UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

SPATIO-TEMPORAL TEXTUAL DATA MODELING

TESIS PARA OPTAR AL GRADO DE
DOCTOR EN COMPUTACIÓN

JUGLAR DIAZ ZAMORA

PROFESORES GUÍA:
FELIPE BRAVO MARQUEZ
BARBARA POBLETE LABRA

MIEMBROS DE LA COMISIÓN:
AIDAN HOGAN
MARCELO MENDOZA ROCHA
LUCA MARIA AIELLO

Este trabajo ha sido parcialmente financiado por CONICYT-PCHA/Doctorado Nacional/2016-21160142, Fondecyt Grant No. 1181896, FONDEF Project ID16I10222 y el Instituto Milenio Fundamentos de los Datos

SANTIAGO DE CHILE
2023

Resumen

La popularidad y el uso de redes sociales en dispositivos móviles con GPS proporciona una fuente de textos enriquecidos con contexto espacio-temporal. Otros dominios, como consultas a motores de búsqueda y descripciones de incidentes delictivos, son también fuentes de textos para los que se conoce cuándo y dónde fueron generados. Texto, tiempo y espacio tienen diferentes formas de representación; por lo que no es trivial desarrollar un modelo que los represente de forma conjunta. La representación conjunta de texto, tiempo y espacio se ha basado en técnicas que ignoran la estructura secuencial de los textos y propiedades de tiempo y espacio, como vecindad y jerarquía. Esto puede limitar la expresividad de un modelo para representar ciertos patrones. En esta tesis presentamos dos nuevos modelos para recuperación de información multi-modal y modelado de lenguaje condicionado espacio-temporalmente. Los modelos propuestos encuentran aplicaciones prácticas en recuperación en texto-espacio-tiempo y caracterización de zonas urbanas.

Para la tarea de recuperación multi-modal, proponemos un modelo basado en una red neuronal *Acceptor* que permite consultar con pares del trío texto-espacio-tiempo para recuperar el tercero. Esto resulta en tres tareas de recuperación que se entrenan simultáneamente. Nuestros experimentos muestran que modelar la estructura secuencial de los textos tiene un impacto positivo en la recuperación de tiempos y lugares. El modelo supera trabajos previos en márgenes desde el 1% al 21% en experimentos desarrollados sobre conjuntos de datos extraídos de las redes sociales Twitter y Foursquare. Nuestras evaluaciones cualitativas demuestran la utilidad del modelo propuesto para descubrir patrones espacio-temporales de delincuencia a partir de reportes de incidentes delictivos.

Para la tarea de modelado de lenguaje condicionado espacio-temporalmente, presentamos una red neuronal que nos permite representar tiempo y espacio como contexto para generación de texto en diferentes granularidades. Nuestros resultados experimentales muestran diferencias significativas en cómo el espacio y el tiempo afectan la generación de lenguaje. Para los datos extraídos de Twitter, el punto ideal para la representación espacial es celdas de 800m × 800m aproximadamente; mientras que para los datos de Foursquare, los mejores resultados se obtienen a medida que las celdas espaciales se hacen más pequeñas. Considerando la representación del contexto temporal, los resultados sobre los datos de Twitter mostraron mejoras marginales pero no fueron tan significativos como el contexto espacial; para los datos de Foursquare, incluir el contexto temporal es mejor que no incluirlo, pero cuando se combina con el contexto espacial muestra no ser un factor positivo. Desarrollamos análisis cualitativos que ejemplifican el uso del modelo propuesto para caracterizar zonas urbanas y cómo una red neuronal basada en atención permite visualizar las relaciones entre el lenguaje natural y el contexto espacio-temporal dónde se genera.

En esta tesis presentamos dos modelos para representación de texto, tiempo y espacio. El modelo de lenguaje permite modelar tiempo y espacio en diferentes granularidades para generación de texto. El modelo para recuperación multi-modal permite consultar con pares de espacio, tiempo y texto; para recuperar el tercero.

Abstract

The popularity of mobile devices with GPS capabilities and the wide adoption of social media has created a rich source of textual data combined with spatio-temporal information. In addition, other domains such as search engine queries and crime incident descriptions are sources of text data associated with timestamps and geo-coordinates. These data sources can be used to gain space-time insights into human behavior. From a data modeling perspective: text, time, and space have different representation approaches; hence it is not trivial to represent them in a unified model. Spatio-temporal textual data representation has relied on techniques ignoring the sequential structure of texts and properties of time and space like neighborhood and hierarchy. This can limit a model's expressiveness for representing certain patterns extracted from spatio-temporal textual data. This thesis is centered around two problems of spatio-temporal textual data processing: multi-modal retrieval and spatio-temporal conditioned language modeling. This results in two spatio-temporal modeling tasks with practical applications on space-time-text retrieval and characterization of urban areas with natural language.

For the multi-modal retrieval task, we propose an *Acceptor* recurrent neural network that allows us to query the model with pairs of elements of space, time, and text to retrieve the third one. This results in three retrieval tasks that are trained simultaneously. Our experiments show that modeling the sequential structure of texts positively impacts retrieving times and places. The model outperforms prior works ranging from a 1% to a 21% improvement for place retrieval and text retrieval on two social media datasets from Twitter and Foursquare. We also conduct qualitative evaluations where we demonstrate the utility of the presented model for finding spatio-temporal patterns of crime from a dataset of crime incident reports.

For the spatio-temporal conditioned language modeling task, we present an end-to-end neural network that allows us to represent time and space as a context for text generation at different granularities. Our results show significant differences in how space and time influence language modeling. For the Twitter dataset, the optimal when modeling space is to discretize around 800m × 800m cells; while for the Foursquare dataset, we observed the best results as the spatial cell got smaller. For the temporal context, including it for the Twitter dataset resulted in small improvements but was not as important as including the spatial context; for the Foursquare dataset, including the temporal context is better than not including it at all, but when combined with the spatial context proved not to be a positive factor as context for language generation. We present qualitative analyses where the proposed model is used to characterize urban places from the perspective of social media. We demonstrate how an attention-based neural network can be used to visualize relations between text and the spatio-temporal context where it is generated.

The models presented in this thesis tackle different needs and complement each other. The language model allows for language generation while modeling time and space at different granularities; the retrieval model allows for querying a multi-modal retrieval model with any pair of space, time, and text; to retrieve the third one.

A mis padres por su apoyo constante y amor incondicional

Acknowledgements

I want to take this opportunity to thank all the people who contributed in one way or another to this achievement.

First off, thanks to my family for their unconditional love and support. My mom, my dad, and my sister, this is for you.

I would like to thank my supervisors Barbara and Felipe, for all the opportune encouragement, guidance, support, and friendship. I could not ask for better role models on how to be a great researcher, professional, and educator.

Thanks to my friends for making me feel like a family in a new country away from my loved ones. For the parties, the games, the movies, the trips, and the wonderful times.

Thanks to all the professors at the DCC for receiving me to a new country and always making me feel welcome. Thanks to ANID and Chile for the scholarship that funded my Ph.D.

