An Integrated Model for Textual Social Media Data with Spatio-Temporal Dimensions

Juglar Diaz, Barbara Poblete and Felipe Bravo-Marquez

Department of Computer Science, University of Chile & IMFD, Santiago, Chile

Abstract

GPS-enabled devices and social media popularity have created an unprecedented opportunity for researchers to collect, explore, and analyze text data with fine-grained spatial and temporal metadata. In this sense, *text*, *time* and *space* are different domains with their own representation scales and methods. This poses a challenge on how to detect relevant patterns that may only arise from the combination of text with spatio-temporal elements. In particular, spatio-temporal textual data representation has relied on feature embedding techniques. This can limit a model's expressiveness for representing certain patterns extracted from the sequence structure of textual data. To deal with the aforementioned problems, we propose an *Acceptor* recurrent neural network model that jointly models spatio-temporal textual data. Our goal is to focus on representing the mutual influence and relationships that can exist between written language and the time-and-place where it was produced. We represent space, time, and text as tuples, and use pairs of elements to predict a third one. This results in three predictive tasks that are trained simultaneously. We conduct experiments on two social media datasets and on a crime dataset; we use Mean Reciprocal Rank as evaluation metric. Our experiments show that our model outperforms state-of-the-art methods ranging from a 5.5% to a 24.7% improvement for location and time prediction.

Keywords: Social media, Spatio-temporal data, Recurrent neural networks

1. Introduction

Online social media has gained wide adoption worldwide, and is now considered as having an influential role in public opinion. Within this context,

social platforms such as Twitter[1], Instagram[2] and Facebook[3] have allowed users to start sharing the textual and multimedia content that they generate (e.g., opinions, interests, reviews and every day activities) with enriched spatio-temporal information. This data can be represented as a record in the form of a $\langle where, when, what \rangle$ tuple, in which the *where* means a location's latitude-longitude coordinates, the *when* is its timestamp and the *what* is its content.

Pattern analysis of spatio-temporal data extracted from social media can help us understand complex human behavior like mobility [1, 2, 3], also *when* and *where* popular social activities are taking place [4, 5, 6, 7]. In addition, timestamps and coordinates that are associated to textual data can be used as filters to detect real-world emerging events, such as earthquakes [8, 9] and civil unrest [10]. Moreover, these types of multi-modal data sources have been successfully used for natural language based financial forecasting [11, 12, 13, 14]. Besides social media, there are other data sources that relate semantic content with spatio-temporal information. An example are crime reports that include a natural language description of the crime, as well as the time-and-place it occurred. The textual crime descriptions can come either in the form of free text provided by the victim, or based on keywords and more standardized phrases used by the police. Overall, the increased access to this type of data can allow us to study and model textual information in relation to its spatio-temporal context.

In this work, we focus on *spatio-temporal textual data*. In this sense, the key component in any data mining problem is data representation, hence, the multi-modality of *space, time, and text* provides an additional challenge. In particular, text, its timestamp, and geographical coordinates, are commonly represented in different scales and magnitudes. For instance, text is discrete and has been represented, for example, using vector spaces, as opposed to timestamp and coordinates, which are continuous variables. Hence, it is not trivial to combine these components within a unified model.

State-of-the-art models [15, 7, 16] use feature embeddings to represent the elements of the tuple $\langle time, location, text \rangle$. This representation considers text as a bag-of-words where each unique word is represented by dense

---

[1]https://twitter.com/
[2]https://www.instagram.com/
[3]https://www.facebook.com/

vectors or embeddings. Then, at inference time, a text is represented as the average of its word embeddings. An important limitation of this type of approach is that language structure (i.e., the order of words within a sentence) is ignored. Hence, potentially relevant language patterns derived from the sequential nature of text data are discarded. In this work, we propose an *Acceptor* recurrent neural network (RNN) architecture [17], which we refer to as STT-RNN. STT-RNN is designed to providee an integrated view of spatio-temporal textual data. The *Acceptor* is an RNN usage pattern in which an RNN encodes a sequence into a single vector that corresponds to the output vector of the last token in the sequence. This vector is usually fed into a fully connected layer to produce a prediction [17]. Specifically, STT-RNN is designed to retrieve one element of the tuple $\langle time, location, text \rangle$ by only knowing the other two. Hence, the use of RNNs allows us to represent text in a more expressive manner without loss of sequential language structures. The goal of our proposed model is to provide a representation that allows us to extract patterns related to spatio-temporal human activities. Specifically, we propose a model that can be trained on spatio-temporal text records, which can be used to gain insight into the following 3 information seeking or retrieval tasks:

1. What is the most likely time period associated with a given text passage and a spatial location?
2. What is the most likely location associated with a given text passage and time period?
3. What is the most likely text associated with a given location and time?

To illustrate the usefulness of the proposed model we present examples of applications for each type of retrieval task. For instance, a possible application for task (1) is helping local police optimize the allocation of their agents to areas that are more prone to certain crimes at certain times of the day. The specific task, in this case, could be *to find the times at which 'car thefts' are more likely to take place in 'shopping mall A'* (i.e., find *time* given *loc* and *text*). For task (2), on the other hand, a possible application is to find places where certain activities take place at a certain time interval. A concrete example, regarding criminal activity, would be *to find areas in a city in which 'drug related crimes' occur at night* (i.e., find *loc* given *text* and *time*). In addition, task (3) can help characterize which activities take place in a certain urban area at a certain time (i.e., activity modeling). For example, *given a particular park and time frame, find the top-recreational activities practiced*

there (i.e., find *text* given *loc* and *time*). We envision many other possibilities for representing effectively time, location and text in a unified manner. Urban planning might improve from understanding which activities people like to do at different places and times in a city. Also, commercial search engines will certainly benefit from context-aware results for users. Since humans follow spatio-temporal routines in their everyday life, a recommender system could provide recommendations to users by only knowing their location at the present moment.

In summary, the main contributions of this paper are:

- We propose an *Acceptor* recurrent neural network architecture that jointly models spatial variables, temporal variables and text.

