ECML/PKDD 2013

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

The Fourth International Workshop on Mining Ubiquitous and Social Environments

MUSE'13

September 23, 2013

Prague, Czech Republic

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 4^{th} International Workshop on Mining Ubiquitous and Social Environments (MUSE 2013) is held in Prague, Czech Republic, on September 23^{rd} 2013 in conjunction with the The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2013).

The emergence of ubiquitous computing has started to create new environments consisting of small, heterogeneous, and distributed devices. These foster the social interaction of users in several dimensions. Mining in ubiquitous and social environments is an established area of research focusing on advanced systems for data mining in such distributed and network-organized systems. However, the characteristics of ubiquitous and social mining are in general rather different from common mainstream machine learning and data mining approaches. Therefore, the analysis of the collected data, the adaptation of well-known data mining and machine learning approaches, and finally the engineering and development of new algorithms are challenging and exciting steps.

The goal of this workshop is to promote an interdisciplinary forum for researchers working in the fields of ubiquitous computing, social web, Web 2.0, and social networks which are interested in utilizing data mining in a ubiquitous setting. The workshop presents contributions adopting state-of-the-art mining algorithms for analyzing ubiquitous social data, to challenging applications, and open datasets. With this workshop, we thus want to accelerate the process of identifying the power of advanced data mining operating on data collected in such ubiquitous and social environments.

Submissions and Sessions

This proceedings volume comprises the papers of the MUSE 2013 workshop. In total, we received twelve submissions, from which we were able to accept five submissions, two full papers and three short papers, based on a rigorous reviewing process. Additionally, the scientific program also featured an invited talk on the mining of information propagation traces in social networks by Francesco Bonchi (Yahoo! Research, Barcelona, Spain). Based on the set of accepted papers, and the invited talk, we set up three sessions. The first session features the invited talk by Francesco Bonchi.

The second session is concerned with mining ubiquitous and social environments. The work *Aperiodic and Periodic Model for Long-Term Human Mobility Prediction Using Ambient Simple Sensors* by Danaipat Sodkomkham discusses limitations in predictability of collective human mobility. Next, the paper *Subgroup Analytics and Interactive Assessment on Ubiquitous Data* by Martin Atzmueller and Juergen Mueller discusses the discovery and assessment of descriptive patterns using subgroup analytics. The context of this work is given by ubiquitous data, specifically noise measurements, assigned tags, and subjective perceptions. Finally, *Open Smartphone Data for Mobility and Utilization Analysis in Ubiquitous Environments* by Jochen Streicher, Nico Piatkowski, Katharina Morik and Olaf Spinczyk presents and discusses a new collected mobility dataset.

The third session starts with the work *Content-Based Geo-Location Detection for Placing Tweets Pertaining To Trending News on Map* by Saurabh Khanwalkar, Marc Seldin, Amit Srivastava, Anoop Kumar and Sean Colbath. In this paper, the authors present a novel approach for content-based geo-location. Concluding the session the paper *Learning the Shortest Path for Text Summarisation* by Emmanouil Tzouridis and Ulf Brefeld presents new insights in learning shortest paths in compression graphs for summarising related sentences.

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 and external referees for their thorough work in reviewing submitted contributions with expertise and patience. Also, we wish to thank the ECML PKDD Workshop Chairs and the members of ECML PKDD Organizing Committee who made this event possible. Special thanks go to Michelangelo Ceci for his help in organizing an independent review process of a selected publication.

We are looking forward to a very exciting and interesting workshop.

Kassel, September 2013

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
Ciro Cattuto	ISI Foundation, Italy
Michelangelo Ceci	University of Bari, Italy
Jill Freyne	CSIRO, Australia
Daniel Gayo Avello	University of Oviedo, Spain
Ido Guy	IBM Research
Andreas Hotho	University of Wuerzburg, Germany
Kristian Kersting	Fraunhofer IAIS and University of Bonn, Germany
Florian Lemmerich	University of Wuerzburg, Germany
Claudia Mueller-Birn	FU Berlin, Germany
Haggai Roitman	IBM Research Haifa, Israel
Giovanni Semeraro	University of Bari, Italy
Philipp Singer	Graz University of Technology, Austria
Maarten van Someren	University of Amsterdam, The Netherlands
Gerd Stumme	University of Kassel, Germany
Arkaitz Zubiaga	City University of New York, USA

Invited Talk

Mining information propagation traces in social networks

Francesco Bonchi, Yahoo! Research, Barcelona, Spain

With the success of online social networks and microblogging platforms such as Facebook, Flickr and Twitter, the phenomenon of influence-driven propagations, has recently attracted the interest of computer scientists, information technologists, and marketing specialists.

In this talk we take a data mining perspective and we discuss what (and how) can be learned from a social network and a database of traces of past propagations over the social network. Starting from one of the key problems in this area, i.e. the identification of influential users, by targeting whom certain desirable marketing outcomes can be achieved, we provide an overview of some recent progresses in this area and discuss some open problems.

Biography Francesco Bonchi is a senior research scientist at Yahoo! Research, where he is leading the Web Mining Research group. His recent research interests include mining query-logs, social networks, and social media, as well as the privacy issues related to mining these kinds of sensible data. In the past he has been interested in data mining query languages, constrained pattern mining, mining spatiotemporal and mobility data, and privacy preserving data mining.

He is member of the ECML PKDD Steering Committee, member of the Editorial Board of ACM Transactions on Intelligent Systems and Technology (TIST), and Associate Editor of IEEE Transactions on Knowledge and Data Engineering (TKDE). He's also the organizer of the *Yahoo! Research Barcelona Seminars* series. He has been program co-chair of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2010). Dr. Bonchi has also served as program co-chair of the first and second ACM SIGKDD International Workshop on Privacy, Security, and Trust in KDD (PinKDD 2007 and 2008), the 1st IEEE International Workshop on Privacy Aspects of Data Mining (PADM 2006), and the 4th International Workshop on Knowledge Discovery in Inductive Databases (KDID 2005). He earned his Ph.D. in computer science from the University of Pisa in 2003.

APP: Aperiodic and Periodic Model for Long-Term Human Mobility Prediction Using Ambient Simple Sensors

Danaipat Sodkomkham, Roberto Legaspi, Ken-ichi Fukui,
Koichi Moriyama, Satoshi Kurihara, and Masayuki Numao

Institute of Scientific and Industrial Research, Osaka University, Japan,
danaipat@ai.sanken.osaka-u.ac.jp

Abstract. The predictive technique proposed in this project was initially designed for the indoor smart environment wherein intrusive tracking techniques, such as cameras, mobile phone and GPS tracking system, could not be utilized appropriately. Instead, we installed simple motion detection sensors in the experimental space and observed occurred movements at each area. However, the movement data recorded with this setting cannot provide as much information about human mobility as the data from the GPS or mobile phone is capable of. In this paper, we conducted an exhaustive analysis on this specific dataset to determine the predictability of future users' mobility using only this limited dataset and regardless of the predictive technique. Furthermore, we also proposed the predictive technique, named APP, for long-term human mobility prediction that works well on our limited dataset. Finally, evaluation on the real dataset collected inside the smart space over 3 months of movements and daily activities data shows that our model is able to predict future mobility and activities of participants in the smart environment setting with high accuracy even for a month ahead.

Keywords: Human mobility, Smart environment, Long-term prediction

1 Introduction

Understanding and predicting human mobility are crucial components in a number of real world applications. We will mention a few examples here. The PUCK architecture[4] was introduced to intelligently provide reminders in the smart environment since it automatically recognizes habitual activities and then reminds the occupant if he/she forgot some important tasks, such as forgetting to take a medicine after a meal. This could be helpful for participants who have dementia or mind cognitive impairment. Moreover, the ability to predict future locations of people is also an important element in transportation planning [13, 11], bandwidth provisioning in wireless local area network [18], and targeted advertisement dissemination [7].

In our specific example, we have an actual office environment built-in with various sensors and actuators to enable the pervasive computing technology to

control different settings of the environment. Our prototype smart office environment was initially designed to create a working environment that can learn users' behavioral activities and react to these activities smartly. Goals of our developing smart environment is to simplify mundane repetitive tasks, and to make the participant live more comfortable. For example, the smart office that predicts future occupancy of the meeting room and automatically gets electronic facilities in the room prepared right before the meeting. The smart office that predicts participants' needs from their daily activities so that it always has hot coffee ready to be served at the time they need. All of the applications mentioned above require the ability to foresee user's future whereabouts and mobility into far future, as known as the open problem of long-term human mobility prediction.

A smart environment, normally, has sensors installed to sense activities and mobilities of participants inside. Different machine learning algorithms are then employed to explore meaningful information about user behaviors and routines. These information will be later used to build a predictor that foresees users' needs and suggest a proper reaction to them. Therefore, one challenging problem for every anybody who works on smart environments researches would be *"to determine the best approach to observing participants' mobility with the least obtrusiveness while providing enough information to build an accurate predictive model"*.

There is apparent trade-off between informativeness and conspicuousness of the sensing technique. For a concrete example, using colored pictures from cameras with a help from image processing technique and semi-supervised classification algorithm, Yu et al. [19] was able to create a system that recognizes people and their positions. Moreover, they were able to map each individual's movement directly into a floor map. From this interesting example, rich mobility information for individual users must be traded with users' discomfort because of surrounding cameras. Apart from camera techniques [1, 3] discussed earlier, mobile phone data [6, 17], GPS [16, 13], and RFID tagging [2, 10] requires users to carry (or put on) the tracking device while inside the environment, which is not feasible in real-world implementation. On the other hand, simple sensors such as infrared distance sensor, ultrasonic distance sensor, and magnetic sensor, are small enough to blend into the environment, and seamlessly observe human mobility inside the environment. It is the case, however, that simple sensors' data is less informative than such high precision sensing technologies and they limit the capability of the mobility predictive model that was built using their data.

Therefore, in this paper, we investigated limits of the predictability over this specific type of mobility dataset. Furthermore, in the latter part of this paper, we present a novel prediction technique, named APP, particularly for the long-term human mobility prediction problem. More specifically, the APP predicts future location of a user at any specific *time frame* in far future, e.g. 21 days from now between 10:00 and 10:05. The prediction is obtained by probabilistic models that compute how likely a certain location will be revisited in the future at the specific

time frame. The prediction part in APP consists of two probabilistic models. Both probabilistic models keep track of visited time stamps, extracts contextual features from each visit (such as what time of the day, what day of the week, or how long after the last visit), and model their relations. The first predictive model, named *Periodic model*, is based on a hypothesis that user keeps visiting some specific locations periodically, such as every three hours, everyday, or every month. Firstly, we analyze periodicity at each location and test this hypothesis. If the hypothesis is accepted, we use the periodic model to predict; otherwise, the second model is used. The second predictive model does not rely on periodicity property in human mobility behavior; instead, it extracts significant patterns of repetitive movements representing user mobility behavior in the past. Hence, it is called the *Aperiodic model*. The aperiodic model postulates that the same (or similar) mobility pattern tends to repeat again at any specific time frame in the future whenever their contextual features are similar. Combining these two models (APeriodic and Periodic, abbreviated to APP) results in a predictor that predicts user's location with acceptably high accuracy and precision, even for a month ahead.

2 Limits of Predictability in Collective Human Mobility

The breakthrough analysis of predictability of human mobility has been studied in [17]. Song et al. explored the limitation in predictability of individual's movements, disregarding quality of the prediction techniques. Despite the difference in user's daily behavior, their analysis over a large population monitored by their mobile phone data shows 93% potential predictability in individual's mobility. In other words, predicting individual's movements can be achieved effectively when history data of individual's movements is available.

When it comes to the situation when historical data of individual's user is not provided, but collective mobility data from multi-users, the fundamental question of predictability on this class of data arises again, i.e.*to what degree is collective mobility predictable?*

2.1 Collective Human Mobility Data

For our experimental smart environment, we have a functioning working space, which includes individuals' cubicle work stations, recreational space and meeting areas, where a total number of 20 graduate students and faculty staff members come to work regularly. Each individual may has different duties, different class schedules and different daily routine, which result in different mobility patterns directionally and temporally. We installed different types of sensors (see figure 5(a)) in the environment to monitor activities and movements that happened inside. Specifically, we used two types of sensors in the experiment. First, infrared distance sensors were mainly used to detect movement at each specific location. Second, magnetic sensors were attached to the hinge of the refrigerator and the oven to observe their usages. All of these sensors are connected

through the lab's network and continuously feed live streams of mobility data to a database. By employing these ambient sensors, participants needed not to be equipped with a tracking device during the experiment period and can normally move along without any concern of being monitored. We observed visitations and mobility inside the experimental environment over 24 hours a day, for 3 months (precisely 92 days) during the autumn semester. The floor plan in figure 5(b) shows placements of sensors used to capture visitations at each location in the space. The number of interested locations N is 30. We modeled human mobility with two representations for different purposes as follows.

As temporal sequences of repeated visitations. Collective mobility, at each location, is represented with the temporal sequence of repetitive visitations visited by unknown users during the observing period. State of visitation at a moment is denoted by a binary value: either 1 for *visited*, or 0 for *not visited*. For instance, a sequence v_x represents mobility at location x from 00:00 to 23:59, with the sample rate μ of 1 sample per hour.

$$v_x = \big((t'_0, 0), (t'_1, 1), (t'_2, 0), \ldots, (t'_{23}, 1)\big)$$

when t'_i represents the observed *time frame* from $t_0 + i\mu$ to $t_0 + (i+1)\mu$, and t_0 is the starting time, i.e., $t_0 = 00{:}00$ and $t'_0 = [00:00,\ 01:00)$.

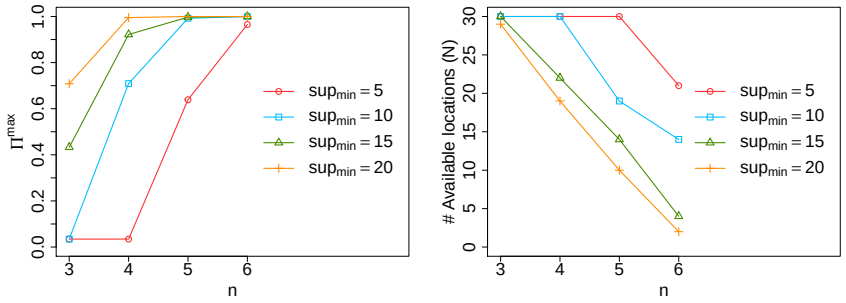
As trajectories. By increasing sample rate of the sensors μ up to 1 sample per 200 milliseconds, we were able to count every visitations. Then, from a temporal sequence of visitations, $((x_0, t_0), (x_1, t_1), \ldots, (x_{w-1}, t_{w-1}))$, we linearly searched for each transition point in the sequence where the transition time $t_{i+1} - t_i > 30$ seconds to cut it into smaller sequences that represent trajectories.

Despite unobtrusiveness and simplicity of the ambient sensing method, the obtained data is primarily noisy. To handle noises (such as false triggered events, sensors blocked by obstacles, and simultaneous trajectories from different users) and extract movement trajectories from the collective mobility dataset efficiently, we applied the data mining algorithm, called PrefixSpan [9], to extract only sub-trajectories of length-n that appeared in T more frequently than a certain minimum number of times $support_{min}$ during the experiment.

2.2 Limits of Predictability

Here we evaluated the predictability over the collective mobility dataset using the same methodologies introduced by Song et al. in [17]. Namely, by employing Fano's' inequality [5, 14], we estimated the upper limit of the probability of the destination of a moving user can be predicted correctly given the most recent trajectory and the past collective mobility data.

Let T'_i denote a movement trajectory and let D_i be a destination of T'_i from the observations, $T = ((T'_0, D_0), (T'_1, D_1), \ldots, (T'_m, D_m))$. Given a predictive technique $f(T'_i)$ that works well in predicting future location D_i of a moving



(a) The probability Π^{max} as functions of trajectory length n

(b) Relation between the parameter $support_{min}$, trajectory length n, and the number of predictable destinations N left after noise removal

Fig. 1. The predictability of the collective human mobility in the smart environment. The Π^{max} is the upper bound of the probability that a particular predictive algorithm is able to predict user's location correctly using only the collective dataset.

user based on recent length-n movement trajectory T' and a set of length-n trajectories T from historical mobility data, let e denote the event of failed prediction, i.e. $f(T') \neq D$, and let $P(e)$ be its probability. According to Fano's inequality, the lower bound on the error probability $P(e)$ can be found in the following inequality.

$$H(D|T') \leq H(e) + P(e)\log(N-1)$$

Thus, the probability of predicting correctly, denoted by Π, is $1-P(e)$. Namely,

$$H(D|T') \leq H(e) + (1-\Pi)\log(N-1), \tag{1}$$

where the destination D can take up to N possible locations and $H(e)$ is the corresponding binary entropy which is defined as follow.

$$\begin{aligned} H(e) &= -P(e)\log(P(e)) - (1-P(e))\log(1-P(e)) \\ &= -(1-\Pi)\log(1-\Pi) - \Pi\log(\Pi) \end{aligned} \tag{2}$$

The conditional entropy $H(D|T')$ appeared in the inequality (1) quantifies the amount of information needed to predict the destination D given the correlated recent trajectory T'. Given the probability $P(T')$ of the set of past trajectories T containing T' and the joint probability $P(T', d)$, the conditional entropy $H(D|T')$ is defined as follow.

$$H(D|T') = \sum_{d \in D, T' \in T} P(T', d) \log\left(\frac{P(T')}{P(T', d)}\right) \tag{3}$$

Then we calculated the entropy $H(D|T')$ individually for each length n of trajectories in T, and analyzed the maximum potential predictability (denoted

by Π^{max}) or the probability of predicting correctly destination of a user given the collective mobility dataset by solving for the Π^{max}, where $\Pi \leq \Pi^{max}$ in the following equation, according to (1), (2) and (3).

$$H(D|T') = -(1-\Pi^{max})\log(1-\Pi^{max}) - \Pi^{max}\log(\Pi^{max}) + (1-\Pi^{max})\log(N-1)$$

Figure 1(a) shows Π^{max} as functions of n, where n denotes length of the considering trajectories. It is unsurprising that n increases the predictability since longer trajectory gives more evidences to the predictor that help narrowing down search space of the most probable locations. The $support_{min}$ also shows its potential to eliminate unusual trajectories in the dataset, and gives significantly higher potential predictability. However, there is a trade-off between the degree of predictability and the number of predictable locations, as Figure 1(b) shows. High threshold of the minimum support ($support_{min}$) results less number of locations N available to the predictive algorithm to predict.

