Kossinets, G., Lin, A., Dokshin, F., Gay, G., Smith, M.: Finding social roles in wikipedia. In: Proceedings of iConference'11. pp. 122–129. ACM (2011)

TweetSurf: a System for Filtering and Visualizing Tweets

Alina Beck and Philippe Suignard

1 avenue du General de Gaulle, 92140 Clamart, France
{alina.beck,philippe.suignard}@edf.fr

Abstract. This paper presents TweetSurf, a system designed to continually download, filter and display tweets mentioning a given company. The filtering phase is very important in the case of companies whose names are ambiguous (e.g. apple the fruit vs. Apple Inc.). In order to filter noisy tweets, the system uses a new online algorithm based on tweets contents and also on a graph modeling users activity. Tweets are then displayed in an interface that allows users to easily have an overview of the data, visualize trends and select tweets for further analysis.
The system has been used for collecting, filtering and visualizing tweets about EDF, a French utility company. The word "EDF" is ambiguous as it designates both the company EDF and French sport teams. The system gives very satisfying results, being capable of correctly classifying tweets as company-related or noisy in most cases.

Keywords: microblogging, noise filtering, online algorithms, contents visualization

1 Introduction

For companies selling products or services, it is very important to be aware of their customers' opinions on the different items they propose and on the company in general. The Internet represents an important source of information since clients often use Web social media to ask questions, discuss news and express opinions. Among the different websites, Internet users turn to microblogging services for publishing and gathering real-time news or opinions about people, companies, events etc. Twitter is such a microblogging site that has been widely used since its creation. It thus represents a very important source of information for companies that want to learn their customers opinions on company-related news and events.

Companies that want to use Twitter in their customer relationship management must be able to collect and process tweets on the fly. They must be responsive and innovative in a competitive environment where the data to analyze may be big and noisy. Indeed, even though specific keywords are used for monitoring tweets related to the company, the collected tweets may be very numerous and some of them may not be relevant for the company, especially if the company's name is ambiguous.

The company EDF (Electricité de France, one of the biggest energy suppliers in the world) has such an ambiguous name as the word "EDF" designates both "Electricité de France" and "équipe de France" (French for French team). If the search of tweets talking about the company is based on the keyword "EDF", the collected dataset is sure to contain noisy tweets i.e. tweets that do not talk about the company (even though they mention its name). This situation may occur for many other companies or products whose names are ambiguous, like Orange, Apple, Blackberry, Jaguar, Metro, Seat, Amazon etc.

Noisy contents must be filtered from the datasets because they can lead to false analysis: false statistics on users' interest for the given subject, false conclusions on the opinions etc. The filtering step can be difficult because of the tweets characteristics (very short content, specific language, possibly incorrect grammar, specific abbreviations etc.).

In this paper we present TweetSurf, a system we have developed for continually collecting, filtering and visualizing tweets related to a given subject and its practical application to the case of EDF. The main scientific contribution of this paper lies in the filtering part. The new method we propose for filtering noisy tweets is scalable, leverages both tweets vocabulary and users' activity, and resolves new tweets on the fly. What's more, it continues performing well over time, without any human intervention.

The remainder of the paper is organized as follows. Section 2 presents the architecture of TweetSurf and its main functions. Next, Section 3 discusses related work. Sections 4, 5 and 6 describe the different components of TweetSurf along with the methods they implement. Section 7 details the results obtained by using the system within the company EDF. Finally, Section 8 presents the conclusion and some ideas for future work.

2 TweetSurf's architecture

TweetSurf consists of four components which work continually, each new tweet being handled on the fly. When component i finished processing a given tweet, component $i + 1$ starts processing the tweet. We present here the four components.

1. Download. The main goal of TweetSurf is to collect tweets talking about a given company, in our case EDF. For that, the first component downloads tweets containing the keyword "EDF". The advantage of using the name of the company alone when searching for tweets is that the research is not limited to some predefined topics. This would be problematic since users could forget or even not be aware of some interesting topics. Also, new topics may emerge, so users would have to define regularly the topics to search.

2. Cluster. The second component of TweetSurf implements an online algorithm for clustering almost identical tweets. Tweets with almost the same text are quite frequent in Twitter datasets. For instance, a retweet has often the same text as the original tweet, possibly with "RT@author" added at the beginning. Also, many online media (i.e. journals) propose shortcuts that allow

users to send a tweet containing the title of an article. If several visitors to the site use the shortcut of the same article, the corresponding tweets contain the same text. Finally, people may hear a special phrase or title of a report on the television, for instance, and write a tweet reproducing it exactly. In this case too, the corresponding tweets have (almost) the same text.

TweetSurf puts each newly downloaded tweet in a cluster, which is useful for reducing the quantity of tweets to analyze. For instance, qualitative or text analysis can be performed directly on the resulting clusters. As we will show in this paper, the clustering of tweets is also useful when filtering noisy contents.

3. Label. This component labels tweets as "company-related" or "noisy". TweetSurf accomplishes this task on the fly, when each new tweet is downloaded, using a new method we propose.

4. Display. Once the tweets are clustered and labeled as "company-related" or "noisy", the system displays existing tweets along with meta-data (e.g. publication date, author, cluster, label etc.). Users can select tweets based on different queries and also visualize several statistics on tweets. They can also follow the evolution of different elements (e.g. words, hash-tags, clusters, authors) throughout time. This component of TweetSurf helps analysts in the task of mining textual contents and, if desired, in the task of preparing datasets for further analysis.

Even though TweetSurf was used for the case of EDF only, it can be easily adapted to the case of other companies whose names are ambiguous. Actually, except for the "Label" component, all the other components can be used exactly as they are for the search, collection, clustering and visualizing of tweets on any topic. As for the labeling part, the only elements that need to be changed, in the case of other disambiguation problems, are two predefined lists of words.

3 Related work

Named-entity disambiguation in tweets. Identifying mentions that do not refer to a given company can be seen as a named entity disambiguation problem. This task has received a lot of attention from the Natural Language Processing community in the last decade [4, 6, 7]. The transposition of this problem to Twitter has also been addressed in the literature. What's more, in 2010 this task represented the subject of a WEPS-3 challenge (http://nlp.uned.es/weps/weps-3). Five research groups submitted results for the task [1]. The best results (an accuracy of 0.83) were obtained by the algorithm LSIR-EPFL [26]; see [27] for the extended version. This system uses several sources (Wordnet, meta-data from the company Web page, Google results, Wikipedia pages) in order to define lists of weighted words describing the company and the noise. The second best system was ITC-UT [28] which used an initial classification step in order to predict the ambiguity of the company name. The classification step consisted of a set of rules based on Part of Speech tagging and Named company recognition. The third system was SINAI [10] which, as LSIR-EPFL, used additional sources based on named companies extracted from the tweets.

All these systems are based on textual information extracted from tweets and, possibly, additional sources. However, no information about the authors of tweets and the links between them (e.g. the links in the Twitter network) is used. In this direction, the authors of [22] used a graph-based approach by taking advantage of the authors of tweets and the links between them. They used a predefined list of words in order to filter out noisy tweets directly. The remaining tweets were evaluated using the hypothesis that most noisy tweets have already been eliminated. This, however, is true only if the predefined list of words is very long. And even then, the list may not suffice if the system has to evaluate a new set of tweets that was collected some time after the definition of the list (as new noise-related events, described by new vocabulary, may have occurred in between). The model proposed in [25] is also graph-based. It has the advantage of not needing any prior knowledge; however, it needs data on several similar companies to the given one.

The main drawback of all these algorithms is that they do not work online. To our knowledge, the algorithm we propose in this paper is the first one capable of labeling tweets on the fly. It is scalable, uses both the contents and the authors of tweets and is capable of adapting to new events, described by new vocabulary, with no human intervention.

Tweets visualization. Nowadays, there is a growing need to analyze text data from an ever-increasing number of emails, blogs, forums and social networks. Tools and visualization techniques can be very useful in the analysis process. For Twitter data, displaying the Twitter "time-line" is not enough and many initiatives are emerging in this area. Although the interest in visualizing Twitter data is quite recent, the domain of visualization in general, and the visualization of topics evolution in particular, is not new. Thus, several representation systems have been proposed as, for instance, ThemeRiver [11], StreamGraph [5] and StoryFlow [18]. A more recent one, called TextFlow [8], identifies how new topics emerge, develop and disintegrate or dissolve into other topics. One can cite other interactive software like Jigsaw [21] or PosVis [24]. There are also libraries or graphic visualization techniques, more or less general, such as ManyEyes [23] or Prefuse [12].

In addition, some visualizations have been specially developed for Twitter. Nokia Internet Pulse [14] automatically scans the microblogging site using specific keywords. The data is visualized on an online platform which highlights the interesting keywords colored according to the expressed opinion. VoxCivitas [9] is a highly interactive tool that allows users to navigate within a database composed of videos and tweets about these videos. The application displays tweets gradually as the video runs. TweetTopicExplorer[1] uses "Word Cluster Diagrams" to show the most frequently used words by a tweet user. The words are displayed more or less close to each other depending on how often they occur together in the same tweets. TwitInfo [16] groups and displays tweets for event exploration through the detection and labeling of peaks of words. The Scalable Reasoning System [2] shows trends in data over time. It displays the evolution

[1] http://tweettopicexplorer.neoformix.com

of given items based on the quantity of tweets talking about them. Finally, the online visualization system proposed in [15] allows users to explore the information propagation on the microblogging site. It displays propagation structures, influence scores of individuals, timelines, and geographical information for any user-query terms.

4 The "Cluster" component

This component clusters tweets with very similar content: retweets, tweets containing article titles or maybe "catchy phrases". The method implemented by TweetSurf is an improved online version of the one used in [22].

The method proposed in [22] receives as input a set of tweets that it clusters. It defines the *set of words* of each tweet as the set of all the words in its text, except words prefixed by "RT@", "@" and "#" (along with the prefixes), URLs and common empty words. It also defines the distance between two tweets as the *Jaccard distance*[2] between their sets of words. Based on this distance, a hierarchical clustering is performed. As opposed to this previous approach, our method also takes advantage of the words contained in the URL(s) of the tweet. If a tweet contains the URL of an article, for instance, the last part of the URL is generally very similar to the article title. Indeed, the words in the title are often present in the last part of the URL, being separated by "-". This information can be very useful as many tweets contain only a very short uninformative text in addition to the URL.

Our new algorithm works as follows. It starts with a set of tweets T_C that is empty at first. When each new tweet t is downloaded, its *set of words* is computed as: the set of all the words in its text and in the last part of its URL(s), except words prefixed by "RT@", "@" and "#" (along with the prefixes), URLs and common empty words. Then, the distance between t and the tweets in T_C is computed using the same definition as in [22]. Let t' be the closest tweet from t. If $distance(t, t')$ is smaller than a given threshold, the tweet t is assigned the same cluster as t'. Otherwise, t is put in a new cluster. In either case, t is added to T_C. Thus, our method clusters tweets on the fly, as opposed to the previous approach that needed the entire collection of tweets as input.

In order to be scalable, the method we propose uses a given maximum number of clusters, old data not being taken into consideration. If the maximum number of clusters has been reached and a new cluster needs to be created, the tweets of the "oldest" cluster are deleted from T_C. The "oldest" cluster c is considered to be the cluster that has the following property: for all tweets whose cluster c' is not c, there is a tweet in c' that was published more recently than all the tweets in c. In other words, the oldest cluster is the one that has not had a new tweet for the longest time. Note that the deletion of the oldest cluster and the creation of a new one are done in constant time. For each new tweet, the algorithm has a time complexity of O(number of tweets in the considered clusters), which is limited given that a maximum number of clusters is used.

[2] For two sets A and B, the Jaccard distance is $1 - \frac{|A \cap B|}{|A \cup B|}$.

5 The "Label" component

This component implements a new online method for labeling tweets mentioning an ambiguous company name as "company-related" or "noisy". Basically, the method we propose works as follows. First, it receives as input two lists of words that are strongly related to the company and to the noise respectively. When a new tweet needs to be evaluated, the method first checks if the tweet contains words in one of the two lists. In this case, the tweet is labeled directly as "company-related" or "noisy" respectively. The tweet is also used in order to update a third list of words. If the tweet does not contain words in the two initial lists (or contains words in both initial lists), its label is estimated based on the third list of words. However, it may happen that no clear decision can be made based on the list of words solely. In this case, the method also uses a graph modeling users' previous activity. Mainly, it leverages the following information: Is the previous activity of the author of the tweet related to the company or to the noise? If so, to what extent?

Before presenting the different steps of the new method, let us first recall several basic graph notions.

5.1 Basic graph notions

A directed graph is a pair $G = (V, A)$. The set V represents the set of vertices (or nodes) of the graph. The set A contains ordered pairs of vertices, called arcs or directed edges. An arc $(u, v) \in A$ with $u \in V$ and $v \in V$ is considered to be directed from u to v; u is said to be the source of the arc and v is said to be its destination. Also, v is said to be an outgoing neighbor of u and u is said to be an incoming neighbor of v. The set of outgoing neighbors of the vertex u represents its out-neighborhood (denoted by $N_{out}(u)$). The cardinal of this set represents the out-degree of u (denoted by $d_{out}(u)$). Similarly, the set of incoming neighbors of v represents its in-neighborhood (denoted by $N_{in}(v)$). The cardinal of this set represents the in-degree of v (denoted by $d_{in}(v)$).

5.2 The labeling algorithm

As said before, the algorithm receives as input two lists of words L_C and L_N. These lists are defined such that tweets containing words in the first list are most probably related to the company and vice-versa. For instance, for the case of EDF, the company-related list L_C contains words like "nuclear", "waterpower", "electrician", the name of the CEO of EDF, names of other companies that share some properties with EDF (only if these names are unambiguous) etc. The sport-related list L_N contains names of different sports, players, coaches, acronyms of sport competitions etc. These lists are defined prior to any computation and do not change during the execution of TweetSurf.

Besides the two lists L_C and L_N, the algorithm uses a third list L of words which is empty at the beginning. It also uses a directed graph $G = (V, A)$. At the

beginning the graph is empty except for two vertices: a vertex v_C corresponding to the studied company and a vertex v_N corresponding to the noise.

Remember that when a tweet t is downloaded, TweetSurf first computes the cluster c of t as explained in Section 4. The graph G is then updated: Two vertices are added to the set V if not already present in this set: a vertex v_a corresponding to the author of t and a vertex v_c corresponding to the cluster c of t. Two arcs are added to the set A if not already present: an arc directed from the vertex v_a to the vertex v_c and an arc directed from v_c to v_a. Remember also that TweetSurf only uses a maximum number of clusters. If the oldest cluster must be deleted, the corresponding vertex is deleted from V and all its arcs are deleted from A.

Next, the labeling algorithm computes the label of t. The algorithm first checks if t contains words in L_C but not in L_N. In this case, t receives directly the label "company-related". Also, the words of t, except common empty words and words in L_C, are added to the list L. An arc connecting v_c to v_C is added to the set A if not already present. Equivalent actions are performed if t contains words in L_N but not in L_C. Figure 1 presents an example of the graph G.

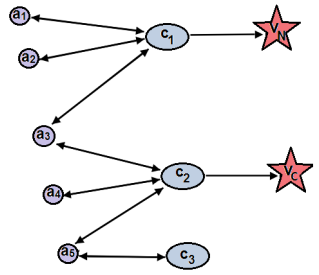


Fig. 1. An example of the directed graph modeling the tweets dataset: vertices a_1 to a_5 correspond to authors of tweets, c_1 to c_3 to clusters and v_N and v_C to the two labels, "noise" and "company".

If t contains words in both lists L_C and L_N or in neither of them, the label of t must be estimated. For that, TweetSurf computes two words-based measures using the list L. If the two measures are close (the absolute difference between them is smaller than a given constant ϵ), TweetSurf also computes two graph-based measures using the graph G. We present the four measures next.

Words-based measures. The two measures are based on a naive Bayesian approach. This approach has been successfully used, for instance, for detecting spam in e-mails [13, 17, 19, 20]. Given the tweet t containing words w_1, w_2, ..., w_m, the probability that t is company-related is

$$p(company|t) = \frac{p(company) \cdot p(t|company)}{p(t)}.$$

The probability that t is noisy is computed in the same way. The two probabilities are then compared. If $p(company|t) > p(noise|t)$, then $\ln p(company|t) > \ln p(noise|t)$ which means that $\ln p(company) + \ln p(t|company) > \ln p(noise) + \ln p(t|noise)$.

