Generating storylines from sensor data

Jordan Frank
School of Computer Science
McGill University
Montreal, Canada
jordan.frank@cs.mcgill.ca

Shie Mannor
Dept. of Electrical Engineering
Technion
Haifa, Isreal
shie@ee.technion.ac.il

Doina Precup
School of Computer Science
McGill University
Montreal, Canada
dprecup@cs.mcgill.ca

ABSTRACT

We present an approach for producing narratives, or storylines, from sensor data collected from a mobile phone. Given a training set of English-language descriptions of events and a set of corresponding sensor data, we learn a probabilistic translation model. Then, given new sensor data, our model can produce English-language descriptions of the events present in the data. Our approach is evaluated on the data provided as a part of the Nokia Mobile Data Challenge (MDC), focusing, in particular, on location labeling. We also present a set of tools for visualising the MDC data, that were used to generate training data for our evaluation.

1. INTRODUCTION

In the last decade, we have seen the instrumentation of society at a pace unrivaled in our history. Smartphones, devices capable of accurately monitoring movement, location, communication, and information consumption, have become ubiquitous. This work considers the problem of adding intelligibility to data available on these devices. We have developed a framework for translating between the raw sensor data available on the device and a human-language (English, in our experiments) description of the events present in the data. A novel location clustering approach is presented which allows us to extract location-based events, such as arriving at or leaving a place of interest, and the labels for locations are learned from event descriptions provided by the user. While we focused on location-based events in this paper, we also describe how our framework can be extended to annotate other types of events, such as phone calls, text messages, and application usage.

We present our translation framework in Section 2 and a brief description of our location-labeling algorithm in Section 3. Our framework is evaluated on data from the Nokia Mobile Data Challenge [7], and the results are presented in Section 4. We conclude by discussing some avenues for future work in Section 5.

This material was prepared for the Mobile Data Challenge 2012 (by Nokia) Workshop; June 18-19, 2012; Newcastle, UK.

2. TRANSLATION FRAMEWORK

Our approach builds on techniques from statistical machine translation (SMT) [6]. However, as opposed to the typical SMT setting, where a pair of parallel corpora are used as training data, the input and output languages differ greatly in our setting. For training the translator, we use as input English-language descriptions of events, such as "I left home and went to my office, arriving at 5:00PM". The output consists of raw sensor data, for example, a gps sensor reading such as $\{\text{time: } 1272222153, \text{lat: } 46.527, \text{lon: } 6.5831, \dots\}$. In the typical SMT setting, the training corpora consist of aligned sentences in the input and output languages, whereas we must deduce the alignment automatically. Additionally, the sensor data consist of a large number of values that can be ignored. For instance, the battery level may be reported every minute, but in the description of a user's day, one would not want mentioned every battery level reading. As our goal is to present a concise summary of the events that occurred, the first step is to preprocess the data in order to find events of interest.

The events of interest depend greatly on the data modality. For instance, for call log data, every event is likely of interest, while for data pertaining to location, one might only be interested in when the user moves from one location to another. In the following section, we discuss how we extract location-based events of interest. For many other modalities, such as application usage events, call and text messaging events, and status event (e.g., ringer disabled, phone plugged in, etc.), it is assumed that all entries in the sensor data are of interest, although performing some additional preprocessing would likely be valuable. To ease our notation, we will refer to the English-language description of events as *event descriptions*, and the set of events of interest as the *event data*.

Four assumptions are made about the event descriptions on which the translator is trained. First, it is assumed that every sentence contains at least one reference to the time at which the event being described occurs. Second, a sentence can be taken as a stand-alone description of at least one event, and does not reference from other sentences (e.g. no pronouns refer to nouns in other sentences). Third, it is assumed that every sentence has some sensor data associated with it, that is there are no sentences describing events that could not be deduced from the available sensor data. By restricting ourselves to a simpler subset of English, good results can be achieved with few event descriptions. As

training data is expensive to collect, and requires accurate recollections of events, we believe that this tradeoff is justified. Finally, it is assumed that there is a large amount of event data, but a much smaller number of event descriptions.