To summarize from the analysis, despite the fact that the collective human mobility contains cumulative movements and behaviors from different users and seems diverge to the experimenter in the first place, the accurate prediction of user's location is achievable with acceptably high probability. However, this analysis does not provide any clue about the potential predictability in the long-term prediction configuration when the inference of user's mobility cannot rely on recent movement patterns and frequent historical trajectories. Moreover, a number of researches [16, 15] have shown that predictive techniques that work well in the short-term human mobility prediction cannot be extended to the long-term prediction effectively. Thus, in the next section, we studied the possibility to employ the periodicity in human behavior to foresee their mobility in far future instead of directly modeling trajectories.

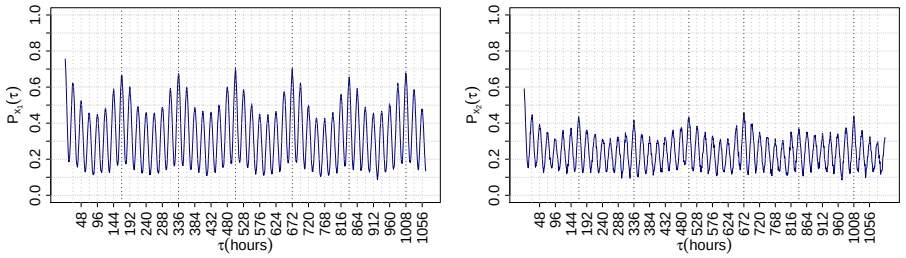
3 Periodicity in Collective Human Mobility

It can be seen easily even without a guide from data mining tool that most of human activities are periodic to some degree. If a certain action, or movement pattern is repeated regularly with a particular interval τ, and if this behavior is consistent over time, it is certainly predictable with the time period τ. In addition, the probability in predicting the correct location of a user in the future depends on the tendency of such mobility patterns recurring at intervals. Thus we define the *periodicity probability* to formally quantify this property.

Definition 1. *Let $P_x(\tau)$ denotes the periodicity, which is the probability of a particular event x reoccurring regularly with the constant time interval τ, where τ is a positive integer. Given the temporal sequence, as described in section 2.1, of events from t'_0 to t'_m in which the location x was visited, the periodicity probability $P_x(\tau)$ is defined as follow.*

$$P_x(\tau) = P\big(v_x(t'_{i+\tau}) = 1 | v_x(t'_i) = 1\big),\ t'_i \in \{t'_0, t'_1, \ldots, t'_{m-1}\} \tag{4}$$

where $v_x(t'_i)$ indicates state of the visitation at x during the time frame t'_i.



(a) The periodicity $P_{x_1}(\tau)$ (b) The periodicity $P_{x_2}(\tau)$

Fig. 2. The periodicity $P_{x_1}(\tau)$ and $P_{x_2}(\tau)$ of repetitive visitations at the location x_1 and x_2 as a function of time period τ.

To find significant periodicity in the collective human mobility, we searched for τ that maximizes the periodicity probability at each location separately. Figure 2 shows two sample locations where periodic behavior can be observed. The small peaks in these plots reveal relatively high probability that these particular locations were visited regularly with the time period τ, when τ are multiples of 24 hours. Moreover, the maximum of predictability probabilities are found at multiples of 168 hours. Undoubtedly, this indicates firm evidences of *daily* and *weekly* behaviors exist in the collective mobility data. With more algorithmic way of finding significant period τ, the Fourier analysis also suggested that $\tau = 24$ hours and 168 hours correspond to two most significant frequencies of $\approx 4.167 \times 10^{-2}$ Hz and $\approx 5.925 \times 10^{-2}$ Hz respectively.

In the next section, we analyze the possibility of the collective human mobility being predictable with the periodic behavioral patterns.

3.1 Predictability of The Periodic Model

Intuitively, the periodicity $P_x(\tau)$ already estimated the precision of a periodic-based predictive model, which is based on a strong assumption of periodically repeated visitations. Hence, the periodicity $P_x(\tau)$ can be considered as a measurement for the predictability of the periodic model. In addition, we also want to provide another predictability analysis employing an academic concept in information theory to the periodic model.

Firstly, we assign the *periodic entropy* to the history data of repetitive visitations at each location to determine the amount of information needed to foresee future visit given records of repetitive visitations in history. At each location x, the periodic entropy is computed as follows.

Definition 2. *Given the collective mobility data, the entropy S_x^τ which quantifies the degree of uncertainty of the periodicity $P_x(\tau)$ in the dataset is*

$$S_x^\tau = \sum_{\nu \in \{0,1\}} P(v_x) H(v_x(t'_{i+\tau} | v_x(t'_i) = \nu)),\ t'_i \in \{t'_0, \ldots, t'_{m-1}\}, \tag{5}$$

where $P(v_x)$ is the probability of a location x being visited, and the conditional entropy $H(v_x(t'_{i+\tau}|v_x(t'_i) = \nu)$ is

$$H(v_x(t'_{i+\tau}|v_x(t'_i) = \nu) = \sum_{\varphi \in \{0,1\}} P(\varphi|\nu) \log\left(\frac{1}{P(\varphi|\nu)}\right), \tag{6}$$

where $P(\varphi|\nu)$ stands for $P(v_x(t'_{i+\tau}) = \varphi|v_x(t'_i) = \nu)$.

Additionally, let S^τ_{xf} be the entropy of future visitations; namely,

$$S^\tau_{xf} = -\sum_{\varphi \in \{0,1\}} P(v_x(t'_{i+\tau}) = \varphi) \log(P(v_x(t'_{i+\tau}) = \varphi)),\ t'_i \in \{t'_0, \ldots, t'_{m-1}\} \tag{7}$$

Next, we determine the predictability for each location x of the periodic model with the probability $\Pi_{x,\tau}$, which is defined as follow.

Definition 3. *Let $\Pi_{x,\tau}$ be the probability that the periodic model predicts times of future visitations at x correctly by always predicting visited at all moments that are $k\tau$ apart from the last visit, for $k = 1, 2, \ldots$. Thus the associated entropy $H(\Pi_{x,\tau})$ of the predictability $\Pi_{x,\tau}$ is*

$$H(\Pi_{x,\tau}) = -\Pi_{x,\tau} \log_2(\Pi_{x,\tau}) - (1 - \Pi_{x,\tau}) \log_2(1 - \Pi_{x,\tau}). \tag{8}$$

The maximum predictability $\Pi^{max}_{x,\tau}$ can be determined using the Fano's inequality in accordance with (1).

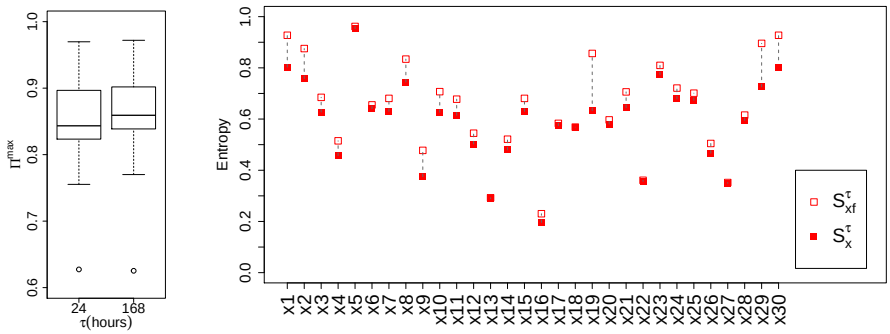
$$S^\tau_x \leq H(\Pi_{x,\tau}) + (1 - \Pi_{x,\tau}) \log_2(N - 1) \tag{9}$$

Because $\Pi_{x,\tau} \leq \Pi^{max}_{x,\tau}$ and $N = 2$ prevents this bound to the binary classification, then the following correction is required.

$$\begin{aligned} S^\tau_x &\leq H(\Pi_{x,\tau}) + (1 - \Pi_{x,\tau}) \log_2(N - 1) \leq H(\Pi_{x,\tau}) + (1 - \Pi_{x,\tau}) \log_2(N) \\ &= -\Pi^{max}_{x,\tau} \log_2(\Pi^{max}_{x,\tau}) - (1 - \Pi^{max}_{x,\tau}) \log_2(1 - \Pi^{max}_{x,\tau}) + (1 - \Pi_{x,\tau}) \log_2(N) \end{aligned} \tag{10}$$

After solving for $\Pi^{max}_{x,\tau}$ in (10), the predictability $\Pi^{max}_{x,\tau}$ determines the upper limit of the probability of predicting future visits of users at location x in far future given an appropriate periodic model (with the time period τ). We evaluated S^τ and Π^{max}_τ separately for each location, and the associated distribution of Π^{max}_τ is shown in figure 3(a). Both distributions of the predictability Π^{max}_τ indicate the average predictability over all locations approximately above 80%, in both daily and weekly model. The average predictability of the weekly model is slightly higher and has lower variance than the daily model. One may conclude from the result that the weekly model fit the collective mobility data better than the daily periodic model.

Figure 3(b) shows differences between the periodic entropy $S^{\tau=24}_x$ and the entropy of future visitations $S^{\tau=24}_{xf}$ at each location x. Note that as S^τ_x closer to zero



(a) The predictability Π_τ^{max} of periodic model.

(b) Differences between the periodic entropy and the entropy of future visits at each location.

Fig. 3. The predictability Π_τ^{max} and its corresponding periodic entropy.

and further from S_{xf}^τ, future visitations are more likely to depend on previous visitations periodically. Result from this figure clearly suggests that not visiting behavior at every location in our environment is periodic. For instance, the locations $x_7, x_{14}, x_{17}, x_{23}$, and x_{28} were not periodic, while the locations x_1, x_2, and x_{20} appeared to be more periodic than others. Therefore, the periodicity-based predictive model alone would not work in every location; hence, we have developed the integrated aperiodic and periodic model for long-term human mobility prediction.

4 APP: Aperiodic and Periodic model for long-term human mobility prediction

Our long-term human mobility predictive model combines two predicting paradigms together. The first approach (Periodic approach) employs the periodic property in human mobility to foresee future visits. On the other hand, the second approach (Aperiodic approach) does not rely on the periodicity; instead, it presumes that mobility patterns are similar to the day in the past that has similar features. The APP uses either one of the two approaches to predict human mobility at a certain location x depending on the periodicity probability $P_x(\tau)$ at that specific location x. If $P_x(\tau)$ is more than the user-specific threshold P_{min}^τ, then the APP uses the periodic approach. Otherwise, it switches to the aperiodic approach.

4.1 APP: The Periodic Approach

The APP with the periodic predictive approach was designed to foresee times of future visitations at each location in the smart space. To predict future locations

of multi-users, the predictions are computed independently for each location, then all the results are combined together providing a set of locations that are likely to be visited at the specific time in far future.

The fundamental idea behind the prediction is based on the assumption of periodicity. Say, if the visitations at x recur regularly, again and again, with a constant time interval τ, and if this periodic behavior appears consistently over time, then the probability $P_x^\tau(t'_f)$ that the future visitation will occur within the time frame t'_f in the future, when the last visit happened at t'_{m-1}, can be computed as

$$P_x^\tau(t') = P(v_x(t'_{m-1-(k)\tau+\delta}) = 1 | v_x(t'_{m-1-(k+1)\tau+\delta}) = 1),\ k = 1, 2, \ldots, \lfloor m/\tau \rfloor$$

where $\delta = (f - m + 1) \bmod \tau$.

This simple, yet accurate, predictive approach works well only at certain locations, where users' mobility has apparent periodicity. Otherwise, the periodic approach gets poor prediction because the mobility in those particular locations are not governed by the periodic behavior. To address this problem, we proposed the optional predictive approach contributed to the APP that is independent of the periodic behavior.

4.2 APP: The Aperiodic Approach

In this second predictive technique implemented in the APP, we extract significant patterns of repetitive visitations at each location that happened on different days. Next, the days that have similar visiting pattern are clustered together, resulting groups of *similar days* in the past history that users behaved similarly. Then, we extract contextual features from each group of similar days. In this project, the interested features consist of: (1) what day of week, (2) whether it is a holiday or not. Note that unlimited additional features that might relate to the mobility pattern can be used to characterize the day more comprehensively, such as temperature, traffic, weather condition, or meeting schedule. However, due to the limit in the dataset we have, only these two features are applied.

The intuition that supports this predictive approach is derived from the weekly model in section 4.1, human mobility patterns on the same *day of the week* are likely similar. Additionally, human activities on national holidays are apparently different from normal workdays; so, we need an additional bit to explicitly specify this property. Hence, mobility pattern of *a day* can be modeled individually by the visitations at each location. Recall the temporal sequence v_x in section 2.1, mobility at a certain location x can be represented with a vector:

$$\begin{aligned} d_x =& [v_x, day_{week}, holiday] \\ =& [v_{t'_0}, \ldots, v_{t'_{23}}, Sun, Mon, \ldots, Sat, Hol] \end{aligned}$$

The day vector d_x consists of 32 bits. The first 24 bits model visitations at location x during a specific time frame of a day, which is divided hourly. The next 7 bits indicate day of week, and the last bit indicates a holiday.

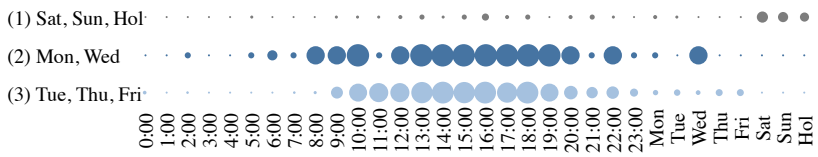


Fig. 4. Three cluster centroids that represent three mobility patterns.

Similarity between two day vectors is basically measured by the Hamming distance [8]. Then the k-means clustering algorithm [12] is applied to a set of day vectors to find clusters of similar days. The parameter k of the algorithm directly implies to the number of different mobility patterns that happened on different days. The centroid of each cluster now represents common mobility pattern that provides predictions (probability) of visitations to be found on any certain days in the future that have similar features. A concrete example of the similar day clusters discovered from the real dataset is shown in figure 4. It is striking that the cluster centroids in figure 4 clearly show 3 different visiting patterns at that particular location. The cluster (1) contains a set of days in the past history when the visitations rarely happened, and the majority of this set are Saturdays, Sundays and the days specified as holiday. On the other hand, the cluster (2) and (3) contain more active days. The days in the cluster (2), which most of them are Monday and Wednesday, have very low visitations records during 11.00-12.00 and 21.00-22.00; moreover, the visitations seems to occur earlier than on the days in the cluster (3). Interestingly, this follows the fact that we have meetings arranged in the experimental space every Monday and Wednesday, and causes the mobility pattern to appear differently to other days.

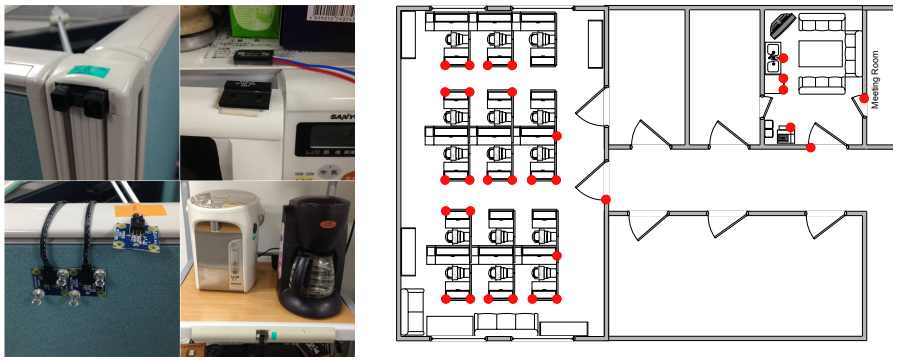
5 Evaluating Prediction Performance

In this section, we evaluated prediction performance of our proposed long-term human mobility predictor on a physical dataset of collective human mobility inside the working environment. As described in section 2.1, the dataset contains 92 days of mutual movements from every participants in the space. Data are collected consecutively 24 hours a day, 7 days a week from $\sim$ 20 users using infrared sensors, and magnetic sensors (figure 5(a)). These sensors were installed at 30 locations over the experimental space to detect activities and mobility at each area. Outline of the space and installed sensors are shown in figure 5(b). Movements and activities committed in the experiment were not scripted beforehand; all actions happened deliberately regarding each individual's routine, work schedule, and needs at that moment.

Firstly, we evaluated the periodic approach for long-term human mobility prediction. Two months of collective mobility data was used to build the predictive model and the remaining 30 days of mobility data was used to test the model. Details of the dataset are summarized in table 1.

Table 1. Human Mobility Dataset

Dataset	Train	Test
Sample rate	hourly	
Number of participants	≤ 20	
Observed locations	30	
Size of data	62 days (1,488 hours)	30 days (720 hours)



(a) Infrared and Magnetic sensors used to observe mobility in the experimental space

(b) Floor plan

Fig. 5. (a) Infrared and Magnetic sensors used in the experiment. (b) Placement of the sensors.

5.1 Periodicity and Prediction Performance

We determined relations between the periodicity probability ($P_x(\tau = 24)$ and $P_x(\tau = 168)$) and the prediction accuracy, precision, and recall rate at each location separately. Figure 6(a) and 6(d) exhibit decreasing trend of prediction accuracy when the periodicity probability increased; yet, the periodic predictor gets higher precision and recall rates as the dataset has higher probability of such movements being repeated periodically. Nevertheless, the measurement of prediction accuracy is meaningless to us because the datasets, which contain visitations records at each location in past history, have negative skew. In other words, naive predictor can achieve at least 60% chance of predicting visitations (either *"visited"* or *"not visited"*) of users at a specific time frame in the future correctly by always guessing *"not visited"*. Figure 6(b) and 6(e) show direct relationship between the precision rate and the periodicity probability. Likewise, the recall rates in figure 6(c) and 6(f) show that the datasets with higher periodicity are more predictable than the others. Moreover, when the periodicity probabilities are lower than 0.4, the daily periodic approach (see figure 6(c)) clearly gets poor results. These confirm our hypothesis that the periodic approach alone is not effective in predicting with low periodicity probability.