Thanks to all of you.

Table of Content

List of Tables

List of Figures

Chapter 1

Introduction

Online social media plays a crucial role in modern societies; it has gained adoption worldwide and is now considered influential in public opinion. Within this context, social platforms such as Twitter[1], Instagram[2], Facebook[3] and Foursquare[4] have enabled users to share the textual and multimedia content they generate (e.g., opinions, interests, reviews, and everyday 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 geo-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 patterns [121, 70, 115], also *when* and *where* popular social activities are taking place [107, 113, 91, 120]. In addition, social media has been successfully used to detect and understand real-world events such as earthquakes, typhoons, and civil unrest [83, 123]. Besides social media, other data sources relate semantic content with spatio-temporal information. An example is crime reports that include a natural language description of the crime and the time and place it occurred. The textual crime descriptions are either in the form of free text provided by the victim or based on keywords and more standardized phrases used by the police. Overall, access to this type of data can allow us to study and model textual information in relation to its spatio-temporal context.

This thesis focuses on *spatio-temporal textual data modeling.* The multi-modality of *space, time* and *text* provides a challenge in modeling their joint representation. In particular, text, its timestamp, and geographical coordinates are commonly represented in different scales and magnitudes. For instance, the text is discrete and has been represented using vector spaces, while timestamps and geo-coordinates are continuous variables. Hence, it is not trivial to combine these components into a unified model. In this thesis, we tackle two traditional well-defined tasks in the

[1] https://twitter.com/

[2] https://www.instagram.com/

[3] https://www.facebook.com/

[4] https://foursquare.com//

context of spatio-temporal textual data generation: information retrieval (IR) and language modeling (LM).

We propose two models: 1) a multi-modal retrieval model where we focus on retrieving one element of the tuple $\langle time, location, text \rangle$ giving the other two as query, and 2) a spatio-temporal conditioned language model where we focus on language modeling given a spatio-temporal context. Model/task 1) allows us to query the model with any combination of keywords, geo-coordinates, or timestamps and retrieve any of the three variables, while model/task 2) allows us to generate a spatio-temporal characterization of urban areas using natural language. We conducted our experiments with freely available public datasets collected from social media platforms Twitter and Foursquare, as well as crime incident reports from the city of New York.

The remainder of this chapter is organized as follows. Section 1.1 presents the research problem. Section 1.2 briefly presents previous solutions and their limitations. The research question and the proposed solutions are introduced in Section 1.3. The main contributions and main results are presented in Sections 1.4 and 1.5. The publications derived from this work are listed in Section 1.6, and in Section 1.7, we present the outline of the thesis's structure.

1.1 Research problem

One challenge related to modeling spatio-temporal conditioned textual data is its multi-modality. Timestamps, geo-coordinates, and texts exhibit different magnitudes and representation approaches, making it difficult to combine them effectively. Timestamps and geo-coordinates are continuous variables, while the text is a sequence of discrete items (i.e. words). An additional challenge is associated with the individual representation of each type of variable. Previous approaches for modeling how text is generated in a spatio-temporal context use a single granularity representation for time or space, either using hand-crafted discretizations [64, 120, 117], automatic models like clustering algorithms [119], or probabilistic models [27, 90, 100, 47, 3, 52]. Spatio-temporal patterns for text data generation should capture patterns at different granularities like hours, weeks, months, or years in the case of time; and houses, blocks, neighborhoods, or cities in the case of space. In this sense, we expect a model that leverages text data generated under spatio-temporal conditions to be flexible enough to capture these granularities at different levels. Previous works on the joint representation of space, time, and text are based on feature embedding representation or topic modeling. These approaches ignore the sequential structure of texts, which can have a negative impact on performance. Endowing our models with the ability to represent such sequential structure and the multi-granularity of time and space is the major contribution of this work.

Given a collection of records that provide textual descriptions of a geographical area at different moments in time, the general problem that we aim to solve is how to effectively model text jointly with the timestamp and geo-coordinates that it was generated at. We separate this general problem into two specific sub-problems based on IR and LM:

1. Let $H = \{r_1, \ldots, r_n\}$ be 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. We aim to solve the sub-problem: given an incomplete record where either t_i, r_i, or e_i is missing, to retrieve the missing item. This results in three retrieval tasks in which we rank the candidates in the collection given the query:

   (a) to retrieve the time for which a certain text was produced in a particular location,

   (b) to retrieve the location from which a text was generated at a certain time, and

   (c) to retrieve the text that is created from a certain location and time.

   An important point to clarify is the difference between retrieving where a text is written vs retrieving the event/place that a text is written about. There are two reasons why we focus our research on retrieving where a text is written instead of retrieving the event/place that a text is written about. i) Predicting the place/event that a text is written about is a problem where we would need to know the set of place/event labels that are covered in the dataset and annotate each text with its corresponding label. We would need human annotations to train a model to make this type of prediction. ii) Our research focuses on finding patterns of how texts are generated given a timestamp and geo-coordinates. If the pattern of people in place X writing about events in place Y is strong enough, it will be captured by the model, hence the pattern of people from place X writing about place Y will be found.

2. Let $H = \{r_1, \ldots, r_n\}$ be a set of spatio-temporal annotated text records (e.g., a tweet). 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. Since e_i is a sequence of words $w_{i0}, \ldots, w_{in}$; the sub-problem that we aim to solve is assigning a probability to $w_{i0}, \ldots, w_{in}$ given the spatio-temporal context $\langle t_i, l_i \rangle$, which is an instance of a conditioned language modeling task (see Section 2.3.2). This task can be written as $p((w_{i0}, \ldots, w_{in})/\langle t_i, l_i \rangle)$

Each of these problems tackles different needs. The multi-modal retrieval model can be helpful in scenarios where we are interested in knowing the most likely third element given two of the tuple $\langle \textit{time}, \textit{location}, \textit{text} \rangle$. The following are examples of these scenarios:

- Helping local police optimize the allocation of their agents to areas 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*).

- Finding places where certain activities occur 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*).

- To 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*).

On the other hand, the spatio-temporal conditioned language modeling task can help to describe/summarize spatio-temporal human activities with natural language beyond just keywords. Consider the following example:

- Given that a sporting event like a basketball game is taking place at a venue like the Staples Center, a coherent natural language description of the event can provide insights into people's feelings about what they are experiencing and the general mood at the event. Table 5.8 provides additional examples of this type of application.

1.2 Prior works and differentiation

Prior works have evolved from topic modeling to feature embedding. Early approaches [3, 27, 47, 52, 64, 90, 100] based on topic modeling aim to discover topics related to geographical areas. Works based on learned embedded representations [117, 119, 120] use feature embeddings methods to find learned representations for the elements of the tuple $\langle time, location, text\rangle$.