We begin by considering each sentence to be an event description. An annotated parse tree of each sentence is generated using the Stanford English PCFG parser [5] and Stanford Named Entity Recognizer [3]. Through the annotations, the number of time references in each sentence can be determined. If there is more than one reference to a time, for example in the sentence *"I left my office at 12:07PM and walked to the cafe, arriving at 12:11PM."*, a simple splitting rule is used to split the sentence by finding the deepest common ancestor x in the parse tree of the words referencing the times, typically a conjunction, and returning two trees, each with one of the children of x removed. In our example sentence, we would get two sentences "*I left my office at 12:07PM.*", and "*I walked to the cafe, arriving at 12:11PM.*" A final tree transformation is applied to automatically split sentences such as "*I emailed Jane and left my office at 12:30PM*", into two sentences "*I emailed Jane at 12:30PM*" and "*I left my office at 12:30PM*". This is done automatically using the tree transformation `(S NP (VP VP1 CC VP2) .)` $\rightarrow$ `(S NP VP1 .),(S NP VP2 .)`[1].

Next, features are extracted from each sentence using a set of heuristics that were tailored to this particular domain and the limited subset of the English language being considered. The verb at the root of the dependency tree for each sentence is extracted, as it generally represents the *event action* The dependent subject and object are extracted, as they generally represent the caller and callee, or text message sender and recipient, respectively. Location-related actions, indicated by the verbs *leave*, *drive*, *walk*, *take*, *go*, *return*, and *arrive*, are treated as a single *location* action. For sentences describing location-related actions the parse tree is searched for a noun phrase with a preposition parent, as such a phrase often contains the label for the location to which the action applies. Named entities that represent times are extracted, indicating the time at which the action occurred, and in some cases the duration of the action being described (e.g., the length of a call).

Let A denote the set of all event actions observed in the training sentences. Each event description is represented as a tuple (t, a, ϕ), where t is the timestamp, $a \in A$ denotes the action, and ϕ denotes the features (whose domain varies depending on the value of a, as described previously). Each sensor datum is represented as a tuple (t, e, θ), where t is the timestamp, e is the event type, and θ denote the features (whose domain depends on e). The event types and corresponding features are described in [7].

As previously discussed, one of the main challenges is aligning event descriptions with the appropriate event data. It is assumed that the event times given in the event descriptions are accurate to within one minute. For each event description, a window of the event data in the range of one minute prior to and one minute after the event time is retreived. Each event datum has an associated type (e.g., `gps`, `calllog`, `application`, etc.). From this data the maximum likelihood distribution, $p(e|a)$, for event types conditioned on the event action is computed. Additionally, for each event type that appears in the window, a distribution $p(\theta|e, a, \phi)$ is computed. The most general form for this latter distribution can be computed by taking the cross product of the feature values θ and ϕ and computing conditional distributions $p(\theta_i|\phi_j)$ for each pair $(i, j) \in \{1, \ldots, |\theta|\} \times \{1, \ldots, |\phi|\}$.

However, the large number of parameters would necessitate a large amount of training data. It is reasonable to assume that some knowledge of the semantics of the event data features ϕ is available, and this knowledge can be used to construct distributions for each event type by comparing only the appropriate pairs of features. For instance, for call log data, where

$$\theta = (\mathit{length}, \mathit{description}, \mathit{direction}, \mathit{number})$$

and

$$\phi = (\mathit{object}, \mathit{subject}, \mathit{duration}),$$

we have

$$\begin{aligned} &p(\theta|\phi, e = \texttt{calllog}, a = \texttt{call}) = \\ &\quad \delta(\mathit{direction} = \texttt{incoming})\delta(\mathit{object} = \texttt{me})\times \\ &\quad \delta(|\mathit{length} - \mathit{duration}| < 60)p(\mathit{number}|\mathit{name} = \texttt{subject})+ \\ &\quad \delta(\mathit{direction} = \texttt{outgoing})\delta(\mathit{subject} = \texttt{me})\times \\ &\quad \delta(|\mathit{length} - \mathit{duration}| < 60)p(\mathit{number}|\mathit{name} = \texttt{object}), \end{aligned}$$

where δ is the Kronecker delta function and $p(\mathit{number}|\mathit{name})$ can be estimated from the training data and potentially augmented with information from the user's address book.