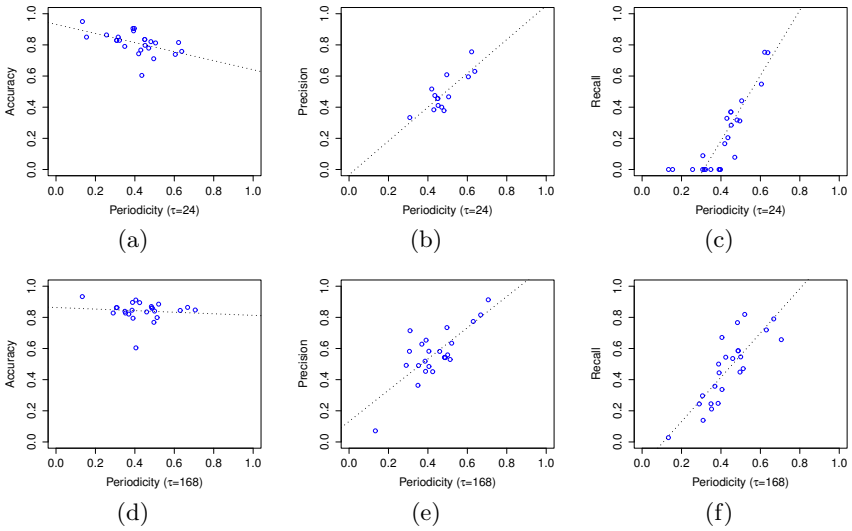


Fig. 6. Periodicity and Prediction Performance.

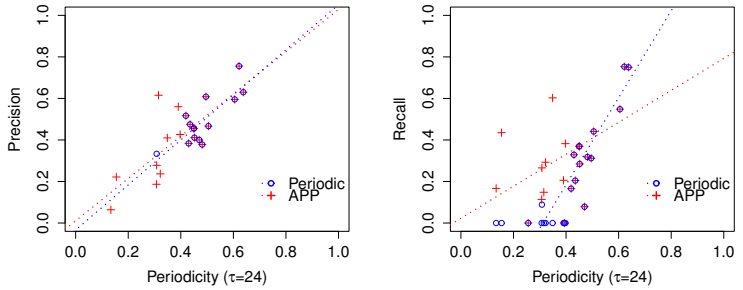
5.2 Prediction Performance of the Similar-day Approach

The aperiodic part in the proposed predictive technique, *APP*, is implemented with the similar-day predictive approach described in section 4.2. In the previous experiment, the periodic approach underperformed on the datasets where the mobility were not really periodic. Especially in the daily periodic model (see figure 6(a), 6(b), and 6(c)) when most of locations in the experimental space have lower periodicity probability than 0.4. Hence, in this experiment, the aperiodic part of the APP is activated where the periodicity is lower than the minimum threshold $P^{\tau}_{min} = 0.4$, the experimenter-specified threshold.

Figure 7 reveals benefit of implementing the aperiodic part into the APPpredictive model. In figure 7(a), the precision rates of the APP, after implementing the similar-day approach in low-periodicity data, are improved significantly, comparing with the periodic approach alone. The precision plots of the periodic approach on the left (periodicity ≤ 0.4) were mostly omitted because the periodic predictor never predicted *"visited"* on those locations, resulting undefined precision rates.

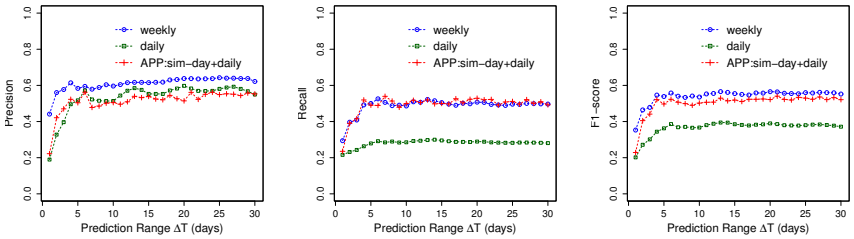
The APP also improves the recall rates, as shown in figure 7(b). It is striking that the recall rates used to get nearly 0.0 in the periodic approach rise up to 0.6 when predicted with the APP predictive technique.

In summary, the aperiodic part in our proposed long-term human mobility predictive technique helps improving the prediction performance especially when the periodicity probability is too low to infer future visitations. However, the similar-day approach that we implemented into the aperiodic part is not effective enough to improve the predictive technique that employs the weekly periodic approach ($\tau = 168$). The reason is that the *day of week* attributes used in the



(a) Precision (b) Recall

Fig. 7. Prediction Performance of the Similar-day Approach.



(a) Precision (b) Recall (c) F1-score

Fig. 8. Prediction Performance on Long-term Prediction.

cluster analysis already corresponded to the weekly basis, and the the *holiday* is not really significant feature since there were few holidays during only 3 months of dataset. So, the implementation of the similar-day approach (aperiodic part) and the weekly periodic approach was not able to achieve much improvement comparing with the weekly periodic predictive approach alone.

5.3 Performance in Long-term Prediction

In this section, we tested the robustness of the APP, Aperiodic and Periodic approach for human mobility predictor, over long range of prediction. The settings of this experiment refer to table 1. The results in figure 8 show the steady prediction performance even when predicting for 30 days ahead. The F1-score, which is the harmonic mean of the precision and the recall rate (in figure 8(c)) summarizes the prediction performance of 3 proposed predictive techniques as follows. Firstly, our collective mobility dataset that seems random in the first place contains enough information to be predicted accurately even in far future. Activities and corresponding mobility in the dataset are likely to be periodic on the weekly basis; hence, the weekly periodic predictive approach alone can get the average F1-score at 0.55. On the other hand, the daily model performs

relatively poor (average F1-score at 0.37) on this dataset because of low periodicity probability on the daily basis. However, after implementing the similar-day approach together with the daily predictive model, the integrated model can achieve average F1-score at 0.52.

6 Conclusions and Future Work

In this paper, awaiting answer to the question about limitations of the predictability of the collective human mobility from simple ambient sensors inside the smart environment has been revealed. We took a challenging decision implementing non-intrusive tracking method. Our choice of tracking method with ambient simple sensors have bold advantages from its unobtrusiveness and simplicity. The limitation, however, is that they cannot distinguish people's identities, such that its data limits a predictive model because it is impossible to create a predictive model individually for each user's mobility pattern.

The predictability analysis reveals potential of building a short-term next location predictive model that predicts next movement of a moving user accurately using only the collective dataset. However, implementing the short-term predictor is outside the scope of this paper. Furthermore, we also discovered acceptably high predictability in long-term prediction by modeling periodic behaviors hidden in the collective mobility data.

Next, we proposed the APP: Aperiodic and Periodic predictive model for long-term human mobility prediction. Results from the experiment shows that performance of the APP predictor improved significantly when predicting in low periodicity situations using the aperiodic approach. However, the aperiodic approach implemented in this project is not effective enough to increase the performance of the weekly periodic predictive approach yet because very limited features can be extracted from the collective dataset. We leave this to be our future work.

References

[1] Anton Andriyenko, Konrad Schindler, and Stefan Roth. "Discrete-continuous optimization for multi-target tracking". In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012, pp. 1926–1933.

[2] Seung-Ho Baeg et al. "Building a smart home environment for service robots based on RFID and sensor networks". In: *Control, Automation and Systems, 2007. ICCAS'07. International Conference on*. IEEE. 2007, pp. 1078–1082.

[3] Csaba Beleznai, David Schreiber, and Michael Rauter. "Pedestrian detection using GPU-accelerated multiple cue computation". In: *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*. IEEE. 2011, pp. 58–65.

[4] B. Das et al. “Automated prompting in a smart home environment”. In: *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*. IEEE. 2010, pp. 1045–1052.

[5] Robert M Fano. *Transmission of Information: A Statistical Theory of Communication*. 1961.

[6] M.C. Gonzalez, C.A. Hidalgo, and A.L. Barabasi. “Understanding individual human mobility patterns”. In: *Nature* 453.7196 (2008), pp. 779–782.

[7] H. Haddadi, P. Hui, and I. Brown. “MobiAd: private and scalable mobile advertising”. In: *Proceedings of the fifth ACM international workshop on Mobility in the evolving internet architecture*. ACM. 2010, pp. 33–38.

[8] Richard W Hamming. “Error detecting and error correcting codes”. In: *Bell System technical journal* 29.2 (1950), pp. 147–160.

[9] Jiawei Han et al. “PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth”. In: *ICDE’2001, April* (2001), pp. 215–24.

[10] Sajid Hussain, Scott Schaffner, and Dyllon Moseychuck. “Applications of wireless sensor networks and RFID in a smart home environment”. In: *Communication Networks and Services Research Conference, 2009. CNSR’09. Seventh Annual*. IEEE. 2009, pp. 153–157.

[11] J. Krumm and E. Horvitz. “Predestination: Inferring destinations from partial trajectories”. In: *UbiComp 2006: Ubiquitous Computing* (2006), pp. 243–260.

[12] JB MacQueen. “Some methods for classification and analysis of”. In: (1967).

[13] A. Monreale et al. “WhereNext: a location predictor on trajectory pattern mining”. In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2009, pp. 637–646.

[14] Nicolas Navet and Shu-Heng Chen. “On predictability and profitability: Would gp induced trading rules be sensitive to the observed entropy of time series?” In: *Natural Computing in Computational Finance* (2008), pp. 197–210.

[15] A. Sadilek and J. Krumm. “Far out: Predicting long-term human mobility”. In: *Twenty-Sixth AAAI Conference on Artificial Intelligence*. 2012.

[16] S. Scellato et al. “Nextplace: A spatio-temporal prediction framework for pervasive systems”. In: *Pervasive Computing* (2011), pp. 152–169.

[17] C. Song et al. “Limits of predictability in human mobility”. In: *Science* 327.5968 (2010), pp. 1018–1021.

[18] L. Song et al. “Predictability of WLAN mobility and its effects on bandwidth provisioning”. In: *Proceedings of INFOCOM*. Vol. 6. 2006.

[19] Shoou-I Yu, Yi Yang, and Alexander Hauptmann. “Harry Potters Marauders Map: Localizing and Tracking Multiple Persons-of-Interest by Nonnegative Discretization”. In: *http://www.cs.cmu.edu/ iyu/files/cvpr13/cvpr13.pdf* (2013).

Subgroup Analytics and Interactive Assessment on Ubiquitous Data

Martin Atzmueller[1] and Juergen Mueller[1,2]

[1] University of Kassel, Knowledge and Data Engineering Group
[2] L3S Research Center
{atzmueller, mueller}@cs.uni-kassel.de

Abstract. This paper applies subgroup discovery for obtaining interesting descriptive patterns in ubiquitous data. Furthermore, we provide a novel graph-based analysis approach for assessing the relations between the obtained subgroup set, and for comparing subgroups according to their relations to other subgroups. We present and discuss first results utilizing real-world data, given by noise measurements with associated subjective perceptions and a set of tags describing the semantic context.

1 Introduction

Ubiquitous data mining has many facets including descriptive approaches: These can help for obtaining a first overview on a dataset, for summarization, for uncovering a set of interesting patterns and their inter-relations.

This paper reports first results on analyzing ubiquitous data: objective (sensor) data and subjective perceptions. We apply subgroup discovery as a versatile method for descriptive data mining. We utilize the VIKAMINE [3] tool [6] for subgroup discovery and analytics in order to implement a semi-automatic pattern discovery process. For the analysis, we apply real-world data from the EveryAware project[4] – focusing on noise measurements. The individual data points include the measured noise in decibel (dB), associated subjective perceptions (feeling, disturbance, isolation, artificiality) and a set of tags for providing semantic context for the individual measurements. We focus on the interrelation between sensor measurements, subjective perceptions, and descriptive tags. For assessing the relations between the result set of subgroups, we propose a novel graph-based analysis approach. This method is applied for visualizing subgroup relations, and can be utilized for comparing subgroups according to their relationships to other subgroups. We present first results analyzing subgroups patterns for hot-spots of low/high noise levels.

The remainder of the paper is organized as follows: Section 2 discusses related work. Next, Section 3 introduces necessary basic notions. After that, Section 4 proposes the novel approach for graph-based subgroup analytics, and presents first analysis results in our application setting. Finally, Section 5 concludes with a summary and presents interesting options for future work.

[3] http://www.vikamine.org
[4] www.everyaware.eu

2 Related Work

Ubiquitous data mining covers many subfields, including spatio-temporal data mining [15], mining sensor data or mining social media with geo-referenced data, c.f., [3]. Applications include destination recommenders, e.g., for tourist information systems [10], or geographical topic discovery [21]. Often established problem statements and methods have been transferred to this setting, for example, considering association rules [2]. Related approaches consider, for example, social image mining methods, cf., [17] for a survey. The concept of collecting information in ubiquitous systems, especially for crowd-sourced and citizen-driven applications is discussed in [18]. Basic issues of measuring noise pollution using mobile phones are presented in [19].

In contrast to the approaches discussed above, in this paper we focus on descriptive patterns. This allows for the flexible adaptation to the preferences of the users, since their interestingness can be flexibly tuned by altering the applied quality function and target concept. There are several variants of pattern mining techniques, e.g., frequent pattern mining [11], mining association rules [1], as well as subgroup discovery [13], which is the method applied in this work.

For analyzing a set of subgroups, these are typically clustered according to their similarity, e. g., [8], or based on their predictive power [14]. Other methods for pattern set refinement and selection, e. g., [16] focus on similarities on the instance and/or description level. In contrast to these approaches, the proposed approach for subgroup set analytics generalizes those methods. We provide a general approach for analyzing subgroup relations based on a freely configurable "relationship" function, embedded in a graph-based framework for the assessment of sets of subgroups.

3 Preliminaries

Data mining includes descriptive and predictive approaches [12]. In the following, we focus on descriptive pattern mining methods. We apply subgroup discovery [13], a broadly applicable data mining method which aims at identifying interesting patterns with respect to a given target property of interest according to a specific quality function. This section first introduces the necessary notions concerning the data representation, subgroup patterns, basics on graphs, and similarity measures.

Formally, a *database* $DB = (I, A)$ is given by a set of individuals I and a set of attributes A. A *selector* or *basic pattern* $sel_{a_i=v_j}$ is a Boolean function $I \to \{0, 1\}$ that is true if the value of attribute $a_i \in A$ is equal to v_j for the respective individual. The set of all basic patterns is denoted by S. For a numeric attribute a_{num} selectors $sel_{a_{num} \in [min_j; max_j]}$ can be defined analogously for each interval $[min_j; max_j]$ in the domain of a_{num}. The Boolean function is then set to true if the value of attribute a_{num} is within the respective range.

A *subgroup description* or (complex) *pattern* sd is then given by a set of basic patterns $sd = \{sel_1, \ldots, sel_l\}$, which is interpreted as a conjunction, i.e., $sd(I) = sel_1 \wedge \ldots \wedge sel_l$, with $length(sd) = l$.

Without loss of generality, we focus on a conjunctive pattern language using nominal attribute–value pairs as defined above in this paper; internal disjunctions can also be generated by appropriate attribute–value construction methods, if necessary. A *subgroup (extension)*

$$sg_{sd} := ext(sd) := \{i \in I | sd(i) = true\}$$

is the set of all individuals which are covered by the subgroup description sd. As search space for subgroup discovery the set of all possible patterns 2^S is used, that is, all combinations of the basic patterns contained in S.

A *quality function* $q\colon 2^S \to \mathbb{R}$ maps every pattern in the search space to a real number that reflects the interestingness of a pattern (or the extension of the pattern, respectively). The result of a subgroup discovery task is the set of k subgroup descriptions $res_1, \ldots, res_k$ with the highest interestingness according to the quality function. While a large number of quality functions has been proposed in literature, many quality measures trade-off the size $|ext(sd)|$ of a subgroup and the deviation $t_{sd} - t_0$, where t_{sd} is the average value of a given target concept in the subgroup identified by the pattern sd and t_0 the average value of the target concept in the general population. Thus, typical quality functions are of the form

$$q_a(sd) = |ext(sd)|^a \cdot (t_{sd} - t_0), a \in [0; 1]\,. \tag{1}$$

For binary target concepts, this includes for example the *weighted relative accuracy* for the size parameter $a = 1$ or a simplified binomial function, for $a = 0.5$.

An (undirected) *graph* $G = (V, E)$ is an ordered pair, consisting of a finite set V containing the *vertices/nodes*, and a set E of *edges/connections* between the vertices. We freely use the term *network* as a synonym for graph. A *weighted* graph is a graph $G = (V, E)$ together with a function $\mathrm{w} : E \to \mathbb{R}^+$ that assigns a positive weight to each edge. The *degree* $\mathrm{d}(u)$ of a node u in a network measures the number of connections it has to other nodes. In weighted graphs the *strength* $\mathrm{s}(u)$ is the sum of the weights of all edges containing u, i. e., $\mathrm{s}(u) := \sum_{\{u,v\} \in E} \mathrm{w}(\{u, v\})$.

Given two vectors $\boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathbb{R}^X$, there are a variety of *similarity measures* for assessing the similarity between the contained values, e. g., [20].We can measure vector similarity, for example, by the (normalized) Manhattan distance, defined as follows:

$$\mathrm{sim}_{\mathrm{man}}(\boldsymbol{v}_1, \boldsymbol{v}_2) := \frac{\sum_{i=1}^{X} |\boldsymbol{v}_{1i} - \boldsymbol{v}_{2i}|}{X}\,, \tag{2}$$

where v_{ij} denotes the j-th component of vector v_i.

Another prominent measure from information retrieval is the cosine measure. The cosine similarity between two vectors $\boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathbb{R}^X$ is then defined as:

$$\mathrm{sim}_{\cos}(\boldsymbol{v}_1, \boldsymbol{v}_2) := \cos \measuredangle(\boldsymbol{v}_1, \boldsymbol{v}_2) = \frac{\boldsymbol{v}_1 \cdot \boldsymbol{v}_2}{||\boldsymbol{v}_1||_2 \cdot ||\boldsymbol{v}_2||_2}\,. \tag{3}$$

4 Exploratory Subgroup Analytics

In the following, we first present the novel subgroup analytics approach using a graph-based representation for inspecting and assessing a set of subgroups. Next, we describe the utilized dataset. After that, we discuss first results of the analysis in our ubiquitous application context.