- We present an empirical evaluation of our model and comparison to similar state-of-the-art approaches.

- We study how the three elements of $\langle time, location, text \rangle$ correlate to each other in social media domains like Twitter and Foursquare[4].

The rest of this paper is organized as follows: Section 2 describes related work in spatio-temporal textual data modeling. Section 3 presents the description of the proposed model. Section 4 shows the validation experiments and finally in Section 5 we present the conclusions.

2. Related work

In this section we provide an overview of the literature relevant to our proposal. First, we describe models that detect geographical topics. Then, we describe works that focus on modeling spatio-temporal activities. After that, we describe multimodal embedding methods for spatio-temporal text data and finally we briefly overview recurrent neural networks.

Geographical topic modeling focuses on detecting topics that characterize geographical areas [18, 19, 5, 20, 21]. Mei et al. in [18] propose a generalization of Probabilistic Latent Semantic Indexing [22] where topics can be generated either by the combination of *timestamp* and *location* or *text*. Yin et al. propose LGTA in [5]. LGTA is a generative model where

_____

[4]https://foursquare.com

4

there are latent regions that are geographically distributed by a Gaussian. Each region has a multinomial distribution over topics and each topic has a multinomial distribution over keywords. Kling et al. propose MGTM in [20], a model based on multi-Dirichlet processes. They use a three-level hierarchical Dirichlet process with a Fischer distribution for detecting geographical clusters. Also, a Dirichlet-multinomial document-topic distribution and a Dirichlet-multinomial topic-word distribution. Wang et al. propose LATM in [19]. LATM is an extension of Latent Dirichlet Allocation (LDA) [23, 24], capable of learning the relationships between locations and words. In the model each word has an associated location. For generating words, the model produces the word and also the location, in both cases with a multinomial distribution conditioned on a topic that is generated by a Dirichlet distribution. In [21], Hong et al. introduce the user as a variable in the model. In each tuple *user,location, text*, texts are represented under a bag-of-words assumption, also geographical locations are clustered into latent regions. Hong et al. use three language models: a background language model, a region-dependent language model and a topic language model. The latent regions are generated by a multinomial distribution depending on the user and a global region distribution. The locations are generated by regions using multivariate Gaussian distributions. After that the topic is generated considering all together the global topic distribution, the user and the region. Finally, the words are generated by the topic.

Our work differs from this approaches in that we do not make assumptions distribution over the data. Spatio-temporal text data generation is influenced by many factors specific to different places and moments, these patterns are difficult to model with predefined distributions.

Spatio-temporal activity modeling [4, 25] is about finding which activities are reported in the different areas of a city. In this section we highlight works that label places using semantic data, such as social media text. Wu et al. in [4] annotate user visits to *points of interest* in the physical world. They test four methods for labeling places with text: term frequency, term frequency-inverse document frequency, a Gaussian mixture model and a kernel density estimation model. The kernel density estimation method got the best results in their experiments. Ye et al. also annotate places with labels in [25]. They model the problem as a multitask classification problem and use a Support Vector Machine classifier. Places are represented with two types of features. Some features are extracted from similar places, whereas the other

ones are extracted from spatial and temporal information. Combining the two types of features proved to be the best approach.

The works in this section annotate places with keywords. In our approach, we jointly model time, places and texts. We can annotate places with keywords but also we can query the model with any combinations of the three variables with text as part of the query.

Embedding methods are used to find a dense, low-dimensional continuous vector representation for discrete variables. These methods have been successfully applied for representing words [26, 27, 28] and nodes in graphs [29]. In the case of spatio-temporal textual data, embedding methods allow to represent the three elements of the tuple $\langle time, location, text \rangle$ in the same space using co-occurrence patterns. It is important to remark that spatial and temporal variables must be discretized in order to employ embedding methods on them.

In [15], Zhang et al. propose CrossMap, a multimodal embedding method. CrossMap applies a discretization method to timestamps and coordinates based on Kernel Density Estimation. High density regions found for timetamps and locations are used as discretization categories. Afterwards, authors use two strategies to compute the embeddings, named *Recon* and *Graph*. *Recon* assumes that each tuple $\langle time, location, text \rangle$ is a relation and then learns embeddings for *timestamps*, *locations* and *words* such that the relation can be reconstructed. *Graph* builds a graph of co-relations and then learns embeddings for *timestamps*, *locations* and *words* such that the structure of the graph is preserved. Later in [7] the authors extend the model to obtain embeddings from streaming data. Unlike Crossmap, they use the hour-of-the-day to discretize timestamps and $300m \times 300m$ grids to discretize geographical coordinates. The main contribution of this approach is two new strategies to compute the vectors from the streaming data source, one based on life-decay learning and the other on constrained learning. In [16], Zhang et al. propose an embedding method capable of learning from multiple sources. Each source is a dataset that defines a graph of co-relations. The embeddings are trained to preserve the structure of the graphs. There is a main graph that is defined by the tuples: $\langle time, location, text \rangle$, the embeddings associated to the main graph are shared with secondary graphs inducted by secondary datasets. In addition, each secondary graph defines an embedding space that is concatenated to the embeddings of the main graph. During training, the model alternates between learning the embeddings for the main graph and

the embeddings for the additional data sources.

These works model the text as the average of word embeddings. In our approach, we jointly model times, places and texts with recurrent neural networks where we represent contexts beyond a shallow average of embeddings.

Overall, our work differs from prior work in that we use an *Acceptor* recurrent neural network architecture that allows representing the sequential structure of text together with spatial and temporal information. Also, we study different levels of granularity for temporal and spatial representation and analyze correlations between the three elements of $\langle time, location, text \rangle$. The proposed model can be queried with any combination of time, place and text and retrieve any of the three variables.