Let T be the set of tweets that have been labeled using the two initial lists of words solely[3]. The two probabilities $p(company)$ and $p(noise)$ are simply equal to the ratio between the number of company or noisy tweets respectively in the set T and the total number of tweets in T. For computing $\ln p(t|company)$ and $\ln p(t|noise)$ one can use different formulas (see for instance [17] for definitions and comparisons of five such formulas). We use the *Multinomial Naive Bayes with Boolean attributes* which performs the best on several datasets according to [17] and [20]. Thus, $\ln p(t|company) = \sum_{i=1}^{m} \ln p(w_i|company)$ where $p(w_i|company)$ is estimated using a Laplacian prior as

$$p(w_i|comp.) = \frac{1 + \text{no. occurrences of } w_i \text{ in company tweets in T}}{\text{no. different words in tweets in T} + N_E}$$

and N_E is the total number of words in company-related tweets in T [4].

Let $wWords(company) = \ln p(company|t)$ (i.e. the logarithm of the probability that t is company-related) and $wWords(noise) = \ln p(noise|t)$. The two weights are the words-based measures used by our method. Following the approach used for filtering spam in e-mails, if $wWords(company) > wWords(noise)$, the label should be *company* and vice-versa. However, if the two measures have close values, the estimated label may not be correct. This can happen, for instance, if the text of the tweet is very short or if it contains words that have been equally observed in company-related tweets and in noisy ones. In this case, we choose to use two graph-based measures in order to estimate the label. As we will show in Section 7, this approach greatly improves the accuracy of the labeling.

Graph-based measures. In order to estimate the label of a tweet based on the graph G, we measure the "proximity" between the node v_c representing the cluster of the tweet and the two nodes v_N and v_C. For that, we use an approach inspired by the PageRank [3]. Basically, a weight is broadcast through the graph starting from the node v_c. At the beginning, the node v_c is given the weight of 1 and all the other nodes are given the weight of 0. Then, five steps are performed: at step s, each node $v \in V$ distributes its weight at step $s-1$ to its outgoing neighbors (if it has any) and receives a fraction of the weights at step $s-1$ of its incoming neighbors:

$$w_s(v) = w_{s-1}(v) + \sum_{u \in N_{in}(v)} \frac{weight_{s-1}(u)}{d_{out}(u)} \text{ if } d_{out}(v) = 0$$

[3] We only use the tweets that were labeled directly. We do not use the other tweets, for which the label was estimated, because the estimation may be wrong, which would lead to false computations.

[4] Words present in only a small number of tweets are not used in the computations.

and

$$w_s(v) = \sum_{u \in N_{in}(v)} \frac{weight_{s-1}(u)}{d_{out}(u)} \text{ if } d_{out}(v) > 0$$

(in this second case, the weight of v at step $s-1$ was distributed to the outgoing neighbors). In other words, at each step, the initial weight of 1 "flows" through the arcs of the graph, from the source of the arcs to their destination. Note that all the vertices in the graph give and receive weights at each step, except vertices v_C and v_N that only receive (since they only have incoming edges). The nodes v_N and v_C can be seen as the destination of the initial weight; they do not contribute to the flow of weight. As an example, Figure 2 shows the vertices that have a non-zero weight at each step, supposing that the tweet that needs to be labeled belongs to cluster c_4.

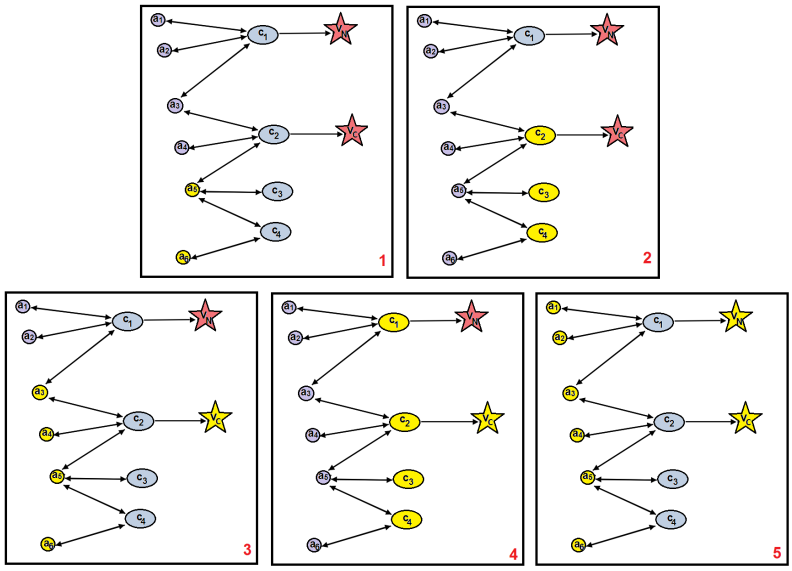


Fig. 2. The broadcast of the initial weight through the graph G. At each step from 1 to 5, the yellow-colored vertices have a non-zero weight.

By performing these steps, the vertices v_C and v_N receive a certain weight if:

- the authors of the evaluated cluster have already written company-related or noisy tweets (the weights of v_C and / or v_N become positive at the end of the third step);

– the authors of the evaluated cluster have already shared clusters with authors of company-related or noisy tweets (the weights of v_C and / or v_N are increased at the end of the fifth step).

We only use five steps because we do not want, when we evaluate the cluster c, to take into consideration authors who are "far" from c. Six steps would not change the weights of v_C and v_N, while seven steps would take into account the activity of authors who share clusters with authors who share clusters with the authors of c, so rather distant authors.

The weights of the vertices v_N and v_C at the end of the five steps represent the two graph-based measures used by TweetSurf: $wGraph(company) = w_5(v_C)$ and $wGraph(noise) = w_5(v_N)$.

5.3 Conclusions on the new algorithm

Let us summarize several properties of the new method we introduced here. First, it is an online method, each tweet being analyzed and labeled on the fly, when it is downloaded. To our knowledge, all the other existing methods need the entire set of tweets in order to perform the labeling.

Second, the method uses words already observed in tweets in order to label new tweets. It thus leverages the tweets style (specific vocabulary, abbreviations, acronyms etc.) which may be very different from the style of formal Web sites like Wikipedia or company official sites.

Third, our method only needs two lists of words that are given as input. By labeling tweets, it builds and updates a new list and a graph that it then uses for labeling new tweets. It is thus capable of adapting to new facts (e.g. new vocabulary related to new events, products etc.) on its own, without any human intervention. As we show in Section 7, the method continues to perform well over time. Note that one could label tweets using the two initial lists of words solely (as proposed in [10,26]). For that, however, the lists would have to be very long. And even then, the lists may not be complete. In our case, for instance, it would be impossible to define a complete list of words related to sport (it would mean listing the names of all sports, players, coaches, teams etc.). Also, the labeling may become less accurate over time if Twitter users employ new vocabulary (when talking, for instance, of new events, products, facts related to the company or to the noise) and the lists are not updated.

Fourth, the algorithm we propose leverages users activity with respect to the two classes (company or noise). This is very useful when the textual contents of the evaluated tweets are not very informative (e.g. very short text, vocabulary often used with both classes etc.). As we show in Section 7, this feature improves the results considerably.

6 The "Display" component

To visualize and analyze the downloaded tweets, we have developed an user interface that allows users to navigate through the set of tweets and the corresponding metadata (e.g. publication date, label, cluster, author etc.). This

interface is an interactive application based on Open Source components such as Apache Lucene (http://lucene.apache.org), a full-featured text search engine library, and JfreeChart (http://www.jfree.org/jfreechart/), a free library used for plotting curves and histograms.

An analyst can query the interface, retrieve tweets containing one or more keywords and view them in the interface. Users can also select subsets of tweets based on successive queries. The software displays the selected tweets in the form of lists, "tag clouds", histograms or flows, as explained in the following section.

6.1 The visualization algorithm

Among the existing visualization techniques, StoryFlow [18] seems particularly well adapted to our context, because it can represent the evolution of items through time and emphasize the fluctuation of items prominence. The displayed items can be words, hash-tags, clusters, authors, mainly anything that is observed at several time points. Our flow visualization system is a modified version of SRS [2] which is based on StoryFlow. As items may disappear and reappear through time (i.e. they are not observed during a given period of time, then they are seen again), we modified the SRS algorithm in the following way:

- For each period (e.g. a week W), we compute the list of items and their score (i.e. their number of occurrences);
- We order items in descending order of their score and display them by placing the one with the highest score at the top of the screen;
- If an item has already been encountered in the past, it is assigned the same color as before, if not, it is assigned a new color;
- If the item does not appear in the given period but appears in following periods (as A in week W in Figure 3), it is assigned the smallest size and is displayed under the x-axis, which helps to maintain a visual continuity. This feature does not exist in the SRS algorithm.
- Finally, same items are connected through Bezier curves in order to produce a smooth rendering.

By varying "Bar Width" and "Bar Space", we observe different phenomena. A low "Bar Space" highlights item scores (as in Figure 4 (up)), while high "Bar Space" and low "Bar Width" highlight items evolution (as in Figure 4 (bottom)).

7 Results

Our system began working mid-June 2013. We present here the results obtained from mid-June 2013 to mid-February 2014. During this period, the system collected approximatively $136,000$ tweets containing the keyword "EDF". For each downloaded tweet, the system computed the corresponding cluster as explained in Section 4. In order to choose the value of the distance threshold, we performed several tests on small random subsets of tweets. The best value seemed to be 0.5, which imposes that there is a tweet t' in the cluster such that at least half of the words of t are present in t' and vice-versa. The system worked with the maximum number of clusters limited to $5,000$.

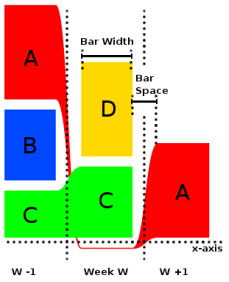


Fig. 3. An example of items visualization based on StoryFlow.

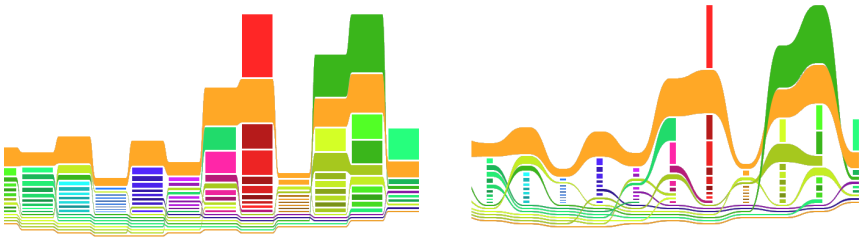


Fig. 4. An example of visualization of the same items with: low "Bar Space" (left) and low "Bar Width" (right).

7.1 The "Label" component

Out of the 136,000 downloaded tweets, approximatively 86,000 were labeled using the initial lists L_C and L_N solely. For the other tweets, the system estimated a label using two words-based measures and two graph-based ones. Normally, the system computes the two graph-based measures only if the values of the words-based ones are close: the absolute value of $\Delta Words = wWords(company) - wWords(noise)$ is smaller than a given value ε. For evaluation purposes, we computed the two graph-based measures for all tweets that needed to be labeled.

If one used the words-based measures solely, the estimated label would be "company- related" if $wWords(company) > wWords(noise)$ and vice-versa; this is the words' "point of view". Similarly, if one used the graph-based measures solely, the estimated label would be "company-related" if $wGraph(company) > wGraph(noise)$ and vice-versa; this is the graph's "point of view". As explained before, we made the following hypothesis: if the absolute value of $\Delta Words$ is high, then the words' point of view is most probably correct. However, if the difference is low, the words-based estimation is not very accurate. This may happen if the text of the tweet is very short or if it contains words that were

observed in both company-related tweets and noisy tweets. In this case, the use of the graph-based measures should give better results.

In order to test this hypothesis, we manually evaluated tweets for different absolute values of $\Delta Words$. First, we read the tweets for which there was a disagreement between the two points of view. As shown in Table 1, the fraction of tweets for which there is a disagreement gets smaller and smaller when the absolute value of $\Delta Words$ increases. This table also shows that, for tweets for which there is a disagreement between the two types of measures, the words' point of view is right in most cases for $|\Delta Words| \geqslant 3$ and the graph's point of view is right in most cases when $|\Delta Words| < 3$.

Table 1. For different values of $|\Delta Words|$, the percentage of tweets for which the words-based measures and the graph-based measures agree and disagree. Among the tweets for which they disagree, the percentage of tweets for which the words-based measures are right.

$\lvert\Delta Words\rvert$	**% agreement**	**% disagreement**	**% correct words-based among disagreement**
$(0, 1)$	36.8	63.2	13.2
$[1, 2)$	74.7	25.3	22.3
$[2, 3)$	83.5	16.4	38
$[3, 4)$	82.8	17.2	54
$[4, 5)$	88.6	11.4	77.7
$[5, 6)$	90.8	9.2	85
$[6, 7)$	93	7	86.3
$[7, 8)$	93.8	6.2	90
$[8, 9)$	91.5	8.5	93.3
$[9, 10)$	94.7	5.3	100
$[10, \infty)$	95.8	4.2	99.8

For tweets having $|\Delta Words| < 3$, we also read all tweets for which there was an agreement between the two points of view. Table 2 shows the fraction of correctly labeled tweets when the words' point of view and the graph's point of view are used respectively. Note that the sum of the percentages on one line is not equal to 100 because there are tweets for which the two types of measures agree. One can see that the use of the graph-based measures improves the results considerably.

We thus decided to use the value $\varepsilon = 3$ as input for the "Label" component. We tried next to evaluate the global accuracy of the labeling algorithm. Given the high number of tweets, an exact measure of the accuracy is difficult to obtain. In order to have an estimation, we read one hundred randomly-picked tweets for each weak from July 2013 to January 2014 included. Figure 5 shows the fraction of correctly labeled tweets for the different weeks. One can see that the accuracy is high (in comparison, the algorithm presented in [26] obtained a 83% accuracy) and quite stable over time. Thus, TweetSurf continues to perform well several months after it got going.

Table 2. For different values of $|\Delta Words| < 3$, the percentage of tweets correctly labeled using the words-based measures and the graph-based measures respectively.

$\lvert\Delta Words\rvert$	% correct labels words-based	% correct labels graph-based
$(0, 1)$	41.7	88.2
$[1, 2)$	73.3	87.4
$[2, 3)$	83.7	87.9

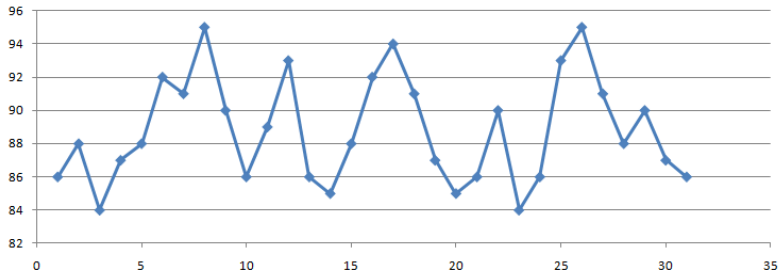


Fig. 5. The estimated accuracy of the labeling: for each week from July 2013 to January 2014 included, the number of correctly labeled tweets out of one hundred randomly-picked tweets.

7.2 The "Display" component

The visualization interface we developed allows users to easily navigate through the tweets corpus. One can have a quick overview of the data, compute statistics and extract subsets for further analyses. The visualization of trends shows how different tweets characteristics evolve. For us, the interface was very useful when evaluating the labeling results. For instance, using the different statistics computed by the interface (e.g. word tags, flows, histograms) we were able to quickly see if, in general, tweets were correctly labeled.

The interface displaying the EDF-related tweets is currently used by EDF employees from several departments such as, for instance, sociologists and text analysts from the R&D department, text analysts from the commercial entity, persons in charge of communication etc. They all find the interface user-friendly, easy to become familiar with, easy to use and extremely convenient for reading tweets. The ability to select tweets based on metadata and to visualize trends is particularly appreciated.

8 Conclusions and future work

In this paper we presented TweetSurf, a system designed for continually collecting and displaying tweets talking about a given company. One of the most important functions of TweetSurf is the identification of noisy tweets (i.e. tweets

not talking about the company) when the company's name is ambiguous. In order to accomplish this task, TweetSurf implements a new algorithm we propose. When tested on the EDF case, the algorithm gives very satisfying results thanks to the use of both tweets contents and a graph modeling the dataset. We intend to evaluate the algorithm on other use cases too, like "Apple" or organization names that have more than two meanings. We will also try to improve the algorithm by using more information extracted from the tweets contents, such as the presence of URLs or hash-tags (since they represent explicit labels chosen by the authors themselves). The use of more social features (authors' influence, users' similarity) may improve the results and help to rank tweets.