4.1 Overview

For subgroup analytics, we first obtain a set of the top-k subgroups for a specific target variable. Typically, an efficient subgroup discovery algorithm needs to be applied. In our experiments, we apply the SD-Map* [5] algorithm for efficient subgroup discovery, which is suitable for sparse tagging data, c.f., [7]. After that, the set of subgroups needs to be assessed and put into relation to each other.

The proposed approach especially focuses on this specific step: It considers a relation between subgroups such that their "connections" according to this relation can be modeled as a graph. More formally, given a certain criterion implemented by a relation function $rel : I \times I \rightarrow \mathbb{R}$ we obtain a value estimating the relationship between pairs of subgroups, identified by their respective subgroup descriptions. Possible relations include, for example, geographic distance, or semantic criteria. In our application setting, we focus on the latter, since we will use the given perceptions for noise measurements as semantic proxies for subgroup relatedness.

For assessing our result set of subgroups R, we obtain the *rel*-value for each pair of subgroups (u, v). After that, we construct a *subgroup assessment graph* G_R for R: The nodes of G_R are given by the subgroups contained in R. The edges between node pairs (u, v) are constructed according to the respective $rel(u, v)$ value: If the respective value between the subgroup pair is zero, then the edge is dropped; otherwise, an edge weighted by $rel(u, v)$ is added to the graph.

It is easy to see that – depending on the applied relationship function *rel*– this construction process can result in a fully connected graph which is hard to interpret. Therefore, a refinement of this process utilizes a certain threshold τ_{rel} which is used for pruning edges in the graph. If the relation "strength" $rel(u, v)$ between a subgroup pair (u, v) is below the threshold, i. e., $rel(u, v) < \tau_{rel}$ then we do not consider the edge between u and v, such that the edge is dropped. By carefully selecting a suitable threshold τ_{rel} the resulting subgroup network can then be easily inspected and assessed.

Typically, the situation becomes interesting when the graph is split into different components corresponding to certain clusters of subgroups. We will discuss examples of constructed networks below. For selecting a suitable threshold, a *threshold-component* visualization can be applied, see Figure 5 for an example. This visualization plots the number of connected components of the graph depending on the applied threshold. Then, the "steps" within the plot can indicate interesting thresholds that can be interactively inspected. A related visualization plots the used threshold against the graph density for obtaining a first impression of the ranges of suitable threshold selections.

4.2 Applied Dataset

In this paper, we utilize data from the EveryAware project, specifically, on collectively organized noise measurements collected using the *WideNoise Plus* application between December 14, 2011 and June 12, 2013.

WideNoise Plus allows the storage of noise measurements using ubiquitous mobile devices, and includes sensor data from the microphone given as noise level in dB and data from the location sensors (i.e., GPS-sensor, GSM- and WLAN-locating) represented as latitude and longitude coordinate as well as a timestamp. Furthermore, *WideNoise Plus* captures the user's perceptions about the recordings, expressed using the four slider feeling (love to hate), disturbance (calm to hectic), isolation (alone to social), and artificiality (nature to man-made). In addition, tags can be assigned to the recording. The data are stored and processed using the EveryAware platform [9], which is based on the UBICON framework [4].

In our analysis, we utilize the following objective and subjective information for each measurement:

- Objective: Level of noise (dB).
- Subjective perceptions about the environment:
  - "Feeling" (hate/love) encoded in the interval $[-5; 5]$, where -5 is most extreme for "hate" and 5 is most extreme for "love".
  - "Disturbance" (hectic/calm), encoded in the interval $[-5; 5]$, where -5 is most extreme for "hectic" and 5 is most extreme for "calm".
  - "Isolation" (alone/social), encoded in the interval $[-5; 5]$, where -5 is most extreme for "alone" and 5 is most extreme for "social".
  - "Artificiality" (man-made/nature), encoded in the interval $[-5; 5]$, where -5 is most extreme for "man-made" and 5 is most extreme for "nature".
- Tags, e. g., "noisy", "indoor", or "calm", providing the semantic context of the specific measurement.

The applied dataset contains 5,237 data records and 1,056 distinct tags: The available tagging information was cleaned such that only tags with a length of at least three characters were considered. Only data records with valid tag assignments were included. Furthermore, we applied stemming and split multi-word tags into distinct single word tags.

Figures 1-4 provide basic statistics about the tag count and measured noise distributions, as well as the value distributions of the perceptions and the number of tags assigned to a measurement. As can be observed in Figure 1 and Figure 4, the tag assignment data is rather sparse, especially concerning larger sets of assigned tags. However, it already allows to draw some conclusions on the tagging semantics and perceptions. In this context, the relation between (subjective) perceptions and (objective) noise measurements is of high interest. Therefore, we present first analysis results of interesting patterns in the case study described below. We focus on the relation between semantics and perceptions as indicated by the different subjective perception values.

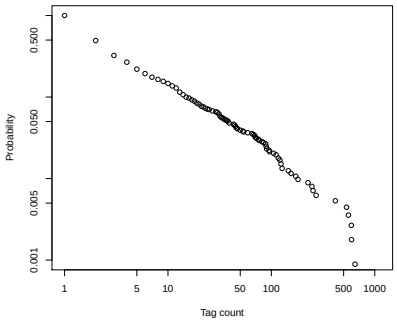


Fig. 1. Cumulated tag count distribution in the dataset. The y-axis provides the probability of observing a tag count larger than a certain threshold on the x-axis.

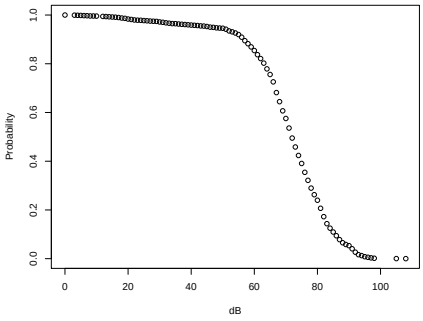


Fig. 2. Cumulated distribution of noise measurement (dB). The y-axis provides the probability for observing a measurement with a dB value larger than a certain threshold on the x-axis.

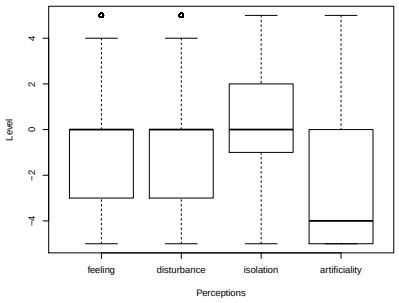


Fig. 3. Overview on the value distribution of the different perceptions.

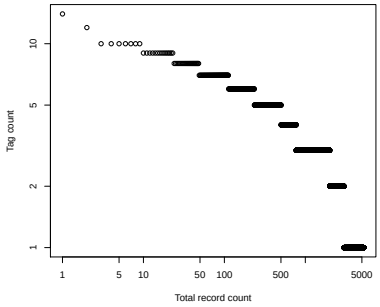


Fig. 4. Distribution of assigned tags per resource/data record.

4.3 Case Study: First Results and Discussion

In the following, we present first analysis results in the context of the *WideNoise Plus* data. According to the proposed approach, we applied subgroup discovery for the target variable *noise (dB)* focusing on subgroups with a large deviation comparing the mean of the target in the subgroup and the target in the whole database. We applied the simple binominal quality function, c.f., Section 3. Table 1 shows the resulting 20 patterns combining the two top-10 result sets.

In the table, we can identify several distinctive tags for noisy environments, for example, *craft*, *aircraft*, *plane*, *heathrow AND plane* which relate to Heathrow noise monitoring, c.f., [4] for more details. These results confirm the basic analysis in [4]. For more quiet environments, we can also observe typical patterns, e. g., focusing on the tags *indoor*, *background* and *work*, and combinations. Some

Table 1. Patterns: 1-10 - target: large mean noise (dB); 11-20 - target: small mean noise (dB); Overall mean (population): 70.12 dB. The last two columns include the node degree in the subgroup assessment graph, for $\tau_{rel} = 0.90$ and $\tau_{rel} = 0.95$.

id	description	size	mean dB	feeling	disturbance	isolation	artificiality	deg (t=0.9)	deg (t=0.95)
1	craft	67	92.10	-3.06	-3.21	3.21	-4.61	4	1
2	air	72	89.72	-3.07	-3.10	2.97	-4.57	4	1
3	arriva	252	78.64	-0.02	-0.01	0.01	0.00	9	8
4	plane	415	76.26	-3.47	-2.61	-0.59	-3.75	5	0
5	heathrow AND plane	31	87.81	-4.61	-4.48	-0.32	-4.65	3	2
6	runway	107	79.62	-3.78	-3.45	-1.45	-3.94	3	2
7	runway AND plane	92	79.92	-3.75	-3.67	-1.38	-3.78	3	2
8	aeroporto	13	94.08	-5.00	0.00	0.00	0.00	6	1
9	ciampino	16	91.13	-4.06	0.00	0.00	0.00	10	2
10	departure	14	92.50	-0.71	0.57	-0.29	-1.36	11	8
11	home	124	45.58	1.10	1.31	-0.96	-0.99	9	7
12	bosco	17	35.35	3.29	3.53	-1.65	1.88	0	0
13	indoor	111	56.69	0.81	0.71	-0.17	-1.29	9	8
14	office	172	59.78	0.10	0.68	-0.35	-1.68	11	9
15	borgo	12	31.33	3.00	3.25	-1.00	1.67	0	0
16	background	35	48.06	0.40	2.11	-2.46	-0.97	10	9
17	work	74	55.76	-0.49	0.19	-0.35	-1.86	11	5
18	indoor AND background	22	44.32	0.55	1.91	-2.14	-0.73	10	8
19	kassel	96	58.67	-0.17	0.64	0.17	-1.41	10	9
20	work AND background	23	47.43	0.61	1.74	-2.00	-0.74	10	8

further interesting subgroups are described by the tags *bosco* (forest) and *borgo* (village). These also show a quit distinct perception profile, shown in the respective columns of Table 1. This can also be observed in the last two columns of the table indicating the degree in the subgroup assessment graph (see below): The subgroups described by *borgo* and *bosco* are quite isolated.

In order to analyze subgroup relations with respect to the perceptions, we apply the Manhattan similarity as defined in Section 3 as our assessment relation *rel*. We measure the similarity using the averaged perception vectors of the respective subgroup patterns, with normalized values in the interval $[0; 1]$. Using the Manhattan distance, we consider the overall "closeness" of the vectors; alternatively, the cosine similarity would focus on similar perception "profiles", i. e., uniformly expressed perceptions.

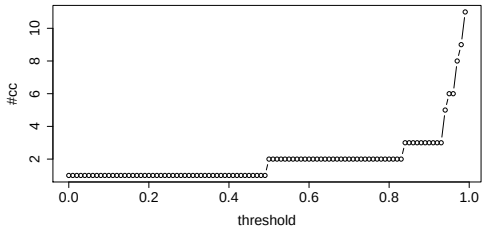


Fig. 5. Thresholded connected component plot based on a minimal *rel* value.

For determining appropriate thresholds τ_{rel}, Figure 5 shows a threshold vs. connected component plot using the Manhattan similarity defined above. Then, appropriate thresholds can be selected by the analyst. As can be observed in Figures 6-7 the respective networks for thresholds 0.90 and 0.95 show a distinct structure. Starting with the lowest threshold $\tau_{rel} = 0.90$ we can already observe the special structure of patterns 12 and 15. At this level, the remaining graph stays connected. With threshold $\tau_{rel} = 0.95$, several clusters emerge – the "Heathrow cluster" (5, 6, 7), as well as the large cluster covering most of the *lower noise* patterns. However, this cluster also contains some patterns from the

higher noise patterns (3, 8, 9, 10), which are rather unexpected and therefore quite interesting for subsequent analysis. The connecting subgroup patterns can then be simply extracted by tracing the connections in the graph.

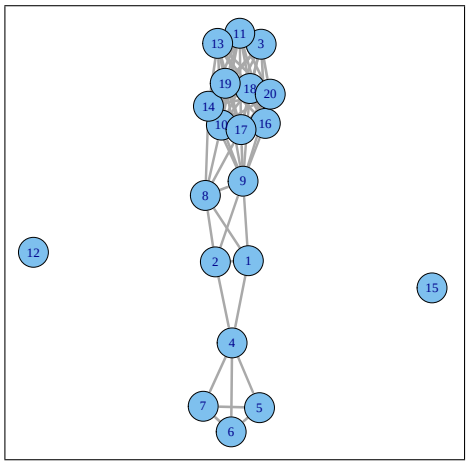


Fig. 6. Assessment graph: $\tau_{rel} = 0.90$.

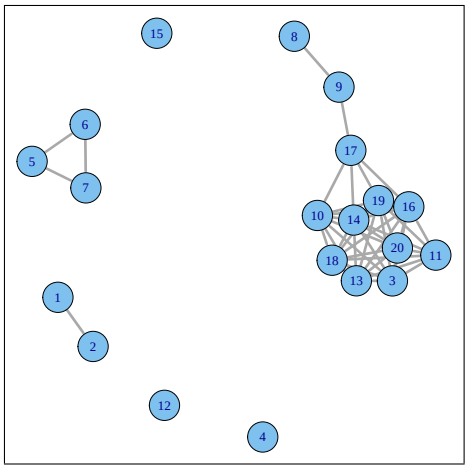


Fig. 7. Assessment graph: $\tau_{rel} = 0.95$.

5 Conclusions

In this paper, we presented exploratory subgroup analytics for obtaining interesting descriptive patterns in ubiquitous data. Specifically, we provided a novel graph-based analysis approach for assessing the relations between sets of subgroups. Using real-world data from a ubiquitous application we presented and discussed first analysis results. The analyzed noise measurements and associated subjective perceptions described by a set of tags confirmed the semantic context and provided interesting patterns towards a more comprehensive analysis.

For future work, we aim to extend the approach to diverse relationship and similarity measures. Furthermore, we plan to investigate multi-relational representations, i. e., multi-graphs capturing a set of relationships for assessing a set of subgroups. A further direction for analysis concerns the interrelations between perceptions, tags, and sentiments based on the tagging data. These can then also be applied, for example, for enhanced event detection, recommendations, or community mining, in combination with spatio-temporal patterns.

Acknowledgements

This work has been supported by the VENUS research cluster at the interdisciplinary Research Center for Information System Design (ITeG) at Kassel University, and by the EU project EveryAware.

References

1. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules. In: Proc. 20th Int. Conf. Very Large Data Bases. pp. 487–499. Morgan Kaufmann (1994)
2. Appice, A., Ceci, M., Lanza, A., Lisi, F., Malerba, D.: Discovery of Spatial Association Rules in Geo-Referenced Census Data: A Relational Mining Approach. Intelligent Data Analysis 7(6), 541–566 (2003)
3. Atzmueller, M.: Mining Social Media: Key Players, Sentiments, and Communities. WIREs: Data Mining and Knowledge Discovery 1069 (2012)
4. Atzmueller, M., Becker, M., Doerfel, S., Kibanov, M., Hotho, A., Macek, B.E., Mitzlaff, F., Mueller, J., Scholz, C., Stumme, G.: Ubicon: Observing Social and Physical Activities. In: Proc. IEEE CPSCom (2012)
5. Atzmueller, M., Lemmerich, F.: Fast Subgroup Discovery for Continuous Target Concepts. In: Proc. ISMIS 2009. LNCS (2009)
6. Atzmueller, M., Lemmerich, F.: VIKAMINE - Open-Source Subgroup Discovery, Pattern Mining, and Analytics. In: ECML/PKDD 2012. Springer, Berlin (2012)
7. Atzmueller, M., Lemmerich, F.: Exploratory Pattern Mining on Social Media using Geo-References and Social Tagging Information. IJWS 2(1/2) (2013)
8. Atzmueller, M., Puppe, F.: Semi-Automatic Visual Subgroup Mining using VIKAMINE. Journal of Universal Computer Science) 11(11), 1752–1765 (2005)
9. Becker, M., Mueller, J., Hotho, A., Stumme, G.: A Generic Platform for Ubiquitous and Subjective Data. In: Proc. 1st Intl. Workshop on Pervasive Urban Crowdsensing Architecture and Applications, PUCAA 2013 (2013)
10. Ceci, M., Appice, A., Malerba, D.: Time-Slice Density Estimation for Semantic-Based Tourist Destination Suggestion. In: Proc. ECAI 2010. pp. 1107–1108. IOS Press, Amsterdam, The Netherlands, The Netherlands (2010)
11. Han, J., Cheng, H., Xin, D., Yan, X.: Frequent Pattern Mining: Current Status and Future Directions. Data Mining and Knowledge Discovery 15, 55–86 (2007)
12. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 2nd Edition. Morgan Kaufmann, San Francisco, USA (2006)
13. Klösgen, W.: Explora: A Multipattern and Multistrategy Discovery Assistant. In: Advances in Knowledge Discovery and Data Mining, pp. 249–271. AAAI (1996)
14. Knobbe, A., Fürnkranz, J., Cremilleux, B., Scholz, M.: From Local Patterns to Global Models: The LeGo Approach to Data Mining. In: Proc. ECML/PKDD'08 LeGO Workshop (2008)
15. Koperski, K., Han, J., Adhikary, J.: Mining Knowledge in Geographical Data. Communications of the ACM 26 (1998)
16. van Leeuwen, M., Knobbe, A.J.: Diverse Subgroup Set Discovery. Data Min. Knowl. Discov. 25(2), 208–242 (2012)
17. Liu, Z.: A Survey on Social Image Mining. Intelligent Computing and Information Science pp. 662–667 (2011)
18. Richter, K.F., Winter, S.: Citizens as Database: Conscious Ubiquity in Data Collection. In: Advances in Spatial and Temporal Databases, Lecture Notes in Computer Science, vol. 6849, pp. 445–448. Springer, Berlin (2011)
19. Santini, S., Ostermaier, B., Adelmann, R.: On the Use of Sensor Nodes and Mobile Phones for the Assessment of Noise Pollution Levels in Urban Environments. In: Proc. Intl. Conf. on Networked Sensing Systems (INSS),. pp. 1 –8 (2009)
20. Strehl, A., Ghosh, J., Mooney, R.: Impact of Similarity Measures on Web-Page Clustering. In: AAAI WS AI for Web Search. pp. 58–64. Austin, TX, USA (2000)
21. Yin, Z., Cao, L., Han, J., Zhai, C., Huang, T.: Geographical Topic Discovery and Comparison. In: WWW 2011. pp. 247–256. ACM, New York, NY, USA (2011)

Open Smartphone Data for Mobility and Utilization Analysis in Ubiquitous Environments

Jochen Streicher[1], Nico Piatkowski[2], Katharina Morik[2], and Olaf Spinczyk[1]

Department of Computer Science, [1]LS12 and [2]LS8
TU Dortmund University
D-44227 Dortmund, Germany
{jochen.streicher,nico.piatkowski,
katharina.morik,olaf.spinczyk}@tu-dortmund.de
http://{ess,www-ai}.cs.uni-dortmund.de

Abstract. The development and evaluation of new data mining methods for ubiquitous environments and systems requires real data that were collected from real users. In this work, we present an open smartphone utilization and mobility dataset that was generated with several devices and participants during a 4-month study. A particularity of this dataset is the inclusion of low-level operating system data. Additionally to the description of the data, we also describe the process of collection and the privacy measures we applied. To demonstrate the utility of the data, we performed two example analyses, which are also presented in this paper.

