Learning Embeddings of Spatial, Textual and Temporal Entities in Geotagged Tweets

Hong Wei
Department of Computer Science
University of Maryland
College Park, Maryland 20742
hyw@cs.umd.edu

Janit Anjaria
Department of Computer Science
University of Maryland
College Park, Maryland 20742
janit@cs.umd.edu

Hanan Samet
Department of Computer Science
University of Maryland
College Park, Maryland 20742
hjs@cs.umd.edu

ABSTRACT

With online social networks being extended to geographical space, location context plays a key role in many applications such as local event detection and location recommendation. Geotagged tweets in Twitter serve as an invaluable source to understand people's activities in urban space. Analyzing geotagged tweets to identify implicit contexts among location, time and text is an interesting problem. In this paper, we present LeGo-CM, a methodology for **Le**earning embeddings of **Geo**tagged tweets for **C**ross-**M**odal search such as locations, time units (hour-of-day and day-of-week) and textual words in tweets. The resulting compact vector representations of these entities make it easy to perform searches like "find which locations are mostly related to the given topics". In LeGo-CM, we first build a graph of entities extracted from tweets in which each edge carries the weight of co-occurrences between two entities. The embeddings of graph nodes are then learned in the same latent space under the guidance of approximating stationary residing probabilities between nodes which are computed using personalized random walk procedures. We evaluate LeGo-CM on datasets of New York City and Los Angeles, showing that the proposed method generally outperforms competitive baseline approaches.

KEYWORDS

Twitter, Geotagged Tweets, Spatial Clustering, Random Walking, Embeddings, Cross-Modal Search

ACM Reference format:
Hong Wei, Janit Anjaria, and Hanan Samet. 2019. Learning Embeddings of Spatial, Textual and Temporal Entities in Geotagged Tweets. In *Proceedings of 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Chicago, IL, USA, November 5–8, 2019 (SIGSPATIAL '19),* 4 pages.

1 INTRODUCTION

Accessing news tweets by location is of great interest (e.g.,see [1, 2] which are based on the NewsStand system [3–5]). Geotagged tweets are particularly interesting in the sense that they provide the complement information about the places of interest [6–13], e.g., where the activities occur. Such location information is crucial when profiling human activities by completing the three pieces of information regarding *where*, *when* and *what*.

In this paper, we aim to uncover the correlation between locations, time and topics in human's urban activities hidden in geotagged tweets. It is, however, challenging to extract location, time and topic context from geotagged tweets. First, although geotagged tweets provide GPS coordinates indicating where people participate in activities, these coordinates often impose certain disagreement even for the same event at the same place, due to the flexibility of people's movement and sometimes the noise of GPS satellite signals. Second, it is hard to effectively and efficiently capture the cross-modal correlations between the spatial, temporal and textual aspects of people's daily-life activities. For example, the techniques of document-term matrix, TF-IDF and Single Value Decomposition (SVD) are often applied to analyze the co-occurrence relationship between locations and words. Such methods, however, can not be easily modified to cope with data of three or more dimensions. Tensor rank decomposition is more promising in modeling high-dimension data but less applicable for large-scale dataset due to its high computational complexity.

This paper aims to learn to represent the spatial, temporal and textual entities in the geotagged tweets by means of embedding vectors in the same semantic space. We propose LeGo-CM to accomplish this learning task. The general idea of LeGo-CM works as follows. First, LeGo-CM extracts essential spatial, temporal and textual entities from the geotagged tweets. Spatial entities refer to locations of interest which witness the aggregation of people. They are usually identified using a clustering algorithm [14, 15] and are in the form of groups of tweet locations. For the publish time of tweets, LeGo-CM uses the features like hour-of-day and day-of-week as temporal entities.As for textual entities in tweets, we address the extracted keywords and phrases after removing stopwords. Second, LeGo-CM systematically constructs a co-occurrence graph that spans spatial, temporal and textual entities in tweets. In particular, the nodes represent the entities, and the edges are weighted by the number of times that two nodes co-occur in tweets. Third, LeGo-CM exploits a graph learning algorithm that approximates the stationary residing probabilities between nodes which result from performing personalized random walk procedures.