As for the visualization interface, we would like to compute and display topics of tweets and their time evolution. This would allow users to easily see "what tweets are about" before performing a deeper analysis. The already computed clusters of tweets could be useful for this task as they reduce the quantity of tweets to process and help avoid redundancy. Users feedback also suggested improving the interface by adding more functions. For instance, users find that it would be interesting to have more statistics on authors' activity and on clusters (clusters with the greatest number of tweets, clusters that last the longest etc.).

References

1. Weps-3 evaluation campaign: Overview of the web people search clustering and attribute extraction tasks. In: Braschler, M., Harman, D., Pianta, E. (eds.) CLEF (2010)
2. Best, D.M., Bruce, J., Dowson, S., Love, O., McGrath, L.: Web-based visual analytics for social media. In: AAAI Technical Report WS-12-03 Social Media Visualization (2012)
3. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems 30(1-7), 107–117 (Apr 1998)
4. Bunescu, R., Pasca, M.: Using encyclopedic knowledge for named entity disambiguation. In: Proceedings of EACL. vol. 6, pp. 9–16 (2006)
5. Byron, L., Wattenberg, M.: Stacked Graphs - Geometry & Aesthetics. IEEE Trans. Vis. Comput. Graph. 14(6), 1245–1252 (2008)
6. de Carvalho, M.G., Gonçalves, M.A., Laender, A.H.F., da Silva, A.S.: Learning to deduplicate. In: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries. pp. 41–50. JCDL '06, ACM, New York, NY, USA (2006)
7. Cucerzan, S.: Large-scale named entity disambiguation based on wikipedia data. In: Proceedings of EMNLP-CoNLL. vol. 2007, pp. 708–716 (2007)
8. Cui, W., Liu, S., Tan, L., Shi, C., Song, Y., Gao, Z., Qu, H., Tong, X.: Textflow: Towards better understanding of evolving topics in text. IEEE Trans. Vis. Comput. Graph. 17(12), 2412–2421 (2011)
9. Diakopoulos, N., Naaman, M., Kivran-Swaine, F.: Diamonds in the rough: Social media visual analytics for journalistic inquiry. In: IEEE VAST. pp. 115–122. IEEE (2010)
10. Garcia-Cumbreras, M.A., Garcia-Vega, M., Martinez-Santiago, F., Perea-Ortega, J.M.: Sinai at weps-3: Online reputation management. In: CLEF (2010)
11. Havre, S., Hetzler, B., Nowell, L.: ThemeRiver: Visualizing Theme Changes over Time. In: INFOVIS. p. 115. IEEE Computer Society (2000)

12. Heer, J., Card, S.K., Landay, J.A.: Prefuse: a toolkit for interactive information visualization. In: Proc. CHI 2005, Human Factors in Computing Systems (2005)
13. Hovold, J.: Naive bayes spam filtering using word-position-based attributes. In: Proceedings of the Second Conference on Email and Anti-Spam (2005)
14. Kaye, J., Lillie, A., Jagdish, D., Walkup, J., Parada, R., Mori, K.: Nokia internet pulse: a long term deployment and iteration of a twitter visualization. In: CHI Extended Abstracts. pp. 829–844. ACM (2012)
15. Li, C.T., Kuo, T.T., Ho, C.T., Hung, S.C., Lin, W.S., Lin, S.D.: Modeling and evaluating information propagation in a microblogging social network. Social Network Analysis and Mining 3(3), 341–357 (2013)
16. Marcus, A., Bernstein, M., Badar, O., Karger, D., Madden, S., Miller, R.: TwitInfo: Aggregating and visualizing microblogs for event exploration. In: Proceedings of the 2011 annual conference on Human factors in computing systems. pp. 227–236. ACM (2011)
17. Mets, V., Androutsopoulo, I., Palioura, G.: Spam Filtering with Naive Bayes - Which Naive Bayes? In: CEAS (2006)
18. Rose, S.J., Butner, S., Cowley, W., Gregory, M.L., Walker, J.: Describing story evolution from dynamic information streams. In: IEEE VAST. pp. 99–106. IEEE (2009)
19. Schneider, K.M.: A comparison of event models for naive bayes anti-spam e-mail filtering. In: EACL. pp. 307–314. The Association for Computer Linguistics (2003)
20. Schneider, K.M.: On Word Frequency Information and Negative Evidence in Naive Bayes Text Classification. In: EsTAL. Lecture Notes in Computer Science, vol. 3230, pp. 474–486. Springer (2004)
21. Stasko, J.T., Gorg, C., Liu, Z., Singhal, K.: Jigsaw: Supporting investigative analysis through interactive visualization. In: IEEE VAST. pp. 131–138. IEEE (2007)
22. Stoica, A.: Filtering Noisy Web Data by Identifying and Leveraging Users' Contributions. In: ICWSM (2012)
23. Viegas, F.B., Wattenberg, M., van Ham, F., Kriss, J., Mckeon, M.: Manyeyes: a site for visualization at internet scale. IEEE Trans. Vis. Comput. Graph. 13(6), 1121–1128 (2007)
24. Vuillemot, R., Clement, T., Plaisant, C., Kumar, A.: What's being said near "Martha"? Exploring name entities in literary text collections. In: IEEE VAST. pp. 107–114. IEEE (2009)
25. Wang, C., Chakrabarti, K., Cheng, T., Chaudhuri, S.: Targeted disambiguation of ad-hoc, homogeneous sets of named entities. In: Proc. of the 21st Int. World Wide Web Conference (WWW 2012), Lyon (France) (2012)
26. Yerva, S.R., Miklos, Z., Aberer, K.: It was easy, when apples and blackberries were only fruits. In: Braschler, M., Harman, D., Pianta, E. (eds.) CLEF (2010)
27. Yerva, S.R., Miklós, Z., Aberer, K.: Entity-based Classification of Twitter Messages. International Journal of Computer Science and Applications 9(1), 88–115 (2012)
28. Yoshida, M., Matsushima, S., Ono, S., Sato, I., Nakagawa, H.: ITC-UT: Tweet Categorization by Query Categorization for On-line Reputation Management. In: Braschler, M., Harman, D., Pianta, E. (eds.) CLEF (2010)

On Spatial Measures for Geotagged Social Media Contents

Xin Wang, Tristan Gaugel, and Matthias Keller

Steinbuch Centre for Computing and Institute of Telematics
Karlsruhe Institute of Technology
76131 Karlsruhe, Germany
uidhw@student.kit.edu,{tristan.gaugel,matthias.keller}@kit.edu

Abstract. Recently, geotagged social media contents became increasingly available to researchers and were subject to more and more studies. Different spatial measures such as Focus, Entropy and Spread have been applied to describe geospatial characteristics of social media contents. In this paper, we draw the attention to the fact that these popular measures do not necessarily show the geographic relevance or dependence of social content, but mix up geographic relevance, the distribution of the user population, and sample size. Therefore, results based on these measures cannot be interpreted as geographic effects alone. We show potential misinterpretations and propose normalized measures that show less dependency on user population.

1 Introduction

Although today's technical infrastructure allows communication with almost anybody, anywhere in the world, geographic location plays a role in the selection of communication partners and topics. Geospatial aspects of online communication have attracted more and more interest from the research community: Which role does location play in the selection of online friends (e.g. [10])? How strong is the local focus of online social media contents and which geospatial spreading patterns can be observed (e.g., [2, 4])? These and other questions have been examined to gain insights into the mechanisms of social media usage and content dispersion. Social media have increasing impact on the spreading of news [9], but in contrast to traditional newswire, the mechanisms of distribution are complex and rarely understood yet.

In this paper, we demonstrate that popular measures, such as Spread, Entropy, and Focus do not necessarily only show the geographic relevance or dependence of social content, but are severely influenced by the distribution of the user population, and sample size. Therefore, results based on these measures cannot be interpreted as geographic effects alone. Furthermore, we propose normalized measures that show less dependency on user population. For the remainder of this paper, we assume that we are dealing with geographic coordinates but abstract from the way they have been retrieved. A set of coordinates can represent, for instance, the occurrences of a hashtag. For readability, we will call such a

set of coordinates the occurrences of a meme, although it could represent, for example, the locations of the friends of a Facebook user as well. Many studies are not based on the analysis of a single meme but on the analysis of a large set of memes, so that all individual sets of meme occurrences cannot be visualized or manually inspected. Hence, measures are required that capture geographic properties of the memes, e.g. to find the hashtags that have the strongest local concentration.

Based on such kind of measures, recent studies revealed surprising patterns of geo-spatial distribution: For instance, Kamath et al. [4] found in a study of geotagged tweets that hashtags seem to be most concentrated at a single location at their peak time, reaching the point of lowest concentration in average after 20 minutes, to slowly return to a single location afterwards. In an analysis of YouTube videos [2] a very similar distribution pattern was observed, called spray-and-diffuse-pattern by the authors. Kamath et al. [4] suggest that this is a fundamental pattern of social media spread. Can we conclude that the users' interest in a meme is traveling, e.g. that at the peak time, a meme is relevant only to users in a single location while it gets increasingly interesting for distant users within the next 20 minutes?

In order to answer such questions, one should make sure that the employed measures are able to capture these aspects and are not influenced by other effects, such as, e.g., the spatial distribution of the user population. This is particularly challenging, since we need to consider that a meme occurrence at coordinates (x, y) is included in the data set if a) a user was present at position (x, y) b) the meme was shared by the user because it is relevant to him and c) location information for this meme is available. Studying geospatial properties of memes is based on the hypothesis that the relevance of a meme to a user depends on his location (x, y). In this paper, we consider the location-based relevance of a meme as the probability that a user contributes to the meme in dependence of his location (x, y). The location-based relevance represents the user behavior. Hence, if we are interested in drawing conclusions about the user behavior, we should focus on the location-based relevance (b) and try to abstract from the user population distribution (a) and the sample size (c).

We assume a basic, circular model for the location-based relevance: only users within a circular area of radius r contribute to a specific meme with a probability $p > 0$. If the radius r is small, the users' interest is concentrated and if it is large, the user's interest in the meme is more widespread. This model is surely a simplification, but it allows to examine how well traditional geospatial measures reflect the location-based relevance and to develop more suitable measures for scenarios with focus on location-based relevance.

The relationship between the concentration of the location-based relevance represented by the radius r and the traditional measures is often counterintuitive as we exemplify with Figure 1: A traditional measure applied in geographic statistics that was also used to analyze social media data (e.g. [4]) is the mean spread, the average distance of samples to their geographic mean point. Mean spread is interpreted as measure for the geographic dispersion of a set of meme

occurrences. In Figure 1-A, we assume that a specific topic, identified by a particular meme, has a strong local focus in central France. We modeled this by choosing a center point, and assume that $r = 150km$. Furthermore, we assume that 50 geotagged occurrences of memes (e.g. with the same particular hashtag) appear within the focal radius. To model the distribution of user population, we collected real geotagged Tweets from this area and selected a random subset of 50 samples. Based on this sample, we calculated a mean spread of 90.6 km. Now, we assume that the meme gets interesting to users in a larger area and set $r = 250km$, generate a sample in exactly the same way and calculate the mean spread again (Figure 1-B). However, the measured mean spread does not increase but drop to 66.2 km. This is caused by the fact that Paris is now within the circular area and the topic has spread to an area in which the population is much more concentrated.

In this paper, we analyze the impact of the users' distribution and the potentially limited number of available samples on the way in which the measures reflect the location-based relevance of memes. The structure and the contributions of this paper are as follows: In Section 2, we review related work and introduce three common measures Focus, Spread and Entropy. Section 3 then formalizes the problems of the influence of i) the user distribution and ii) too small sample sizes in order to specify corresponding requirements for the measures. Furthermore, we introduce our methodology of assessing the impact of these problems in realistic scenarios with Monte Carlo simulations. In Section 4, we discuss the simulation results and show that both, the user distribution and the sample size severely limit the meaningfulness of geospatial measures with respect to location-based relevance and show that a relatively high number of samples is necessary to achieve indicative results. In Section 5, we propose and evaluate modified measures that are more robust against population density effects. Furthermore, we explain, why it is not possible to correct for systematic errors of Focus and Entropy caused by a varying numbers of samples. Section 6 concludes the paper and we summarize the implications for existing studies and the application of the measures in the future.

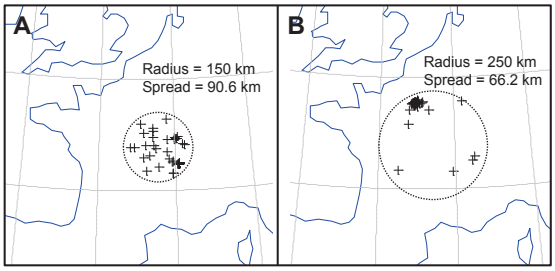


Fig. 1. Influence of user population on spatial measures: Although the focus area is larger in plot B, a smaller Spread is measured.

2 Related Work

In Section 2.1, we discuss related work. In Section 2.2 we specify the three common spatial measures Focus, Entropy and Spread formally.

2.1 Geospatial Analysis of Social Media Contents

Due to the recent increasing interest in analyzing geospatial properties of social media, plenty of measures have been employed by the research community in order to quantify different aspects of geo-spatiality. The variety of measures also emerges from the fact that the underlying data varies in its characteristics. While for some works [4, 7] the analyzed geospatial data originates from Twitter, other studies employed similar analyses for YouTube videos [2] or web resources [3]. The specific characteristics required slightly varying definitions for similar measures. The three measures which are considered and analyzed in this work, namely the Spread, Entropy, and Focus have all been used in a slightly adjusted way in several works before.

Spread is used as a measure for uniformity and in general states the mean distance over all geo-locations to the geographical midpoint. It is for example being used in [4] for an analysis of the dispersion of Twitter users or in [3] for analyzing geographical scopes of web resources. Entropy is generally used in accordance with its information theoretical meaning of stating the bits required to represent the spatial spread. The granularity of the spatial resolution as well as the actual spatial distribution of, e.g., the users or tweets thus plays an important factor in determining the entropy. A high value is seen as an indication for a more uniform spread of the data under consideration. Entropy is hereby employed by [4] and [2] as an indication for the randomness in spatial distribution for tweets and YouTube video views, respectively. Focus is used as a measure for the centricity of tweets [4] and YouTube video views [2] by calculating the maximum proportion of occurrences of geo-locations falling into any individual area.

Additionally, there are other works that focus less on calculating a single measure for representing the spatiality, but represent it, e.g., by means of a CDF of the geographic distance between users of an social network [10]. In [7] again a density measure is used to visualize areas with high or low occurrences of tweets. Then again other work primarily focuses on identifying the center point and the corresponding dispersion for information that is spatially distributed, e.g. search queries [1] or web resources [3]. Although these studies are dealing with a similar topic of quantifying spatiality, their goal is not to represent this by means of one single measure.

2.2 Spatial Measures

In this paper, we analyze the measures Focus, Entropy and Spread, which were used in previous work. Focus and Entropy are zone-based measures that require the globe to be divided into a set R of zones (e.g., countries [2] or identical squares of 10 km by 10 km [4]). If the set O denotes the set of all meme occurrences and

$O_i \in O$ denotes the occurrences in zone $i \in R$, the relative frequency P_i of zone i is given by:

$$P_i = \frac{|O_i|}{|O|} \tag{1}$$

Focus measures whether there is a single zone in which the meme occurrences are concentrated. Focus is defined as the maximum P_i for any region i. Then, i is called focus location. As a consequence, the higher the focus value is, the more occurrences appear in the focus location and the fewer in other regions.

$$Focus, F = \max_{i \in R} P_i \tag{2}$$

Entropy measures whether the occurrences are concentrated in only a few zones (low entropy) or whether the samples are more equally distributed over a large amount of zones (high entropy).

$$Entropy, E = -\sum_{i \in R} P_i log_2 P_i \tag{3}$$

Spread measures how close the occurrences are located to each other, without the usage of a zone system. Spread is defined as the mean distance of all meme occurrences to their geographical mean point. Higher values of Spread correspond to a larger spatial coverage of the distribution. With $D(c_1, c_2)$ being the geographic distance between two points c_1 and c_2, Spread is defined as:

$$Spread, S = \frac{\sum_{o \in O} D(o, MeanPoint)}{|O|} \tag{4}$$

where the $MeanPoint$ equals the geometric median, the location with the smallest possible average distance to all $o \in O$.