Works following the topic modeling approach take inspiration from topic models such as Probabilistic Latent Semantic Analysis [10] and Latent Dirichlet Allocation [11]. These works extend traditional models by assigning distribution probabilities over locations to topics or introducing latent geographical regions. Feature embedding methods find distributed learned representations for discrete variables. Learned embedded representations are very popular in natural language processing [65, 76], graph node representation [49], and computer vision [32]. For spatio-temporal textual data, embedded representations learn a joint representation for the elements of the tuple $\langle time, location, text\rangle$. At inference time, a text is represented as the average of its word embedding representations. In Chapter 3, we provide a more in-depth description of works modeling spatio-temporal textual data following the two approaches: topic modeling and feature embedding representations.

Both topic models and feature embedding methods assume a bag-of-words approach for text modeling, which ignores the sequential structure of texts. Hence, potentially relevant language patterns derived from the sequential nature of text data are discarded. When considering time and space modeling, each work models timestamps and geo-coordinates at a single level of granularity using hand-crafted spatial cells and temporal windows or clustering algorithms. Only Ahmed et al. [3] model hierarchy, but only for space; to the best of our knowledge, there are no studies of how representing time and space at different levels of granularity impact the modeling of text generation under spatio-temporal conditions.

Overall, we can conclude that existing approaches ignore two dimensions of the problem:

1. the sequential structure of language.
2. jointly modeling texts together with time and space represented at different granularities.

1.3 Proposed solution

This thesis addresses the problem of modeling textual data generated under spatio-temporal conditions. The research question is as follows:

> Can the joint representation of text, time (timestamp), and space (geographic coordinates), be better modeled by capturing the sequential structure of texts and representing the spatio-temporal variables at multiple levels of granularity?

Modeling space and time at different levels of granularity should allow for better spatio-temporal representation models by adapting to different data sources with different spatio-temporal patterns of language generation. Also, potentially relevant language patterns derived from the sequential nature of the text, that are discarded when modeling text following a bag-of-words approach, can improve the modeling of texts in conjunction with time and space. We will answer the research question by developing and conducting quantitative and qualitative evaluations over spatio-temporal textual data representation models that capture the properties of text, time, and space mentioned before.

We separate the problem of modeling spatio-temporal annotated textual data into two specific problems: 1) a retrieval problem where the goal is to retrieve one element of the tuple $\langle \textit{time}, \textit{location}, \textit{text} \rangle$ given the others as query, and 2) a spatio-temporal conditioned language modeling problem where the goal is to model language generation under spatio-temporal conditions. We propose two neural network architectures to tackle these problems. Next, we provide an overview of these models.

1.3.1 Multi-modal retrieval model

We propose an *Acceptor* [35] recurrent neural network (RNN) architecture which we refer to as STT-RNN. 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 [35]. STT-RNN follows an *Acceptor* usage pattern and is designed to provide an integrated view of spatio-temporal textual data. Specifically, STT-RNN is designed to retrieve one element of the tuple $\langle time, location, text \rangle$ by only knowing the other two. Our proposed model aims to provide a representation that allows us to extract patterns related to spatio-temporal human activities. We propose a model that can be trained on spatio-temporal text records and, can be used to gain insight into the following three 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?

1.3.2 Spatio-temporal conditioned language model

With the spatio-temporal conditioned language model, we propose an end-to-end neural network for encoding spatial and temporal contexts and decoding/generating text. In contrast to the previous method, our design is targeted to model the spatio-temporal context at different granularities. We employ an encoder-decoder architecture where we test state-of-the-art sequence representation methods based on recurrent neural networks [21] and attention-based neural networks [99].

By modeling time and space at different granularities, the proposed architecture is adaptable to the specific characteristics of each data source. This has proven to be paramount according to our experiments. Also, we can analyze how each granularity level is weighted in the representation model. Attention-based neural networks like the transformer architecture have the benefit of providing insights into the importance of components of the spatio-temporal context by visualizing the attention weights. The proposed model can be used to summarize activities in urban environments with natural language generation. This application highlights the importance of modeling the sequential structure of texts in order to generate coherent descriptions for spatio-temporal contexts (the sequential structure of texts is ignored in the retrieval model for text retrieval).

1.4 Main contributions

The main contributions of this thesis center around the two models proposed: the multi-modal retrieval model and the spatio-temporal conditioned language model. Table 1.1 presents a comparison between these models considering their capabilities, what information properties they can model, and what tasks they solve. The language model allows for language generation and for modeling time and space at different granularities as context for language generation; while the retrieval model allows for querying a multi-modal retrieval model with any combination of $\langle \textit{time}, \textit{location}, \textit{text} \rangle$ and to retrieve the third one.

Table 1.1: Model comparison.

	Retrieval Model	Language Model
Hierarchy	✗	✓
Neighborhood	✗	✓
Multi-modal Retrieval	✓	✗
Natural Language Generation	✗	✓

The multi-modal retrieval model finds practical applications in information-seeking tasks, for example: to find patterns of crime incidents given a dataset of crime incident descriptions (see Section 4.2.4). The spatio-temporal conditioned language model allows us to characterize urban locations from the perspective of social media (see Section 5.2.4) with natural language as well as visualizing relations between texts and the spatio-temporal context where it is generated (see Section 5.2.4).

1.5 Main results

Our main results are related to exploratory analyses as well as to qualitative and quantitative evaluations conducted over the proposed multi-modal retrieval model and the spatio-temporal conditioned language model. Exploratory analyses conducted over two social media datasets from Twitter and Foursquare, we found that words are more related to places than to times; covering 91% and 86% of the maximum entropy with temporal windows while only 50% and 34% with spatial cells (see Section 2.6.1). From quantitative evaluations, the multi-modal retrieval model outperforms previous works on Mean Reciprocal Rank (see Section 2.4.1) from 0.6758 to 0.7175 (6%) for place retrieval and from 0.3895 to 0.3939 (1%) for time retrieval, in the Twitter dataset (see Section 4.2.2). The improvements for the Foursquare dataset are from 0.9168 to 0.9547 (4%) and from 0.3716 to 0.4505 (21%) (see Section 4.2.2). For the spatio-temporal conditioned language model, when considering the spatial context, we observed with the Twitter dataset that the optimal for spatial cells is around 800m × 800m cells (around 0.008 geo-coordinates values in cell-size); while for the Foursquare dataset, the observed pattern is that the lower the spatial cell, the better the modeling of the spatial context (see Section 5.2.3). The temporal context proved not to be a principal factor.

1.6 Publications

Journal papers:

- **Juglar Diaz**, Felipe Bravo-Marquez and Barbara Poblete. "An Integrated Model for Textual Social Media Data with Spatio-Temporal Dimensions." *Information Processing & Management* 57, no. 5 (2020): 102219.

- **Juglar Diaz**, Felipe Bravo-Marquez and Barbara Poblete. "Language Modeling on Location-Based Social Networks." *ISPRS International Journal of Geo-Information* 11, no. 2 (2022): 147.

Workshop and short papers:

- **Juglar Diaz** and Barbara Poblete. "Car Theft Reports: a Temporal Analysis from a Social Media Perspective." *In Companion Proceedings of the 2019 World Wide Web Conference* (WWW'19 Companion), (2019), San Francisco, CA, USA, 4 pages.