We describe the procedure for computing $p(\theta|e = \texttt{wifi}, a = \texttt{location}, \phi)$ in the next section.

3. LOCATION-LABELING ALGORITHM

In this section we describe a novel approach to finding places of interest based on wifi signals. An advantage of using wifi signals is that locations can be detected in indoor environments, when GPS reception is unavailable. Localization using wifi signals or, more generally, RF-signals, is an area of active research. Approaches for localization from RF-signals can roughly be categorized into those that explicitly build an RF-signal map (examples include RADAR [1] and Horus [10]), and those that rely on an RF signal propagation model (examples include ARIADNE [4] and EZ [2]). What differentiates our approach from many existing approaches is that we are not interested in placing the user on a map or in physical space, but in identifying locations of interest and determining when the user is present in a particular location of interest. Put another way, we are not concerned with predicting coordinates for a user at a particular time, we are concerned with predicting the label that a user would use to describe the location they are in at a particular time.

The typical assumption made when inferring locations from wifi signals is that for a fixed location, the received signal strength (RSS) measurements from each access point in range of the receiver can be modelled by a constant plus Gaussian noise [10]. We have found this assumption to be frequently violated in practice. For example, we consider the wifi signals from one of the users in the MDC data set

[1]See [6, chapter 11] for an explanation of this syntax

between the hours of 02:00 and 06:00 every day for the 14 months of available data, for a total of 12,812 observations. During this time, the user was assumed to be at home, and the phone was stationary. Of the 92 visible access points, the Shapiro-Wilks normality test rejects 65 of the sets of RSS values as being normally distributed with significance level 0.01. While this alone does not constitute a thorough analysis, it certainly provides evidence that the Gaussian assumption for RSS signals is a concern.

Instead, we take an approach motivated by topic modeling for text data, and model the data by a hierarchical Dirichlet process (HDP) [8]. The HDP is more forgiving than a Gaussian model of access points that occasionally disappear but have high RSS when they are visible[2]. An advantage of the HDP is that it is nonparametric, and ideally uses only as many clusters as are present in the data. We use the online HDP learning algorithm [9] to learn an individual clustering model for each user. A common criticism of nonparametric clustering models is that they are expensive to learn, but this particular algorithm is efficient enough to cluster over a year of data from a particular user (over 84,000 observations) in approximately 25 minutes on a standard desktop computer.

A *wifi observation* consists of a set of visible access points, and the corresponding RSS, for a particular instance in time. In the language of topic modeling, the HDP posits a generative model where a document with N words is produced by first selecting a distribution X over the topics, then, N times, sampling a topic t from X and a word from t. For our purposes, a wifi observation corresponds to a document where the *words* are the access points that are visible in a wifi observation and the word counts correspond to the RSS values (suitably discetized). Clusters (i.e., topics) correspond to distributions over wifi access points. The HDP is a mixture model, and thus assigns to each wifi observation a distribution over clusters.

Due to the nature of the HDP model, where the expected number of clusters scales logarithmically with the size of the data set, locations that are frequented more often tend to be represented by more clusters. This has the advantage of allowing for a finer level of discrimination in locations where the user spends a lot of time. The downside of this is that there is rarely a one-to-one correspondence between clusters and locations. To address this, we build a weighted graph of co-occurring clusters. Let $n_{i,j}$ be the number of observations that are assigned nontrivial probabilities ($p > \varepsilon = 0.0001$) to clusters i and j, and let n_i be the number of observations where cluster i has nontrivial probability. Each cluster with $n_i > 0$ is represented by a vertex, and two vertices i and j are connected by an edge with weight $n_{i,j}/(n_i n_j)$ if and only if $n_{i,j} > 0$. Figure 1 shows the graph for a particular user. A full-sized version of the plot is included in the supplementary material, but the the two groups of densely connected vertices representing locations associated with *home* and *work* are clearly visible in Figure 1. The supplementary material also includes two videos of the movement of a user visualised by the cluster occupancy for each hour of the day.