The contributions of this paper are summarized as follows.

- First, we comprehensively profile people's activities in Twitter from 4 aspects: location, words, hour-of-day and day-of-week.
- Second, for cross-modal search, we construct a co-occurrence graph to calculate stationary residing probabilities between nodes, which subsequently guides the learning process in the graph embedding algorithm.

SIGSPATIAL '19, Chicago, IL, USA

DOI: 10.1145/3347146.3359108

2 RELATED WORK

There has been much work on identifying correlations between locations and textual contents and sometimes time factors. Some work has focused on discovering geographical topics [16, 17]. Our method is different from these works because they rely on probabilistic graphical models which impose prior distribution assusmptions on the existing data while we rely on the simple co-occurrence relationship to learn embeddings.

Therefore, we are more interested in studies which similarly represent locations and topics in the form of vectors. The techniques based on document-term matrix (such as TF-IDF, SVD) provide a typical way of presenting multi-dimensional data as vectors. However, such techniques are difficult to extend to high-dimensional data. Recently, there have been efforts bringing the technique of word2vec [18] to location-based social networks in order to learn embedding representation of locations and users [19–21]. The foundation of these methods lies in a graph embedding strategy proposed in DeepWalk [22].

Similar to our method, CrossMap [23] also exploits co-occurrence relationships to jointly learns embeddings for location, time and text. The key differences are two-fold: (1) We try to minimize the gap between embedding-based probabilities and graph-based stationary residing probabilities while CrossMap minimizes the difference between embedding-based probabilities and outdegree-based probabilities; (2) We include the features of hour-of-day and day-of-week in time while CrossMap relies on detected temporal hotspots. ReAct [24] also uses co-occurrences between location, time and text in tweets to learn embeddings. However, its location is in the form of grid cells of 300mx300m. Although such a tessellation of space may simplify the processing of geospatial location, its assumption of a uniform distribution may not fit well to real-life tweet data and is sensitive to parameters like grid cell size and sometimes noise. In contrast, our method identifies spatial clusters of tweets as locations of interest beforehand.

Figure 1: System overview

3 METHOD

3.1 Spatial, Temporal and Textual Entity Extraction

3.1.1 Spatial Entity Extraction. We use mean shift to group together GPS points in tweets and thus identify locations of interest. Mean shift is a clustering algorithm that assigns circular regions of data points to clusters by iteratively shifting towards the modes. The mode can be understood as a local maxima of the density function upon the samples of data points. Formally, let z^t be the estimation of mode at iteration t, z^{t+1} can be defined as:

$$z^{t+1} = \frac{\sum_{p \in \mathcal{N}_b(z^t)} K(\frac{p-z^t}{b}) p}{\sum_{p \in \mathcal{N}_b(z^t)} K(\frac{p-z^t}{b})} \tag{1}$$

where $\mathcal{N}_b(z^t)$ represents a set of points falling inside the circular region centered at z^t with a radius of b (also called the bandwidth in the mean shift); $K(\frac{p-z^t}{b})$ is a kernel function that determines the weight of nearby points on the basis of their distance to the mode estimation. In this paper, we use a flat kernel as follows:

$$K(\frac{p-z^t}{b}) := \begin{cases} 1, & \text{if } \|\frac{p-z^t}{b}\| \leq 1 \\ 0, & \text{if } \|\frac{p-z^t}{b}\| > 1 \end{cases} \tag{2}$$

The mean shift continues to iterate until z^t converges to a small variance, e.g., $\|z^{t+1} - z^t\|$ goes below a small threshold, and thereby yields a location of interest.

3.1.2 Temporal and Textual Entity Extraction. Comparing to extracting locations of interest from tweets, it is quite intuitive and straightforward to extract temporal and textual entities. For example, we can directly calculate the local hour-of-day and day-of-week from the UNIX timestamp in a tweet's publication time. As for essential words in tweets, we exploit the off-the-shelf tool [25, 26] [1] to attain entities and noun phrases as textual entities [15].