Recurrent neural networks [30] are a class of neural networks for modeling sequential data. They have been successfully applied to natural language processing problems like speech recognition [31] and machine translation [32, 33, 34], as well in other text mining problems such as analyzing sentiment time series from social media for financial asset allocation [35]. In the case of spatio-temporal data, they have been mostly used for mobility modeling [36, 37, 38, 39]. LSTM [40] and GRU [41] are popular variants of RNN architectures. In our case, we use GRU because they have fewer parameters and have shown similar performance to LSTM according to [41]. In the basic architecture for an RNN there is a vector h that represents the sequence. At each timestep t the model takes as input h_{t-1} and the *t-th* element of the sequence x_t; then computes h_t. GRU introduces gates to this process to select what to pass to the next hidden state h_t and what to forget. Equations 1,2,3 and 4 show the computing steps of a GRU network.

$$r_t = \sigma(W_{ir}x_t + b_{ir} + W_{hr}h_{(t-1)} + b_{hr}) \tag{1}$$

$$z_t = \sigma(W_{iz}x_t + b_{iz} + W_{hz}h_{(t-1)} + b_{hz}) \tag{2}$$

$$n_t = \tanh(W_{in}x_t + b_{in} + r_t(W_{hn}h_{(t-1)} + b_{hn})) \tag{3}$$

$$h_t = (1 - z_t)n_t + z_t h_{(t-1)} \tag{4}$$

RNNs can successfully model the order of words within a sentence by treating a sentence as a sequence of word embeddings. Each hidden state h_t can be viewed as a memory unit that encodes the sentence until word x_t.

Hence, the hidden state of the last word encodes the whole information of the sentence while paying attention to the structured properties of it [17]. In our model we incorporate location and time as special tokens into the sequence.

3. Proposed Approach

3.1. Problem Formulation

Given a collection of records that provide textual descriptions of a geographical area at different moments in time; our goal is to create a model capable of representing this multi-modal data. We require the resulting model to be able to predict missing values of either text, time, or space.

More formally, let be $H = \{r_1, ..., r_n\}$ a set of spatio-temporal annotated text records (e.g., a tweet, a crime incident description). Each r_i is a tuple $\langle t_i, l_i, e_i \rangle$, where: t_i is the timestamp associated with r_i, l_i is a two-dimensional vector representing the location corresponding to r_i, and e_i denotes the text in r_i. Given an incomplete record where either t_i, r_i or e_i is missing, the resulting model should be able to reconstruct the missing item. This results in three predictive tasks: 1) predicting time for which a certain text was produced in a particular location, 2) predicting location from which a text was generated at a certain time, and 3) predicting text that is created from a certain location and time.

3.2. The STT-RNN Model

As mentioned, state-of-the-art models [15, 16, 7] use feature embeddings to represent the elements of the tuple $\langle t_i, l_i, e_i \rangle$. For the case of text, embedding representations discard the order of words, which can limit the expressive power of the model. We propose STT-RNN, an *Acceptor* RNN model that goes beyond feature embeddings.

Figure 1 shows the STT-RNN architecture. As a first preprocessing step: time and space are discretized and text is tokenized. The elements of the tuple $\langle time, location, text \rangle$ are passed through a multi-modal embedding layer. The multi-modal embedding layer projects words, time and location into a unified representation space. After discretization, each item from $\langle time, location, text \rangle$ is represented by an integer that is used as an index in the multi-modal embedding layer. The multi-modal embedding layer is a look-up table shared by the three elements, each one of these, $\langle time \rangle$, $\langle location \rangle$ and $\langle text \rangle$, occupies a segment of the table. Afterwards, a single

8

element from the tuple is selected as the *target*, whereas the two remaining ones are selected as the *source-context*. Only the elements selected as *source-context* are passed through the multi-modal embedding layer.

The elements that are selected as *source-context* are concatenated as a sequence input to the GRU-RNN. In this way, the RNN processes the input as a sequence of tokens formed by words, times or locations. The output of the GRU-RNN is passed as input to the predictor component. The predictor component is formed by three different fully connected layers to predict either *time*, *place* or *words*; depending on the task. We call them *PredictorTime*, *PredictorLoc*, and *PredictorWord*. We select which fully connected layer to use depending on the target. The fully connected layers are passed through a softmax function with a cross-entropy loss over the output space of the corresponding task: words for *PredictorWord*, locations for *PredictorLoc*, and time for *PredictorTime*. The probability of a *text* passage is computed as the average probability of the predictions made by *PredictorWord* for each of its *word*.

We trained the model with pairs $\langle time, loc \rangle \rightarrow text$, $\langle time, text \rangle \rightarrow loc$ and $\langle loc, text \rangle \rightarrow time$. The contexts are represented with the same GRU-RNN sharing parameters for the three cases. This allows each task to benefit from each other and allows us to query the model with any combination of $\langle time, location, text \rangle$. We can ask to return any of the three variables as output. Any combination of $\langle time, location, text \rangle$ can be processed as a input sequence by the GRU-RNN, for example: we could query the model with any of the sequences that the model is trained to represent $\langle time, loc \rangle$, $\langle time, text \rangle$ and $\langle loc, text \rangle$, but also with sequences like just text $\langle text \rangle$, or just a token of time $\langle time \rangle$ or a token of location $\langle loc \rangle$ and ask the model to predict any of the three variables.

As mentioned above, we designed STT-RNN to perform three predictive tasks: 1) predict time for which a certain text was produced in a particular location, 2) predict location from which a text was generated at a certain time, and 3) predict text that is created from a certain location and time. Although, in tasks 1 and 2, $\langle text \rangle$ is part of the context and it makes sense to use a recurrent neural network to capture the sequential structure of text; in task 3, $\langle text \rangle$ is not part of the context, and hence, the context tuple $\langle time, space \rangle$ lacks a sequential structure. Nevertheless, we stay with our design for the following reasons. First, to maintain comparability with previous works that evaluate models in the three tasks mentioned. Second, to allow our model to be queried with any combination of $\langle text \rangle$, $\langle time \rangle$ and $\langle space \rangle$

and retrieve any of those items. Finally, for the sake of completeness, we think it is important to show all scenarios on which our proposed model can be evaluated to establish a complete background for future work.