- **Juglar Diaz**. "Spatio-temporal Conditioned Language Models." *In Proceedings of the 43rd International Conference on Research and Development in Information Retrieval* (ACM SIGIR), (2020), Virtual Event, China, pp. 2478-2478.

- **Juglar Diaz** and Barbara Poblete. "Spatio-temporal data representation: place, time and text embedded in the same space." *Alberto Mendelson Workshop* (2018), Cali, Colombia. (Presentation only, opted out of proceedings)

1.7 Thesis outline

This thesis is organized as follows:

Chapter 2 describes the preliminary concepts needed to better understand the content of this thesis. We describe topic modeling, neural network representations, retrieval models, language models, evaluation metrics, and datasets. We present an exploratory analysis of the relation of words with time and space, as well as how the data distributes over temporal windows and spatial cells in the presented datasets.

Chapter 3 provides a background of the literature relevant to this thesis. In the first part of the section, we provide as context a discussion of applications that leverage spatio-temporal textual data; after that, we delve into models that jointly represent the three variables and highlight existing drawbacks from previous approaches that need to be addressed.

Chapter 4 describes our proposed retrieval model that tackles the task of retrieving one element of the tuple *space, time* and *text* giving the other two as a query. Also, we present the experimental results conducted to validate this model.

Chapter 5 describes our proposed spatio-temporal conditioned language model. We present our experimental results where we perform quantitative and qualitative evaluations.

Chapter 6 presents the conclusions highlighting the main findings of this thesis, as well as directions for future work.

Chapter 2

Preliminaries

This chapter presents the conceptual and theoretical background necessary for understanding this work. We begin by describing some preliminaries central to this thesis's topic; in particular, we present properties of text, time, and space that need to be considered when modeling spatio-temporal textual data. We further describe machine learning representation models related to spatio-temporal text data representation. After that, we describe evaluation measures and the datasets we use to conduct qualitative and quantitative evaluations. Finally, we describe discretization approaches for timestamps and geo-coordinates and show exploratory analyses on how text correlates to time and space; as well as how the examples in our datasets distribute over different times and places.

2.1 Spatio-temporal text data properties

This section describes the properties of text, time, and space central to our research question (see Section 1.3). While the sequential structure of texts is explicitly captured by models designed with this intention, properties of time and space like neighborhood and hierarchy are captured by modeling time and space at different levels of granularity. Next, we delve into these properties:

- What is meant by *"neighborhood"*? - As Tobler's first law of geography states: *"everything is related to everything else, but near things are more related than distant things"* [96]. By leveraging neighborhood, we aim to capture that texts generated near in time or space should be more similar than text generated far in time or space.
- What is meant by *"hierarchy"*? - Both time and space show a hierarchical composition. In the case of time: hours make days, days make weeks, weeks make months, months make seasons, and seasons make years. For space: buildings make blocks, blocks make neighborhoods, neighborhoods make municipalities,

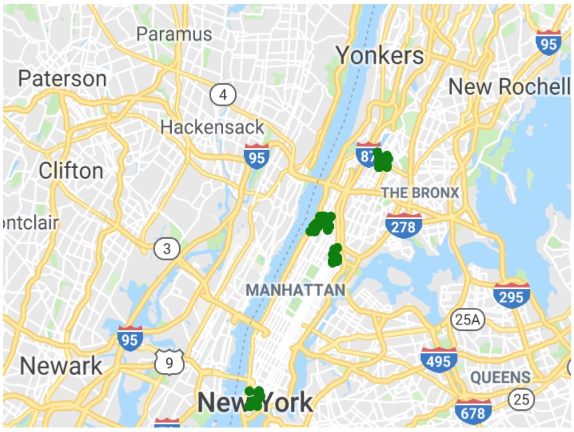


Figure 2.1: Non-penal crime incident descriptions in New York.

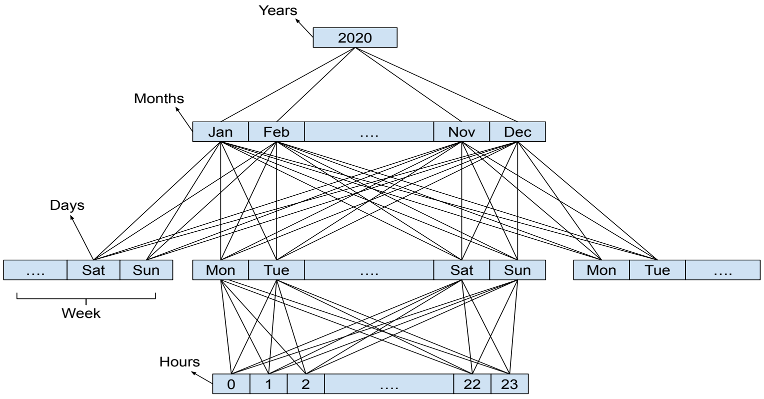


Figure 2.2: Hierarchy representation.

etc. These human-created semantic arrangements of time and space can impact how text is generated in spatio-temporal contexts.

- What is meant by *"sequential structure of text"*? - The bag-of-words approach ignores the sequential structure of texts. Discarding word order in text representation ignores the context and can fall short of capturing sentence meaning (semantics) [50, 8, 9]. Context and meaning could tell the difference between the same words differently arranged ("this is a nice place" vs "is this a nice place?"), capture lexical relations like synonymy ("beautiful location" vs "great location") or antonymy ("I am having a good time in a beautiful location" vs "I am having a bad time in a terrible location"), and much more.

2.2 Machine learning models

In this section, we overview machine learning models relevant to this thesis. These models have been successfully applied in previous approaches or are the base for novel models proposed in this thesis. We describe topic models and neural network representations like feature embedding, recurrent neural networks, and self-attention neural networks.

2.2.1 Topic modeling

Topic modeling is an unsupervised machine learning technique capable of scanning a set of documents and automatically detecting topics from texts. The basic assumption of topic models is that each document consists of a mixture of topics, and each topic consists of a collection of words. Latent Semantic Analysis (LSA) [26], Probabilistic Latent Semantic Analysis (PLSA) [45], and Latent Dirichlet Allocation (LDA) [11] are the most frequent topic modeling techniques. The core idea of LSA is to take a matrix of what is the input: documents, and words and decompose it into a separate document-topic matrix and a topic-word matrix; usually by Singular Value Decomposition. PLSA adds a probabilistic treatment of topics and words on top of LSA. The core idea is to find a probabilistic model with latent topics that can generate the data observed in the document-word matrix. LDA is a generalization of PLSA, and it introduces sparse Dirichlet prior distributions over document-topic and topic-word distributions, encoding the intuition that documents cover a small number of topics and that topics often use a small number of words. All these models treat documents as bag-of-words, discarding language patterns derived from the sequential nature of the text.