3.2 Co-occurrence Graph Construction

Up to now, we have 4 types of entities: location, hour-of-day (hour), day-of-week (wday) and word as illustrated in Figure 2. These entities function as the vertices in the co-occurrence graph $G = (V, E)$. When building the co-occurrence graph G, an edge e_{ij} between node v_i and node v_j establishes and its weight is added by 1 if v_i and v_j co-exist in the tweet d. Note that the nodes of "word" have additional edges within themselves to capture the co-occurrence between words in tweets, which is different from the other 3 types of entities.

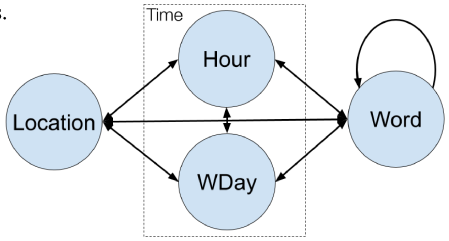

Figure 2: Illustration of basic co-occurrence graph.

3.3 Cross-Modal Search

The objective of cross-modal search is to answer this question: given an entity from one modal, which entities in other modals are most likely to be associated with it? Formally, given a source entity v_i^m from modal m and a target entity v_j^n from modal n, what is possibility of observing v_j^n from v_i^m? In the following, we try to approach this possibility from two perspectives: co-occurrence graph and vectorized embeddings, and then utilize their relationship to guide the embedding learning process.

3.3.1 From the Perspective of Co-occurrence Graph. On the co-occurrence graph G, we resort to modifying personalized graph random walk procedures [27–30] to approach the above probability, which is actually equivalent to the possibility of the random surfer residing at vertex v_j^n if he initially starts at v_i^m. For convenience, let us denote this probability by $p(v_i^m \rightarrow v_j^n)$.

Suppose that we now have the converged residing probabilities $\mathbb{R} = \begin{bmatrix} r_{v_1} & r_{v_1} & \cdots & r_{v_N} \end{bmatrix}$, instead of directly using the residing probability of vertex v_j^n (i.e., $r_{v_j^n}$), we calculate its normalized

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

variant as the $p(v_i^m \rightarrow v_j^n)$:

$$p(v_i^m \rightarrow v_j^n) = \frac{r_{v_j^n}}{\sum_{v_k \in V^n} r_{v_k}} \tag{3}$$

where V^n denotes the set of vertices from the modal n.

3.3.2 From the Perspective of Vectorized Embeddings. Remember that our goal is to approach the possibility of generating the entity v_j^n from the entity v_i^m which is from a different modal. Suppose that we have the vectorized embeddings of the entities, it is relatively easy to model the objective probability using the embeddings compared to co-occurrence graph. For example, we may use $p(v_j^n|v_i^m)$ as the objective probability, which is defined as:

$$p(v_j^n|v_i^m) = \frac{e^{\mathbf{v}_i^m \cdot \mathbf{v}_j^n}}{\sum_{v_k \in V^n} e^{\mathbf{v}_i^m \cdot \mathbf{v}_k}} \tag{4}$$

where $\mathbf{v}$ is the vectorized embedding of entity v and V^n similarly denotes the set of entities from the modal n.

3.3.3 Learning Embeddings. Given the probability of $p(v_i^m \rightarrow v_j^n)$ from the co-occurrence graph and the probability of $p(v_j^n|v_i^m)$ from the initial embeddings, the goal of learning embeddings is to iteratively update values in embeddings so that $p(v_j^n|v_i^m)$ becomes closer and closer to $p(v_i^m \rightarrow v_j^n)$. In doing so, eventually the computed vectorized embeddings will be able to preserve the structure information of the co-occurrence graph. We use the Kullback-Leibler divergence $KL(\cdot)$ to measure the difference between two probability distributions. Subsequently, we define the loss function between any two modals of entities as:

$$\begin{aligned}\mathcal{L}(m,n) = &\sum_{v_i^m \in V^m} KL(p(v_i^m \rightarrow \cdot)\|p(\cdot|v_i^m)) \\ &+ \sum_{v_j^n \in V^n} KL(p(v_j^n \rightarrow \cdot)\|p(\cdot|v_j^n))\end{aligned} \tag{5}$$

Such a loss function basically means to minimize both of the distribution differences to generate one modal of entities from entities in another modal and conversely to generate another modal of entities from entities in this modal. At last, the total loss function is the sum of different $\mathcal{L}(m,n)$ with respect to all different edges types in Figure 2. Note that the computation of such loss functions can be solved efficiently using stochastic gradient descent and negative sampling [18, 23].