It is worth mentioning that we conducted preliminary experiments considering variations of the proposed architecture for both, the *GRU-RNN* and the *Predictor* components. All these variants either performed worst or did not exhibit any improvement over our model while adding complexity in some cases. For the GRU-RNN we tested using LSTMs [40] instead of GRUs and also experimented with attention mechanisms [42]. Both cases added complexity to the model (in terms of the number of parameters to estimate) without obtaining any significant improvements. In the case of the LSTM, our results were consistent with previous results [41]. Regarding the attention mechanism, since our sentences were truncated to 15 and 10 tokens for the social media and crime incidents description datasets correspondingly, we believe that our sentences were not long enough to observe the benefits of adding an attention mechanism to the network.

Regarding the predictor component, we tested text generation with a decoder-GRU recurrent neural network using the GRU-RNN as encoder. We got poor results with this approach when comparing to generating each word independently.

Our preliminary experiments show the main benefit of our architecture: it is simple and competitive to other more complex variations.

3.2.1. Model Parameters

We use 64-dimensional embedding representation for *timestamp*, *location* and *words*. The GRU-RNN representation uses a single-layer GRU with a hidden layer size of 128.

3.3. Training Algorithm

Algorithm 1 shows the training process for STT-RNN. First we build a text indexer and discretize timestamps and coordinates. The text indexer builds a vocabulary, keeping only those terms that are alpha-words (words made up of alphabet letters only), appear more than 100 times and are not stop-words (words like articles and prepositions without semantic meaning). Timestamps and coordinates are discretized using a clustering-based approach described in Section 3.4.

The model is trained for a number of *epochs* using mini-batch gradient descent with Adam optimizer [43]. At each step we store the model's weights
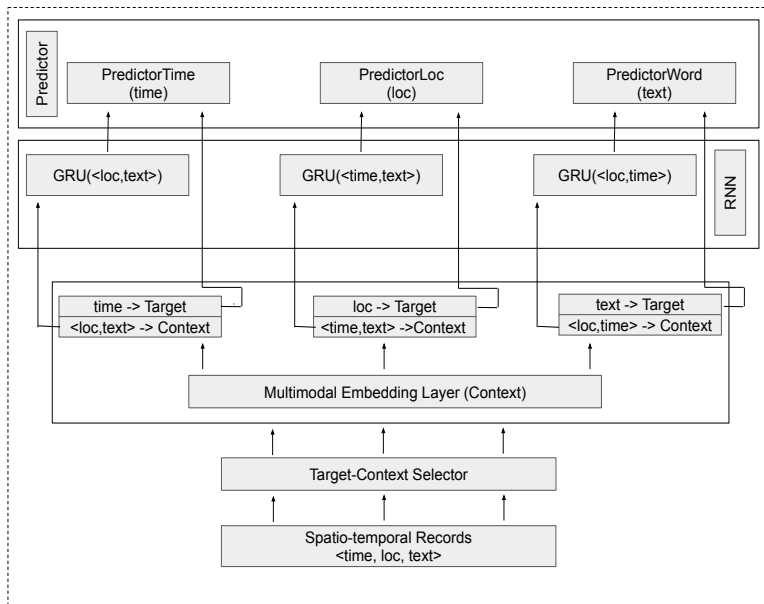
Figure 1: Architecture of STT-RNN.

as well as the results of evaluating the model on a held-out validation set. The returned model is the one that performed best on the validation set across the various training steps.

3.4. Timestamp and coordinate discretization

As mentioned before, texts, timestamps and coordinates are variables from different domains with different scales and representation methods. Text is a sequence of discrete tokens (i.e. words, characters), while timestamps and coordinates are continuous variables. To jointly model the three variables the approach that we followed is to discretize timestamps and coordinates.This approach allows us to deal with sparseness issues and to represent the tuple $\langle time, location, text \rangle$ under the unified paradigm of a sequence of discrete tokens.

In our experiments we used two different approaches for temporal and spatial discretization: 1) a density-based approach and 2) equal-width binning. The density-based approach was proposed by Zhang et al. in [15] and we use it to make our results comparable to previous work. Timestamps

---

Algorithm 1: Training algorithm for STT-RNN

Input: Set H of spatio-temporal records tuples of the form $\langle t_i, l_i, e_i \rangle$.

Output: Trained model T.

Build TimeDiscretizer(H).

Build LocDiscretizer(H).

Build TextIndexer(H).

$//Discretize$

for $h_i \in H$ **do**
    TimeDiscretizer(t_i)
    LocDiscretizer(l_i)
    TextIndexer(e_i)

$//Training$

Initialize Parameters θ

for $epoch \in \{1,2,...,EPOCHS\}$ **do**
    for $batch \in \{1,2,...,batch\_size\}$ **do**
        for $target \in \{t, l, e\}$ **do**
            $context$ is $\{t, l, e\}$ - $target$
        Update θ using the three batches of $\langle context, target \rangle$ from
        Train using the optimization algorithm and the objective
          function
    Save θ at this step.
    Save the results of evaluating model T with parameters θ over the
    Validation set

Output trained model T with weights θ at step with best results over the Validation set

---

are converted to numbers in the range [0-86,400][5] by calculating their offset in seconds with respect to 12:00am. Then, a density-based automatic discretization technique is applied to both the transformed temporal variables and coordinates. This leads to high density temporal windows and spatial cells (for more details refer to [15]).

The second discretization approach is to apply equal-width binning to both temporal and spatial variables. The main benefits of this approach are: 1) discretization bins are easier to interpret, and 2) it allows us to study the

---

[5]86,400 is the number of seconds in a day.

impact of the discretization granularity in the model's performance by modifying the size of the bins. For equal-width binning timestamp discretization we consider the 168 hours of a week (24×7) as the representation domain. That means that two events occurring on the same hour and day of the week would be mapped to the same time number. Then we use bins of k continuous hours to discretize the 168 hour window. The greater the value of k the lower the number of bins.

For equal-width spatial discretization we use equal size cells obtained after performing the following arithmetic operation on the latitude and longitude floating number coordinates: $l - (l \bmod c)$, where l can be latitude or longitude and c refers to the cell size.

For example: coordinates (-72.45772, 33.358423) would be assigned to cell (-72.457, 33.358) using 0.001 as the cell size, or to cell (-72.456, 33.358) using 0.002 as the cell size. Table 1 shows an example of the discretization of a tweet.