2.2.2 Neural network representations

Artificial neural networks (ANNs) [42] are a sub-field of machine learning inspired by the working of the human brain. ANNs are comprised of artificial neuron layers containing an input layer, one or more hidden layers, and an output layer. Each artificial neuron receives the output of artificial neurons in the previous layer; these values are weighted and passed through an activation function which generates the output that is passed to the next layer of the network.

Next, we describe three neural network models that are related to this thesis. First, feature embedding models, in particular, word2vec [65, 66]. After that, we describe the two neural network architectures that have shown the best results for natural language processing tasks: recurrent neural networks [23, 38, 43] and self-attention based neural networks [24, 79, 99, 109]. We focus on the gated recurrent unit [21] recurrent neural network and the transformer [99] self-attention neural network.

Feature embedding: word2vec

Feature embedding models are used to find a dense, low-dimensional continuous vector representation for discrete variables. These methods have been successfully applied for representing words [65, 66, 76] and nodes in graphs [49]. In the case of spatio-temporal textual data, embedding methods allow for representing 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 to employ embedding methods. Next, we describe word2vec [65, 66], a popular word embedding method whose approach is similar to previous feature embedding models for spatio-temporal textual data representation [117, 119, 120] (see Section 3.1.4).

Word2vec takes as its input a text corpus and produces as output word vectors for each word in the vocabulary. These word vectors present properties such that words that appear in similar contexts in the corpus are located close to one another in the vector space. There are two word2vec model architectures: skip-gram (Skip-gram) and continuous bag-of-words (CBOW). They both focus on capturing relations between a center word and its context in sliding windows over the corpus. These models are two-layer neural networks, where each word has an associated vector in each layer depending on whether the word is acting as a center word or as a context word. In the Skip-gram architecture, the model uses the current word to predict the surrounding words in the context window. In the CBOW architecture, the model predicts the current word from the window given the surrounding context words. The order of context words does not influence the prediction, hence the bag-of-words assumption.

In a general view, given a target word that is associated with vector w_t, and a predicted word vector v_p; the probability of the target word conditioned on the predicted word is calculated by the softmax function in equation 2.1, where W is the set of all target word vectors. Equation 2.2 shows cost functions for one target word where the goal is to minimize the negative log-likelihood of the target word vector given its corresponding predicted word.

$$P(w_t|v_p) = \frac{\exp(w_t^T v_p)}{\sum_{w \in W} \exp(w^T v_p)}, \quad w_t, v_p, w \in \mathbb{R}^d; \quad t, p \in (1, 2, ..., VOCAB) \tag{2.1}$$

$$Loss(w_t, v_p) = -\log P(w_t|v_p), \quad w_t, v_p \in \mathbb{R}^d; \quad t, p \in (1, 2, ..., VOCAB) \tag{2.2}$$

In the Skip-gram model, for an index i and a window size c, Skip-gram predicts the context words $\{w_j\}$, $(i - c \leq j \leq i + c, j \neq i)$ given the center word v_i. Hence, $w_t = w_j$ and $v_p = v_i$ for this case in the general model. Equation 2.3 shows the derivation of the Skip-gram cost function.

$$Loss_{skipgram}(c, i) = \sum_{i-c \leq j \leq i+c, i \neq j} - \log P(w_j | v_i), \quad w_j, v_i \in \mathbb{R}^d \tag{2.3}$$

CBOW predicts a word given its context. The target word vector is now the output vector of the word at index i, while the predicted word vector v_p is the sum over all context input vectors (Equation 2.4). Equation 2.5 shows the derivation of the CBOW cost function.

$$v_p = \sum_{i-c \leq j \leq i+c, i \neq j} v_j, \qquad v_p, v_j \in \mathbb{R}^d \tag{2.4}$$

$$Loss_{CBOW}(c, i) = - \log P(w_i | v_p), \qquad w_i, v_p \in \mathbb{R}^d \tag{2.5}$$

Authors Mikolov et al. claim that CBOW is faster to train while Skip-gram finds better representations for infrequent words[1].

Recurrent neural networks: gated recurrent unit

Recurrent neural networks (RNN) [38] are a family of neural network architectures that capture temporal dynamic behavior. RNNs have been successfully applied to natural language processing problems like speech recognition [39] and machine translation [93, 22, 59]. In the case of spatio-temporal data, they have been mostly used for mobility modeling [58, 108, 111, 30]. In the basic building block for a RNN, a vector h represents the state of an input sequence, allowing it to perform sequence processing tasks. At each timestep t, the model takes as an input h_{t-1} and the t-th element of the sequence x_t; then computes h_t. This enables making predictions at each time step t for each object, or at the last object for the full sequence. For language modeling, at each time step t, h_t is used as input to a feed-forward network that predicts the next token x_{t+1}. These building blocks can be stacked into L levels; where the output of each block is used as input to the next block, and predictions are computed over the outputs of the last level in the stack. Also, in bi-directional RNNs [86], predictions can be made based on past and future contexts. This is done by concatenating the outputs of two RNNs; one processing the input from left to right, and the other one from right to left.

A simple instantiation of a RNN is Elman's network. In Elman's network, h_t is computed as the result of a sigmoid function applied to the sum of vector b after the matrix-vector multiplication of weights matrices W and U with input sequence element x_t and previous state h_{t-1} (Equation 2.6).

$$h_t = \text{Sigmoid}(W_h x_t + U_h h_{t-1} + b_h), \quad W_h, U_h \in \mathbb{R}^{k \times d}; b_h \in \mathbb{R}^d \tag{2.6}$$

[1] https://code.google.com/archive/p/word2vec/

The most popular architectures of RNNs are the Long-Short Term Memory recurrent neural network (LSTM-RNN) [43] and the Gated Recurrent Unit recurrent neural network (GRU-RNN) [21]. Both variants introduce gate mechanisms that control the information flow between the hidden states representing the sequence with the goal of preventing vanishing or exploding gradients problems in the training process. Next, we describe GRU-RNN; which has been proven to give similar results to LSTM-RNN with fewer parameters, which makes it faster to train [23]. We develop our recurrent neural network experiments in this thesis with GRU-RNN (see Sections 4.2.2 and 5.2.2).

Equations 2.7, 2.8, 2.9, and 2.10 describe a GRU-RNN block. A GRU-RNN block has two gates, a reset gate r (Equation 2.7) and an update gate z (Equation 2.9). The reset gate decides what information from the past (h_{t-1}) to forget (Equation 2.8). The update gate decides how much the state representation h_t updates its content (Equation 2.10). The state of the GRU-RNN at time t (h_t), is a linear interpolation between the previous state h_{t-1} and the candidate state c_t (Equation 2.8).