4 EVALUATION

4.1 Experimental Settings

4.1.1 Datasets. The evaluation is performed on two sets of geo-tagged tweets collected from 2014-08-01 to 2014-11-30 in two corresponding cities: New York City (NYC) and Los Angeles, CA (LA) [24]. The total number of tweets, is about 1.5 million in NYC and 1.2 million in LA, respectively. We randomly take 10,000 tweets for testing and the rest for learning the embeddings of location, words, hours-of-day and days-of-week.

4.1.2 Baseline Approaches. We compare with the following baseline approaches: TF-IDF, SVD, Doc2Vec [31], ReAct [24], and CrossMap [23].

By default, TF-IDF, SVD and Doc2Vec handle data of only two dimensions. We perform the following preprocessing in these methods in order to incorporate all the entities of location, words, hours-of-day and days-of-week. We treat each location as a document and its sentences comprise the tweets falling inside the location. The hour-of-day and day-of-week values extracted from posting time of each tweet are parsed as special words and appended to that tweet's bag of words.

4.1.3 Parameter Settings. The major parameters in LeGo-CM are set as follows. For embedding dimension length, we set N_{dim} = 200. For time in tweets, we extract its natural integral hours-of-day and days-of-week, i.e., $hour = \{0, 1, 2, \cdots, 21, 22, 23\}$ and $wday = \{Mon, Tue, Wed, Thu, Fri, Sat, Sun\}$, in order to reflect patterns of people's daily life in urban areas. We set the bandwidth b of mean shift[2] for clustering tweet locations to $160m$, which yields around 18,000 location clusters in NYC and 17,000 location clusters in LA. As for the random walk procedure to calculate stationary residing probabilities between vertices in the co-occurrence graph, we use a default damping factor h = 0.8 and run 20 iterations in all cases. In the embedding learning process, we set the number of epochs for training N_{epoch} = 256 and the learning rate α_{learn} = 0.02.

For comparison, all methods are tested using the same N_{dim} except for TF-IDF. Also note that TF-IDF, SVD and Doc2Vec use the same representations of location and time as LeGo-CM. Although ReAct is also fed with the same form of locations, it uses natural integral hours and time hotspots for time representations as in their implementation, respectively.

4.2 Quantitative Analysis

4.2.1 Effectiveness. We evaluate the effectiveness of different embedding methods by performing the tasks of ranking tweets with negative attributes. To quantify the ranking orders of testing tweets, we adopt the metric of Mean Reciprocal Rank (MRR) [23, 24], which is defined as:

$$MRR = \frac{\sum_{d \in \mathcal{D}_{Test}} \frac{1}{\mathcal{R}_d}}{|\mathcal{D}_{Test}|} \tag{6}$$

where $\mathcal{D}_{Test}$ represents the testing dataset of tweets. It is easy to see that higher-quality embeddings will yield larger MRR values. In our settings, we set $|\mathcal{D}_{Test}|$= 10,000 and then compute such an MRR for each of the attributes in $\langle loc_d, hour_d, wday_d, word_d \rangle$.

Table 1: Comparison results using Mean Reciprocal Rank.

Method	NYC				LA			
	Loc	Word	Hour	WDay	Loc	Word	Hour	WDay
TF-IDF	0.275	0.274	0.279	0.280	0.277	0.279	0.283	0.286
SVD	0.402	0.321	0.321	0.321	0.350	0.317	0.341	0.342
Doc2Vec	0.448	0.491	0.342	0.345	0.469	0.523	0.338	0.336
ReAct	0.470	0.459	0.167	N/A	0.560	0.561	0.167	N/A
CrossMap	0.516	0.619	N/A	N/A	0.514	0.642	N/A	N/A
LeGo-CM	**0.589**	0.598	**0.348**	**0.348**	**0.616**	0.612	0.339	0.339