1 Introduction

Today's mobile phones are able to produce a vast amount of valuable data. Produced by several physical and logical sensors, the data provides knowledge about the owner as well as his environment. Several studies have shown that smartphones can be used as an effective tool to gain insights into patterns of human behavior and interaction that were not available before. Notable examples are the datasets of the MIT Human Dynamics Lab like the Reality Mining dataset [1] or the Lausanne Data Collection Campaign [2]. The latter was however only available for participants of the 2012 Nokia data challenge. [3]

While smartphones are certainly an excellent *tool* for research, it is not less important to consider how data collection campaigns can help to improve these devices and the respective infrastructure. Previous research has shown that insights into utilization and mobility patterns of mobile devices are indeed of value for that purpose. This concerns the mobile network infrastructure [4] as well as the user experience with respect to the devices. A limiting factor to user experience is certainly the lifetime of a smartphone's battery. Much research has been conducted towards the use of user-specific mobility and utilization patterns to increase the energy-efficiency of mobile devices, like the reduction of GPS utilization via location prediction [5] or accelerated file prefetching. [6]

Research on these problems requires real data collected on real devices and from real users. While high-level data like location and phone call logs might

be sufficient for some of the problems, others also need data concerning the device, not only its owner. *Device Analyzer*[1] is an Android application collecting data from thousands of devices all over the world in order to get insights into utilization patterns, with the explicit goal to provide crucial information for the improvement of future smartphones. To guarantee complete anonymity all privacy-critical identifiers (e.g., cell tower IDs or MAC-addresses) are hashed with individual salts. This makes the dataset unavailable for the analysis of social interaction. Also, it is not clear when the data will be available.

Our proposed dataset shares commonalities with all three mentioned examples, but features all of the following: 1.) It contains operating system level data, 2.) not all identifiers are hashed with individual salts, and 3.) it is openly available at http://sfb876.tu-dortmund.de/mobidata.

The remainder of this paper is structured as follows: In Section 2 we will shortly describe how we collected the data and ensured the privacy of our participants, while Section 3 describes the resulting dataset. In Section 4 we present two exemplary analyses performed on these data. Section 5 concludes the paper.

2 The Collection Process

We started with 11 participants, who all were members of our collaborative research center. During our summer school in 2012, we additonally collected data from 11 attendees. For this purpose we used *MobiDAC*, our flexible infrastructure for data collection on Android-based smartphones. MobiDAC allows experimenters to use the participating devices like programmable sensor nodes. Operators write *sensing modules* that perform the actual data acquisition on the device. These modules may be uploaded to respective devices and remotely started or stopped. When a module is running, it is collecting, possibly preprocessing and saving data locally on the device. Data is sent back to the experimenter when certain conditions are met, like an established Wi-Fi connection. Currently, a modified version of the Scripting Layer for Android (SL4A)[2] is used to execute the sensing modules.

2.1 Modus Operandi

We used both the Android-API as well as Linux' virtual file systems (VFS) "`/proc`" and "`/sys`" as data sources. When the device was awake, most of the data was collected high-frequently (temporal resolution of two seconds) or via callbacks from Android. To reduce the amount of data that had to be transmitted from the device, we only recorded changes to data values. Every 60 seconds, we took a snapshot of all data from the virtual file systems and started sensor sampling for two seconds with the highest possible frequencies. A Bluetooth scan was started every five minutes. Every two hours, the periodically sampled

[1] Device Analyzer website: http://deviceanalyzer.cl.cam.ac.uk/
[2] SL4A can be found at: http://code.google.com/p/android-scripting

data was recorded completely (not only changes). As opposed to Symbian-based phones, which were used for some of the data collection campaigns mentioned in the introduction, Android phones try to *suspend* whenever possible. This happens, when the screen is off and no application is keeping the device awake by means of a *wake lock*. During the suspended state, no data can be collected at all. Thus, we explicitly wake the device from its sleep every 60 seconds and perform a full acquisition of all data values with the respective intervals.

2.2 Privacy Preservation

Since this version of our dataset is truly open and available to anyone, we were obliged to be especially careful in the process of ensuring privacy. We treated data in the following ways:

- Everything that uniquely identifies a participant is *globally consistently* replaced with a random value. This is also true for all identifiers from interaction with other entities (e.g., MAC-addresses and SSIDs) as well as for the names of installed and running application packages and processes.
- Mobile network cell information was replaced by *locally consistent* random values for each participant. This means that the mapping of cell identification (CID) and location area code (LAC) is different for every participant.

3 The Data

We collected data from various hardware and software subsystems, namely communication (Wi-Fi, Bluetooth and mobile), sensors, power supply, the Linux kernel and Android's application framework. This section coarsely describes the contents of the dataset resulting after the privacy-preserving measures.

3.1 Contents

The data may be categorized into high-level user context, external sensing, and system internals.

High-Level User Context is utilization data that contains direct hints to the participant's current activity and context. This includes the state of the display (on/off, brightness) and the phone (idle, ringing, or off the hook). Also the currently running packages belong to this category. Settings can also indirectly tell about the participant's context. For example, turning the phone to silent mode, when it was set to play a ringtone before, is a hint that the situation changed to one that prohibits phone noise, like a meeting or a cinema. Besides audio settings, also the communication settings, whether Bluetooth or Wi-Fi is enabled, or whether the device is in airplane mode, belong to this category.

Sensing data is obtained from various physical sensors as well as positioning and communication hardware. The physical sensors measured acceleration, magnetic field strength, orientation and light intensity. When the participant allowed it, also information about current altitude and speed were obtained from the GPS hardware. Also, communication devices can be used to sense the presence or even the signal strength of (potential) peers. Whereas Wi-Fi and baseband processors deliver information about stationary communication peers (access points and cell towers) and are thus feasible for positioning, Bluetooth delivers information about mobile communication peers.

System Internal data mainly describes the overall usage of the system's resources like the CPU, the battery, the main memory and the network interfaces. The use of Android *wakelocks* also belongs to this category. Since wakelocks are used to prevent the device from suspending, (application) bugs regarding their handling can severely increase energy consumption.

3.2 Structure

For every device, our dataset contains a stream of events. Every event is composed of a timestamp, an attribute name, and the new value for this attribute. Table 2 shows an excerpt of such a stream. For entity types with multiple instances, like Wi-Fi access points, attribute names contain a unique identifier for this resource (e.g., the BSSID for Wi-Fi). The appearance and disappearance of such entities is denoted with "1" or "0" respectively. For example, at time 1346837529394, package "gBRth" is started, whereas at 1346837579524, the access point "PSQdw" has gotten out of reach. The complete dataset consists of 250 million of these events. Figure 1 illustrates their distribution regarding event type and participant. Table 1 contains all attributes in condensed form. A detailed and exhaustive description can be found at the dataset's website.

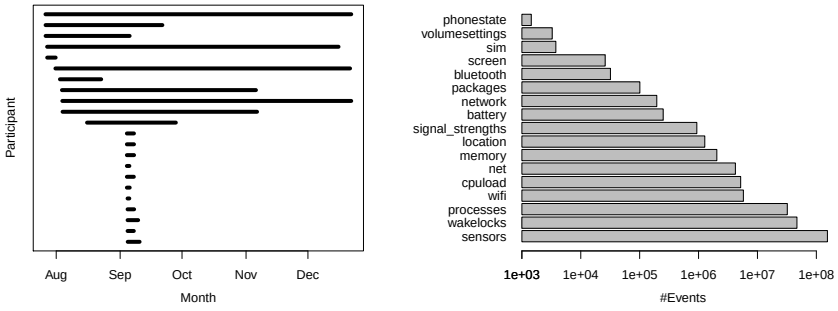


Fig. 1. Left: Period of participation for every participant. Right: Total number of events for every event type (log scale).

Table 1. Condensed names of collected attributes including the data **Cat**egory (**H**igh-Level, **S**ensing, or **I**nternal), the sampling **Int**erval, and the data **Sour**ce (Android **A**PI or Linux **V**FS). The sampling interval is either given in seconds or as the fact that we received the data as callback event (**E**) whenever it changed. Values in square brackets are placeholders for actual identifiers. An asterisk means that the according value (or identifier if in the bracket) was replaced for privacy.

Attributes	Cat.	Int.	Src.
airplanemode, phonestate	H	2,E	A
battery:{health, level, plugged, status, technology, temperature, voltage}	I	E	A
bluetooth:{, connected:[*mac_address**], device:[*mac_address**]:{, name*, class, bondstate, prev_bondstate}}	H,S		A
cpuload:{1min, 5min}	I	2,60	V
location:{gps, network}:{time, speed, altitude, accuracy}	S	60	A
media:{maxvolume, volume}	H	2	A
memory:{Buffers, Cached, Dirty, MemFree, MemTotal, Writeback}	I	60	V
net:[*device*]:{, {r, t}x_{bytes, packets, dropped, errors}, ... }	I	60	V
network:{roaming, cell:{cid*, lac*}, operatorid*}	S	2	A
notifications:vibrate, ringer:{maxvolume, silent, vibrate, volume}	H	2	A
packages:{launchable:[*package**], running:[*package**]}	H	60	A
processes:[*pid*]:{, cmdline:{*, parameters*}, state, tcomm*, {u, s, cu, cs}time, priority, nice, num_threads, start_time, vsize}	I	60	V
screen:{brightness, on, timeout}	H	2	A
self:{skip, start}			
sensors:{time, azimuth, light, pitch, roll, time, {x,y,z}force, {x,y,z}Mag}	S	120	A
signal_strenghts:{gsm_signal_strength, cdma_dbm, evdo_dbm}	S	E	A
sim:{state, operatorid*, serial*, subscriberid*}	I	300	A
wakelocks:[*name*]:{active_since, {expire, wake, }_count, last_change, {max, sleep, total}_time}	I	2	V
wifi:{, connection:{bssid*, hidden_ssid*, ip_address*, link_speed, network_id, rssi, ssid*, supplicant_state}, scan:[*bssid**]:{, capabilities, frequency, level, ssid*}}	H,S	60	A

Table 2. Example data for one device.

Timestamp	Attribute	Value	Timestamp	Attribute	Value
1346837529316	network:cell:cid	O8aal	1346837529469	cpu:load:5min	3.49
1346837529346	wifi	True	1346837529512	network:operatorid	obTLz
1346837529366	wifi:connection:ssid	WfQ4k	1346837529633	net:wlan0:tx_bytes	31342252
1346837529394	wifi:scan:PSQdw:ssid	ZvAet	1346837530254	packages:running:gBRth	1
1346837529428	ringer:vibrate	True	1346837530317	sim:serial	FUxuY
1346837529451	screen:on	False	1346837530351	phone:state	idle
1346837529454	screen:brightness	100	1346837534507	bluetooth:devices:ZOchS	1
1346837529468	battery:level	39	1346837579524	wifi:scan:PSQdw	0

4 Exemplary Analysis

Many different kinds of analysis can be imagined on the dataset presented here. Among them semantic place prediction [7], network cell prediction [4], transportation mode detection [8], frequent subsequence mining as well as the generative modeling of user or hardware behavior [9], [10]. Due to the streaming nature of our dataset, the streams abstraction [11] was used for preprocessing[3]. In the following, we will explain how the data can be incorporated in a network cell prediction task [4] as well as for smartphones power modeling [9].

Network Cell Prediction. Every mobile network connection has a unique network cell identifier. A-priori knowledge about cells that a user will visit in near future can deliver an indicator about upcoming changes in network routing and load. The network operator can automatically and pro-actively react to these changes, for example, by reserving capacity in the predicted cell. For a given set of Information I_t at time t, the network cell prediction task is to predict the next network cell that the corresponding user will visit. Forward feature selection shows, that the most important source of information is knowledge about the last k cells. For each user, one task specific dataset is generated. Therefore, the sequence of visited cells is extracted for each user and a sliding window of 10 minutes width is applied to this sequence. If the cell identifier changed multiple times within a single 10 minute window, the most frequent CID is selected. Each k consecutive windows are then concatenated to form a data row $(\text{cell}_0, \text{cell}_{-1}, \text{cell}_{-2}, \text{cell}_{-3})$, with label cell_0 and attributes cell_{-1} to cell_{-3}. Here, two windows are considered as consecutive if their time stamps do not differ more than 10 minutes. Applying a simple Naive Bayes Classifier for this problem yields only $\approx 30\%$ accuracy on the just described data. To enhance the performance for this task, cells that are only visited once can be removed from the dataset and more sophisticated methods like Support Vector Machines or Markov Random Fields can be applied [4].

Energy Modeling. Researchers have proposed a number of power models for ubiquitous systems [9], [10], [12]. Usually, power models are derived manually by using a power meter attached to one specific system instance. As a result of the model derivation process, the generated power model is at best accurate for one type of embedded system and at worst accurate only for the specific ubiquitous system instance for which it was built. It would require great effort and time to manually generate power models for the wide range of phones now available.

We now show how a simple linear regression power model can be estimated with our dataset, whereby we follow the approach that was presented by Zhang et al. [9]. The event stream is converted to a set of consecutive windows as described above. Since the energy consumption should be predicted, we consider the change of battery level as label, i.e. $y = \text{batLevel}_t - \text{batLevel}_{t-1}$. The following measurements are considered as features: mobile network and Wi-Fi signal

[3] The stream container and processors that have been written to preprocess the data for both tasks are available online at: http://sfb876.tu-dortmund.de/mobistream.

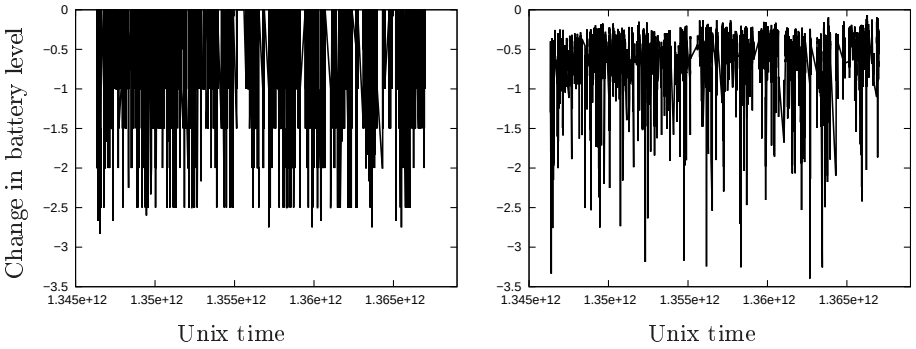


Fig. 2. Energy consumption of a smartphone as measured by the change in battery level over time. Left: measured consumption. Right: predicted energy consumption.

strength, Wi-Fi speed, number of outgoing/incoming Wi-Fi and mobile network packets in the last window, display brightness, GPS usage and CPU utilization. Figure 2 shows the measured energy consumption for one user over time on the left, and the corresponding prediction on the right. The 10-fold cross validated root mean squared error of the estimated linear model is 0.604 with a deviation of ± 0.02 and the absolute error is 0.525 (± 0.013).

Using such prediction models as a building block within a larger learning task can help to estimate the energy consumption of certain decisions since it can be used to assign costs to mobile network, display, CPU, Wi-Fi and GPS usage.

5 Conclusion

We presented our open smartphone utilization dataset collected within our collaborative research center and during its summer school and presented some analysis to show its utility. We believe that open datasets greatly help to evaluate and improve analysis methods like these. It is interesting to see that, the further down the software-hardware stack a data source resides, the more data is generated and the more data is actually needed to obtain meaningful results. We see this as an indication towards the need for data collection frameworks that allow for flexible preprocessing and data aggregation.

Acknowledgments

This work has been supported by Deutsche Forschungsgemeinschaft (DFG) within the Collaborative Research Center SFB 876 "Providing Information by Resource-Constrained Data Analysis", project A1. We would also like to thank our collaboration partners from the EcoSense project at Aarhus University for providing technical support.