3 Methodology

In this section, we first formalize the requirements for spatial measures that reflect the location-based relevance (Section 3.1). In order to analyze the impact of a non-uniform user population, we collected an extensive amount of geo-tagged tweets from Twitter which we are using as an approximate of the user population U (cf. Section 3.2). Considering the impact of this real-world data set would be hard to realize with an analytical approach, we conduct Monte Carlo simulations (Section 3.3). Since both, Focus and Entropy are zone-based measures, we describe challenges related to partitioning in Section 3.4.

3.1 Formalization of Requirements

By location-based relevance of a meme, we mean the probability that a user publishes a meme in a certain time span in dependency of the user location.

We assume that this probability can be quantified by $f(x, y)$, expressing the probability for a user to contribute to a meme, if located at (x, y). Consequently, $f(x, y)$ models the geographic preference of a meme. While we assume that such a geographic preference function exists, we must keep in mind that we can only observe it indirectly trough a limited number of samples in practice.

Without limiting the generality of the model, we abstract from user mobility. Hence, the user population (the set of potential contributors, e.g. the members of a social network) can be modeled as a set of coordinates $U = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, (x_z, y_z)\}$, reflecting the locations of the users in the time span of the analysis. The set of meme occurrences $C \subseteq U$ can be considered as outcome of a random variable gained from evaluating $f(x, y)$ for all users U.

An important consideration is that in most analyzed scenarios, it is very unlikely that we have access to the entire set C of meme occurrences and the entire set U of user locations. For instance, the locations of all Twitter users are not available and not all occurrences of a hashtag are geotagged. Hence, we should assume that only a subset $C_n \subset C$ is available for analysis that reflects the communication behavior of only $n \leq z$ users. We denote the true Entropy, Focus and Spread based on the entire (but not observable) set of occurrences C as E^C, F^C and S^C respectively. The true frequency count in the zone $i \in R$ will be denoted as P_i^C. We can now formulate two basic requirements:

- **Requirement P1:** The true but not observable measures E^C, F^C and S^C should reflect characteristics of the location-based relevance function $f(x, y)$ and should not be biased too strongly by the distribution of the users U.
- **Requirement P2:** The observable measures E, F and S should not deviate substantially from E^C, F^C and S^C.

It is obvious that both requirements are not fulfilled entirely by the measurements under examination in a sense that they are unbiased by the distribution of the population and the selection of samples. However, the strength of these biases has not been analyzed before and therefore it is also not known to which degree the considered measurements are appropriate for analyzing the location-based relevance of geotagged social media contents at all.

3.2 Sampling Dataset

Geographical data of a large set of randomly collected tweets are considered as a reasonable estimate of the varying density of the user population across different regions. We collected a corpus of over 4 million geo-tagged tweets between 21th of June 2013 and 5th of August 2013. We only considered a single tweet per user (the first one in our data set) since our aim is to model the user density and not the tweet density. This resulted in a set of 2,620,058 coordinates, which is denoted as real data set in the remainder of this paper. Additionally, we generated a reference data set of about the same number of random coordinates which are uniformly distributed on the earth, which we refer to as baseline data set.

3.3 Simulation of Spatial Distributions

To assess the measures with respect to the requirements, we generate sample sets of meme occurrences based on the circular model using Monte Carlo simulations. To apply the circular model with radius r, first a random center point Cp is chosen. It is assumed that $f(x, y) = k$, with k being a constant, for all (x, y) with $D((x, y), Cp) < r$ and $f(x, y) = 0$ otherwise. Instead of defining k, we specify the number of samples n and generate meme occurrences in the following way: 1) A random point from the sampling data set is drawn and considered as center point Cp; 2) n points with $D((x, y), Cp) < r$ are randomly drawn from the sampling data set (Figure 2).

In a next step (Section 4), we then evaluated how well Focus, Entropy and Spread reflect the concentration of the location-based relevance represented by the radius r. Independent from the location of Cp, larger radii should be indicated by a higher Entropy, a lower Focus and a higher Spread. These relationships should not be lost due to the influence of the user population and the sample size since, otherwise, the measures would not be able to reflect the location-based relevance even in the most basic scenario (circular model).

Keep in mind, that due to effects of real geography, it could be possible that the drawn Cp is located in an isolated user population cluster, i.e. on an island. In this case, the samples will be always strongly concentrated in a single location, even though the radius r is increased. In such a case, increasing radii cannot be captured by spatial measurements at all and it would not be an appropriate scenario to assess the quality of spatial measurements. To avoid such situations, criteria were defined to remove these outliers.

3.4 Region Partition

In order to measure Focus and Entropy, we apply Leopardi's algorithm to divide the globe surface into a certain number of tiles with equal area as unit regions [6]. However, the size and boundaries of unit regions will affect the frequency counts and hence the results of the measurements, also known as modifiable areal unit problem (MAUP) in spatial analysis [8]. So it is worth noting the choice of a proper scale of unit. When studying real social media contents, this choice usually

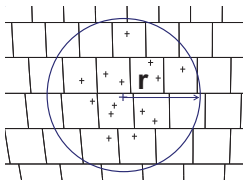


Fig. 2. The distribution area of a meme is modeled as circular area with radius r. The figure also shows the zonal partitioning according to Leopardi's algorithm (cf. Section 3.4)

depends on the intended level of analysis and interpretation, i.e., whether one is studying the distribution across countries, cities, or postcode districts, etc. Since the main interest of our tests is not concrete values but the variation and stability of measuring results, the analysis is less sensitive to the scale of unit regions. Yet, we still need to choose a scale so that unit regions are neither too big (bigger than the area of distribution) nor too small (too few points in each region). In this study, we set the number of unit regions to 10000, with an area of about $50000km^2$ for each region, similar to a median size state in the U.S.

The position of a center point Cp in relation to the zone boundaries may influence spatial measurements and hence, observed variances among generated sets of meme occurrences with a fixed radius r might be caused by the underlying zone instead of the user population density. We use the baseline data set for being able to isolate zone effect in our simulations.

4 Analysis of Existing Measures

We will discuss and analyze Requirement P1 in Section 4.1 and Requirement P2 in Section 4.2

4.1 P1 - Dependency on User Distribution

Simulation Setting. In the basic circular model, the shape of $f(x, y)$ is specified by the center point Cp and the radius r. If the measures are not dependent on the underlying distribution of the user population, the measured values should be influenced only by r and not by the location of Cp. Interpreted as measure for location-based relevance, Focus should drop with increasing r, since it measures the tendency of concentration in a single area. Entropy should increase with increasing r, since it measures the tendency of dispersion in multiple areas. Spread should increase as well, since it measures the spatial coverage of the meme. If these correlations are low, the measures are not appropriate to draw conclusions about the location-based relevance, even in the most basic scenario (the circular model).

Hence, we generated a series of sets of meme occurrences as described in Section 3.2 starting with a radius of $r = 150km$ and increasing the radius in 50km steps up to $r = 1150$. For each of the 20 different radius values, we conducted 30 Monte Carlo runs, each time selecting a different random center point Cp. With 500 samples per iteration we chose a number that is large enough to avoid sample size effects (P2) but still allowed us to conduct the series of simulations in a reasonable time.

Beside the distribution of the population, the MAUP could also have strong impact on the comparability of two sets of meme occurrences with different center points. To isolate this effect, we generated a second series in which we draw the user locations from the uniformly distributed baseline data set.

Results. The plots shown in Figure 3 summarize the results and can be interpreted as follows: Each dot represents an individual set of meme occurrences consisting of 500 samples. The x-axes represent the measured values of Focus (first row), Entropy (second row) and Spread (third row). The y-axes are given by the corresponding radii r of the set of meme occurrences from which the values result. The plots in the first column result from the uniform baseline user distribution. In this idealized scenario, each measured value allows us to draw relatively precise conclusions about the radius r, because at each location on the x-axes, the values on the y-axes are spread only within small intervals. The plots in the second column show that the population density has a strong effect on the measures which masks the influence of r. For example, the results indicate that if we measure a Focus of 0.4, it could result from memes with strong local concentration (with a radius of 200km) as well as from memes that are spread within a much wider area (with a radius above 1000km) if we take a realistic setting into account. The experiments allow to draw the following conclusions:

- In the baseline scenario, the measured values correlate well with the radii r, with not much differences between the zone-based measures and Spread. Hence, effects caused by MAUP (i.e., whether the center point Cp is close to a zone center or a zone border) seem to be insignificant in the considered parameter range.
- In realistic scenarios, all three measurements are strongly influenced by the distribution of the user population. If we apply the measurements to compare two sets of meme occurrences, differences in the values are only indicative if they are considerably large. While, e.g., a Focus value of 0.9 indicates indeed a higher concentration (i.e., a smaller r) than a Focus value of 0.1, the measures seem not to be suitable for, e.g., a precise ranking.
- The experiment indicates that Spread seems to be most robust against effects of the user population density, while Focus seems to be most prone to biases.
- We also analyzed the mean values of the sets of meme occurrences for each radius r and the confidence intervals. Interestingly, we found no indications for systematic errors, i.e., that larger radii caused the average Focus to increase or the average Spread or Entropy to drop (cf. example in Section 1). Consequently, studies based on averaging over a large number of memes should not be severely biased by the distribution of user population.

4.2 Influence of Sample Size

Simulation Setting. With respect to the influence of the sample size, we want the measure to not become unstable and unreliable for a small amount of samples. In other words, smaller numbers of samples n should not cause significant deviations of the measures if the other parameters are fixed. Hence, we generate multiple set of meme occurrences with fixed r and varying n to analyze the stability of the measurements.

The examination of the effect of different sample sizes is especially meaningful for measures such as Focus and Entropy, because according to their definitions,

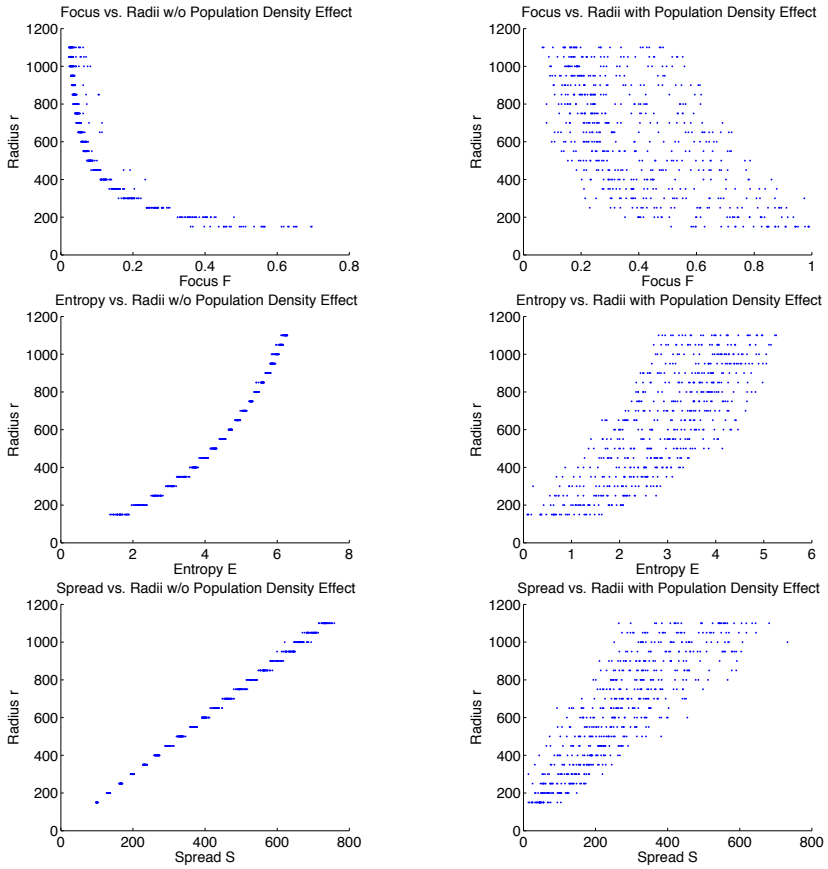


Fig. 3. Under the assumption of a uniformly distributed user population, the measured values and the radii of the distribution area correlate well (left column). Consideration of a realistic user distribution adds strong noise (right column) and the measured values do not reflect geographic preferences accurately.

the minimum value of Focus ($1/n$) and the maximum value of Entropy ($log_2 n$) are both dependent on the total number of data points.

To exclude the influence of population density, we conducted sample size tests based on the uniform baseline data set. We chose three different radii: 150 km, 1000 km and 2000 km. For each radius, we generated sets of meme occurrences ranging from 5 samples to 295 samples in steps of 10. In each step, we generated 30 sets of memes and computed the average Focus, Entropy and Spread respectively.

Results. In Figure 4 the results of our evaluation are depicted, which reveal strong systematic errors for Focus and Entropy. Mean Focus increases and mean Entropy decreases with the drop of sample size. These trends are particularly significant when the sample size is small (under 50 in our tests). Values of Spread are relatively stable, but a trend of increase can still be observed when the sample size is smaller than 50. The experiment indicates:

- If applied to small numbers of samples, there are strong tendencies of the measured Focus to overestimate the true Focus F^C. The results indicate that at least 50 or better 100 samples are required. Still, large differences in the number of samples can limit the comparability of two sets of meme occurrences.
- Even if more than 100 samples are used, the measures are biased by the sample size, which is especially a problem when Entropy is used and the number of samples in the individual set of meme occurrences differ strongly (e.g., 100 samples vs. 1000 samples). In such cases, an upper bound for the number of samples should be defined and if this number is exceeded only a subset of the samples should be used.
- If more than approximately 25 samples are used, Spread seems not to be significantly biased by the number of samples. However, the variances (not shown in the plots) of the individual sets of meme occurrences are much higher if 25 samples are used instead of 100 or more.

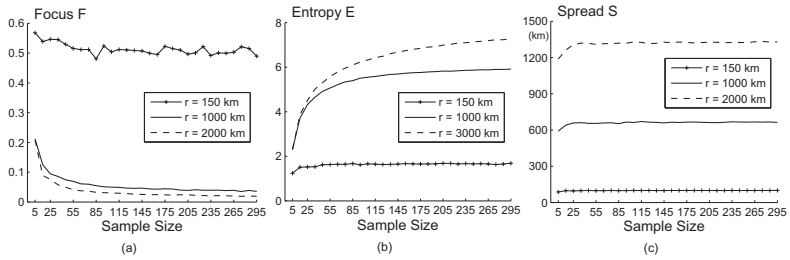


Fig. 4. (a) Focus, (b) Entropy and (c) Spread for sets of meme occurrences created with the baseline data set with various sample sizes, radii $r = 150km$, $r = 1000km$ and $r = 2000km$

5 Reducing Biases

In Section 5.1, we propose adjusted measures that are more robust against effects of the user population. In Section 5.2, we discuss the difficulties of adjusting for the systematic errors caused by the sample size.