Location	34.0430 ,-118.2673	34.04 ,-118.26
Time	Feb 1,2019, 1:31:00AM	(Friday)$5 \times 24 + 1 = 121 \in 120$
Message	LeBron is back LakeShow	lebron back lakeshow

Table 1: Example of discretization of a geo-tagged tweet using one-hour time window size and 0.02 spatial cell size.

3.4.1. Training Parameters

As loss function, we used the cross-entropy loss. Also, we used back propagation through time [44], mini-batch gradient descent and Adam [45] with an initial learning rate of 0.001. We used a batch size of 256 and trained for 30 epochs with early stopping.

4. Experiments

In this section we describe our experimental framework. The goal of this evaluation is to measure our model's ability to predict a missing element from a tuple of the form $\langle time, location, text \rangle$, in which either *time*, *location*, or *text* is missing. For the purpose of comparing to prior work, we followed the evaluation methodology used by Zhang et al. in [15].

Next, we describe the datasets used in our evaluation and baseline models. Then, we present the evaluation methodology and a study of the sensitivity

to the discretization techniques. We conclude our analysis with a case study of a real-world crime description dataset.

4.1. Dataset description

Our experiments aim to answer two research questions: 1) is our model competitive to previous approaches? and 2) can our model be used in more than one domain?

In that sense, we evaluated our model using two types of data sources, *social media user posts*, and *official crime incident reports*. Social media datasets coming from Twitter and Foursquare are used for quantitative comparison between our approach and state-of-the-art baselines using the same settings used in the original evaluations. Crime reports, on the other hand, are included to add more diversity to our analysis and to show a real-world application of our model in the form of a case study. Each dataset is described below.

Twitter dataset ('Tweets'): This dataset was provided by Zhang et al. in [15] and corresponds to Twitter messages collected from Los Angeles, USA. This dataset consists of 1,584,307 geo-tagged tweets (short-text messages) covering the period of time from 2014.08.01 to 2014.11.30.

Foursquare dataset ('4S'): This data was also provided by Zhang et al. in [15] and consists of Foursquare check-ins reported on Twitter by users in the city of New York, USA. The data contains 479,297 records that indicate places in the city that were visited by users along with their location, for the period of 2010.02.25 to 2012.08.16.

Crime incident dataset ('Crime'): This dataset contains crime reports from the city of New York, USA[6]. It was obtained from the New York City Open Data repository[7]. The dataset contains textual descriptions used by police agents to classify crime incidents along with their geolocation. The dataset consists of 1,016,008 crime incident records, after filtering, that cover the dates starting 2000.01.01 to 2015.12.31.

Table 2, shows a summary of each dataset described.

---

[6]https://data.cityofnewyork.us/api/views/qgea-i56i/rows.csv?accessType=DOWNLOAD
[7]https://opendata.cityofnewyork.us/

14

	Records	City	Start Date	End Date
Tweets	1,584,307	Los Angeles	2014.08.01	2014.11.30
4S	479,297	New York	2010.02.25	2012.08.16
Crime	1,016,008	New York	2000.01.01	2015.12.31

Table 2: Spatio-temporal Annotated Datasets

4.2. Baselines

In this section we describe the baseline methods that we used for validating our approach, STT-RNN. As baselines, we used current state-of-the-art for modeling spatio-temporal textual data, and also existing approaches for geographic topic modeling. Next, we detail these baselines.

- **LGTA [5]** is a generative model where there are latent regions that are geographically distributed by a Gaussian distribution. Each region has a multinomial distribution over topics and each topic is a multinomial distribution over words.

- **MGTM [20]:** is a generative model based on multi-Dirichlet process. The authors use a three leveled hierarchical Dirichlet process with a Fischer distribution for detecting geographical clusters, a Dirichlet-multinomial document-topic distribution and Dirichlet-multinomial topic-word distribution.

- **SVD** performs Singular Value Decomposition on the co-occurrence matrix of *timestamps*, *location* and *words*.

- **Recon [15]** assumes each tuple $\langle time, location, text \rangle$ is a relation and then learns embeddings for *timestamps*, *locations* and *words* such that the relation can be reconstructed.

- **Graph [15]** builds a graph of co-relations and then learns embeddings for *timestamps*, *locations* and *words* such that the structure of the graph can be reconstructed.

4.3. Results

In this section we show a comparison with the state-of-the-art methods. Later, we study how parameters like spatial granularity and temporal granularity affect the quality of predictions.

For each $\langle t, l, e \rangle$ in the dataset, the input to the model is the set of tuples in the form: $\langle time, location, text \rangle$. We kept only those terms that are alpha-words, appear more then 100 times and are not stop-words. Then, we split the dataset in training-validation-test, keeping 20% of each dataset as test, 20% for validation and 60% for training.

To evaluate the model, for each tuple in the test we want to predict an element, given the others of the tuple. For each test prediction, we randomly selected k=10 negative examples. We ranked the negative examples and the target according to the model using the context elements as input.

We used the mean reciprocal rank (MRR) to evaluate the quality of the ranks produced by the model. Given a set Q of queries, the MRR is defined as:

$$MRR = \left(\frac{\sum_{i=1}^{|Q|} \frac{1}{rank_i}}{|Q|} \right) \tag{5}$$

It is worth mentioning that we chose this evaluation setting to keep the evaluation methodology consistent with the evaluation setting of state-of-the-art works [15]. We used the same discretization techniques with the same parameters and the same methodology described in [15].

Method	Text		Location		Time	
	Tweets	4S	Tweets	4S	Tweets	4S
LGTA	0.376	0.6107	0.3792	0.6083	-	-
MGTM	0.3874	0.5974	0.4474	0.5753	-	-
SVD	0.4475	0.4475	0.3953	0.646	0.3256	0.3187
STT-RNN	0.4947	0.7227	**0.7175**	**0.9547**	**0.3939**	**0.4505**
Recon	0.687	0.9219	0.6526	0.9044	0.3582	0.3612
Graph	**0.7011**	**0.9449**	0.6758	0.9168	0.3895	0.3716

Table 3: Mean Reciprocal Rank for spatio-temporal textual data modeling. The three tasks evaluated are predicting each one of the elements of the tuple $\langle time, location, text \rangle$ knowing the other two. In this table, we show results for the social media datasets of the proposed model and state-of-the-art methods.