The results of LeGo-CM for cross-modal search are listed in Table 1, and the MRR value in our method is bold if it is the highest value in the comparison results. It shows that LeGo-CM outperforms almost all baseline approaches including TF-IDF, SVD

[2] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.MeanShift.html

and Doc2Vec and achieves better results than the state-of-the-art methods in most cases. In particular, a significant improvement is observed over ReAct with respect to the MRR values of locations. Note that the MRR values with respect to day-of-week are not reported for ReAct because this method does not use this feature. Similarly, CrossMap uses the hotspots in the temporal dimension to represent time and thus does not report the MRR values for the integral features of hour-of-day and day-of-week. In general, TF-IDF has the worst performance in most cases due to its direct use of sparse row/column vectors extracted from the document-term matrix. SVD improves over TF-IDF by performing dimentionality reduction and thereby only preserves the most essential information in the compacted row/column vectors in the document-term matrix. In comparison, Doc2Vec gets much better results on location and word by encoding them in the same latent space. ReAct is not as good as we expected. This is probably resulted from its online learning process which only addresses the most recent information happening at a location and chooses to forget the past information in an exponential time-decay manner. This also explains its low MRR values of hour-of-day. Although CrossMap achieves slightly better MRR values than our method LeGo-CM with respect to word, it has significant lower MRR values with respect to location.

4.2.2 Efficiency. To fairly investigate the efficiency of learning process, we omit all the data preparation operations and only address the step of model training. The experiments are conducted on an AWS EC2 instance with 240GB memory and an Intel Xeon CPU (E5-2686 2.30GHz). In each method, we record the time spent in processing the training tweets. The results are reported on the NYC dataset as it contains relatively more tweets.

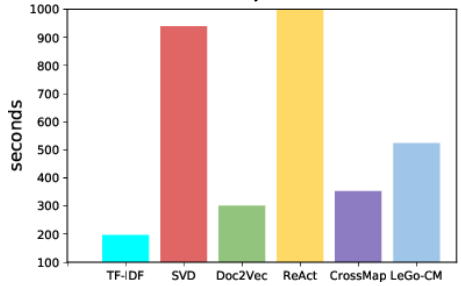

Figure 3: Model training time consumption.

Figure 3 presents the training time of different methods in seconds. It shows that TF-IDF runs the fastest because of its simplicity. Our method LeGo-CM achieves moderate efficiency comparing to CrossMap considering that we address 4 types of nodes in the graph while ReAct addresses 3 types of nodes. The method ReAct runs the slowest because of its small batch size which leads to frequent weight updating in its online training procedure.

5 CONCLUSIONS

In this paper, we presented LeGo-CM for learning embeddings of spatial, textual and temporal entities in geotagged tweets. Prior to the learning process, a mean shift-based spatial clustering procedure is performed to detect locations of interest. For the time dimension, we extract hour-of-day and day-of-week as temporal entities which are consistent with people's daily-life habits and patterns. We then utilize the co-occurrence between locations, words, hours-of-day and days-of-week to build graphs for LeGo-CM. LeGo-CM learns the embeddings of graph nodes by approximating the stable residing probabilities between nodes. The evaluation results on two selected cities show that LeGo-CM outperforms competitive baselines in most cases, thereby showing the effectiveness of the proposed method. For future work, we plan to extending the co-occurrence graph by adding edges between locations to reflect their spatial proximity and topical closeness and thus conduct location similarity searches.

6 ACKNOWLEDGEMENT

This work was supported in part by the National Science Foundation under grant IIS-1816889.