References

1. Eagle, N., Pentland, A.: Reality mining: Sensing complex social systems. Personal and Ubiquitous Computing **10**(4) (2006) 255–268
2. Kiukkonen, N., Blom, J., Dousse, O., Gatica-Perez, D., Laurila, J.: Towards rich mobile phone datasets: Lausanne data collection campaign. Proc. ICPS, Berlin (2010)
3. Laurila, J.K., Gatica-Perez, D., Aad, I., Jan Blom, Olivier Bornet, T.M.T.D., Dousse, O., Eberle, J., Miettinen, M.: The mobile data challenge: Big data for mobile computing research. In: Mobile Data Challenge by Nokia Workshop, in conjunction with Int. Conf. on Pervasive Computing. (June 2012)
4. Michaelis, S., Piatkowski, N., Morik, K.: Predicting next network cell IDs for moving users with Discriminative and Generative Models. In: Mobile Data Challenge by Nokia Workshop in conjunction with Int. Conf. on Pervasive Computing. (June 2012)
5. Chon, Y., Talipov, E., Shin, H., Cha, H.: Mobility prediction-based smartphone energy optimization for everyday location monitoring. In: Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems. SenSys '11, New York, NY, USA, ACM (2011) 82–95
6. Fricke, P., Jungermann, F., Morik, K., Piatkowski, N., Spinczyk, O., Stolpe, M., Streicher, J.: Towards Adjusting Mobile Devices To User's Behaviour. In: Analysis of Social Media and Ubiquitous Data. Volume 6904 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, Heidelberg (2011) 99–118
7. Huang, C.M., Jia-Chin Ying, J., Tseng, V.: Mining users' behavior and environment for semantic place prediction. In: Mobile Data Challenge by Nokia Workshop in conjunction with Int. Conf. on Pervasive Computing. (June 2012)
8. Stenneth, L., Wolfson, O., Yu, P.S., Xu, B.: Transportation mode detection using mobile phones and GIS information. In: Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. GIS '11, New York, NY, USA, ACM (2011) 54–63
9. Zhang, L., Tiwana, B., Qian, Z., Wang, Z., Dick, R.P., Mao, Z.M., Yang, L.: Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In: Proceedings of the 8th IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis. CODES/ISSS '10, New York, NY, USA, ACM (2010) 105–114
10. Dong, M., Zhong, L.: Self-constructive high-rate system energy modeling for battery-powered mobile systems. In: Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services. MobiSys '11, New York, NY, USA, ACM (2011) 335–348
11. Bockermann, C., Blom, H.: The streams framework. Technical Report 5, TU Dortmund University (12 2012)
12. Kjærgaard, M.B., Blunck, H.: Unsupervised Power Profiling for Mobile Devices. In: Proceedings of the 8th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (Mobiquitous 2011), Springer (2011)

Content-Based Geo-Location Detection for Placing Tweets Pertaining To Trending News on Map

Saurabh Khanwalkar[1], Marc Seldin[1], Amit Srivastava[1], Anoop Kumar[1], Sean Colbath[1]

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

Abstract. In the last few years, social media services such as Twitter have proven to provide first-line information on current news events such as civil unrest, protests, elections, etc. The limited availability of self-identified geo-location information makes it challenging to place such current and trending news on the globe. In this paper, we demonstrate a novel approach for content-based geo-location of Arabic and English language tweets by collating contextual tweets into a document using a user-tweeting-frequency based temporal window. We compute the distances of the content-based geo-located tweets against the device-based geospatial points provided natively via Twitter API as well as the Twitter User Profile based locations. We show that content-based geo-location detection provides an effective way of geo-localizing trending news topics, geo-political entities and Hashtags.

Keywords: Geo-location, twitter, social media

1 Introduction

Micro-blogging services such as Twitter have become a very popular communication tool among Internet users, being employed for a wide range of purposes including marketing, expressing opinions, broadcasting events or simply conversing with friends. Each day, more than 200 million active users publish more than 400 million tweets per day in the social network, sharing significant events in their daily lives [1]. Additionally, Twitter allows researchers unprecedented access to digital trails of data as users share information and communicate online. This is helpful to parties seeking to understand trends and patterns ranging from customer feedback to the mapping of health pandemics [2]. As explained by T. Sakaki et al. [3], every Twitter user can be described as a sensor that can provide spatiotemporal information capable of detecting major news events such as earthquakes or hurricanes.

Location is a crucial attribute to understanding the ways in which online flow of information might reveal underlying economic, social, political, and environmental trends. Localization facilitates temporal analyses of trending news topics from a geospatial perspective, which is often useful in further analysis. Studies such as [4] and [5] have addressed the capability to track emergency events and how they evolve, as people usually first post news on Twitter, and are later broadcast by traditional media corporations [6]. One of the biggest challenges is identifying the location where events are taking place.

Twitter supports per-tweet geo-tagging feature which provides extremely fine-tuned Twitter user tracking by associating each tweet with latitude and longitude co-ordinate points. In our sampling of 20 million tweets, less than 0.70% of all tweets actually use the geospatial tagging functionality. When this feature is enabled, it generally functions automatically when a tweet is published with the coordinate data coming either from user's device itself via GPS, or from detecting the location of the user's Internet (IP) address. Additionally, these features do not provide location estimates based on the content of the user-posted tweet messages.

Effective geo-location of tweets based purely on their textual content is a difficult task, and although Twitter provides vast amounts of data, it introduces several natural language processing (NLP) challenges:

- Multilingual posts and code-switching [7] between languages makes it harder to develop language models and often needs Machine Translation (MT).
- With the limitation of 140 characters per-tweet, Twitter users often use short-hand and non-standard vocabulary which makes named-entity detection and geo-location via gazetteer more challenging.
- Twitter content tends to be very volatile, and pieces of content become popular and fade away within a matter of hours.

In this paper, we present a content-based, geo-location detection approach that is capable of geo-locating multilingual tweets, within a time window, by exclusively using the textual content of these tweets. Our premise is that tweets encode geospatial location-specific content; either specific place names or named-entities. Additionally, our intuition is that within a time window, Twitter users tweet specific to their current location or specific to localized trending events which are of interest.

The rest of the paper is organized as follows: in Section 2, we review the related work on content-based geo-location detection. In Section 3, we describe the dataset, and the evaluation metrics we used to benchmark our geo-location detection performance. In Section 4, we explain our approach to content-based geo-location detection for placing tweets on a map. In Section 5, we present results from performance evaluation of our geo-location detection algorithm, followed by examples of using geo-location detection for placing tweets pertaining to trending news on map in Section 6.

2 Related Work

Recently, content-based geo-location detection techniques have been explored, some focusing on supervised and language model based approaches, while others focusing on location name-based approaches. Applications vary from providing relevant advertisements, to public health awareness, user modeling and tracking trending news events. Cheng et al. [8] propose a probabilistic framework for estimating a Twitter user's city-level location based purely on the content of that user's tweets.

Roller et al. [12] present a supervised, text-based geo-location using language models on an adaptive grid. Given training documents labeled with latitude and longitude coordinates, *pseudo-documents* are constructed by concatenating the documents within a grid cell overlaid on Earth; then a location for a test document is chosen

based on the most similar pseudo-document. Paradesi [13] explores research that identifies the locations referenced in a tweet and show relevant tweets to a user based on that user's location. For example, a user traveling to a new place would not necessarily know all the events happening in that place unless they appear in the mainstream media. The proposed system, called TwitterTagger geo-tags tweets in near real-time and shows tweets related to surrounding areas.

The contributions of this paper towards content-based geo-location research are:

- Novel approach for collating multilingual tweets into a temporal document using user-tweeting-frequency based time window;
- Named-entity detection and geospatial points-based clustering;
- Location-specific feature set calculation and document scoring for best-match content-based location identification for a document.

3 Dataset, Evaluation Setup and Metrics

We developed a process to identify Social Media users across the Middle East region who are influential contributors on the Twitter social media platform. Through this process, we created a list of Twitter users, culled from mainstream journalism feeds, diplomatic circles, and political circles having wide Arabic regional appeal. We collected tweets over a period of 3 months (January 2013 to March 2013) using the Twitter Spritzer streaming API with a filter for our selected users of interest. Using this setup, we collected approximately 17 million multilingual tweets distributed into 85% Arabic, and 15% English from 2.6 million Twitter users.

To evaluate the performance of our tweet geo-location detection algorithm, the first metric we consider is the ***Error Distance***, which quantifies the distance in miles between the actual geo-location of the tweet $l_{act}(t)$ and the estimated geo-location $l_{est}(t)$ [8]. The Error Distance for tweet *t* is defined in equation (1) as –

$$ErrDist(t) = d\big(l_{act}(t),\ l_{est}(t)\big) \tag{1}$$

The overall performance of the content-based tweet geo-location detector can further be measured using the Average Error Distance across all the geo-located tweets T using equation (2) –

$$AvgErrDist(T) = \frac{\sum_{t \in T} ErrDist(t)}{|T|} \tag{2}$$

A low Average Error Distance indicates that the detector may geo-locate tweets close to their geo-location on average as provided by the user profile or user device. This metric does not provide more insight into the distribution of the geo-location detection errors. We apply maximum allowed distance in miles thresholding at three points; 100 miles, 500 miles and 1000 miles and calculate the next metric, Accuracy100, Accuracy500 and Accuracy1000 using equation (3) –

$$Accuracy_K(T) = \frac{|t|t \in ErrDist(t) \leq K|}{|T|} \ where\ K\ is\ distance\ in\ miles \tag{3}$$

4 Technical Approach

In this section, we present our approach for content-based geo-location detection as outlined by the processing flowchart in Fig. 1.

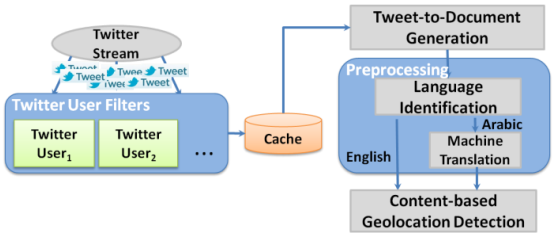


Fig. 1. Processing flowchart for content-based geo-location showing all the stages, starting from tweets collection, document conversion, preprocessing and geo-location detection

4.1 Tweets-to-Documents Generation

We developed an approach for generating cohesive documents from tweets that can be used as a subject of analysis by Information Extraction (IE) algorithms. The motivation for defining *document* was two-fold; (1) a single tweet is limited to 140 characters and may not have sufficient content for estimating location that corresponds to a specific topic, and, (2) most Twitter users post tweets on specific trending topics and move on to other topics within a certain temporal window [10]. This approach is formulated in Algorithm 1.

Algorithm 1 Tweet-to-Document Conversion

Input: *tweets:* List of n tweets from m Twitter users in time window t
minWindowSize: The minimum size of the time window in hours
maxWindowSize: The maximum size of the time window in hours
minTweetsInWindow: The minimum number of tweets per-user in a time window
maxTweetsInDocument: The maximum number of tweets allowed in a document
Output: *documentList*: List of documents in time window t
Notation: { } – List, [ ] - Array

```
1.  startTime = currentTimeInHours
2.  epochStartTime = startTime
3.  While (True)
4.     timeSpan = startTime – epochStartTime
5.     windowSize =(timeSpan>=maxWindowSize)?maxWindowSize : minWindowSize
6.     endTime = currentTimeInHours + windowSize
7.     foreach userTweetTime in userTweetTimeTable
8.        if (created_at   >= startTime && created_at  < endTime)
9.           userTweetList.Insert(userTweet)
10.    foreach userTweet in userTweetList
11.       tweets = userTweetList[userTweet]
12.       foreach tweet in tweets
13.          Document.Add(tweet);
14.          if(Document.Size >= maxTweetsInDocument)
15.             DocumentList.Insert(Document)
16.    startTime = endTime
```

4.2 Preprocessing

Once all the tweets in a time-delineated window are converted into documents, such that each document contains multiple tweet posts from a specific user, we preprocess all the documents in preparation for content-based geo-location detection. First, we perform n-gram based language identification [10] to identify Arabic versus English tweets and translate Arabic tweets into English using the SDL Language Weaver Machine Translation (MT) system. The geo-location detection algorithm operates on source English tweets and the MT-English equivalent of the Arabic tweets.

4.3 Content-based Geo-location Detection

Our geo-location detection algorithm has three distinct phases as shown in Fig. 2. In the first phase, tweets that were grouped into a time-delineated content window via the document generation algorithm described in section 4.1 are submitted to a named entity detection algorithm [11].

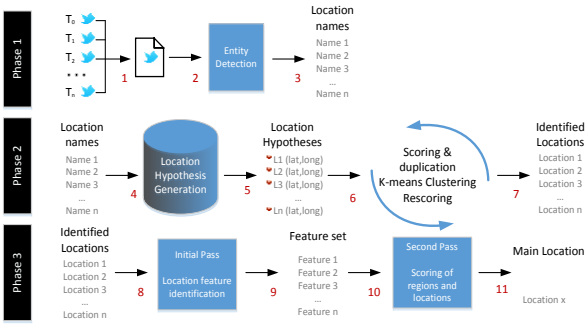


Fig. 2. Three phases of the Content-based Geo-location Clustering and Detection Algorithm

In phase two, the list of named entities which were discovered in phase one is now employed to select location records from several gazetteers. Each match is then given a preliminary score based on features both internal to the location record and features from external sources. Points are then duplicated proportionally to their scores to create a weighting scheme for k-means clustering. The randomly assigned points are then rescored based on how close they are to their cluster's center or centroid location. Finally, location identities are assigned to location names according to their membership in the cluster with the highest score containing that name.

The third phase is concerned with selecting the best overall location associated with the document. This phase begins by iterating through the locations identified in the previous step. During this initial pass, common features such as political administrative unit membership are identified, as well as other features such as order of occurrence. In a second pass, each location is scored by comparing it to the results of the first pass; certain features are biased and others receive an anti-bias. After each point is scored, the highest scoring location is returned as the estimated location.

5 Results and Discussion

A key point to be noted is that our geo-location detection evaluation is based solely on the location of the users where they were tweeting. While these results help us assess the performance of geo-location detector, we believe that creating a manually annotated set would allow use to demonstrate greater accuracy. This is due to the discrepancy between a user's physical location and the topic a user may be tweeting about. For example, a user from Boston, MA, USA might be traveling in Egypt, while tweeting about trending news in Syria.

5.1 Comparison against device-based geo-location provided by Twitter

To minimize outliers, we filtered tweets that are from potential spammers based on 2 criteria; (1) filter tweets that are not from our core selected users, and, (2) filter tweets that are auto-generated by advert spreading tools. After filtering, we had approximately 50K tweets with Twitter-provided device-based geospatial data in terms of latitude and longitude points. Table 1 shows the results of our content-based geo-location detection algorithm using metrics defined in section 3.2.

Table 1. Performance of our content-based geo-location detection

AvgErrDist (Miles)	$Accuracy_{100}$	$Accuracy_{500}$	$Accuracy_{1000}$
1881.98	0.122	0.321	0.497

We found that only 12% of the 50K tweets in the test set could be geo-located within 100 miles of their device-provided geospatial points and that the AvgErrDist across all 50K was 1,881 miles. The accuracy does improve close to 50% for tweets that could be geo-located within 1000 miles of their device-provided location.

5.2 Comparison of results after varying algorithmic parameters

For our baseline evaluation, we set the parameters *minWindowSize* and *maxWindowSize* of our Tweet-to-Document generation (Algorithm 1 described in section 4.1) to 4 hours and 8 hours respectively. These values were motivated by an initial assessment that users tweet on a specific topic for a short period and move on to other topics of interest that are trending on that specific day. The *maxWindowSize* parameter controls the maximum time window allowed for the user's tweets such that they are considered localized to specific topic or news story. In Table 2, we present some results with variation of these parameters.

Table 2. Impact of Tweet-to-Document Generation parameter adjustments on content-based tweet geo-location

Method	AvgErrDist(Miles)	$Accuracy_{100}$	$Accuracy_{500}$	$Accuracy_{1000}$
Base (min:4,max:8)	1881.98	0.122	0.321	0.497
Var1 (min:2,max:8)	773.43	0.313	0.392	0.574
Var2 (min:2,max:4)	**693.24**	**0.377**	**0.412**	**0.581**

In Variant 1, we changed *minWindowSize* parameter to 2 hours which reduced the contextual time window, leading to smaller length documents. We noticed that $Accuracy_{100}$ increased by 156% relative to our baseline parameters and the AvgErrDist also reduced to 773 miles from 1,881 miles. This indicates that, even though shorter time window leads to smaller length documents, the content is more localized to a specific region. In Variant 2, we changed both, the *minWindowSize* and *maxWindowSize* parameters to 2 hours and 4 hours respectively. This lead to a further improvement in $Accuracy_{100}$; 209% relative to baseline and 20% relative to Variant 1. This improvement indicates that a time window of 4 hours leads to a more optimal context for all tweets that pertain to topic or news story.

6 Twitter Trends on a Map

Our application of content-based geo-location detection is to segregate tweets pertaining to specific hashtags or trending news story and localize them on the global map. Such geo-location leads to detection of news or events that are trending in a specific city, country or region.

Fig. 3. Examples of news trends on a map displays the output of our geo-location detection system as clusters of geospatially distributed tweets matching a search query

As shown in Fig. 3 leftmost map, we searched our database of more than 20 million tweets using the keyword "muslim brotherhood" and displayed the top 1000 tweet results on the global map. As expected, the largest number of hits for this keyword query put the tweets on Egypt. The map in the middle shows an example of an event "roadside bomb" that was trending in and around countries in Middle East on July 3, 2013 and Google News reported roadside bombs in Baghdad, Afghanistan and southern Thailand. The majority of tweets are distributed around Afghanistan and Iraq with a few outliers that mention the keyword "roadside bomb" and are geo-located in India and Yemen. Finally, the rightmost map shows an example of Hashtag #30June that was trending during July 3, 2013 and pertained to trending event "protests in Egypt" that happened on June 30, 2013.

7 Conclusion

In this paper, we present an approach that incorporates two novel algorithms; (1) user-tweeting-frequency based time window to collate multilingual tweets into a document, and, (2) location-entity clustering and disambiguation, for content-based geo-

location detection. We compare our geo-location detection with Twitter-provided device-based and user-profile-based geospatial coordinates and show that we are able geo-locate 58% of the tweets in the test set within 1000 miles with algorithmic parameter adjustments. Furthermore, our content-based geo-location algorithm operates not only on native English but also on machine-translated English tweets, thereby, enabling multilingual tweet geo-location. We demonstrate an application of content-based geo-location of tweets through examples of country-localized trending keyword and geo-political entity.