In Table 3, we show the results for the social media datasets. We can see that STT-RNN outperformed all the models for location prediction and time prediction. The improvement for location prediction are: 0.0649 for

the dataset Tweets and 0.0503 for the dataset 4S. The improvement for time prediction are: 0.0357 for the dataset Tweets and 0.0893 for the dataset 4S. These improvements in Mean Reciprocal Rank correspond to 9.9%, 5.5%, 9.9% and 24.7% with respect to previous state-of-the-art results. This is consistent with our idea that RNNs will produce a better representation than the average of word embeddings for texts. Since the *Text* is only considered as input for *Location* prediction and *Time* prediction, these are the tasks that STT-RNN performed the best.

In the case of *Text* prediction STT-RNN was only outperformed by the state-of-the-art models **Recon** and **Graph**. We believe this is due to the context for *Text* prediction being a sequence of only two elements $\langle Location, Time \rangle$ and a RNN performs at its best with longer sequences when comparing to simple average embedding of words, though STT-RNN outperformed the rest of the baselines. Consistent with previous works, it showed better results for 4S than for Twitter. Also, *Time* prediction proved to be the hardest task. To further analyze these results, we computed the average entropy of the distribution of words for temporal windows and spatial cells for both datasets. In Table 4, we see that for both datasets the entropy of word distribution over spatial cells covers a higher percentage of the maximum entropy than the entropy of the word distribution over temporal windows. This means that words are more strongly correlated to spatial cells than to temporal windows.

	Tweets		4S	
Metric	Time	Location	Time	Location
Ave Entropy	4.50	6.76	4.01	4.23
Maximum Entropy	4.90	13.31	4.64	12.37
% Max Entropy	**0.91**	**0.50**	**0.86**	**0.34**
No Cells	24	5297	29	10159

Table 4: Maximum entropy, number of cells and percent of the maximum entropy covered by the average entropy of words distribution over spatial cells and temporal windows for both social media datasets Tweets (tweets from LA) and 4S (Foursquare check-ins from NY).

4.3.2. Sensitivity analysis to spatial and temporal granularities

In this section, we show how the spatial and temporal granularities affect the results of STT-RNN. We studied how robust the model is to changing the granularity of time windows and spatial cells.

Temporal-granularity analysis. In Table 5, we show results by changing the temporal window size. For these experiments we used a combination of *hour-of-the-day* and *day-of-the-week* resulting in a set of 24x7=128 hour ranges. In our experiment we used temporal windows sizes 1,2,4,8,12 and 24 (See Sect. 3.4). In Table 5 we can see that location prediction is not affected by changes in the temporal variable, while text prediction shows a small drop. For time prediction, the clear tendency is to decrease the MRR while increasing the windows size. We consider that this is due to the fact that increasing the temporal window size introduces noise because a bigger temporal window size means a bigger spreading of when the text was generated and additional places to consider inside the temporal window. Also, this corroborates the idea that the temporal variable is poorly correlated with the other two and changing the temporal discretization almost does not affect the prediction for places and texts.

	Text	Location	Time
Window-Size	4S	4S	4S
1	0.6373	0.9524	0.4489
2	0.6319	0.9532	0.4432
4	0.6334	0.9553	0.4362
8	0.6288	0.9551	0.3938
12	0.6262	0.9542	0.3647
24	0.6209	0.953	0.2966

Table 5: Mean Reciprocal Rank for spatio-temporal textual data modeling. The three tasks evaluated are predicting each one of the elements of the tuple $\langle time, location, text \rangle$ knowing the other two. In this table, we show how STT-RNN performs while changing the temporal window, here we evaluate on the Foursquare dataset.

Spatial-granularity analysis. In Table 6, we show the results by changing the spatial cell size. We used squared equal-size spatial cell by manipulating the continuous values representing the latitudes and longitudes (See Sect. 3.4). We experimented with cells size 0.01, 0.02, 0.03, 0.04, 0.05 and 0.06 which are equivalent to around 10, 20, 30, 40, 50 and 60 blocks. Predicting the temporal variable does not get affected by changing the size of the spatial cell, confirming previous findings about the relations between the temporal variable and the spatial variable. Similar to what happens with time prediction when expanding the temporal window, expanding the spatial cell makes it harder to predict correctly the spatial cell. Also, expanding the spatial

cell makes harder the task of text prediction, confirming a strong correlation between places and texts.

	Text	Location	Time
Cell-Size	4S	4S	4S
0.01	0.6373	0.9524	0.4489
0.02	0.5612	0.941	0.4534
0.03	0.5352	0.9291	0.4521
0.04	0.5013	0.9253	0.4539
0.05	0.4735	0.9125	0.4534
0.06	0.4755	0.9054	0.4541

Table 6: Mean Reciprocal Rank for spatio-temporal textual data modeling. The three tasks evaluated are predicting each one of the elements of the tuple $\langle time, location, text \rangle$ knowing the other two. In this table, we show how STT-RNN performs by changing the spatial granularity. Here we evaluate on the Foursquare dataset.

4.3.3. Crime data analysis

In this section, we show a case study of the application of STT-RNN to a dataset of crime descriptions. We chose this dataset to show the usefulness of applying STT-RNN to different domains. Crime descriptions are texts describing a crime, either with natural language descriptions used by a victim or keywords and phrases used by police agents. We used a dataset of crime descriptions from the city of New York (See Sect. 4.1) which contains texts used by police agents to describe the incident, timestamps of *when* the crime took place and coordinates of *where*.