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r), \qquad W_r \in \mathbb{R}^{s\times d}, x_t \in \mathbb{R}^d, U_r \in \mathbb{R}^{s\times s}; h_{t-1}, b_r \in \mathbb{R}^s \tag{2.7}$$

$$c_t = \tanh(W_c x_t + U_c(r_t \odot h_{t-1}) + b_c), \qquad W_c \in \mathbb{R}^{s\times d}, U_c \in \mathbb{R}^{s\times s}; b_c \in \mathbb{R}^s \tag{2.8}$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z), \qquad W_z \in \mathbb{R}^{s\times d}, U_z \in \mathbb{R}^{s\times s}; b_z \in \mathbb{R}^s \tag{2.9}$$

$$h_t = (1 - z_t) \odot c_t + z_t \odot h_{(t-1)}, \qquad \odot\text{: element-wise product}; z_t \in \mathbb{R}^s \tag{2.10}$$

The hyper-parameters of the GRU-RNN are d (input embedding size), s (hidden state size) and L (number of layers). In our experiments, we use two different settings: $d = 64$, $s = 128$, and $L = 1$ in Section 4.1 and $d = 128$, $s = 128$, and $L = 2$ in Section 5.1.1. We selected this parameter configuration to fit our available hardware capabilities.

Self-attention based neural networks: the transformer

Self-attention architectures like the transformer [99] have revolutionized the natural language processing field with several works that followed this approach [24, 109, 79]. The transformer architecture discards the recurrent component of RNNs that limits parallelization. This allows faster training with superior quality when compared to previous models based on convolutional neural networks or recurrent neural networks. The transformer was initially proposed for a language translation task, composed of an encoder-decoder architecture. Pre-trained language models following this approach have improved the state of the art for many NLP tasks [24, 109, 79].

A transformer is a stacking of L transformer blocks. A transformer block can be interpreted as a function $f_\theta : \mathbb{R}^{n \times d} \rightarrow \mathbb{R}^{n \times d}$. Given an input $x \in \mathbb{R}^{n \times d}$, x should be interpreted as a collection (often a sequence) of n objects, each with d features. The attention mechanism differentially weights (Equations 2.11, 2.12 and 2.13) the importance of each part of the input data. Each object involved in the attention process is associated with vectors: *value*, *query*, and *key* (Equation 2.11). These vectors are used to produce the representation for each object as a weighted sum of the values, where the weight assigned to each value is computed by a function of the query and the key (Equation 2.12).

$$\begin{array}{ll} Q^{(h)}(x_i) = W_{h,q}(x_i), & W_{h,q} \in \mathbb{R}^{l \times d} \\ K^{(h)}(x_i) = W_{h,k}(x_i), & W_{h,k} \in \mathbb{R}^{l \times d} \\ V^{(h)}(x_i) = W_{h,v}(x_i), & W_{h,v} \in \mathbb{R}^{l \times d} \end{array} \tag{2.11}$$

Each block in a transformer model has H attention heads. Each set h of matrices $(W_{h,q}, W_{h,k}, W_{h,v})$, where $h = 1, \ldots, H$; is called an attention head (Equation 2.11). The output values of the H attention heads are concatenated to compute the output of the multi-head attention layer (Equation 2.13). Multi-head attention allows encoding multiple patterns and nuances in the relationships between objects like words in a text.

$$\alpha_{i,j}^{(h)} = \text{Softmax}(\tfrac{\langle Q^{(h)}(x_i), K^{(h)}(x_i) \rangle}{\sqrt{l}}), \qquad l: \text{ dimension of key vector} \tag{2.12}$$

$$\mathbf{u}'_i = \text{Dropout}_1(W_u(\text{Concat}_{(h=1)}^{H}(\textstyle\sum_{j=1}^{n} \alpha_{i,j}^{(h)} V^{(h)}(x_i))); \rho_1), \quad W_u \in \mathbb{R}^{d \times d} \tag{2.13}$$

$$\mathbf{u}_i = \text{LayerNorm}(\mathbf{x}_i + \mathbf{u}'_i; \gamma_1, \beta_1), \qquad \gamma_1, \beta_1 \in \mathbb{R}^d \tag{2.14}$$

$$\mathbf{y}'_i = \text{Dropout}_2(W_2(\text{ReLU}(W_1(\mathbf{u}_i))); \rho_2), \qquad W_1 \in \mathbb{R}^{d \times m}, W_2 \in \mathbb{R}^{m \times d} \tag{2.15}$$

$$\mathbf{y}_i = \text{LayerNorm}(\mathbf{u}_i + \mathbf{y}'_i; \gamma_2, \beta_2), \qquad \gamma_2, \beta_2 \in \mathbb{R}^d \tag{2.16}$$

$$\begin{aligned} &\mu_{\mathbf{y}} = \tfrac{1}{n}\textstyle\sum_{i=1}^{n}\mathbf{y}_i, \quad \sigma_{\mathbf{y}} = \sqrt{\tfrac{1}{n}\textstyle\sum_{i=1}^{n}(\mathbf{y}_i - \mu_{\mathbf{y}})^2} \\ &\text{LayerNorm}(\mathbf{y};\gamma,\beta) = \gamma\tfrac{(\mathbf{y}-\mu_{\mathbf{y}})}{\sigma_{\mathbf{y}}} + \beta, \qquad \gamma_2, \beta_2 \in \mathbb{R}^d \end{aligned} \tag{2.17}$$

A transformer, as described at this point, is a bag of features model, ignoring the sequential structure in input $x \in \mathbb{R}^{n\times d}$. The approach followed to model positions in a transformer is positional encodings based on sinusoidal position embeddings $p \in \mathbb{R}^{n\times d}$ (Equation 2.18). Authors Vaswani et al. claim that the sinusoidal representation works as well as a learned one and that it generalizes better to sequences longer than the ones processed during training [99].

$$p_{i,2j} = sin(\frac{i}{10000^{\frac{2j}{d}}}), \quad p_{i,2j+1} = cos(\frac{i}{10000^{\frac{2j}{d}}}) \tag{2.18}$$

As mentioned before, the transformer was initially proposed for a language translation task, composed of an encoder-decoder architecture. Each encoder consists of several encoder transformer blocks. Each encoder block processes encodings from the previous block and weighs their relevance to each other to generate output encodings; these output encodings are then passed to the next encoder as its input, as well as to the decoders. The first encoder takes positional information and embeddings of the input sequence as its input. Each decoder consists of several decoder transformer blocks. The decoder adds an additional attention mechanism that extracts information from the same level encoding block, this is called the encoder-decoder attention. Like the first encoder, the first decoder takes positional information and embeddings as its input. Also, it is important to mention that in the decoder, the sequence must be masked to prevent information flow from following objects.

The hyper-parameters of the transformer are: d (input embedding size), m (fully-connected layer size), s (encoding size), H (number of heads), and L (number of layers). In our experiments in Section 5.1.1 we use the setting $d = 128$, $m = 256$, $H = 4$, and $L = 2$. We selected this parameter configuration to fit our available hardware capabilities.

2.3 Modeling approaches

In this section, we describe the two modeling approaches we follow in this thesis, IR and LM. First, we provide a context for what is the purpose of retrieval models and contextualize our proposed model within this context. Later, we describe the goal of language models and how our proposed model fits within the language modeling task.