REFERENCES

[1] N. Gramsky and H. Samet. Seeder Finder: Identifying Additional Needles in the Twitter Haystack. LBSN '13.
[2] J. Sankaranarayanan, H. Samet, B. E. Teitler, et al. TwitterStand: News in Tweets. SIGSPATIAL '09.
[3] H. Samet, J. Sankaranarayanan, M. D. Lieberman, et al. Reading News with Maps by Exploiting Spatial Synonyms. *Commun. ACM*, 2014.
[4] H. Samet, M. D. Adelfio, B. C. Fruin, et al. Porting a Web-based Mapping Application to a Smartphone App. SIGSPATIAL '11.
[5] H. Samet, B. E. Teitler, M. D. Adelfio, et al. Adapting a Map Query Interface for a Gesturing Touch Screen Interface. WWW '11.
[6] M. D. Lieberman, H. Samet, and J. Sankaranayananan. Geotagging: Using Proximity, Sibling, and Prominence Clues to Understand Comma Groups. GIR '10.
[7] M. D. Lieberman and H. Samet. Multifaceted Toponym Recognition for Streaming News. SIGIR '11.
[8] M. D. Lieberman and H. Samet. Adaptive Context Features for Toponym Resolution in Streaming News. SIGIR '12.
[9] H. Wei, J. Sankaranarayanan, and H. Samet. Finding and Tracking Local Twitter Users for News Detection. SIGSPATIAL '17.
[10] H. Wei, H. Zhou, J. Sankaranarayanan, et al. Residual Convolutional LSTM for Tweet Count Prediction. WWW '18 Companion.
[11] H. Wei, H. Zhou, J. Sankaranarayanan, et al. Detecting Latest Local Events from Geotagged Tweet Streams. SIGSPATIAL '18.
[12] H. Wei, J. Sankaranarayanan, and H. Samet. Enhancing Local Live Tweet Stream to Detect News. ACM SIGSPATIAL LENS '18.
[13] G. Quercini, H. Samet, J. Sankaranarayanan, et al. Determining the Spatial Reader Scopes of News Sources Using Local Lexicons. GIS '10.
[14] S. Jenson, M. Reeves, M. Tomasini, et al. Mining Location Information from Users' Spatio-temporal Data. SmartWorld '17.
[15] C. Zhang, G. Zhou, Q. Yuan, et al. GeoBurst: Real-Time Local Event Detection in Geo-Tagged Tweet Streams. SIGIR '16.
[16] L. Hong, A. Ahmed, S. Gurumurthy, et al. Discovering Geographical Topics in the Twitter Stream. WWW '12.
[17] W. Wei, K. Joseph, W. Lo, et al. A Bayesian Graphical Model to Discover Latent Events from Twitter. ICWSM '15.
[18] T. Mikolov, I. Sutskever, K. Chen, et al. Distributed Representations of Words and Phrases and their Compositionality. NIPS '13.
[19] J. Pang and Y. Zhang. DeepCity: A Feature Learning Framework for Mining Location Check-ins. ICWSM '16.
[20] K. Mets. Learning Meaningful Location Embeddings from Unlabeled Visits, https://www.sentiance.com/2018/01/29/unlabeled-visits/#Location_Profiling, last accessed on December 27 2018.
[21] M. Kejriwal and P. Szekely. Neural Embeddings for Populated Geonames Locations. ISWC '17.
[22] B. Perozzi, R. Al-Rfou, and S. Skiena. DeepWalk: Online Learning of Social Representations. KDD '14.
[23] C. Zhang, K. Zhang, Q. Yuan, et al. Regions, Periods, Activities: Uncovering Urban Dynamics via Cross-Modal Representation Learning. WWW '17.
[24] C. Zhang, K. Zhang, Q. Yuan, et al. React: Online Multimodal Embedding for Recency-Aware Spatiotemporal Activity Modeling. SIGIR '17.
[25] A. Ritter, S. Clark, Mausam, et al. Named Entity Recognition in Tweets: An Experimental Study. EMNLP '11.
[26] A. Ritter, Mausam, O. Etzioni, et al. Open Domain Event Extraction from Twitter. KDD '12.
[27] L. Page, S. Brin, R. Motwani, et al. The PageRank Citation Ranking: Bringing Order to the Web. WWW '98.
[28] W. Xing and A. Ghorbani. Weighted PageRank Algorithm. CNSR '04.
[29] P. Lofgren, S. Banerjee, and A. Goel. Personalized PageRank Estimation and Search: A Bidirectional Approach. WSDM '16.
[30] H. Wei, J. Sankaranarayanan, and H. Samet. Measuring Spatial Influence of Twitter Users by Interactions. ACM SIGSPATIAL LENS'17.
[31] Q. Le and T. Mikolov. Distributed Representations of Sentences and Documents. ICML'14.