First, we compared STT-RNN to the state-of-art work [15] following the same methodology described in section 4.3. Similar to the results using the social media datasets, in Table 7 we can see that STT-RNN shows the best results for predicting times and places. Given that STT-RNN is at its best for retrieving places and times, in Table 8 and Table 9 we show results querying STT-RNN when trained with the crime dataset. To show the utility of the model, we queried first with a crime associated with night activity *"alcoholic beverage control law"*. We can see that the results show night hours and weekend days. Second, we queried the model with a crime not associated with night activities *"state laws non penal"*, we can see that the results show afternoon hours and weekdays. For both cases STT-RNN allows us to find hot-spots in the map of the corresponding types of crimes.

19

	Text	Location	Time
Method	Crime	Crime	Crime
Graph	**0.37**	0.3852	0.3191
STT-RNN	0.3109	**0.5586**	**0.3688**

Table 7: Mean Reciprocal Rank for spatio-temporal textual data modeling. In this table we show results for the crime incident dataset of STT-RNN and the state-of-the-art work Graph.

Coordinates	Time-Day	Time-Hour
	Friday	11pm
	Sunday	1am
	Saturday	1am
	Sunday	12am
	Thursday	11pm
	Saturday	10pm
	Saturday	12am
	Tuesday	10pm
	Wednesday	1am
	Thursday	1am

Table 8: Spatial and temporal results for textual queries. **Query="alcoholic beverage control law"**.

5. Discussion and Conclusions

We studied the problem of modeling spatio-temporal annotated textual data and proposed a recurrent neural network that jointly models *text*, *timestamps* and geographical *coordinates*. The proposed model STT-RNN outperformed state-of-the-art methods in our experiments for two of the three tasks evaluated and ranked third for the other task. STT-RNN proved to be an effective method to model spatio-temporal text data. Given a dataset of spatio-temporal text data, a trained STT-RNN model can be queried with any combination of elements from the tuple $\langle time, location, text \rangle$ and recover any of its element. This can be helpful for finding intrinsic spatio-temporal patterns that characterize the textual information exhibited in social media or crime incident descriptions.

We studied the correlation between the three variables $\langle time, location, text \rangle$ in social media data from Twitter and Foursquare. We found that location

20

Coordinates	Time-Day	Time-Hour
	Wednesday	3pm
	Tuesday	3pm
	Monday	3pm
	Wednesday	5pm
	Saturday	6pm
	Thursday	5pm
	Thursday	3pm
	Tuesday	5pm
	Friday	3pm
	Tuesday	4pm

Table 9: Spatial and temporal results for textual queries. **Query="state laws non penal"**.

and text are highly correlated, text and time are slightly correlated and location and place are poorly correlated with each other.

Despite its strengths, STT-RNN has its limitations. Predicting text from $\langle time, location \rangle$ did not outperform the state-of-the-art. We attribute this to the fact that RNNs benefit when processing sequential data as input (e.g., text). Time and space do not exhibit this sequential structure when used as input. In other words, time and space showed to be a weak context for the STT-RNN to extract the information required to predict text.

As future work, we are interested in three lines of further research. First, we would like to explore other architectures for predicting text from the $\langle time, location \rangle$ context. For example, we plan to explore using contextualized word embeddings from large pre-trained neural networks such as ELMo [46] and BERT [47].

Second, we plan to study other automatic discretization techniques. For example, in the case of spatial discretization, we would like to explore using geographical divisions with semantic information like socioeconomic divisions as the discretization criterion.

Third, we intend to study the transferability of our approach by deploying spatio-temporal textual models trained on data from a source domain to a target domain. The source and the target domains can differ from each other in many ways, for example, the source domain can be Twitter and the target domain can be crime reports. The hypothesis is that if the data of the

source domain is larger than that of the target domain, and both domains are related to each other (e.g., the language and the space regions are shared in both domains) then a transfer learning approach can be employed. Neural network models are very suitable for transfer learning as one can pretrain a model from the source domain and adapt it to the target domain via further training. Since neural network leverage statistical strengths from large datasets (the source domain), the transfer learning approach may help to improve performance on the target domain.

Acknowledgements

References

[1] C. Zhang, K. Zhang, Q. Yuan, L. Zhang, T. Hanratty, J. Han, Gmove: Group-level mobility modeling using geo-tagged social media, in: KDD: proceedings. International Conference on Knowledge Discovery & Data Mining, NIH Public Access, 2016, p. 1305.

[2] A. Noulas, S. Scellato, N. Lathia, C. Mascolo, Mining user mobility features for next place prediction in location-based services, in: Data mining (ICDM), 2012 IEEE 12th international conference on, IEEE, 2012, pp. 1038–1043.

[3] Q. Yuan, W. Zhang, C. Zhang, X. Geng, G. Cong, J. Han, Pred: Periodic region detection for mobility modeling of social media users, in: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, ACM, 2017, pp. 263–272.

[4] F. Wu, Z. Li, W.-C. Lee, H. Wang, Z. Huang, Semantic annotation of mobility data using social media, in: Proceedings of the 24th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2015, pp. 1253–1263.

[5] Z. Yin, L. Cao, J. Han, C. Zhai, T. Huang, Geographical topic discovery and comparison, in: Proceedings of the 20th international conference on World wide web, ACM, 2011, pp. 247–256.

[6] S. Sizov, Latent geospatial semantics of social media, ACM Transactions on Intelligent Systems and Technology (TIST) 3 (2012) 64.

[7] C. Zhang, K. Zhang, Q. Yuan, F. Tao, L. Zhang, T. Hanratty, J. Han, React: Online multimodal embedding for recency-aware spatiotemporal activity modeling, in: Proceedings of the 40th International ACM SI-GIR Conference on Research and Development in Information Retrieval, ACM, 2017, pp. 245–254.

[8] T. Sakaki, M. Okazaki, Y. Matsuo, Earthquake shakes twitter users: real-time event detection by social sensors, in: Proceedings of the 19th international conference on World wide web, ACM, 2010, pp. 851–860.

[9] B. Poblete, J. Guzmán, J. Maldonado, F. Tobar, Robust detection of extreme events using twitter: worldwide earthquake monitoring, IEEE Transactions on Multimedia 20 (2018) 2551–2561.