5.1 Requirement P1 - Improved Measures

Focus and Entropy. The number of occurrences $|O_i|$ in zone i depends on both, the number of users N_i in the zone i and the shape of $f(x, y)$ in zone i. The aim is to eliminate or at least to decrease the influence of N_i for a more accurate measurement of the location-based relevance. We first consider the ideal case in which the same number of users N is located in each zone. For now, we do not take the sample size problem into account and assume that all user locations and meme occurrences are known. Since the zone-based measures only capture how the samples are distributed among the zones and not the distribution within the zones, we furthermore assume that the probability that a user contributes to the meme only depends on the zone i and not on the exact coordinates, i.e., $f(x, y)$ has a constant value in each zone i. We will denote this zone-based probability as f_i. The number of samples $|O_i|$ in zone i has the expected value $E(|O_i|) = f_i N$, which will be approximated well by $|O_i|$ if the number of users N in each zone i is large enough (law of large numbers). Under this assumption, the relative frequency P_i does not depend on the number of users N and reflects only the relative value of f_i which we denote as P_i^f:

$$P_i = \frac{|O_i|}{|O|} = \frac{|O_i|}{\sum_{j \in R} |O_j|} \approx \frac{f_i N}{\sum_{j \in R} f_j N} = \frac{f_i}{\sum_{j \in R} f_j} = P_i^f \tag{5}$$

Hence, under the assumption of a uniformly distributed user population and an appropriately large number of users in each zone, P_i^f can be considered as the ground truth, undistorted by the population density, that is approximated by P_i. Thus, to approximate P_i^f if the number of users N_i in each zone i is *not* constant, we can use P_i' as follows instead of the relative frequency P_i:

$$P_i' = \frac{\frac{|O_i|}{N_i}}{\sum_{j \in R} (\frac{|O_j|}{N_j})} \approx \frac{\frac{f_i N_i}{N_i}}{\sum_{j \in R} (\frac{f_j N_j}{N_j})} = P_i^f \tag{6}$$

P_i' approximates the relative value of f_i and abstracts from the distribution of the users, by dividing f_i simply by the number of users in this zone. For versions of Focus and Entropy that are independent from the user distribution, we can use P_i' instead of P_i, if the number of samples in each zone is large enough. However, if the number of samples is too small, P_i' will not approximate P_i^f well and, moreover, individual samples can have a much higher influence on the results in comparison with the original measures if they appear in zones with few users. Hence, we risk boosting random effects. We further modified the adjusted measures to avoids this. Since by dividing by the number of users in

the zone, values of zones with few users are increased and values of zones with many users are decreased, we can treat selected zones neutrally by dividing by the average number of users. Hence, we divide by the average number of users in all zones R' that contain samples instead of dividing by the actual number of users N_i if the number of samples is small. The larger the number of samples $|O_i|$ the more strongly we consider the actual number of users for correction. We use a factor d_i in the range $[0, 1]$ that depends on the number of users N_i to control the strength of the correction. With

$$O'_i = \frac{|O_i|}{d_i N_i + (1 - d_i) \frac{\sum_{j \in R'} N_j}{|R'|}} \tag{7}$$

we calculate the corrected relative frequency as

$$P_i^{corr} = \frac{O'_i}{\sum_{j \in R} O'_j} \tag{8}$$

To set the factor d_i in dependence of $|O_i|$, we use

$$d_i = min\left(1, \frac{|O_i|}{sensitivity}\right) \tag{9}$$

with *sensitivity* being a predefined parameter. We tested different sensitivity-values and found a value of 25 being an appropriate tradeoff between the risk of boosting random effects and the goal of abstracting from the user distribution for our setting.

Evaluation. To evaluate whether the adjusted measures can reflect characteristics of $f(x, y)$ more accurately, we use the Monte Carlo simulation method described in Section 4.1 and test whether we can infer the radius r more precisely from the measured values. To compare the performance of the original and adjusted measures, we divided the range of Focus, Entropy and Spread values into 10 intervals of the same size. If the range of radii from which the set of meme occurrences in an interval [x,y) result is small, we can conversely predict the radius well if we measure a value in this range. To quantify this, we calculated the mean absolute deviation for the radii that produced the values in each interval, i.e. the average distance of the radii to the mean radius. The results are depicted in Figure 5. Both, adjusted Focus F^{corr} and adjusted Entropy E^{corr} allow us to recover r more accurately in comparison with the original measures, while the improvements are more obvious in case of the adjusted Focus. In case of large radii r (i.e. sets of meme occurrences with high Focus and low Entropy), the adjusted measures and original formulations seem to be almost equivalent.

Spread. The zone-based measures have the advantage that the number of users in each zone can be counted and, hence, the user density can be easily incorporated into the model. In order to abstract from the user distribution, we can

formulate a zone-based approximation S' of the Spread measure S by mapping all samples within a zone i to the midpoint of the zone m_i. Then, S' can be calculated based on the zone frequencies because:

$$S' = \frac{\sum_{i \in R} |O_i| \, D\left(m_i, Mp\right)}{|O|} = \sum_{i \in R} \left(\frac{|O_i|}{|O|} D\left(m_i, Mp\right) \right) = \sum_{i \in R} P_i D\left(m_i, Mp\right) \tag{10}$$

Since the distance between the midpoint of the zone i and the geographic mean point of the set of meme occurrence $D(m_i, Mp)$ does obviously not depend on the number of users N_i, we can use P_i^{corr} instead of P_i to reduce the effect of the user distribution, similar to F^{corr} and E^{corr}. Then, we also need to consider the P_i^{corr} as weights to calculate the weighted mean point, denoted as Mp'. Instead of mapping all samples in a zone to the midpoint of the zone, we can use the average distance of the samples in a zone to the weighted mean point Mp' to improve the accuracy. Hence, we use the following corrected Spread measure:

$$S^{corr} = \sum_{i \in R} \left(P_i^{corr} \sum_{o \in O_i} \frac{D\left(o, Mp'\right)}{|O_i|} \right) \tag{11}$$

Evaluation. Similar to the evaluation of F^{corr} and E^{corr}, we plotted the error rates for Spread and adjusted Spread S^{corr}. Figure 6 illustrates considerable improvements, while a significant influence of the population density effect remains.

5.2 Requirement P2 - Inherent Limitations

According to the definition of the measures, minimum Focus and maximum Entropy depend on the number of samples. When measuring a set C_n with n occurrences, the Focus values F will be in the range $[\frac{1}{n}, 1]$ and the Entropy values E in the range $[0, log_2 n]$. Thus, the possible values of both measures depend on the sample size. Obviously, we could simply normalize both measures to an

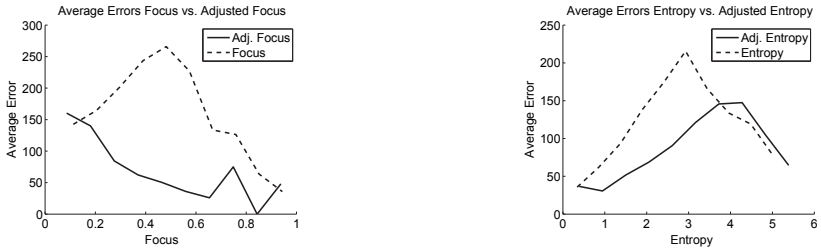


Fig. 5. The adjusted measures decrease the mean absolute deviation and reflect geographic preferences more accurately.

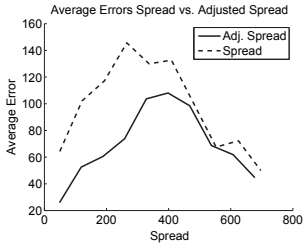


Fig. 6. Adjusted Spread in comparison with the original measure

interval of $[0, 1]$ by using the normalized Shannon Entropy $E_{norm} = E/(log_2 n)$ [5] and a normalized Focus measure $F_{norm} = (F - \frac{1}{n})/(1 - \frac{1}{n})$. However, this does not solve the problem. We would just induce another bias since even if the minimal measurable Focus is $\frac{1}{n}$ and the maximal measurable Entropy is $log_2 n$, the true Focus F^C and Entropy E^C might still lie above or below respectively. Therefore, it is an inherent limitation of these measures that they cannot capture the entire range of possible values when the subset is smaller than the underlying entire set, hence resulting in systematic errors. We can only ensure that an appropriate sample size is used to avoid too strong biases.

6 Conclusion

In this paper, we have analyzed to which degree current measures that are employed for the geospatial analysis of social media are influenced by the distribution of the underlying user population and sample size. Our results indicate that neither Spread, nor Entropy, nor Focus are able to solely characterize location-based relevance, but that the outcome is mixed up with impact from the underlying user distribution as well. Overall, our conducted Monte Carlo simulations demonstrate that too few samples and effects caused by the distribution of the user population can distort spatial measures so that they do not reflect location-based relevance accurately. Even though, we based our analysis only on three selected measures, our results indicate that the described problems are not specific to the measures under examination but apply to other spatial measures as well. Based on our experiments, we provided concrete hints that indicate in which scenarios the measures are still applicable. We proposed adjusted measures that decrease the influence of the user distribution. Hence, the adjusted measures reflect location-based relevance more accurately, even though, the biases are not eliminated entirely. Furthermore, we showed that the sample size problem leads to systematic errors that cannot be adjusted easily because it is not possible to draw conclusions about the location-based relevance at all if the number of samples is too small.

The findings presented in this paper suggest reconsidering current beliefs on the temporal evolution of social media content. Latest studies on the propagation of YouTube videos [2] and twitter hashtags [4] both observed a similar pattern. As

our experiments indicate, the observed results, however, can also be explained by other effects and not solely by location-based relevance: For example, a decrease of entropy and an increase of focus after the peak time can be explained by the decreasing number of samples in each time slot. We conclude that future studies should reinvestigate spatio-temporal patterns of social media distribution with much more dense data sets that provide a sufficient number of samples in each time interval.

References

1. Backstrom, L., Kleinberg, J., Kumar, R., Novak, J.: Spatial variation in search engine queries. In: Proceedings of the 17th international conference on World Wide Web. pp. 357–366. ACM (2008)
2. Brodersen, A., Scellato, S., Wattenhofer, M.: Youtube around the world: Geographic popularity of videos. In: 21st International Conference on World Wide Web. pp. 241–250. WWW '12, ACM (2012)
3. Ding, J., Gravano, L., Shivakumar, N.: Computing geographical scopes of web resources (2000)
4. Kamath, K.Y., Caverlee, J., Lee, K., Cheng, Z.: Spatio-temporal dynamics of online memes: A study of geo-tagged tweets. In: 22nd International Conference on World Wide Web. pp. 667–678. WWW '13, International World Wide Web Conferences Steering Committee (2013)
5. Kumar, U., Kumar, V., KAPUR, J.N.: Normalized measures of entropy. International Journal Of General System 12(1), 55–69 (1986)
6. Leopardi, P.: A partition of the unit sphere into regions of equal area and small diameter. Electronic Transactions on Numerical Analysis 25, 309–327 (2006)
7. Li, L., Goodchild, M.F., Xu, B.: Spatial, temporal, and socioeconomic patterns in the use of twitter and flickr. Cartography and Geographic Information Science 40(2), 61–77 (2013)
8. Openshaw, S.: The Modifiable Areal Unit Problem. Concepts and techniques in modern geography, Geo Books, Norwick (1984)
9. Petrovic, S., Osborne, M., McCreadie, R., Macdonald, C., Ounis, I., Shrimpton, L.: Can twitter replace newswire for breaking news? In: Seventh International AAAI Conference on Weblogs and Social Media. The AAAI Press (2013)
10. Scellato, S., Noulas, A., Lambiotte, R., Mascolo, C.: Socio-spatial properties of online location-based social networks. In: ICWSM (2011)

Context-Aware Location Prediction

Roni Bar-David and Mark Last

Department of Information Systems Engineering
Ben-Gurion University of the Negev
roni12@gmail.com, mlast@bgu.ac.il

Abstract. Over the past few years, predicting the future location of mobile objects has become an important and challenging problem. With the widespread use of mobile devices, applications of location prediction include location-based services, resource allocation, smooth handoffs in cellular networks, animal migration research, and weather forecasting. Most current techniques try to predict the next location of moving objects such as vehicles, people or animals, based on their movement history alone. However, ignoring the dynamic nature of mobile behavior and trying to repeatedly exploit the same common patterns, may yield wrong results, at least part of the time. Analyzing movement in its context and choosing the best movement pattern by the current situation, can reduce some of the errors and improve prediction accuracy. In this paper, we present a context-aware location prediction algorithm that utilizes various types of context information to predict future location of vehicles. We use five contextual features related to either the object environment or its current movement data: current location; object velocity; day of the week; weather conditions; and traffic congestion in the area. Our algorithm incorporates these context features into the trajectory-clustering phase as well as in the location prediction phase. We evaluate our algorithm using two real-world GPS trajectory datasets. The experimental results demonstrate that the context-aware approach can significantly improve the accuracy of location predictions.

1 Introduction

In recent years, ever more data is available about the location of moving objects and their usage context. Playing no small part in this growth is the increasing popularity of mobile devices equipped with GPS receivers that enable users to track their current locations. These devices have now emerged as computing platforms enabling users to consume location-aware services from Internet websites dedicated to life logging, sports activities, travel experiences, geo-tagged photos, etc. In addition, many users have started sharing their outdoor movements using social networks. The interaction of users with such services inevitably leaves some digital traces about their movements in device logs and on various web servers, which provide a rich source of information about people's preferences and activities with regard to their location. This provides the opportunity to collect both spatio-temporal and contextual data, and in turn, to develop innovative methods for analyzing the movement of mobile objects.

We assume that the behavior of moving objects can be learned from their historical data. In any particular environment, people do not move randomly but rather tend to follow common paths, specific repetitive trajectories that correspond to their intentions. Thus, predicting their future location is possible by analyzing their movement history. Location prediction models can enhance many different applications, such as location-aware services [1], mobile recommendation systems [2], animal migration research, weather forecasting [3], handoff management in mobile networks [4], mobile user roaming [5], etc. While a substantial amount of research has already been performed in the area of location prediction, most existing approaches focus on using the most common movement patterns for prediction without taking into account any additional contextual information. This severely limits their applicability since in practice the movement is usually free and uncertain. In many situations, it may not be sufficient to consider only the movement history, since the probability of a future location does not only depend on the number of times it was visited before but also on the object's current context. For example, in different situations, a driver may choose different routes leading to the same destination.

The importance of contextual information has been recognized by researchers and practitioners in many disciplines. Context-aware applications are able to identify and react to real world situations better than applications that do not consider this sort of information. Context information can help systems to ease the interaction with the user and sometimes even allow a device to be used in a way that was not intended by design. In this paper, we argue that relevant contextual information does matter in location prediction and that it is important to consider this information when providing predictions.

The goal of this work is use context to improve the accuracy of predicting movement of mobile objects such as vehicles. We represent movement patterns as sequences of spatio-temporal intervals, where each interval is a minimal bounding box (MBB). We enhance the standard MBB-based representation of movement patterns described in [6] with a context profile. The extended MBB represents a geographic area traversed by an object at a certain time, which also has a semantic meaning. The context profile turns an MBB into a segment that represents a set of data points not only close in time and space, but also having the same or similar values of contextual features. We extract contextual features for each MBB from its summarized data points. We calculate the average velocity, separate weekends from regular workdays, fetch weather conditions for the area where the object is moving and mark traffic congestion areas. We use a context-aware similarity measure to cluster similar trajectories with different types of contextual data. We present a new algorithm for location prediction that utilizes several context information features and overcomes the limitation mentioned above by choosing the relevant movement pattern according to the context. Finally, we compare the prediction performance and show that our context-aware prediction model can provide more accurate predictions than the existing context-free model. The main contribution of this work is the use of spatio-temporal data mining techniques in combination with context information in order to achieve a higher prediction accuracy than a state-of-the-art context-free method. We build an integrated model of context information and we

use several context features in the location prediction domain. The algorithm we develop simultaneously considers several types of contextual information from external and internal data sources, and incorporates them in the location prediction model.

2 Related Work

2.1 Location Prediction Methods

Predicting the future location of a given mobile object is an interesting and challenging problem in data mining. In [7], the authors propose the WhereNext system that is based on association rules, and define a trajectory as an ordered sequence of locations. Given a trajectory of a moving object, it selects the best matching association rule and uses it for prediction. Trajectory patterns are sequences of spatial regions that emerge as frequently visited in a given order. Motion functions represent the trajectory of an object based on its recent movements, and indicate location changes by updating the function parameters, instead of recording object locations at individual timestamps. The most common functions are linear models that assume an object follows linear movements [8]. Given an object's current location, time and velocity the object's future location is estimated at a future time. Although this simple formula avoids many complications, it has limited applicability due to the enormous diversity of motion types. There are non-linear models that capture the object's movements by more sophisticated mathematical formulas. Thus, their prediction accuracies are higher than those of the linear models [9]. The recursive motion function is the most accurate prediction method among motion functions. It predicts an object's location at a specific time in relation to those of the recent past. It can express unknown complex movements that cannot be represented in an obvious manner. In [10] authors propose a hybrid prediction algorithm that takes advantage of an object's pattern information, as well as its motion function. This algorithm mines an object's trajectory patterns that follow the same routes over regular time intervals. When there is no pattern candidate in the prediction process, it calls a recursive motion function to answer the query.

The common Markov Chain models are used to address the problem of short-term location prediction [11]. This paper describes a hidden Markov model for routing management in mobile networks. A Markov model is one in which the current model state is conditionally dependent on the previous k states of the model. Each movement between locations corresponds to a transition between states and it is assigned a certain probability. In a hidden Markov model, the underlying states are not directly observable, and there is only access to an observable set of symbols that are probabilistically related to the hidden states.

Clustering is an unsupervised method of learning, which reveals the overall distribution pattern and interesting correlations in datasets. A cluster is a set of data objects with similar characteristics that can be collectively treated as one group. In [12] the authors introduced the concept of moving micro-clusters. A micro-cluster indicates a group of objects that are not only close to each other in a specific time, but also likely to move together for a while. In order to keep the high quality of clustering, these micro-clusters may split and get reorganized. A moving micro-cluster can be viewed as one

moving object, and the center of it can be treated as its location. In [13] the authors propose a segmentation algorithm for representing a periodic spatio-temporal trajectory, as a compact set of MBBs. An MBB represents a spatio-temporal interval bounded by limits of time and location. This structure enables summarizing close data-points, such that only the minimum and maximum values of the spatial coordinates and time are recorded. The authors define data-amount-based similarity between trajectories according to proximity of trajectories in time and space.