[2]By analogy to topic modeling, the word *relativity* may occur frequently in some documents associated with a *physics* topic, but my be completely absent from others.

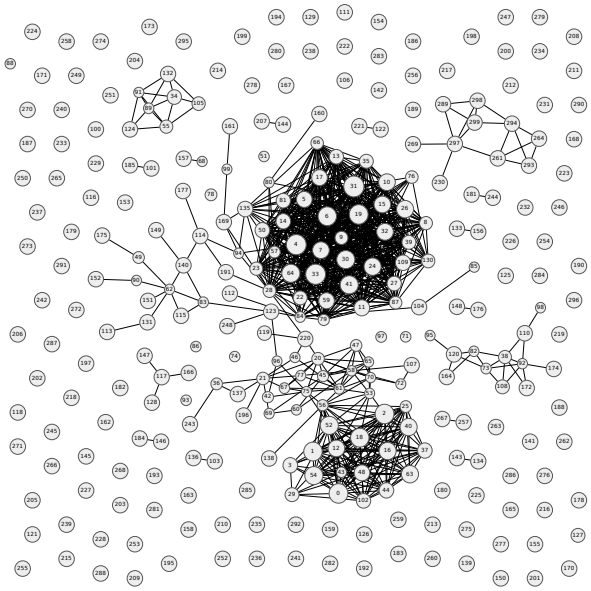

Figure 1: Wifi cluster cooccurrence plot. The vertices represent location clusters, and pairs of location clusters are connected by an edge if an only if they are both observed with nontrivial probability in a single observation. The largest group of densely connected vertices represents locations associated with "home", and the second-largest group represents locations associated with "work".

The cluster occupancy for each hour of the day is averaged over all weekdays and all weekends separately, to visualise the average behaviour of the user on these two categories of days.

Given a set of location-related event descriptions (as defined in the previous section), the distribution $p(c|l)$, the cluster probability given a particular a location label, is estimated for the clusters present labeled data. The clustering is performed on the entire data set, so many of the clusters may not appear in the data associated with the event descriptions. The weighted graph is used to propagate the distribution $p(c|l)$ through connected components of the graph in an iterative manner. At each iteration, for each node that does not have an associated distribution $p(c|l)$, a weighted sum of the distributions of its neighbours is computed, and associated with the node. If none of the neighbours have an associated distribution $p(c|l)$, then the node is not updated. The procedure repeats until no nodes are updated. Some nodes will still not have an associated distribution $p(c|l)$, but this is to be expected as there are certainly locations that occur in the data for which we do not have event descriptions. The supplementary material includes a plot of the clusters with the labels that are explicitly assigned to them by the event descriptions, and a plot depicting the result of propagating the labels through the graph.

We compute $p(c)$ from the clustering of the entire data set, and $p(l)$ from the event descriptions by counting the number of mentions of each location. We initially considered the proportion of time spent at each location by computing

the interval between arriving and leaving each location (in the event descriptions), but found that using the number of times each location was mentioned performed much better.

Once the distributions $p(c)$, $p(l)$, and $p(c|l)$ have been estimated, the distribution $p(l|c) = p(c|l)p(l)/p(c)$ is computed, which is used to label wifi observations based on the observed clusters. For a sequence of event data, most likely locations for each wifi observation are computed and then simple smoothing is performed to eliminate oscillations by replacing occurrences of the sequence $l_1 l_2 l_1$ with $l_1 l_1 l_1$. Events of interest are represented as the points in time when the location changes. Arrival events are generated when a pattern $l_2 l_1 l_1$ is observed, and leaving events are generated when a pattern $l_1 l_1 l_2$ is observed. Despite the simplicity of these rules, the performance is quite good, as is demonstrated in the following section.

4. EVALUATION

We evaluate our approach on the Nokia MDC data [7]. The data do not come with event descriptions, so we generated these by hand. In order to generate reasonable event descriptions, we developed a data visualisation tool. The visualisation tool is a web application that allows one to browse, by user and by day, the MDC data. Data from the GPS, GSM, and Wifi are plotted on Google maps. The location of GSM radio towers are retrieved from the OpenCellID project[3], and wifi locations are computed using both the available `wlan_location` data provided by Nokia and the Google location API[4].