2.3.1 Retrieval models

Information retrieval [62] in computer science and information science is the process of obtaining information from sources related to information needs. An information retrieval process begins with a user entering a query into the system. A query is a formal statement of a need for information, such as a search string in a web search engine. A query does not uniquely identify a single item in a collection; alternatively, multiple objects can match the query, possibly with varying degrees of relevance, so the results are usually ordered. Most IR systems calculate a numerical score for how well each object in the database matches the query, and rank objects based on that value.

In our case, the collection of resources is a set of spatio-temporal annotated text records. The query can be any combination of $\langle time, location, text \rangle$, returning a ranking of the specified needed type of information as output.

2.3.2 Language models

Language models [36, 50] solve the task of language modeling, which is defined as the task of assigning a probability to a sequence of words $\mathbf{w}$: $p(\mathbf{w}) = p(w_0, \ldots, w_j)$. State-of-the-art models[2] for language modeling are based on neural networks. Typically, neural network language models are constructed and trained as discriminative predictive models that learn to predict a probability distribution $p(w_j / w_0, \ldots, w_{j-1})$ for a given word conditioned on the previous words in the sequence. These models are trained on a given corpus of documents. The probability of a sequence of words $p(w_0, \ldots, w_j)$ can be estimated with: $\prod_{i=1}^{i=j} p(w_i / w_0, \ldots, w_{i-1})$.

Conditioned language modeling is defined as the task of assigning a probability to a sequence of words given a context c: $p(\mathbf{w}/c) = p((w_0, \ldots, w_j)/c)$. Then, the probability of each word in the sequence is computed as $p(w_j / c, w_0, \ldots, w_{j-1})$. Conditioned language models have applications in multiple natural language processing tasks, for example, machine translation (generating text in a target language conditioned on text in a source language), description of an image conditioned on the image, a summary conditioned on a text, an answer conditioned on a question and a document, etc.

In our case, the language model that we aim to develop is an instance of a conditioned language model. The context that we will consider as the source for the language generation will be a tuple of a timestamp and geo-coordinates.

[2] http://nlpprogress.com/english/language_modeling.html

2.4 Evaluation

This section describes the evaluation measures used to validate the presented models. First, we describe the ranking evaluation measure to evaluate the multi-modal retrieval model: Mean Reciprocal Rank (MRR), and then the language modeling measure: Perplexity. We also discuss the evaluation procedure considering the temporal component.

2.4.1 Evaluation of rankings

We evaluate the retrieval models assuming one of the items $\langle time \rangle$, $\langle location \rangle$, and $\langle text \rangle$ is missing and use the other two to retrieve the missing one. We use the retrieval model to build a ranking with the missing item, and k negative examples randomly selected from the dataset. We use Mean Reciprocal Rank to evaluate the rankings produced. We expect that the model ranks the real missing item better. Given a set of queries Q, where each query is the two known items, MRR is defined as:

$$MRR = \left(\frac{\sum_{i=1}^{|Q|} \frac{1}{r_i}}{|Q|}\right) \qquad r_i : \text{ ranking of the real missing item } i \tag{2.19}$$

2.4.2 Evaluation of language models

Evaluation of language models can be performed intrinsically or extrinsically. In extrinsic evaluation, the evaluation is performed by measuring how a language model improves other tasks. Language models have been used to solve external tasks like speech recognition, machine translation, optical character recognition, handwriting recognition, and others. The other approach for language model evaluation is using an internal measure. This is the traditional approach followed to evaluate language models and the one followed in this thesis. Intrinsic evaluation of language modeling is usually done using Perplexity [16]. Perplexity measures how well a language model predicts a test sample; it captures how many bits are needed on average per word to represent the test sample. It is important to note that in Perplexity, the lower the score, the better the model. Perplexity, for a test set where all sentences are arranged one after the other in a sequence of words $w_1, \ldots, w_T$ of length T, is defined as:

$$Perplexity = 2^{-\frac{1}{T}\log_2 p(w_1,\ldots,w_T)} \tag{2.20}$$

2.4.3 Evaluation procedure

When using a machine learning model, there are two types of analyses that can be done when considering the temporal component, a retrospective analysis or a predictive analysis:

- in a retrospective analysis, the goal is to study what happened in the past with the help of a model to discover relevant patterns found in a dataset. An example of a retrospective analysis is using a model to discover what happened in Chile during the 2021 presidential elections from a social media perspective by building a model with a dataset covering the time period and place of interest,
- in a predictive analysis, the goal is to predict the future from what happened in the past. An example of predictive analysis is time series forecasting, where the goal is to study patterns of the value that a variable takes over time to predict the values it will take in the future.

Considering model validation and the time dimension, for a predictive analysis the correct approach is to split the dataset by the temporal variable, but for retrospective analysis, it may not be the correct approach. For example, to discover what happened in Chile during the 2021 presidential elections, it would be desirable that the model is tested with texts, times, and places related to the event of interest, the Chilean presidential elections in 2021.

Both models proposed in this thesis, the multi-modal retrieval model and the spatio-temporal conditioned language model can be used for both types of analysis, retrospective and predictive. Nevertheless, the practical applications presented in Sections 4.2.4 and 5.2.4 are examples of retrospective analysis. After our literature review, one issue that we found is the lack of a standard evaluation setting (see Section 3.2) when modeling spatio-temporal annotated text data. Previous works evaluate their proposal on their own dataset, with their own evaluation procedure, and their particular evaluation metric. We decided not to make this situation worse by developing a new evaluation procedure, hence our evaluation setting is similar to previous works [119] where datasets are randomly split for model validation without special consideration for time or place dimensions.

2.5 Datasets description

We conduct experiments over two types of data sources, *social media user posts*, and *official crime incident reports*. Social media datasets from Twitter and Foursquare are used to validate both models. In the retrieval model, the social media datasets are used for quantitative comparison between our approach and previous works. Crime reports, on the other hand, are included to add diversity to our analysis of applications of the model. We conduct quantitative and qualitative evaluations over the two social media datasets for the language model. Table 2.1 shows a summary of these datasets. We next describe each dataset.

Twitter dataset

This dataset ('LA-TW') was presented by Zhang et al. [119] and corresponds to Twitter messages collected from Los Angeles, USA. This dataset consists of 1,584,307 geotagged tweets (short-text messages) covering the period of time from 2014.08.01 to 2014.11.30.

Foursquare dataset

This dataset ('NY-FS') was also presented by Zhang et al. [119] and consists of Foursquare check-ins reported on Twitter by users in the city of New York, USA. The data contains 479,297 records indicating places in the city that were visited by users and their location for the period from 2010.02.25 to 2012.08.16.

Crime incident dataset

This dataset ('NY-Crime') contains crime reports from the city of New York, USA[3]. It was obtained from the New York City Open Data repository[4]. The dataset contains textual descriptions used by police agents to classify crime incidents along with their geolocation. This dataset consists of 1,016,008 crime incident records that cover the dates starting from 2000.01.01 to 2015.12.31.

Table 2.1: Spatio-temporal textual datasets.