2.2 Spatio-Temporal Context Models

Context awareness is defined as the ability to identify and react to situations in the real world by taking advantage of contextual information [14]. Making use of relevant context information from a variety of sources to provide suitable services or task-dependent information is one of the main challenges in ubiquitous computing. In [15] ubiquitous computing defined as a paradigm shift where technologies become virtually invisible or transparent and yet pervasively affect every aspect of our lives. The authors proposed a reference framework to identify key functionalities of context-awareness, and assisting ubiquitous media applications in discovering contextual information and adapting their behavior accordingly. Another context-aware model that supports managing of ubiquitous applications is described in [16]. It includes modules for context acquisition from distributed sources, context storage and actuation in the environment, and context processing for notifying the current state.

Context is a general concept describing the conditions where a process occurs. Several definitions of the term can be found in the literature, and it can be seen that the scope of definitions has expanded over the years. The earliest definition of context perceive it mostly related to location [17] as the user location, the identity of people near the user, the object around, and the changes in these elements. In [18] the authors enriched context definition to include additional user and environment information, such as habits, emotional state, shared position with others and social interactions. Context definition was further extended in [14] to distinguish between active context that influences the behavior of an application, and passive context that can be relevant but not critical to the application.

Context information is used in the location prediction literature mostly with respect to identity, location and time. In [19] authors demonstrated the use of user location as context in their supervised learning approach for detecting spatial trajectory patterns. They mined travel sequences of multiple users from GPS trajectories, and found interesting places in a certain region from the location history. An interesting location is defined using a stay point, a geographic region where the user stayed over a certain time interval. This model uses other people's travel experiences in order to improve travel recommendations. In [20] the authors propose to capture the spatio-temporal context of user check-in behavior in order to improve a location prediction model. They consider temporal periodic patterns in addition to spatial historical trajectories for computing the next visit probability in a location during a certain time interval. The location and time probabilities restrain and complement each other in the form of spatio-temporal context. Appling smoothing techniques they found that a user mostly checks-in

at a location during a specific time interval and rarely visits during other time intervals. In [21] authors propose incorporating temporal context dimensions such as time of the day in a route recognition system. They designed a real-time personal route recognition system that uses context information to predict the route destination. Instead of comparing the current driving trajectory against the trajectories observed in the past and select the most similar route, they incorporated the time of day as context, and adjusted the route recognition output accordingly. They found that the trajectories depend on the time of day feature, even at the beginning of a trip, that typically starts at the same departure point (e.g., home) and creates overlapping trajectories.

In our literature survey, we could not find any location prediction solutions that utilize context information from different data sources. Current techniques often focusing on only one type of spatial or temporal context, and do not consider sudden changes in the movement and the environment. The attempts to integrate both spatial and temporal context information suffer from the overfitting problem due to the large number of spatio-temporal trajectory patterns. In order to cope with the dynamic nature of movement, we are interested in context features that affect the choice of a movement pattern. Existing context-aware models have not considered how to incorporate the context information in the task model in an efficient way, but separated the context model from the task. We try to find a convenient representation of the context information, in order to incorporate it in the movement pattern representation. In our work, we try to integrate into one prediction model both the context features generated from the historical movement data and the context features available from external sources, like web services.

3 Proposed Methodology

Our approach consists of three main steps: (1) extracting periodic trajectories together with the context information of the mobile object; (2) inducing context-aware models that describe the object's movement patterns; and (3) predicting the future location of an object based on the two previous steps and the current context.

3.1 Extracting Contextual Information

We introduce a unique representation of an area and its contextual features, as a combination of a minimal bounding box (MBB) and a context profile. The extended MBB represents a spatio-temporal interval as well as its related context features. It can be regarded as a geographic area traversed by an object at a certain time, which also has a semantic meaning. Formally, MBB has the following properties:

$$i.t_{min} = \min(\forall p \in i, p.t_{min}),\ i.t_{max} = \max(\forall p \in i, p.t_{max})$$
$$i.x_{min} = \min(\forall p \in i, p.x_{min}),\ i.x_{max} = \max(\forall p \in i, p.x_{max})$$
$$i.y_{min} = \min(\forall p \in i, p.y_{min}),\ i.y_{max} = \max(\forall p \in i, p.y_{max})$$

where i represents an MBB; p represents a data point in a box; x and y are spatial coordinates; and $t_{\min}$ and t_{max} are the object's minimum and maximum times in the area.

This allows us to summarize close data points into one MBB, and represent it by these six elements rather than by the multiple original data points. We consider context to be represented by a context profile related to both an area and a mobile object's current state. Context features may be related to the environment surrounding a mobile object or to the mobile object movement itself. A context profile turns an MBB into a segment that represents a set of data points not only close in time and space, but also having the same or similar values of contextual features. A context feature has an identifier, a type and a value. Formally:

$$Context\ Profile = \{ i, amount, location, velocity, day\ of\ the\ week, weather, traffic\ congestion \}$$

where *i* identifies the MBB; and the other properties represent the aggregated feature values. We enhance the MBB representation to support additional context features with the following method:

$$MBB.cf = \text{aggregation}(\forall p \in MBB, p.cf)$$

where *cf* stands for a context feature that is being aggregated; and p represents a point member in an MBB. In case of a discrete feature, such as the day of the week, all data points will have the same contextual value. In case of a numeric feature, all summarized data points will be restricted to a certain interval. For example, a movement pattern may be characterized by a predefined velocity range.

This compact and unique representation of an MBB allows us to obtain quality and time efficient results in subsequent mining stages, and to incorporate the context information in the trajectory-clustering phase. Several context features may influence the mobile object movement. We are looking for occurrences of certain context characteristics that change the movement habits. Awareness of such events can help us choose the right movement pattern, and help to improve future location predictions. A diagram describing the different types of context is described in Fig. 1.

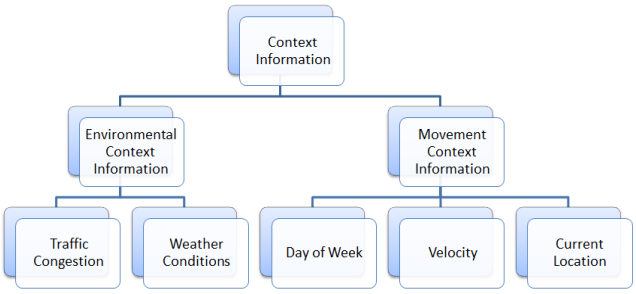


Fig. 1. Different Context Types

Amount. This is the baseline property that represents the number of data points that are summarized by a given MBB. It is calculated as the number of times the area has been accessed. The more times an object visits a particular MBB, the more important

it becomes. The algorithm chooses the most common MBB within the time bounds with the maximal data amount property to be the prediction result. Formally:

$$MBB.amount = \text{count}(\forall p \in MBB, 1)$$

Current location. The current location of every point is recorded in the dataset using Cartesian coordinates. Though it is the simplest feature to extract, it is very powerful because it allows us to discard common, yet infeasible, patterns. We can correct unreasonable predictions by choosing other movement patterns. For instance, if a mobile object usually travels from home to work every morning, and we find it near the other side of town, we should change our prediction. We calculate the center point of each candidate MBB, and choose the closest one to the current object position. We use the Haversine distance formula [22] to calculate the distance between two points.

Velocity. Moving at a certain velocity may imply the next destination of the object. We assume that different velocities can represent different contexts, and if an object's velocity is higher than average, it is an indication that its destination is different than the usual one and vice versa. For example, driving at a high velocity near the work area may imply that the destination is not the office. We assume that each mobile object has its actual velocity recorded at every data point. If it is missing, we can compute it from the spatio-temporal trajectory by dividing the vector travelled between two consecutive points by the time between the two measurements. We estimate the average velocity at each data point as the average velocity over the last five points. The average velocity of each MBB is calculated by the sum velocities of points included in it divided by the number of data points. The algorithm chooses only MBBs with same velocity as the object current velocity.

Day of the week. The current time is recorded as a timestamp for each entry in the dataset. Thus, we know the current weekday and try to find typical routes for the weekend and regular weekdays. In order to model this contextual feature, the week is divided into two periods: the regular weekdays and the weekend. Each point or MBB is associated with one of these periods, according to the trajectory time. We assume that the mobile object movement depends on the day of the week context. For instance, on a regular weekday a person may go to work in the morning and choose to drive on a specific route, while on the weekend he will probably choose a different recreation destination. The algorithm chooses MBBs according to the object current day.

Weather. Weather conditions may affect every object's movement. For example, snow can prevent drivers from accessing certain areas from driving while downpours can cause people to gather in the mall instead of the public open park. Since the movement data does not contain information about weather, we get this information from external online weather Web service called Weather Underground. This site publishes weather data on an hourly basis and describes weather conditions with different numeric and binary variables. We divide the object movement area into a grid, consisting of squares with a certain parameter size. We then query the site for each new point, and save the weather information in order to reuse it as long as the location does not exceed a threshold value. We refer only to weather conditions that may influence mobile object movement namely foggy, rain, and snow. Any other type of weather condition, such as clear,

cloudy, hot, cold, etc. is ignored and labeled as "other" category. We retrieve the weather conditions in an MBB by using the mean coordinates and median timestamp. An MBB weather is defined as the most frequent category in its area and within its data points time bounds.

Traffic congestion. Traffic congestion affects every mobile object movement by slowing its normal tempo and causing a change in routine movement habits. Since the movement data does not contain explicit information about traffic congestion, we implicitly infer this data. Using an external Web service to locate areas that are prone to traffic congestion, we search for a certain events of decreasing velocity near them. Traffic congestion is characterized by a significant difference between the current speed and the permitted speed on the road. When an object is moving much slower than the allowed speed in the area, we infer that a traffic congestion event takes place. In the case when we do not have access to the permitted velocity on the road, we look for areas in which the velocity decreases for a certain distance and certain time. We use the segmentation algorithm described in section 3.2 to extract traffic congestion areas containing the mentioned events, with duration above 5 minutes and distance above 100 meters. We mark these MBBs with a traffic congestion marker accordingly. The clustering algorithm described in section 3.3 find areas where these kinds of events occur repeatedly. Not every decreasing velocity event corresponds to traffic congestion. Very possibly the decreased velocity may be due to an intersection with a fueling or parking area. The marked MBBs are areas containing decrease velocity events that occur closely, both in space and in time. Thus, occasional low velocity events that occur closely in space but at different times will be regarded as noise.

In order to verify our traffic congestion findings, we use an external web service called Google Places that suggests points of interest that are close to a given location. Points of interest are interesting locations such as culturally important places and commonly frequented public areas, like shopping malls and restaurants. Google Places supports 126 metadata types to describe points of interest. We refer only to those in the following three categories: institutions (including religious, health and services), entertainment (including shopping, parks and food) and transportation (including bus/train stations and intersections). We search points of interest in the one-kilometer radius of every MBB center point. If an MBB is located near highway intersections around the city, or on major streets, it is more likely that traffic congestion can be expected. In the prediction phase, when we detect a traffic slowdown event, we adjust the time bounds respectively. Thus, the algorithm can choose only close and feasible MBBs relative to the current object location with the same velocity.

Combined context. This method integrates all the above context features together into one model. We calculate the context similarities for each possible MBB candidate within the time bounds. The final similarity measurement assigns the same weight to all five context features, and utilizes their overall influence on the movement trajectory.

3.2 Building MBB-based Trajectory

We present an incremental segmentation algorithm that summarizes the movements of each mobile object into periodic trajectories while extracting the desirable context features. We are looking for trajectories that continually reoccur according to some temporal pattern in a certain context. The algorithm inputs are a spatio-temporal dataset

that contains GPS log entries, the available context feature, and a scaling parameter to determine MBB bounds. The threshold values are calculated by multiplying the data-points standard deviation in each dimension, with the scaling parameter. The larger the threshold is, the more summarized the trajectories are. The outputs of this phase are an object's periodic trajectories represented as a list of MBBs. These segmented trajectories become the input of the next clustering phase.

Algorithm I: Building an Object's Context-Aware Trajectory

1. *trajectory*.addNewMBB($point_0$) // add the first MBB
2. $i \leftarrow 1$
3. **For Each** $point_i \in dataset$
4. **If** $point_i.date = point_{i-1}.date$
5. **If** $|point_i$.x – *trajectory*.lastMBB.maxX$| < threshold_X$ **And**
6. $|point_i$.y – *trajectory*.lastMBB.maxY$| < threshold_Y$ **And**
7. $|point_i$.time – *trajectory*.lastMBB.maxTime$| < threshold_{Time}$ **And**
8. $point_i.cf =$ *trajectory*.lastMBB.cf
9. *trajectory*.lastMBB.addPoint($point_i$)
10. **Else**
11. *trajectory*.addNewMBB($point_i$)
12. **Else**
13. *result*.add(*trajectory*)
14. *trajectory* ← null
15. *trajectory*.addNewMBB($point_i$) // start new periodic trajectory
16. **Return** *result*

In lines 1-4, we process each data point on a daily basis. In lines 5-9, as long as the point has the same context and is within the MBB bounds, we insert it to an existing current MBB. In lines 10-11, we decide that the point stretches the MBB beyond the threshold and create a new MBB. The addNewMbb function initializes a new MBB in the trajectory, and set its bounds and context according to the data point. The addPoint function increments the MBB amount property and updates the existing MBB with the new point boundaries.

3.3 Context-Aware Similarity Measure

We define a similarity measure that takes into account the context features in the movement patterns clustering algorithm. We incorporate the context into the model by increasing the similarity for MBBs in the same context. For example, if one MBB has a raining weather condition, all the MBBs with the same weather value will get bonuses towards their similarities with the current MBB. All the other MBBs will remain in the set without decreasing their similarity. We define the similarity between two trajectories as the sum of similarities between their MBBs. We calculate the similarity between each MBB feature. Our context-based similarity is calculated as the product of three factors: (1) the overlapping time of two MBBs; (2) the similarity between the data point

amounts summarized within the two MBBs; and (3) the minimal distance between the two MBB properties. Formally:

$$MBBsim(MBB_1, MBB_2) =$$

$$|t_m - t_n| \times \#points \times minContextDist(MBB_1, MBB_2)$$

where t_m is the time when the two MBBs start to overlap; and t_n is the time when its overlapping ends. The greater the overlapping time between two MBBs, the larger the similarity between them. The *#points* parameter is the minimum amount of data points within the two compared MBBs. The more data points are included in both of the compared MBBs, the stronger support we have for their similarity. The minimal context distance between two MBBs, consists of the sum of minimal distances of every feature in the MBBs. The minimal distance between two features is defined as the distance in that dimension or as zero if the two MBBs overlap. Formally:

$$minContextDist(MBB_1, MBB_2) = \sum_{i=1}^{n} \min CF\text{Dist}(MBB_1.cf_i, MBB_2.cf_i)$$

$$minCFDist(MBB_1, MBB_2) =$$

$$\max(0, \max(MBB_1.cf_{min}, MBB_2.cf_{min}) - \min(MBB_1.cf_{max}, MBB_2.cf_{max}))$$

where *n* is the number of context features we are measuring; and *cf* is the value of the feature *i*. We combine two distance values: a spatial distance and a context distance using a weighted average approach. For numeric features such as velocity, the distance is defined using this formula: $|v_1 - v_2|/(v_1 + v_2)$ where v_1, v_2 are feature values. When there is no difference between the values, the distance is zero, otherwise it is the normalized distance between the values, so the distances have a comparable scale. For nominal features such as weather and day of the week, the distance is defined as zero if the context values are the same and as one, otherwise.

We enhanced the k-means incremental clustering algorithm described in [6] to use this similarity measurement in order to mine context-aware movement patterns from the periodic trajectories. The algorithm gets as an input a set of periodic trajectories, a number of clusters, iteration bound, and a context feature. It initializes the centroids with the trajectories that are first to arrive. Then, each subsequent trajectory is inserted into its nearest centroid cluster. We update the centroids by adding each MBB into the trajectory that represents the cluster's centroid. We stop the clustering when there is no change or when we reach the iteration bound. The algorithm output is the trajectory cluster's centroids which are the movement patterns.