8 References

1. K. Lerman, R. Ghosh. Information contagion: An empirical study of the spread of news on digg and twitter social networks. In Proceedings of 4th International Conference on Weblogs and Social Media (ICWSM), 2010.
2. Zook, M.; Graham, M.; Shelton, T.; and Gorman, S. 2010. Volunteered geographic information and crowdsourcing disaster relief: A case study of the Haitian earthquake. World Medical & Health Policy 2(2):7–33.
3. T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. WWW '10 Proceedings of the 19th international conference on World Wide Web, pages 851–860, 2010.
4. F. Abel, C. Hau, G. J. Houben, R. Stronkman, and K. Tao. Semantics+filtering+ search=twitcident. Exploring information in social web streams. In Proceedings of the 23rd ACM conference on Hypertext and social media, page 285294, 2012.
5. S. Vieweg, A. L. Hughes, K. Starbird, and L. Palen. Microblogging during two natural hazardsevents: what twitter may contribute to situational awareness? In Proceedings of the 28th international conference on Human factors in computing systems, 2010.
6. A. L. Hughes and L. Palen. Twitter adoption and use in mass convergence and emergency events. International Journal of Emergency Management, 6(3):248260, 2009.
7. Sebba, Mark, Shahrzad Mahootian, and Carla Jonsson. Language Mixing and Code-Switching in Writing: Approaches to Mixed-Language Written Discourse. Routledge Critical Studies in Multilingualism. Routledge, Taylor & Francis Group.
8. Zhiyuan Cheng, James Caverlee, and Kyumin Lee. 2010. You are where you tweet: a content-based approach to geo-locating twitter users. In Proceedings of the 19th ACM international conference on Information and knowledge management (CIKM '10).
9. Fabian Abel, Qi Gao, Geert-Jan Houben, Ke Tao. Analyzing Temporal Dynamics in Twitter Profiles for Personalized Recommendations in the Social Web In Proceedings of ACM WebSci '11, 3rd International Conference on Web Science, Koblenz, Germany (June 2011)
10. William B. Cavnar, John M. Trenkle. N-Gram-Based Text Categorization (1994). In Proceedings of SDAIR-94.
11. S. Miller, J. Guinness, A. Zamanian. Name tagging with word clusters and discriminative training In Proceedings of Human Language Technology conference / North American chapter of the Association for Computational Linguistics annual meeting, Vol. 4 (2004).
12. Roller, Stephen, Speriosu, Michael, Rallapalli, Sarat, Wing, Benjamin and Baldridge, Jason. "Supervised Text-based Geolocation Using Language Models on an Adaptive Grid." Paper presented at the meeting of the EMNLP-CoNLL, 2012.
13. Paradesi, Sharon Myrtle. "Geotagging Tweets Using Their Content." In FLAIRS Conference. 2011.

Learning Shortest Paths for Word Graphs

Emmanouil Tzouridis and Ulf Brefeld

Technische Universität Darmstadt,
Hochschulstr. 10, 64289 Darmstadt, Germany
{tzouridis,brefeld}@kma.informatik.tu-darmstadt.de

Abstract. The vast amount of information on the Web drives the need for aggregation and summarisation techniques. We study event extraction as a text summarisation task using redundant sentences which is also known as sentence compression. Given a set of sentences describing the same event, we aim at generating a summarisation that is (i) a single sentence, (ii) simply structured and easily understandable, and (iii) minimal in terms of the number of words/tokens. Existing approaches for sentence compression are often based on finding the shortest path in word graphs that is spanned by related input sentences. These approaches, however, deploy manually crafted heuristics for edge weights and lack theoretical justification. In this paper, we cast sentence compression as a structured prediction problem. Edges of the compression graph are represented by features drawn from adjacent nodes so that corresponding weights are learned by a generalised linear model. Decoding is performed in polynomial time by a generalised shortest path algorithm using loss augmented inference. We report on preliminary results on artificial and real world data.

1 Introduction

Information is ubiquitous. People are consuming information on the go by reading news articles, blogs entries, checking status updates, or planning the next vacations. However, the informed mobility comes at the cost of relevance. Users need to manually identify relevant pieces of information in the overloaded supply and to aggregate these pieces themselves to find an answer to their query. Thus, there is a real need for techniques that assist the user in separating relevant from irrelevant information and aggregating the pieces of information automatically.

In this paper we study the intelligent aggregation of related sentences to quickly serve the information needs of users. Given a collection of sentences dealing with the same real-world event, we aim at generating a single sentence that is (i) a summarisation of the input sentences, (ii) simply structured and easily understandable, and (iii) minimal in terms of the number of words/tokens. The input collection of sentences is represented as a word graph [4], where words are identified with nodes and directed edges connect adjacent words in at least one sentence. The output is a sequence of words fulfilling conditions (i-iii).

In general, learning such mappings between arbitrary structured and interdependent input and output spaces challenges the standard model of learning a

mapping from independently drawn instances to a small set of labels. In order to capture the involved dependencies it is helpful to represent inputs $\mathbf{x} \in \mathcal{X}$ and outputs $\mathbf{y} \in \mathcal{Y}$ in a joint feature representation. The learning task is therefore rephrased as finding a function $f : \mathcal{X} \times \mathcal{Y} \to \Re$ such that

$$\hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}) \tag{1}$$

is the desired output for any input $\mathbf{x}$ [2, 12]. The function f is a linear model in a joint space $\Phi(\mathbf{x}, \mathbf{y})$ of input and output variables and the computation of the argmax is performed by an appropriate decoding strategy. Prominent applications of such models are part-of-speech tagging, parsing, or image segmentation.

In this paper, we cast sentence compression as a supervised structured prediction problem to learn a mapping from word graphs to their shortest paths. Edges of the graphs are labeled with costs and the shortest path realises the lowest possible costs from a start to an end node. Thus, we aim at finding a function f, such that

$$\hat{\mathbf{y}} = \operatorname*{argmin}_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}). \tag{2}$$

We devise structured perceptrons and support vector machines for learning the shortest path. The latter usually requires the use of two-best decoding algorithms which are expensive in terms of computation time and memory. We proof that an equivalent expression can be obtained by a generalised shortest path algorithm using loss-augmented inference. The latter renders learning much more efficient than using a two-best decoding strategy. Empirically, we compare our approach to the state-of-the-art that uses heuristic edge weights [4] on artificial and real world data and report on preliminary results.

The remainder is structured as follows. Section 2 introduces preliminaries. Our main contribution on learning shortest paths is presented in Section 3 and Section 4 reports on empirical results. Section 5 discusses related work and Section 6 concludes.

2 Preliminaries

2.1 Word Graphs

Word or compression graphs have been studied for instance by [3, 4]. The idea is to build a non-redundant representation for possibly redundant sequences by merging identical observations. From a collection of related sentences $\{s_1, \ldots, s_n\}$, we iteratively construct a word graph by adding sentences one-by-one as follows: We begin with the empty graph and add the first sentence s_1, where every word in the sentence becomes a node and a directed edge connects nodes of adjacent words. After this step, the graph is a sequence representing s_1. Words from the second sentence s_2 are incorporated by differentiating two cases: (i) if the graph already contains the word (e.g., using lowercase representations), the word is

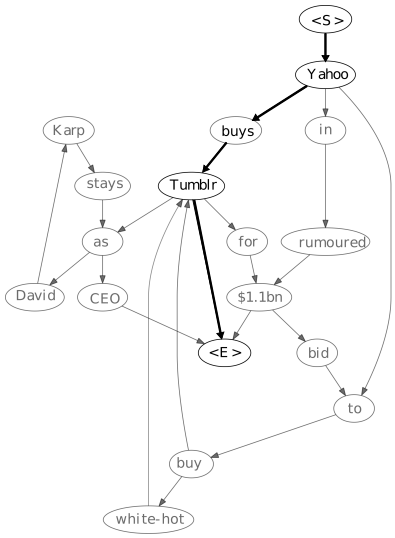


Fig. 1: Word graph constructed from the sentences: "Yahoo in rumoured $1.1bn bid to buy white-hot Tumblr", "Yahoo buys Tumblr as David Karp stays as CEO", "Yahoo to buy Tumblr for $1.1bn". The shortest path is highlighted.

simply mapped to the corresponding node and (ii) otherwise, a new node is instantiated. In both cases, a directed edge is inserted to connect the word to its predecessor from s_2. The procedure continues until all n sentences are incorporated in the graph.

Usually, sentences are augmented by designated auxiliary words indicating the start and the end of the sentence. The sketched procedure merges identical words but preserves the structure of the sentences along the contained paths and the original sentences can often be reconstructed from the compressed representation. There are many different ways to build such graphs, e.g., by excluding punctuations, stop-word removal, or by using part-of-speech information to improve merge operations. Figure 1 shows some related sentences and the corresponding word graph.

2.2 Shortest Path Algorithms

Given a directed weighted graph $\mathbf{x} = (N, E)$, where N is the set of nodes and E the set of edges. As the graph $\mathbf{x}$ defines the sets N and E, we will use $N(\mathbf{x})$ and $E(\mathbf{x})$ in the remainder to denote the set of nodes and edges of graph $\mathbf{x}$, respectively. Every edge $(x_i, x_j) \in E(\mathbf{x})$ is assigned a positive weight given by a cost function $cost : (x_i, x_j) \mapsto \Re^+$. A path $\mathbf{y}$ in the graph $\mathbf{x}$ is a sequence of connected nodes of $\mathbf{x}$ and the cost of such a path is given by the sum of the edge costs for every edge that is on the path. Given the graph $\mathbf{x}$, a start node x_s and

an end node x_e, the shortest path problem is finding the path in $\mathbf{x}$ from x_s to x_e with the lowest costs,

$$\begin{aligned} \underset{\mathbf{y}}{\text{argmin}} \quad & \sum_{(x_i,x_y)\in N(\mathbf{x})} y_{ij} cost(x_i, x_{i+1}) \\ \text{s.t.} \quad & \mathbf{y} \in path(x_s, x_e). \end{aligned}$$

There exist many algorithms for computing shortest paths efficiently [6–8]. Usually, these methods are based on relaxation integer programming, where an approximation of the exact quantity is iteratively updated until it converges to the correct solution. Such algorithms converge in polynomial time if there are no negative cycles inside the graph, otherwise the problem is NP-hard. A prominent algorithm for computing the k-th shortest paths is Yen's algorithm [5]. Intuitively, the approach recursively computes the second best solution by considering deviations from the shortest path, the third best solution from the previous two solutions, and so on. Figure 1 visualises the shortest path for the displayed compression graph.

3 Learning the Shortest Path

3.1 Representation

To learn the shortest path, we need to draw features from adjacent nodes in the word graph to learn the score of the connecting edge. Let x_i and x_j be connected nodes of the compression graph $\mathbf{x}$, that is $x_i, x_j \in N(\mathbf{x})$ and $(x_i, x_j) \in E(\mathbf{x})$. We represent the edge between x_i and x_j by a feature vector $\phi(x_i, x_j)$ that captures characteristic traits of the connected nodes, such as indicator functions detailing whether x_i and x_j are part of the same named entity or part-of-speech transitions.

A path in the graph is represented as an $n \times n$ binary matrix $\mathbf{y}$ with $n = |N(\mathbf{x})|$ and elements $\{\mathbf{y}_{ij}\}$ given by $y_{ij} = [[(x_i, x_j) \in path]]$ where $[[z]]$ is the indicator function returning one if z is true and zero otherwise. The cost of using the edge (x_i, x_j) in a path is given by a linear combination of those features parameterised by $\mathbf{w}$,

$$cost(x_i, x_j) = \mathbf{w}^\top \phi(x_i, x_j).$$

Replacing the constant costs by the parameterised ones, we arrive at the following objective function (ignoring the constraints for a moment) that can be rewritten as a generalised linear model.

$$\sum_{(x_i,x_j)\in E(\mathbf{x})} y_{ij}\, \mathbf{w}^\top \phi(x_i, x_j) = \mathbf{w}^\top \underbrace{\left(\sum_{(x_i,x_j)\in E(\mathbf{x})} y_{ij} \phi(x_i, x_j) \right)}_{=\Phi(\mathbf{x},\mathbf{y})} = \mathbf{w}^\top \Phi(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}, \mathbf{y})$$

Given a word graph $\mathbf{x}$, the shortest path $\hat{\mathbf{y}}$ for a fixed parameter vector $\mathbf{w}$ can now be computed by

$$\hat{\mathbf{y}} = \operatorname*{argmin}_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}),$$

where f is exactly the objective of the shortest path algorithm and the argmin consequently computed by an appropriate solver, such as Yen's algorithm [5].

3.2 Problem Setting

In our setting, word graphs $\mathbf{x} \in \mathcal{X}$ and the best summarising sentence $\mathbf{y} \in \mathcal{Y}$ are represented jointly by a feature map $\Phi(\mathbf{x}, \mathbf{y})$ that allows to capture multiple-way dependencies between inputs and outputs. We apply a generalised linear model $f(\mathbf{x}, \mathbf{y}) = \mathbf{w}^\top \Phi(\mathbf{x}, \mathbf{y})$ to decode the shortest path

$$\hat{\mathbf{y}} = \operatorname*{argmin}_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}).$$

The quality of f is measured by the Hamming loss

$$\Delta_H(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{2} \sum_{(x_i, x_j) \in E(\mathbf{x})} [[y_{ij} \neq \hat{y}_{ij}]]$$

that details the differences between the true $\mathbf{y}$ and the prediction $\hat{\mathbf{y}}$, where $[[\cdot]]$ is again the indicator function from Section 3.1. Thus, the generalisation error is given by

$$R[f] = \int_{\mathcal{X} \times \mathcal{Y}} \Delta_H \left(\mathbf{y}, \operatorname*{argmin}_{\bar{\mathbf{y}}} f(\mathbf{x}, \bar{\mathbf{y}}) \right) dP(\mathbf{x}, \mathbf{y})$$

and approximated by its empirical counterpart

$$\hat{R}[f] = \sum_{i=1}^{m} \Delta_H \left(\mathbf{y}, \operatorname*{argmin}_{\bar{\mathbf{y}}} f(\mathbf{x}, \bar{\mathbf{y}}) \right) \tag{3}$$

on a finite m-sample of pairs $(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), ...(\mathbf{x}_m, \mathbf{y}_m)$ where $\mathbf{x}_i$ is a word graph and $\mathbf{y}_i$ its shortest path (i.e., the best summarising sentence). Minimising the empirical risk in Equation (3) directly leads to an optimisation problem that is not well-posed as there generally exist many equally well solutions that realise an empirical loss of zero. We thus consider also the minimisation of the regularised empirical risk

$$\hat{Q}[f] = \Omega(f) + \sum_{i=1}^{m} \Delta_H \left(\mathbf{y}, \operatorname*{argmin}_{\bar{\mathbf{y}}} f(\mathbf{x}, \bar{\mathbf{y}}) \right) \tag{4}$$

where $\Omega(f)$ places a prior on f, e.g., to enforce smooth solutions. In the remainder we focus on $\Omega(f) = \|\mathbf{w}\|^2$.

3.3 Perceptron-based Learning

Structured Perceptrons [13, 14] directly minimise the empirical loss in Equation (3). The training set is processed iteratively, where in the i-th iteration the actual prediction $\hat{\mathbf{y}} = \operatorname{argmin}_{\mathbf{y}} f(\mathbf{x}_i, \mathbf{y})$ is compared with the true output $\mathbf{y}_i$. If $\Delta_H(\mathbf{y}_i, \hat{\mathbf{y}}) = 0$ the algorithm proceeds with the next instance. However, if $\Delta_H(\mathbf{y}_i, \hat{\mathbf{y}}) \neq 0$ an erroneous prediction is made and the parameters need to be adjusted accordingly. In case of learning shortest paths, we aim at assigning a smaller function value to the true path than to all alternative paths. If this holds for all training examples we have

$$\forall \quad \bar{\mathbf{y}} \neq \mathbf{y}_i : \mathbf{w}^\top \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) - \mathbf{w}^\top \Phi(\mathbf{x}_i, \mathbf{y}_i) > 0. \tag{5}$$

In case one of these constraints is violated, the parameter vector is updated according to

$$\mathbf{w} \leftarrow \mathbf{w} + \Phi(\mathbf{x}_i, \hat{\mathbf{y}}) - \Phi(\mathbf{x}_i, \mathbf{y}_i),$$

where $\hat{\mathbf{y}}$ denotes the erroneously decoded path (the false prediction). One can show that the perceptron converges if an optimal solution realising $\hat{R}[f] = 0$ exists. [13, 14]

3.4 Large-margin Approach

For support vector-based learning, we extend the constraints in Equation (5) by a term that induces a margin between the true path $\mathbf{y}_i$ and all alternative paths. A common technique is called margin-rescaling and implies to scale the margin with the actual loss that is induced by decoding $\bar{\mathbf{y}}$ instead of $\mathbf{y}_i$. Thus, rescaling the margin by the loss implements the intuition that the confidence of rejecting a mistaken output is proportional to its error. In the context of learning shortest paths, margin-rescaling gives us the following constraint

$$\forall \bar{\mathbf{y}} \neq \mathbf{y}_i : \mathbf{w}^\top \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) - \mathbf{w}^\top \Phi(\mathbf{x}_i, \mathbf{y}_i) > \Delta_H(\mathbf{y}_i, \bar{\mathbf{y}}) - \xi_i, \tag{6}$$

where $\xi_i \geq 0$ is a slack-variable that allows pointwise relaxations of the margin. Solving the equation for ξ_i shows that margin rescaling also effects the hinge loss that now augments the structural loss Δ_H,

$$\ell_{\Delta_H}(\mathbf{x}, \mathbf{y}, f) = \max \left[0, \min_{\bar{\mathbf{y}}} \left[\Delta_H(\mathbf{y}_i, \bar{\mathbf{y}}) - \mathbf{w}^\top \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) + \mathbf{w}^\top \Phi(\mathbf{x}_i, \mathbf{y}_i) \right] \right].$$

The effective hinge loss upper bounds the structural loss Δ_H for every pair $(\mathbf{x}_i, \mathbf{y}_i)$ and therefore also

$$\sum_{i=1}^{m} \ell_{\Delta_H}(\mathbf{x}_i, \mathbf{y}_i, f) \geq \sum_{i=1}^{m} \Delta_H(\mathbf{y}_i, \operatorname*{argmin}_{\bar{\mathbf{y}}} f(\mathbf{x}_i, \bar{\mathbf{y}}))$$

holds. Instead of minimising the empirical risk in Equation (3), structural support vector machines [2] aim to minimise its regularised counterpart in Equation

(4). A maximum-margin approach to learning shortest paths therefore leads to the following optimisation problem

$$\begin{aligned} \min_{\mathbf{w},\boldsymbol{\xi}} \quad & \|\mathbf{w}\|^2 + \sum_{i=1}^{m} \xi_i \\ s.t. \quad & \forall i \, \forall \bar{\mathbf{y}} \neq \mathbf{y}_i : \; \mathbf{w}^\top \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) - \mathbf{w}^\top \Phi(\mathbf{x}_i, \mathbf{y}_i) > \Delta_H(\mathbf{y}_i, \bar{\mathbf{y}}) - \xi_i \\ & \forall i : \xi_i \geq 0 \end{aligned}$$

The above optimisation problem can be solved by cutting plane methods (e.g., [2]). The idea behind cutting planes is to instantiate only a minimal subset of the exponentially many constrains. That is, for the i-th training example, we decode the shortest path $\hat{\mathbf{y}}$ given our current model and consider two cases: (i) The case $\hat{\mathbf{y}} \neq \mathbf{y}_i$ the prediction is erroneous and $\hat{\mathbf{y}}$ is called the most strongly violated constraint as it realises the smallest function value, i.e., $f(\mathbf{x}_i, \hat{\mathbf{y}}) < f(\mathbf{x}_i, \mathbf{y})$ for all $\mathbf{y} \neq \hat{\mathbf{y}}$. Consequentially, the respective constraint of the above optimisation problem is instantiated and influences the subsequent iterations. (ii) If instead the prediction is correct, that is $\hat{\mathbf{y}} = \mathbf{y}_i$, we need to verify that the second best prediction $\hat{\mathbf{y}}^{(2)}$ fulfils the margin constraint. If so, we proceed with the next training example, otherwise we instantiate the corresponding constraint, analogously to case (i). Luckily, we do not need to rely on an expensive two-best shortest path algorithm but can compute the most strongly violated constraint directly via the cost function

$$Q(\bar{\mathbf{y}}) = \Delta_H(\mathbf{y}_i, \bar{\mathbf{y}}) - \mathbf{w}^\top \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) + \mathbf{w}^\top \Phi(\mathbf{x}_i, \mathbf{y}_i) \tag{7}$$

that has to be maximised wrt $\mathbf{y}$. The following proposition shows that we can equivalently solve a shortest path problem for finding the maximiser of Q.