The wifi clustering results are plotted, which allows us to qualitatively assess the performance of the HDP model for clustering. It is striking that patterns of movement indoors are visible in the cluster probabilities. While ground truth data is not available, it appears that the cluster probabilities can be used to accurately discriminate between various locations indoors, where GPS data is unavailable.

A screenshot of the wifi location visualisation is shown in Figure 2, which depicts the cluster occupancy for one of the users on a particular Friday, between 11:45 and 14:45. The horizontal axis represents time, and the vertical axis represents the cluster ids that are observed. The colours are used to depict the times when clusters are observed, and the shading depicts the probability of the cluster. One can clearly see that between approximately 13:00 and 14:00, the user visited a new location, after which they returned to the previous location. In fact, on many other Fridays, between 13:00 and 14:00, the user was in this same location, and since the most recent GPS observations place this location somewhere near EPFL, it is likely that this is a class, or regular meeting attended by the user. However, this location information is not present in either the GPS or GSM data, and based on our experience, a density-based clustering (e.g., DBSCAN) of the raw wifi data would merge the office locations (clusters 0, 1, and 2) with this other location (cluster 3), as many of the same wifi access points are visible.

[3] `http://www.opencellid.org/`
[4] `https://developers.google.com/maps/`

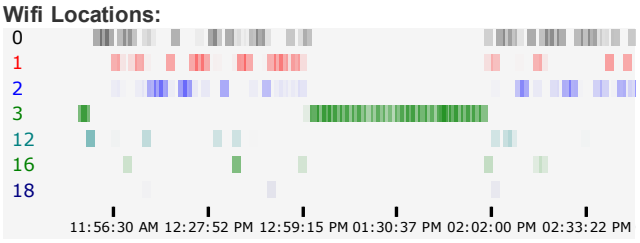

Figure 2: Location clusters showing indoor location discovery. The horizontal axis represents time, and the values on the vertical axis represent individual clusters. The coloured regions indicate that a particular cluster is observed at a particular time, and the shading represents the probability assigned to that cluster (darker represents higher probability).

Using the visualisation tool, we fantasized two weeks worth of journal entries for one particular user (chosen randomly). We invented names for people that the user communicated with, as the data was anonymized, and invented place names based on visual inspection of the wifi location clusters, and by picking the names of nearby establishments from Google maps when GPS data was available. We were very specific, and included the names of bus and metro lines that the user used to get to places, which we could deduce from the GPS data and the Lausanne public transit website[5]. In the interest of the user's privacy, we have changed the names of the locations. A short segment of a journal entry is included:

I left Home at 7:32AM. I walked to Perrelet Bus Station, arriving at 7:35AM. I checked my Calendar at 7:36AM. I checked my messages at 7:37AM. I took the 7 Bus to Renens-Gare Nord Bus Station, arriving at 7:47AM. I walked to Renens-CFF Metro Station, arriving at 7:50AM. I took the M1 to EPFL Metro Station, arriving at 8:01AM. I walked to My Office, arriving at 8:06AM. I missed a call from Gary at 8:53AM. I plugged my phone in at 10:12AM. I checked my calendar at 10:15AM. I unplugged my phone at 1:01PM.

We clustered the wifi location and learned a translation model from the synthesized journal entries. The translation model was used to translate the location data for the week following the last journal entry. A short segment is included:

I left home at 04:20. I arrived at auditorium at 11:48. I left auditorium at 11:50. I arrived at lounge at 11:52. I left lounge at 12:38. I arrived at my office at 12:46. I left my office at 19:51. I arrived at library at 19:53. I left library at 19:57. I arrived at epfl metro station at 20:03. I left epfl metro station at 20:06. I arrived at renens-gare nord bus station at 20:19. I left renens-gare nord bus station at 20:25. I arrived at perrelet bus station at 20:29. I left perrelet bus station at 20:33. I arrived at home at 20:34.

The entire generated storyline and a plot of the location graphs, annotated with the locations and their unnormalized log-likelihoods, are included in the supplementary material. We have also included two animations of the cluster occupancy over time for weekends and weekdays in the sup-

[5] `http://www.t-l.ch/`

plementary material.