3.4 Context-Aware Location Prediction

The algorithm for predicting the future location of mobile objects uses the context-aware movement patterns we mined from the trajectories history, along with the current contextual features. The prediction algorithm filters MBBs by context, leaving only

those that have the same context as the currently observed instance. For example, if it is currently raining, only the MBBs that are marked as raining will be left in the candidate set. The algorithm inputs are a specific future time, a set of cluster centroids that are the movement patterns, and a context profile that contains any of the current context features mentioned. The algorithm returns the MBB's maximal and minimal coordinates as the prediction result.

Algorithm II: Predicting Mobile Object's Future Location

1. **For Each** $c_i \in centroids$
2. $j \leftarrow 0$
3. $MBB \leftarrow c_i.getMBB(j)$
4. **While** $(MBB_{maxTime} < time)$ // proceed to relevent MBBs
5. $MBB \leftarrow c_i.getMBB(j)$
6. $j{+}{+}$
7. **While** $(MBB_{minTime} \leq time \leq MBB_{maxTime})$ // add to candidate set
8. $MBB \leftarrow c_i.getMBB(j)$
9. *candidates*.add(*MBB*)
10. $j{+}{+}$
11. **If** $MBB_{minTime} > time$ **And** *candidates*.size = 0
12. **If** $j > 0$ // if empty set, choose the closest MBB
13. $prev \leftarrow c_i.getMBB(j-1)$
14. **Else**
15. $prev \leftarrow c_i.lastMBB$
16. **If** $(MBB_{MinTime} - time > time - prev_{maxTime})$
17. *candidates*.add(*MBB*)
18. **Else**
19. *candidates*.add(*prev*)
20. *candidates*.sortByAmount()
21. **For Each** $cf_i \in cf$ // iterate context features
22. **For Each** *MBB* ∈ *candidates*
23. **If** $MBB_{cf} \neq cf_i$
24. *candidates*.remove(*MBB*)
25. *removed*.add(*MBB*)
26. **If** *candidates*.size = 0
27. $candidates \leftarrow removed$ // ignore feature
28. **Return** *candidates*.getMaxAmount()

In lines 1-3, we search the centroids for MBBs that are within the time bounds. In lines 4-10, we add relevant MBBs to the candidate set. In lines 12-15, if no MBB matches the input time, we look for the closest MBB. The previous MBB can be the cyclic preceding item in the centroid, as before the first item at time 0:00 comes the last item at time 23:59. Finally, in lines 21-26, we iterate through the context profile and

filter the candidates by the required features. The feature is ignored if none of the candidates is left in the set. In lines 28, we choose the one with the highest amount score and return it as the prediction result.

4 Empirical Evaluation

We evaluated our context-aware prediction algorithm on real-world datasets comprised of spatio-temporal trajectories of mobile objects and data from external web services. Our hypothesis was that the context-aware algorithm should be able to predict future locations in a more accurate manner than a context-free method. Our algorithm predicts future location results in the form of MBBs rather than raw data points. In order to assess the prediction capability, we used two performance measures. The first measure is the prediction accuracy, which is the probability that the actual future location is found within the predicted MBB. Accuracy is defined as:

$$Accuracy = \frac{hits}{hits + misses}$$

According to this measure, larger MBBs should have better prediction accuracy than smaller MBBs. For example, if the clustering phase produces only one large MBB, which includes all predicted locations, we will have 100% accuracy but the prediction will not be useful for any practical purpose. On the other hand, it will be hard to build a useful general model using very small MBBs. The second measure is the Mean Average Error (MAE), or the spatial distance between the real location of the object at the predicted time and the borders of the predicted area. We calculate minimal, average and maximal Euclidean distance between the object and the MBB's boundaries. The maximal distance is as an upper boundary to the prediction error. MAE is defined as:

$$MAE = \frac{\sum maxDistance(object_{X,Y} - MBB_{X,Y})}{N}$$

where $object_{X,Y}$ represents the real position of the object, MBB is the prediction result, *maxDistance* is the maximal Euclidean distance function, and *N* stands for the total number of predictions.

We performed experiments on two real-world spatio-temporal datasets. The INFATI dataset [23] consists of GPS log-data from 20 cars driving by family members in Aalborg, Denmark. The movement of each car was recorded for periods of between 3 weeks to 2 months. The dataset contains a total of 1.9 million records. The GEOLIFE dataset [24] is a trajectory dataset that was collected mostly in Beijing, China, from 182 users, during a five-year period. We referred only to object trajectories traversed by vehicles, which were based on data collected for more than a one-month period.

In the algorithm empirical evaluation, we tried to predict the future location of mobile objects with and without each context feature. The evaluation of each context method was conducted as follows. We used all the data from every object as an independent dataset. We divided the dataset into training and testing sets, using the leave-one-out cross-validation method. In every iteration, we selected two dates as the testing set while keeping the rest of the dates as a training set. The clustering algorithm mined

context-aware movement patterns from the spatio-temporal trajectories in the training set. Afterwards, we iterated the data points from the two-day testing set, updated the context profile and predicted their future location. We used the one-tailed distribution t-test, to test for significant difference between the compared methods. The results of each method are summarized in Table 1.

Table 1. Accuracy and MAE Prediction Results

	INFATI		GEOLIFE	
	Accuracy	**MAE**	**Accuracy**	**MAE**
Baseline	0.738	1,067	0.814	683
Location	0.769*	943	0.830*	584
Velocity	0.799*	921	0.835*	622
Day of the week	0.731	1,239	0.801	751
Weather	0.738	1,096	0.820	641
Traffic congestion	0.812*	898	0.837*	523
Combined context	0.823*	850	0.883*	474

In both datasets, most of the context information methods outperform the baseline method. In the INFATI dataset, when using the location context method which ignores distant predictions, the accuracy increased to 0.769 (p-value < 0.001). When using the velocity context method, the accuracy was even better 0.799 (p-value < 0.0001). In the GEOLIFE dataset, the location context method increased the accuracy to 0.830 (p-value < 0.001). When using the velocity context method, the accuracy was somewhat better 0.835 (p-value < 0.001). The day of the week method did not operate better than the baseline method. The weather method has a slightly better, yet not statistically significant, prediction accuracy of 0.820 (p-value = 0.383). In the INFATI dataset, the weather method has the same prediction accuracy of 0.738 as the baseline method, with no significant improvement. This is probably due to the short period of the data collection. The best single context feature in both datasets is the traffic congestion method that combined location and velocity features and improved the prediction accuracy to 0.812 (p-value < 0.001) in the INFATI dataset, and to 0.837 (p-value < 0.001) in the GEOLIFE dataset. Finally, the combined context method which utilizes all the context features significantly improved prediction accuracy to 0.823 (p-value < 0.001) in the INFATI dataset, and to 0.883 (p-value < 0.001) in the GEOLIFE dataset. It should be noted that some context features, like traffic congestion, might be related to the day of the week. The location, velocity and traffic congestion MAEs of the context-aware methods decreased compared to the baseline method. This confirms the usefulness of location and velocity features compared to the day of the week feature. The combined context method demonstrates the best performance with the minimal MAE values for both datasets. This indicated that context information helps obtaining more accurate prediction results.

We analyze several characteristics of the movement patterns we found. The clustering results depend on segmentation threshold values and on the similarity measures.

The higher they are, the more trajectories are summarized into larger MBBs. The ability to make an accurate prediction depends on the number of MBBs we have to choose from and on each MBB's size. As we can see in Table 2, context-aware methods provide smaller MBBs and more MBBs to choose from in every prediction.

Table 2. Average sizes of MBBs and candidate set

	INFATI		GEOLIFE	
	Candidates	**MBB Size**	**Candidates**	**MBB Size**
Baseline	3.9	1,952	4.7	1,299
Location	3.9	2,952	4.7	1,299
Velocity	9.5	1,906	7.1	1,283
Day of the week	4.5	1,721	5.4	1,352
Weather	6.1	1,853	6.2	1,465
Traffic congestion	9.8	1,622	9.3	1,002
Combined context	9.8	1,302	9.3	921

5 Conclusions

In this paper, we analyzed location prediction from a context awareness perspective. We proposed a novel way to leverage context information of several types in order to predict the future location of mobile objects. We suggested appropriate methods to extract context information based on the movement and the surrounding environment. We enhanced k-means clustering algorithm for discovering movement patterns from spatio-temporal trajectories, to support five types of context features regarding location, velocity, day of the week, weather and traffic congestion. We designed a new algorithm for predicting the future location of a mobile object that utilizes context information based on the above ideas. The algorithm is shown to perform well, with significantly better accuracy results. Our evaluation results show that it is indeed helpful to refer to context information, while choosing movement patterns for prediction. In addition, incorporating context information allows us to collect smaller amounts of historic movement data, before we can start predicting the future location of mobile objects effectively. Thus, we can shorten the data collection period, while keeping the same level of prediction accuracy.

As future work, we suggest extending the algorithm to support other relevant context information features. For example, we can extract significant anchor points, such as home and work, from the object trajectories. We can also extract information about activities or events in the area from external data sources, like social networks. In addition, as this method uses the trajectories of each mobile object alone, one may apply this technique to consider all available trajectories. Instead of using only the individual user history, we could utilize all available object trajectories together. Further work is needed for designing a formal ontology-based context model, which will be able to capture, represent, manipulate and access all characteristics of location and movement.

Ontology should address such issues as semantic context representation, context dependency, and context reasoning.

References

[1] S. Amini, A. J. Brush, J. Krumm and J. Teevan, "Trajectory-aware mobile search," *In Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, pp. (pp. 2561-2564, 2012.

[2] . L. Kuang, Y. Xia and Y. Mao, "Personalized Services Recommendation Based on Context-Aware QoS Prediction," *IEEE 19th International Conference In Web Services*, pp. pp. 400-406, 2012.

[3] J. Lee, J. Han and K. Whang, "Trajectory clustering: a partition-and-group framework," *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, 2007.

[4] L. Zhang, N. Seta, H. Miyajima and H. Hayashi, "Fast Authentication Based on Heuristic Movement Prediction for Seamless Handover in Wireless Access Environment," *In IEEE WCNC, Wireless Communications and Networking Conference*, pp. pp 2889-2893, 2007.

[5] S. Akoush and A. Sameh, "Mobile user movement prediction using bayesian learning for neural networks," *In ACM IWCMC*, p. 191–196, 2007.

[6] S. Elnekave, M. Last and O. Maimon, "Predicting Future Locations Using Clusters Centroids," *In the Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems*, 2008.

[7] A. Monreale, F. Pinelli, R. Trasarti and F. Giannotti, "WhereNext: a Location Predictor on Trajectory Pattern Mining," *15th ACM SIGKDD*, pp. pages 637-646, 2009.

[8] J. Patel, Y. Chen and V. Chakka, "Stripes: an efficient index for predicted trajectories," *in SIGMOD*, p. pp. 635–646, 2004.

[9] Y. Tao, C. Faloutsos, D. Papadias and B. Liu, "Prediction and indexing of moving objects with unknown motion patterns," *In SIGMOD Conference. ACM Press* , pp. 611-622, 2004.

[10] H. Jeung, Q. Liu, H. Shen and X. Zhou, "A hybrid prediction model for moving objects," *IEEE 24th International Conference on Data Engineering*, pp. p.70-79, 2009.

[11] I. Nizetic, K. Fertalj and D. Kalpic, "A Prototype for the Short-term Prediction of Moving Object's Movement using Markov Chains," *In Proceedings of the ITI*, pp. 559-564, 2009.

[12] Y. Han and J. Yang, "Clustering moving objects," *In KDD*, p. pages 617–622, 2004.

[13] S. Elnekave, M. Last and O. Maimon, "Incremental Clustering of mobile objects," *STDM07, IEEE,* 2007.

[14] G. Chen and D. Kotz, "A survey of context-aware mobile computing research," *Technical Report TR2000-381, DartmouthCollege, ComputerScience,* 2000.

[15] D. Zhang, H. Huang, C. F. Lai and X. Liang, "Survey on context-awareness in ubiquitous media," *Multimedia Tools and Applications, 67(1),* pp. 179-211, 2013.

[16] J. Lopes, M. Gusmão, C. Duarte and P. Davet, "Toward a distributed architecture for context awareness in ubiquitous computing," *Journal of Applied Computing Research,* pp. 19-33, 2014.

[17] B. Schilit, N. Adams and R. Want, "Context-aware computing applications," *Proceedings of theWorkshop onMobile Computing Systems and Applications, MD: IEEE Computer Society,* 1994.

[18] A. Schmidt, M. Beigl and H. Gellersen, "There is more to context than location," *Procof the Intl.Workshop on Interactive Applications of Mobile Computing,* 1998.

[19] Y. Zheng, L. Zhang, X. Xie and W. Ma, "Mining interesting locations and travel sequences from GPS trajectories," *Proceedings of the 18th international conference on World wide web, ACM,* 2009.

[20] H. Gao, J. Tang and H. Liu, "Mobile location prediction in spatio-temporal context," *In Nokia mobile data challenge workshop,* 2012.

[21] O. Mazhelis, I. Zliobaite and M. Pechenizkiy, "Context-aware personal route recognition," *In Discovery Science. Springer Berlin Heidelberg,* pp. (pp. 221-235), 2011.

[22] C. Veness, "Calculate distance, bearing and more between points," 2002. [Online]. Available: http://www.movable-type.co.uk/scripts/latlong.html..

[23] C. Jensen, H. Lahrmann, S. Pakalnis and J. Runge, "The INFATI Data," *TimeCenter Technical Report,* pp. pp. 1-10, 2004.

[24] Y. Zheng, X. Xing and W. Ma, "GeoLife: A Collaborative Social Networking Service among User, location and trajectory," *in IEEE Data Engineering Bulletin. 33, 2,* pp. pp. 32-40, 2010.

Content-Based Sentiment and Geolocation Tagging of Social Media Messages for Trend Analysis

Gretchen Markiewicz[1], Saurabh Khanwalkar[1], Guruprasad Saikumar[1], Amit Srivastava[1] and Sean Colbath[1]

[1]Raytheon BBN Technologies, Cambridge, MA, USA
{gmarkiew,skhanwal,gsaikuma,asrivast,scolbath}@bbn.com

Abstract. Analysts have increasingly turned to social media platforms for identifying trends and following news stories. Public opinion and geographical context are key components in understanding evolving stories, but few social media messages provide geolocation information and there are limited services for analyzing trending sentiment. Most automated natural language processing (NLP) systems are trained on newswire domain and are not developed for the noisy and colloquial language of social media. We developed an approach for customizing content-based sentiment tagging and geolocation tagging of social media messages, by employing a social media specific tokenization process and training on in-domain data. Through this effort, we achieved a relative 48% improvement in sentiment tagging performance on Arabic tweets compared to a system developed for newswire, and we increased the accuracy of geolocation tagging on English and Arabic tweets. We demonstrate the application of content-based sentiment and geolocation tagging for social media by placing tweets filtered by sentiment polarity on a world map and tracking sentiment over time.

Keywords: Twitter, sentiment tagging, content-based geolocation, social media

1 Introduction

Social media such as Twitter and Facebook are invaluable resources for extracting up-to-the minute information on current events and reactions as they unfold around the globe. As publicly available media consisting of user-generated content, they often serve as the first platforms for spreading information on events and trends, before traditional media sources report on the same. The social aspect introduces personal opinions and a subjective layer to the information. The sheer volume of Twitter and Facebook posts necessitates automated approaches to analyzing this wealth of data.

Automatic geolocation of tweets and Facebook posts based on content provides insightful geopolitical context. The geographic distribution of social media posts containing the keyword "protest" can quickly alert an analyst to where such events may be developing, regardless of where the tweets originate. Distilling the sentiment of tweets and posts provides analysts with more nuanced snapshots of popular trends and reactions to current events. Moreover, understanding the opinions expressed by social media users surrounding a new government, for instance, and how sentiment around

this topic changes over time, can aid in situational awareness. Given the current international relevance of events in the Arabic-speaking world, work that improves the ability to intelligently analyze Arabic social media content is timely and important.

Messages from regions of interest often contain informal or dialectal non-English text with different characteristics from newswire text. Social media entities, such as hashtags and user-mentions, are prevalent; these entities provide potentially valuable information for analysts, including location information or key sentiment words, but are often not exploited when using traditional analysis tools. From a newswire perspective, they contain unusual punctuation, occur in unexpected contexts, or combine multiple words within a token. The forced brevity of tweets and the assumption of understood subtext on social media introduce further challenges when analyzing this content. One cannot expect strict language separation, even from a single user or within a single post. These language characteristics present challenges for Natural Language Processing (NLP) systems developed for traditional news media.