Proposition 1 (Loss augmented inference for shortest path problems). *The maximum $\mathbf{y}^*$ of Q in Equation (7) can be equivalently computed by minimising a shortest path problem with cost$(x_i, x_j) = y_{ij} + \mathbf{w}^\top \phi(x_i, x_j)$.*

Proof. In the proof, we treat paths $\mathbf{y}$ as graphs and write $N(\mathbf{y})$ for the set of nodes on the path and $E(\mathbf{y})$ to denote the set of edges that lie on the path. If, for instance, an element of the binary adjacency matrix representing path $\mathbf{y}$ equals one, e.g., $y_{ij} = 1$, we write $y_i, y_j \in N(\mathbf{y})$ and $(y_i, y_j) \in E(\mathbf{y})$. First, note that the Hamming loss can be rewritten as

$$\Delta_H(\mathbf{y}_i, \bar{\mathbf{y}}) = \sum_{(y_i, y_j) \in E(\mathbf{y})} (1 - y_{ij} \bar{y}_{ij}) \, . \tag{8}$$

Using Equation (8), we have

$$\begin{aligned}
\hat{\mathbf{y}} &= \operatorname*{argmax}_{\bar{\mathbf{y}}} \Delta_H(\mathbf{y}_i, \bar{\mathbf{y}}) + \mathbf{w}^\top \Phi(\mathbf{x}_i, \mathbf{y}_i) - \mathbf{w}^\top \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) \\
&= \operatorname*{argmax}_{\bar{\mathbf{y}}} \ \Delta_H(\mathbf{y}_i, \bar{\mathbf{y}}) - \mathbf{w}^\top \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) \\
&= \operatorname*{argmax}_{\bar{\mathbf{y}}} \sum_{(y_i,y_j)\in E(\mathbf{y})} (1 - y_{ij}\bar{y}_{ij}) - \mathbf{w}^\top \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) \\
&= \operatorname*{argmax}_{\bar{\mathbf{y}}} - \sum_{(y_i,y_j)\in E(\mathbf{y})} y_{ij}\bar{y}_{ij} - \mathbf{w}^\top \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) \\
&= \operatorname*{argmin}_{\bar{\mathbf{y}}} \sum_{(y_i,y_j)\in E(\mathbf{y})} y_{ij}\bar{y}_{ij} + \mathbf{w}^\top \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) \\
&= \operatorname*{argmin}_{\bar{\mathbf{y}}} \sum_{(y_i,y_j)\in E(\mathbf{y})} y_{ij}\bar{y}_{ij} + \mathbf{w}^\top \left(\sum_{(x_i,x_j)\in E(\mathbf{x})} \bar{y}_{ij}\phi(x_i, x_j) \right) \\
&= \operatorname*{argmin}_{\bar{\mathbf{y}}} \sum_{(x_i,x_j)\in E(\mathbf{x})} y_{ij}\bar{y}_{ij} + \mathbf{w}^\top \left(\sum_{(x_i,x_j)\in E(\mathbf{x})} \bar{y}_{ij}\phi(x_i, x_j) \right) \\
&= \operatorname*{argmin}_{\bar{\mathbf{y}}} \sum_{(x_i,x_j)\in E(\mathbf{x})} (y_{ij} + \mathbf{w}^\top \phi(x_i, x_j))\bar{y}_{ij}
\end{aligned}$$

The output $\hat{\mathbf{y}}$ is the shortest path with costs given by $y_{ij} + \mathbf{w}^\top \phi(x_i, x_j)$. □

Given a parameter vector $\mathbf{w}$ and start and end nodes x_s and x_t, respectively, the optimisation of Q can be performed with the following linear program.

$$\begin{aligned}
\min_{\bar{\mathbf{y}}} \ & \sum_{ij} \left(y_{ij} + \mathbf{w}^\top \phi(x_i, x_j) \right) \bar{y}_{ij} \\
s.t. \quad & \forall k \in N(\mathbf{x})/\{s,t\} : \ \sum_j \bar{y}_{kj} - \sum_i \bar{y}_{ik} \leq 0 \\
& \forall k \in N(\mathbf{x})/\{s,t\} : -\sum_j \bar{y}_{kj} + \sum_i \bar{y}_{ik} \leq 0 \\
& \sum_j \bar{y}_{sj} - \sum_i \bar{y}_{i,s} \leq 1 \quad \wedge \quad -\sum_j \bar{y}_{sj} + \sum_i \bar{y}_{is} \leq -1 \\
& \sum_i \bar{y}_{it} - \sum_j \bar{y}_{tj} \leq 1 \quad \wedge \quad -\sum_i \bar{y}_{it} + \sum_j \bar{y}_{tj} \leq -1 \\
& \forall (i,j) : \ y_{ij} \leq x_{(i,j)} \quad \wedge \quad \forall (i,j) : \mathbf{y}_{ij} \in \{0,1\}
\end{aligned}$$

The first two constraints guarantee that every inner node of the path must have as many incoming as outgoing edges, the third line of constraints guarantees the path to start in x_s and, analogously, the forth line ensures that it terminates in x_t. The last line of constraints forces the edges of the path $\bar{\mathbf{y}}$ to move along existing paths of $\mathbf{x}$.

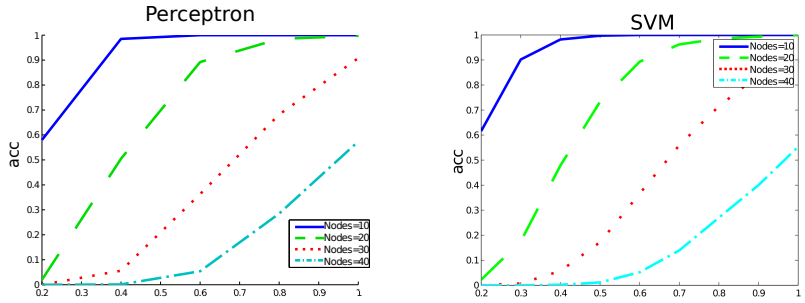


Fig. 2: Performance on artificial data for perceptrons (left) and SVMs (right).

4 Empirical Results

4.1 Artificial Data

To showcase the effectivity of our approach, we generate artificial data as follows. We sample random graphs with $|N| \in \{10, 20, 30, 40\}$ nodes. For every node in a graph, we sample the number of outgoing edges uniformly in the interval $[\frac{|N|}{2}, |N|]$. For every outgoing edge, a receiving node is sampled uniformly from the remaining $|N|-1$ nodes. The optimal path is annotated as follows. We draw the length of the path uniformly in the interval $[\frac{|N|}{2}, |N|]$ and randomly select the respective number of nodes from N. This gives us a sequence of nodes that we define as the shortest path. In case two adjacent nodes are not connected by an edge, we discard the nodes and draw again.

To ensure that the optimal path is actually the one with lowest costs, edge features are sampled from a one-dimensional Gaussian mixture distribution, where the generating component is chosen according to whether the respective edge lies on the the shortest path or not. That is, we introduce two Gaussian components $G_{0,1}$, so that costs for edges lying on the shortest path are drawn from $G_0(\mu_1, \sigma_1^2)$ while costs for all other edges are sampled from $G_1(\mu_2, \sigma_2^2)$.

The difficulty of the experimental setup is controlled by a parameter α that measures the distance of the two means, i.e., $\alpha = |\mu_1 - \mu_2|$. We initialise the components $G_{0,1}$ by drawing one of them (say q) according to a coin flip and sample the corresponding mean from a normal distribution $\mu_q \sim G_q(-\frac{\alpha}{2}, 0.1)$. $G_{\bar{q}}$ is then initialised with $\mu_{\bar{q}} \sim G_{\bar{q}}(\frac{\alpha}{2}, 0.1)$. We use $\sigma_1 = \sigma_2 = 0.01$. We report on averages over 100 repetitions.

The results for perceptrons and SVMs are shown in Figure 2. The distance α is depicted on the x-axis. The y-axis shows the top-one accuracy. The performance of both algorithms highly depends on the distance of the cost-generating components and the size of the graph. The fewer nodes and the larger the distance α, the more accurate is the prediction. Both algorithms perform similarly.

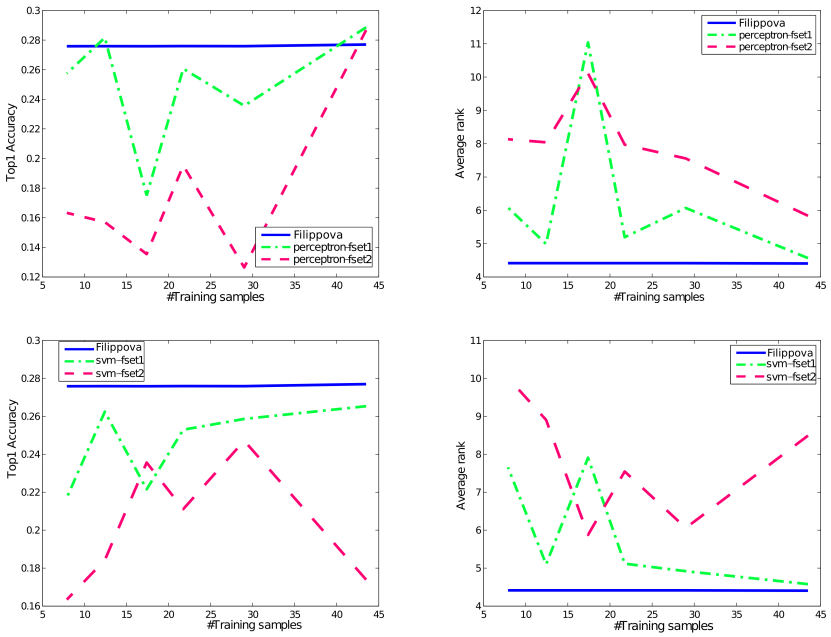


Fig. 3: Performance on news headlines. Accuracies (left column) and average ranks (right column) are evaluated for perceptrons (top row) and SVMs (bottom row)

4.2 News Headlines

The real-world data originates from news articles which have been crawled from several web sites on different days. We use categories *Technology, Sports, Business* and *General* and focus only on the headlines. Related sets of news headlines are manually identified and grouped together. We discard collections with less than 5 headlines and build word graphs for the remaining sets following the procedure described in Section 2.1 where we remove stop words. Ground truth is annotated manually by selecting the best sentence among the 20 shortest paths computed by Yen's algorithm [5] using frequencies as edge weights, this annotation has been done by one of the authors. This process leaves us with 87 training examples.

We aim to learn the costs for the edges that give rise to the optimal compression of the training graphs. We devise three different sets of features. The first feature representation consists of only two features that are inspired by the heuristic in [4]. For an edge (x_i, x_j), we use

$$\phi_1(x_i, x_j) = \left(\frac{\#(x_1)}{\#(x_1, x_2)}, \frac{\#(x_2)}{\#(x_1, x_2)} \right)^\top ,$$

Table 1: Leave-one-out results for news headlines

	avg. acc.	avg. rank
Filipova	0.277	4.378
SVM-fset3	0.252	6.942

where # denotes the frequency of nodes and edges, respectively. The second feature representation extends the previous one by Wordnet similarity $sim_W(x_1, x_2)$ of the nodes and frequencies $\#_L$ taken from the Leipzig corpus, to incorporate the notion of global relatedness,

$$\phi_2(x_i, x_j) = \left(\frac{\#(x_1)}{\#(x_1, x_2)}, \frac{\#(x_2)}{\#(x_1, x_2)}, sim_W(x_1, x_2), \frac{\#_L(x_1)}{\#_L(x_1, x_2)}, \frac{\#_L(x_2)}{\#_L(x_1, x_2)} \right)^\top .$$

Finally, we use a third feature representation which is again inspired by [4]. Instead of precomputing the surrogates, we simply input the ingredients to have the algorithm pick the best combination,

$$\phi_3(x_i, x_j) = \left(\#(x_1), \#(x_2), \#(x_1, x_2), \log\left(\#(x_1)\right), \log\left(\#(x_2)\right), \log\left(\#(x_1, x_2)\right)\right)^\top .$$

We compare our algorithms with the unsupervised approach by Filippova [4]. In that work, the author extracts the shortest path from the word-graph that uses $\frac{\#(x_1)+\#(x_2)}{\#(x_1,x_2)}$ as edge weights.

Figure 3 shows average accuracy (left column) and average rank (right column) for perceptrons (top row) and SVMs (bottom row) for different training set sizes, depicted on the x-axis. Every curve is the result of a cross-validation that uses all available data. Thus, the rightmost points are generated by a 2-fold cross validation while the leftmost points result from using 11-folds. Due to the small training sets, interpreting the figures is difficult. The unsupervised baseline outperforms the learning methods although there are indications that more training data could lead to better performances of perceptrons and SVMs. The first feature representation shows better performances than the second. However, these conjectures need to be verified by an experiment on a larger scale.

Finally, Table 1 shows average accuracies and average ranks for a leave-one-out setup using the third feature representation. The results are promising and not too far from the baseline, however, as before, the evaluation needs to be based on larger sample sizes to allow for interpretations.

5 Related work

Barzilay et al. [9] study sentence compression using dependency trees. Aligned trees are represented by a lattice from which a compression sentence is extracted by an entropy-based criterion over all possible traversals of the lattice. Wan et al. at [11] use a language models in combination with maximum spanning trees to rank candidate aggregations that satisfy grammatical constrains.

While the previous approaches to multi-sentence compression are based on syntactic parsing of the sentences, word graph approaches have been proposed, that do not make use of dependency trees or other linguistic concepts. Filippova [4] casts the problem as finding the shortest path in directed word graphs, where each node is a unique word and directed edges represent the word ordering in the original sentences. The costs of these edges are a heuristic function that is based on word frequencies. Recently, Boudin and Morin [10] propose a re-ranking scheme to identify summarising sentences that contain many keyphrases. The underlying idea is that representative key phrases for a given topic give rise to more informative aggregations.

In contrast to the cited related work, we cast the problem of sentence compression as a supervised structured prediction problem and aim at learning the edge costs from a possibly rich set of features describing adjacent nodes.

6 Conclusion

In this paper, we proposed to learn shortest paths in compression graphs for summarising related sentences. We addressed the previously unsupervised problem in a supervised context and devised structured perceptrons and support vector machines that effectively learn the edge weights of compression graphs, so that a shortest path algorithm decodes the best possible summarisation. We showed that the most strongly violated constrains can be computed directly by loss-augmented inference and rendered the use of expensive two-best algorithms unnecessary. Empirically, we presented preliminary results on artificial and real world data sets. Due to small sample sizes, conclusions cannot be confidently drawn yet, although the results seemingly indicate that learning shortest paths could be an alternative to heuristic and unsupervised approaches. Future work will address this question in greater detail.

References

1. U. Brefeld. Cost-based Ranking in Input Output Spaces. Proceedings of the Workshop on Learning from Non-vectorial Data, 2007.
2. I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, Large Margin Methods for Structured and Interdependent Output Variables, Journal of Machine Learning Research, 6 (Sep):1453-1484, 2005
3. R. Barzilay, L. Lee, Learning to Paraphrase: An Unsupervised Approach Using Multiple-Sequence Alignment, in Proc. of NAACL-HLT, 2003.
4. K. Filippova. Multi-sentence compression: Finding shortest paths in word graphs, COLING, 2010
5. J. Y. Yen, Finding the k Shortest Loopless Paths in a Network. Management Science 17 (11): 712716, 1971
6. R. Bellman (1958). "On a routing problem". Quarterly of Applied Mathematics 16: 8790. MR 0102435.
7. J. Ford, R. Lester (August 14, 1956). Network Flow Theory. Paper P-923. Santa Monica, California: RAND Corporation.

8. E. W. Dijkstra (1959). "A note on two problems in connexion with graphs". Numerische Mathematik 1: 269271. doi:10.1007/BF01386390.
9. R. Barzilay and K. McKeown. Sentence Fusion for Multidocument News Summarization, Computational Linguistics, 2005.
10. F. Boudin and E. Morin. Keyphrase Extraction for N-best Reranking in Multi-Sentence Compression, Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2013.
11. S. Wan, R. Dale, M. Dras, C. Paris. Global revision in summarisation : generating novel sentences with Prim's algorithm, Conference of the Pacific Association for Computational Linguistics, 2007.
12. B. Taskar and D. Klein and M. Collins and D. Koller and C. Manning. Max-margin parsing, Proc. EMNLP, 2004.
13. M. Collins and N. Duffy. New Ranking Algorithms for Parsing and Tagging: Kernels over Discrete Structures, and the Voted Perceptron, ACM, 2002
14. Y. Altun, M. Johnson, T. Hofmann. Investigating loss functions and optimization methods for discriminative learning of label sequences, Proc. EMNLP, 2003.
15. Princeton University "About WordNet." WordNet. Princeton University. 2010. http://wordnet.princeton.edu
16. Leipzig Corpora Collection (LCC). Universität Leipzig, Institut für Informatik, Abteilung Sprachverarbeitung. http://corpora.uni-leipzig.de/