From visual inspection, the location events detected by our algorithm corresponded perfectly with the events that we observed in the wifi data, and the location labels were all correct. Of the 62 location events detected by the algorithm, 6 locations were marked as unknown, and visual inspection of the data confirmed that these constituted clusters that were not visited and did not co-occur with clusters that were visited during the period covered by the training data. However, 4 of these locations were from geographic regions that the user did visit, and so it is possible that these are errors due to deficiencies in the clustering algorithm. Of course, given the nature of this work, and the fact that we generated the training data, a proper quantitative evaluation is impossible. We discuss this further in the following section.

5. DISCUSSION AND FUTURE WORK

In this paper we presented a framework for generating storylines from sensor data using human-generated journal entries as training data. This framework is built on three components, a translation model, a novel approach to detecting location-based events from wifi data, and a powerful visualisation tool for generating training storylines and assessing the output of our algorithm. We include some videos and images in the supplementary material that demonstrate how our location-labeling approach can be used to visualise and analyse mobility patterns.

As is common for practical applications of machine translation, we relied on heuristics and domain knowledge to accommodate a small training set. We focus on a simple grammar and a restricted subset of the English language, making strong assumptions about the content of the event descriptions. Given more training data, these assumptions could likely be lifted. Additionally we intend to learn a language model from the event descriptions, and to use the learned model to generate storylines that more closely match the style and prose of the training data.

This paper describes the first steps in building a translator between sensor data and human-readable descriptions of events. Accurately summarizing location-based events allows one to answer questions such as "*when was the last time I visited the zoo?*" or "*how long, on average, do I spend waiting for the bus?*", for example, and we intend to develop a system for processing the results and supporting these types of queries. Enhancing the location events with data from the GPS and GSM sensors is an area of future work. We also intend to develop the components for generating descriptions of events related to phone calls, text messages, system events such as plugging in and unplugging the phone, and application usage such as accessing the calendar or surfing the Internet.

As in many tasks involving natural language, performing a quantitative evaluation is difficult. Qualitatively, the storylines generated by our system captured the significant location-based events. However, we intend to perform a more thorough quantitative evaluation which will require us to collect data accompanied by user-provided event descriptions, as opposed to generating the event descriptions ourselves. The quality of the generated storylines could then be evaluated by the users themselves.

This line of research has strong implications for personal privacy. Our intention with this work is to develop methods that can be used to enhance the user experience with personalized analytics. Applications range from personal productivity to security to fitness. We also hope that by making public the nature of the personal information that can be extracted from mobile phones, and by extension any applications that are installed without scrutinizing the permissions they request, we can increase awareness and allow people to make more informed decisions regarding their mobile device usage.

6. REFERENCES

[1] P. Bahl and V. Padmanabhan. RADAR: An inbuilding RF-based user location and tracking system. In *INFOCOM*, 2000.

[2] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan. Indoor localization without the pain. In *MobiCom*, 2010.

[3] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*, 2005.

[4] Y. Ji, S. Biaz, S. Pandey, and P. Agrawal. ARIADNE: A dynamic indoor signal map construction and localization system. In *MobiSys*, 2006.

[5] D. Klein and C. D. Manning. Accurate unlexicalized parsing. In *ACL*, 2003.

[6] P. Koehn. *Statistical machine translation*. Cambridge University Press, 2010.

[7] J. K. Laurila, D. Gatica-Perez, I. Aad, J. Blom, O. Bornet, T. Do, O. Dousse, J. Eberle, and M. Miettinen. The mobile data challenge: Big data for mobile computing research. In *Proc. Mobile Data Challenge by Nokia Workshop, in conjunction with Int. Conf.. on Pervasive Computing*, Newcastle, June 2012.

[8] Y. Teh, M. Jordan, M. Beal, and D. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476), 2006.

[9] C. Wang, J. Paisley, and D. Blei. Online variational inference for the hierarchical dirichlet process. In *AISTATS*, 2011.

[10] M. Youssef and A. Agrawala. The Horus WLAN location determination system. In *MobiSys*, 2005.