In this paper, we present a sentiment tagging model customized for Arabic social media content, and we analyze the impact of a social media tokenization on content-based geolocation tagging of tweets. In Section 2, we review related work. In Section 3, we describe the data and evaluation metrics. We outline our technical approach to sentiment and geolocation tagging in Section 4. We discuss results obtained on multilingual tweets in Section 5. We illustrate an application for sentiment and geolocation tagging by tracking sentiment geographically and over time, before concluding the paper.

2 Related Work

Sentiment tagging of Twitter data has gained popularity in recent years. [1] demonstrates that perceptron algorithms perform better than batch-learning methods when tagging sentiment of streaming data, such as tweets. [2] shows that a multi-epoch averaged perceptron model outperforms its corresponding perceptron model in sentiment tagging of book reviews. This work, like many other approaches to sentiment tagging, assigns one polarity to an entire document.

A finer-grained approach is to tag sentiment at the phrase level, allowing the possibility of both polarities to be found in one document. [3] takes this approach, discussing the importance of identifying words that express opinion. They developed their model using newswire data from the Multi-Perspective Question Answering (MPQA) corpus [4]. The phrases they found to contribute to subjective expressions may not be representative of sentiment expression in social media.

[5] demonstrates that content-based geolocation tagging is an effective solution for placing tweets within a geospatial context. They found that only 0.70% of 20 million tweets provided device- or user-based geospatial data, which may also be unrelated to the content. They developed a novel tweet-document creation algorithm based on a temporal window that supplies additional context for detecting geolocation. They then extract location named entities and cluster the hypothesized geographic locations to determine a document-level geolocation tag.

3 Data and Evaluation Metrics

3.1 Experimental Datasets

We developed our initial sentiment tagging model using MPQA corpus [4]. This corpus contains sentiment-annotated English newswire documents from 187 news sources. Additionally, to customize the sentiment tagger for social media domain, we collected Arabic tweets filtered by influential users from the Middle East using the Twitter Spritzer API. Then, we used the human English translation of these tweets to annotate sentiment phrases using the annotation guidelines described in MPQA. Table 1 summarizes these newswire and social media sentiment datasets.

Set	Corpus	Dates	Tokens	Subjective Tokens	Positive Tags	Negative Tags
Train	MPQA	'01-'02	281k	30k	3,757	6,787
	Tweet	2012	12k	3k	388	397
Test	MPQA	'01 –'02	36k	3k	390	718
	Tweet	2012	12k	3k	413	393

Table 1. Summary of data sets used to train and test the sentiment tagging system

Additionally, we collected close to 7M Arabic and English tweets over a one-month period in early 2014 to evaluate geolocation tagging. These tweets are 87% Arabic and 13% English and originate from 1.2M users. Our geolocation test set consists of ~6k such tweets that provided automatic device-based or user-based geotags.

3.2 Evaluation Metrics

Sentiment Evaluation. We use the Automatic Content Extraction (ACE) word-level alignment scoring tool, developed by DARPA [6]. Reference and hypothesis tokens are aligned and their respective sentiment classes are compared. We report precision, recall, and f-measure, or harmonic mean of precision and recall. We follow the same evaluation setup as in [3], except we report results for exact match of the token spans as well as of the polarity type (positive or negative), for a more rigorous evaluation. For example, for the subjective expression "roundly criticized", our method might only identify "criticized" with a negative polarity, but this prediction will be considered incorrect.

Geolocation Evaluation. For geolocation performance evaluation, we first compute the Error Distance on every tweet with a reference location and a hypothesized location. The Error Distance (ErrDist) is the distance in miles between the two geographic latitude-longitude points. We arrive at an overall metric by averaging over all tweets in the test set, for the Average Error Distance (AveErrDist). The lower the AveErrDist, the closer the hypothesized locations are to their reference locations, on average. For further analysis, we also compute the percentage of test set tweets for which the ErrDist is less than or equal to a fixed threshold, for AccuracyN (AccN); we report Acc100, Acc500, and Acc1000. All metrics are defined in [5].

4 Technical Approach

Our sentiment and geolocation taggers are two NLP components of a social media analysis system, depicted in Fig. 1. We process the text through named entity detection, machine translation (MT), sentiment tagging, and geolocation tagging. We perform n-gram based language identification (ID) [7] and translate Arabic text into English using the SDL Language Weaver MT system [8]. As this processing flow shows, we use MT-English for analyzing both sentiment and geolocation. We group tweets into documents for post-MT analysis to provide more context. To make our analysis components social media aware, we employ a social media tokenization technique. The NLP components under consideration are described below.

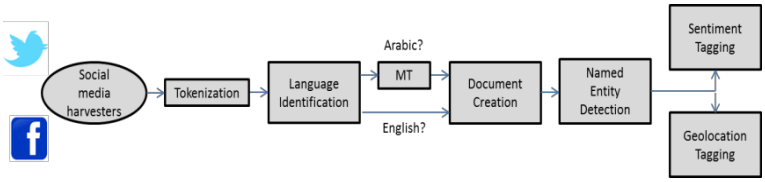


Fig. 1. Overview of our social media analysis system.

4.1 Social Media Tokenization

To understand the differences in newswire and social media domains, we compared the top frequent terms across these domains. We found 87 English Al Jazeera newswire articles containing keyword "Syria" and 69 Facebook messages with keyword "#syria" in our database. All top 10 Al Jazeera terms are common dictionary words, such as Syria, Iraq, government, Islamic, and country, whereas four of the top 10 Facebook terms contain the hash symbol (#), including one that combines two words in a single token, such as #Syria, #Together, #Eastern_gouta, #Ghouta. These language differences emphasize the need for a social media customized sentiment tagger and tokenizer to extract inherent information in entities such as hashtags.

For tokenization, we process hashtags by removing the hash symbol (#) and isolating words between underscores, to increase the likelihood of accurate translation or named entity recognition (NER). MT-English tokens are reconstructed into analogous hashtags. Our tokenization labels user-mention entities and ensures they remain single tokens with the user symbol (@) attached, to preserve the entity after MT. URLs and emoticons are also labeled to reduce the likelihood of alteration after MT.

4.2 Sentiment Tagging

We approach the sentiment tagging task as a discriminative problem of classifying each token in a document as one of five classes: positive-start (PS), positive-continue (PC), negative-start (NS), negative-continue (NC), and none (N). The only constraint dictates that a continue class can only follow a start class of the same polarity. For

example, the sentence "The proposal has really great ideas but was ultimately rejected" could be classified as N-N-N-PS-PC-N-N-N-N-NS.

Because it can handle a large number of variable features, we employ an averaged perceptron model [9]. [10] describes a similar model designed for name tagging. We incorporate hierarchical word clusters generated by an algorithm specified in [11], in which words are represented as a binary string indicating leaf location in a cluster tree. We build upon the name-tagging model in [10] by extracting additional features that improved sentiment tagging, namely the final four categories in Table 3 below.

Feature Category	Description
Word	Features include IDs for current and contextual tokens, hamza, and whether the word is capitalized, has digits, has punctuation, or has ASCII characters
Word cluster	Cluster-prefix features that capture a word's cluster ID at four hierarchical levels for current and contextual tokens.
Prefix/Suffix	1-, 2-, and 3-character prefix and suffix features
SentiWordNet (SWN)	Features based on polarity and strength of sentiment scores in SentiWordNet's subjectivity lexicon [12]
Part-of-speech (POS)	Features indicating one of 37 distinct parts of speech.
N-gram features	Bigram and trigram features that extend above unigram features to capture sequence combinations
SWN/POS features	Features combining SWN and POS information

Table 2. Descriptions of the features extracted in our sentiment tagging model

4.3 Geolocation Tagging

We employ the content-based geolocation tagger described in [5], which takes a three-phased approach, following tokenization, language ID, and MT as in Fig. 1.

Phase 1. To provide more context for analysis, we use a document creation algorithm described in [5] that groups tweets by user and frequency. The algorithm considers a time window size and minimum number of tweets per document. The English (source plus MT) tweet documents are then processed through an NER component [10].

Phase 2. This phase selects location records from gazetteers based on the list of named entities from Phase 1. The locations are assigned an initial score based on the record features and external features; the scores are used for weighting in k-means clustering, which in turn is used to rescore location points based on distance to their cluster's center. Scores are then used to associate location identities with names.

Phase 3. The final phase selects the most representative location for the document. Features are extracted for the location identities and new scores are assigned to each location. The highest ranking location becomes the document-level geolocation.

5 Results and Discussion

5.1 Sentiment Performance

We applied our sentiment decoder trained on newswire domain data to the Arabic tweet test set for a baseline performance of 22.5 f-measure. The relatively poor baseline performance may be due to observations made earlier of how sentiment expression differs across domains, the writing style of Arabic users, or issues of translation. We then incorporated the tweet training data. With only 12k tokens, this is an increase of less than 5% over the 281k tokens in the baseline training. While performance on newswire degrades 11% relative, as Table 3 shows, the social media data was sufficient to improve the f-measure on the Arabic tweets by ~50% relative. We note that the generally low accuracy measures stem from the exact phrase alignment scoring. If the hypothesized phrase is a word off from the reference phrase, the system is penalized. This method, however, provides a more rigorous evaluation.

Test Set	Training Data	Recall	Precision	F-measure
MPQA	MPQA	33.0	38.0	35.3
	MPQA + Tweets	34.0	30.0	31.2
Tweet	MPQA	18.0	30.0	22.5
	MPQA + Tweets	30.0	37.0	33.3

Table 3. Comparison of Sentiment tagging performance on newswire and social media

5.2 Geolocation Performance

We conducted three geolocation experiments by varying the tokenization in MT and NER to improve the data seen by the geocoder model. The three conditions we tested are as follows: (1) whitespace tokenization in MT and NER; (2) social media tokenization in MT, whitespace tokenization in NER; and (3) social media tokenization in MT and NER. The results of these experiments are summarized in Table 4.

Expt	AveErrDist	Acc100	Acc500	Acc1000
1	2,072	10.9	32.3	49.9
2	1,995	11.7	33.2	51.8
3	1,987	11.3	31.4	52.4

Table 4. Impact of social media tokenization on geolocation tagging in multilingual tweets.

While the gains in AveErrDist are marginal, the improving trend indicates that customizing preprocessing steps for social media can have an impact on geolocation. We note that the geolocation references are based on physical location, rather than content. The improving trend in AveErrDist suggests that these references are useful for benchmarking geolocation; however, the accuracy measures may not be a true representation of the geocoder's performance.

6 Application: tracking sentiment in space and time

[5] demonstrated the application of content-based geolocation by searching a database of tweets by keyword and displaying the results on a map. We now add a new sentiment dimension. We searched by keyword "World Cup", filtered for positive and then negative sentiment, displaying the top results on a map in Fig. 2. We see a cluster of positive tweets (686) in Barcelona, for example: "RT @news_kuw1: Barcelona player, Messi, gave a grant to Israel a million dollars to attain his country's team at the World Cup, as he promised in the #Gaza Strip." Filtering for negative sentiment, we see a cluster of tweets in Germany, including: "@ShamiWitness the whole world sinks into war in these hours and days,only here in Germany people still celebrate the world cup win, :) :)".

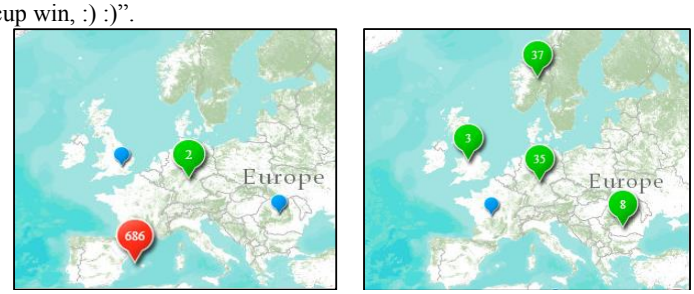


Fig. 2. Geographic distribution of tweets with keyword "World Cup" filtered for positive sentiment (left) and negative sentiment (right).

We can also analyze how sentiment changes over time. The left plot in Fig. 2 shows the count of tweets over a two-week period containing hashtag #Iraq with topic Iraq; we see a spike in tweets around June 13-15th, coinciding with Iraqi counterattacks to ISIS forces in northern Iraq. The 13th has a surge in negative tweets, and two days later, positive tweets dominate. We do not see this shift among tweets with topic Iraq but Arabic hashtag #العراق, shown in the right plot. There, we see more negative than positive tweets. Given that ISIS uses sophisticated social media tactics to control its image, these observations could cue an analyst to delve further into the data.

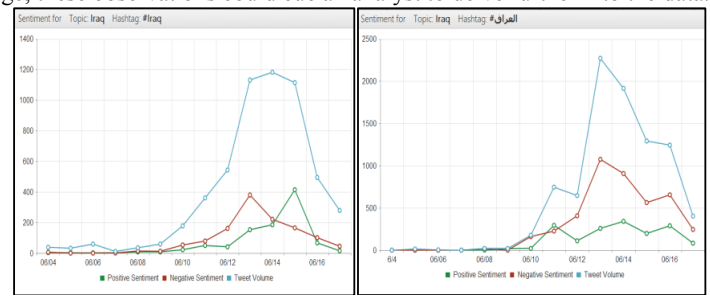


Fig. 3. Tweet volume and positive (green) and negative (red) sentiment of tweets over two weeks, with topic Iraq and hashtag #Iraq (left) or the Arabic hashtag for Iraq, # العراق (right).

7 Conclusion

We examined differences in newswire versus social media domains, noting significant differences in language expression. We customized content-based sentiment tagging and geolocation tagging systems to be social media aware; we demonstrate that systems developed for social media outperform those for traditional news domain, as evaluated on tweet data sets. We improved sentiment tagging by nearly 50% relative over a newswire baseline by incorporating a small amount of in-domain training data. We improved geolocation modestly by pre-processing data with a social media specific tokenization schema during machine translation and named entity recognition. Finally, we demonstrate the value to analysts of adding a sentiment dimension to geolocated tweets on a world map, and of tracking sentiment of filtered tweets over time.

Acknowledgements. This paper presents work funded by the Combating Terrorism Technology Support Office (CTTSO), under COMET contract N41756-11-C-3878.

References

1. Aston, N., Liddle, J., Hu, W.: Twitter Sentiment in Data Streams with Perceptron. Journal of Computer and Communications, vol. 2 (2014)
2. Haimovitch, Y., Crammer, K., Mannor, S.: More is Better: Large Scale Partially-supervised Sentiment Classification. In: JMLR: Proceedings of ACML (2012)
3. Breck, E., Choi, Y., Cardie, C.: Identifying Expressions of Opinion in Context. In: Proceedings of IJCAI-2007 (2007)
4. Wiebe, J., Wilson, T., Cardie, C.: Annotating expression of opinions and emotions in language. Language Resources and Evaluation, vol. 39, pp. 165-210 (2005)
5. Khanwalkar, S., Seldin, M., Srivastava, A., Kumar, A., Colbath, S.: Content-Based Geo-Location Detection for Placing Tweets Pertaining to Trending News on Map. In: 4th International Workshop on MUSE, ECML (2013)
6. Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S., and Weischedel, R.: The Automatic Content Extraction (ACE) Program-Tasks, Data, and Evaluation. In: LREC (2004).
7. Cavnar, W., Trenkle, J.: N-Gram-Based Text Categorization. In: Proceedings of SDAIR-94, 3rd Annual Symposium (1994)
8. Kiecza, D., Srivastava, A., Saikumar, G., Colbath, S. Lillie, M., Natarajan, P., Echihabi, A., Marcu, D., Wong W.: Operational Integration of STT and MT for Rich Transcription and Translation of Speech. In: Olive, J., Christianson, C., McCary, J. (eds.) Handbook of Natural Language Processing and Machine Translation: DARPA GALE, Springer (2011)
9. Collins, M.: Ranking algorithms for named-entity extraction: boosting and the voted perceptron. In: Proceedings of the 40th Annual Meeting on ACL, ACL (2002)
10. Miller, S., Guinness, J., Zamanian, A.: Name Tagging with Word Clusters and Discriminative Training. In: Proceedings of ACL Annual Meeting, vol. 4 (2004).
11. Brown, P.F., deSouza, P.V., Mercer, R.L., Della Pietra, V.J., Lai, J.C.: Class-based N-gram Models of Natural Language. Comput. Linguist., vol. 18, pp. 467-479 (1992)
12. Baccianella, S., Esuli, A., Sebastiani, F.: SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In: LREC, vol. 10 (2010)