[10] L. Zhao, F. Chen, C.-T. Lu, N. Ramakrishnan, Spatiotemporal event forecasting in social media, in: Proceedings of the 2015 SIAM International Conference on Data Mining, SIAM, 2015, pp. 963–971.

[11] C. Chen, W. Dongxing, H. Chunyan, Y. Xiaojie, Exploiting social media for stock market prediction with factorization machine, in: 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), volume 2, IEEE, 2014, pp. 142–149.

[12] J. Li, H. Bu, J. Wu, Sentiment-aware stock market prediction: A deep learning method, in: 2017 International Conference on Service Systems and Service Management, IEEE, 2017, pp. 1–6.

[13] L. Malandri, F. Z. Xing, C. Orsenigo, C. Vercellis, E. Cambria, Public mood–driven asset allocation: The importance of financial sentiment in portfolio management, Cognitive Computation 10 (2018) 1167–1176.

[14] F. Z. Xing, E. Cambria, Y. Zhang, Sentiment-aware volatility forecasting, Knowledge-Based Systems 176 (2019) 68–76.

[15] C. Zhang, K. Zhang, Q. Yuan, H. Peng, Y. Zheng, T. Hanratty, S. Wang, J. Han, Regions, periods, activities: Uncovering urban dynamics via

cross-modal representation learning, in: Proceedings of the 26th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2017, pp. 361–370.

[16] C. Zhang, M. Liu, Z. Liu, C. Yang, L. Zhang, J. Han, Spatiotemporal activity modeling under data scarcity: A graph-regularized cross-modal embedding approach, in: Thirty-Second AAAI Conference on Artificial Intelligence, p. 12.

[17] Y. Goldberg, Neural network methods for natural language processing, Synthesis Lectures on Human Language Technologies 10 (2017) 1–309.

[18] Q. Mei, C. Liu, H. Su, C. Zhai, A probabilistic approach to spatiotemporal theme pattern mining on weblogs, in: Proceedings of the 15th international conference on World Wide Web, ACM, 2006, pp. 533–542.

[19] C. Wang, J. Wang, X. Xie, W.-Y. Ma, Mining geographic knowledge using location aware topic model, in: Proceedings of the 4th ACM workshop on Geographical information retrieval, ACM, 2007, pp. 65–70.

[20] C. C. Kling, J. Kunegis, S. Sizov, S. Staab, Detecting non-gaussian geographical topics in tagged photo collections, in: Proceedings of the 7th ACM international conference on Web search and data mining, ACM, 2014, pp. 603–612.

[21] L. Hong, A. Ahmed, S. Gurumurthy, A. J. Smola, K. Tsioutsiouliklis, Discovering geographical topics in the twitter stream, in: Proceedings of the 21st international conference on World Wide Web, ACM, 2012, pp. 769–778.

[22] T. Hofmann, Probabilistic latent semantic indexing, in: ACM SIGIR Forum, volume 51, ACM, 2017, pp. 211–218.

[23] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, Journal of Machine Learning research 3 (2003) 993–1022.

[24] D. M. Blei, Probabilistic topic models, Communications of the ACM 55 (2012) 77–84.

[25] M. Ye, D. Shou, W.-C. Lee, P. Yin, K. Janowicz, On the semantic annotation of places in location-based social networks, in: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2011, pp. 520–528.

[26] T. Mikolov, W.-t. Yih, G. Zweig, Linguistic regularities in continuous space word representations., in: HLT-NAACL, volume 13, pp. 746–751.

[27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in neural information processing systems, pp. 3111–3119.

[28] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation., in: EMNLP, volume 14, pp. 1532–43.

[29] X. Huang, J. Li, X. Hu, Label informed attributed network embedding, in: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, ACM, 2017, pp. 731–739.

[30] A. Graves, Supervised sequence labelling, in: Supervised sequence labelling with recurrent neural networks, Springer, 2012, pp. 5–13.

[31] A. Graves, N. Jaitly, Towards end-to-end speech recognition with recurrent neural networks, in: International Conference on Machine Learning, pp. 1764–1772.

[32] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks, in: Advances in neural information processing systems, pp. 3104–3112.

[33] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder–decoder for statistical machine translation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1724–1734.

[34] S. Liu, N. Yang, M. Li, M. Zhou, A recursive recurrent neural network for statistical machine translation, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pp. 1491–1500.

[35] F. Z. Xing, E. Cambria, R. E. Welsch, Intelligent asset allocation via market sentiment views, IEEE Computational Intelligence Magazine 13 (2018) 25–34.

[36] Q. Liu, S. Wu, L. Wang, T. Tan, Predicting the next location: A recurrent model with spatial and temporal contexts., in: AAAI, pp. 194–200.

[37] C. Yang, M. Sun, W. X. Zhao, Z. Liu, E. Y. Chang, A neural network approach to jointly modeling social networks and mobile trajectories, ACM Transactions on Information Systems (TOIS) 35 (2017) 36.

[38] D. Yao, C. Zhang, J. Huang, J. Bi, Serm: A recurrent model for next location prediction in semantic trajectories, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, ACM, 2017, pp. 2411–2414.

[39] J. Feng, Y. Li, C. Zhang, F. Sun, F. Meng, A. Guo, D. Jin, Deepmove: Predicting human mobility with attentional recurrent networks, in: Proceedings of the 2018 World Wide Web Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2018, pp. 1459–1468.

[40] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–1780.

[41] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, in: NIPS 2014 Workshop on Deep Learning, December 2014.

[42] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, Y. Bengio, End-to-end attention-based large vocabulary speech recognition, in: 2016 IEEE international conference on acoustics, speech and signal processing (ICASSP), IEEE, 2016, pp. 4945–4949.

[43] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, CoRR abs/1412.6980 (2014).

[44] P. J. Werbos, et al., Backpropagation through time: what it does and how to do it, Proceedings of the IEEE 78 (1990) 1550–1560.

[45] D. Kinga, J. B. Adam, A method for stochastic optimization, in: International Conference on Learning Representations (ICLR), volume 5, p. 12.

[46] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 2227–2237.

[47] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186.