	Records	City	Start Date	End Date
LA-TW	1,584,307	Los Angeles	2014.08.01	2014.11.30
NY-FS	479,297	New York	2010.02.25	2012.08.16
NY-Crime	1,016,008	New York	2000.01.01	2015.12.31

2.6 Text-space-time patterns exploration

In this section; first, we explore how text correlates with time and space; and second, how the examples in the social media datasets distribute over time and space. For both exploratory analyses, timestamps and geo-coordinates are discretized. To cover the two discretization approaches followed in this thesis, for the correlation analysis,

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

[4] https://opendata.cityofnewyork.us/

we use a density-based automatic discretization approach (see Section 4.1.1), while for the distribution analysis, we use hand-crafted discretizations (see Section 5.1.1).

2.6.1 Words relation to time and space

The first step for this exploratory analysis is to discretize timestamps and geo-coordinates. Timestamps are converted to numbers in the range [0-86,400][5] by calculating their offset in seconds with respect to 12:00 am, while geo-coordinates are represented in the 2-D space of latitudes and longitudes. 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 [119]). Each timestamp belongs to a temporal window and each geo-coordinate belongs to a spatial cell (column *Cells* in Table 2.2). We computed the average entropy of the distribution of words for temporal windows and spatial cells for both social media datasets from Twitter and Foursquare (see Section 2.5). In Table 2.2, we see that for both datasets the entropy of word distribution over spatial cells covers a lower percentage of the maximum entropy than the entropy of the word distribution over temporal windows. This means that words are more correlated with spatial cells than with temporal windows, and is an indication that capturing patterns of text in relation to locations should be easier than capturing patterns of text in relation to times.

Table 2.2: Number of spatial cells and temporal windows. Average, maximum entropy and percent of the maximum entropy of words distribution over spatial cells and temporal windows for datasets LA-TW and NY-FS.

Dataset	Dimension	Cells	Ave Entropy	Maximum Entropy	% Max Entropy
LA-TW	Time	24	4.50	4.90	0.91
	Location	5297	6.76	13.31	**0.50**
NY-FS	Time	29	4.01	4.64	0.86
	Location	10159	4.23	12.37	**0.34**

2.6.2 Distribution of the number of tweets over timestamps and geo-coordinates discretizations

In order to better understand how the examples in the datasets distribute over time and space, as well as the spatio-temporal discretizations, we show in Figures 2.3 and 2.4 the distribution of the number of tweets over the 24 hours of the day (0-23) and the discretization of geo-coordinates by (0.001 × 0.001) spatial cells.

We can observe that, for both datasets, early morning hours are the least frequent, starting to increase in the afternoon until the night hours. In total, there are

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

19,157 spatial cells for the NY-FS dataset and 84,693 for the LA-TW dataset. In the case of the NY-FS dataset, around 82% (15,796) of the cells have less than the mean number of texts per cell (dotted line in Figure 2.3). For the LA-TW dataset, the distribution is similar; around 83% (70,529) of the cells have less than the mean number of examples per cell (dotted line in Figure 2.3). These similarities in the patterns observed in the distributions indicate that even when these datasets were collected from different cities and on different time windows, there are patterns for text generation under spatio-temporal contexts that prevail independently of the place and time window within which the data was collected.

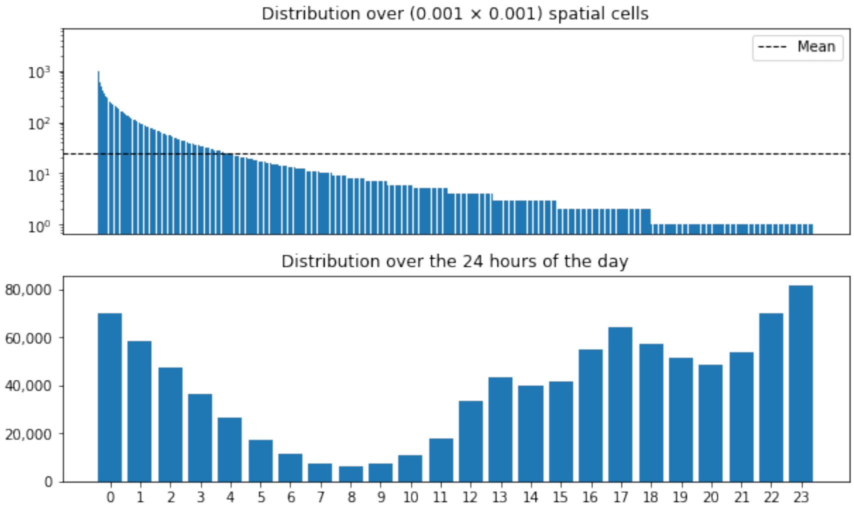


Figure 2.3: Distribution of the number of tweets for the NY-FS dataset.

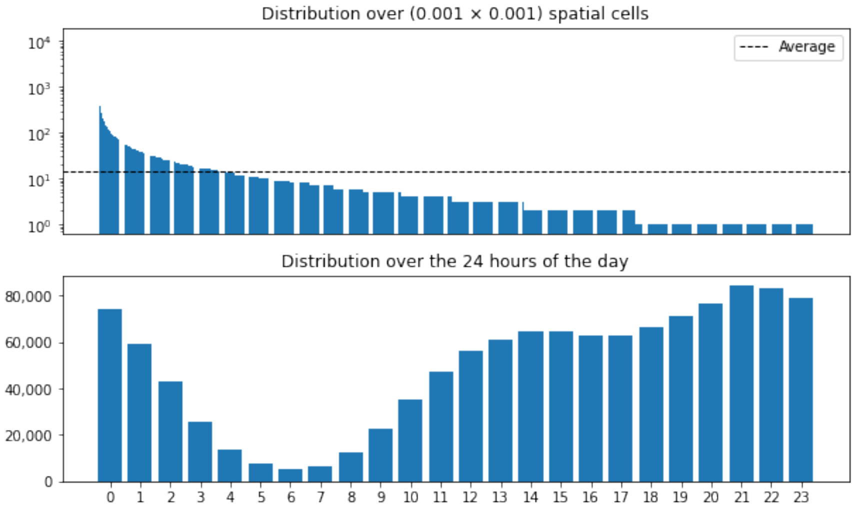


Figure 2.4: Distribution of the number of tweets for the LA-TW dataset.

Chapter 3

Related works

In this section, we provide an overview of the work in the literature related to this thesis. First, we describe the principal applications of spatio-temporal text data on the main source of this type of content: location-based social networks. Later, we delve into the models for spatio-temporal text data, derived from the applications mentioned before. These works study how text, time, and space are jointly modeled and how time and space can be used as context for language generation.

3.1 Applications for spatio-temporal text data

As stated in previous sections, many sources of text data have spatio-temporal dimensions. Nevertheless, most of the work in the literature focuses on the social network domain. It is the most abundant data source and easiest to acquire using APIs. The main applications [118] in the literature are mobility modeling [70, 7, 121], event detection [83, 84, 71, 75, 101] , event forecasting [118, 103, 33, 19, 123], and activity modeling [64, 27, 100, 90, 47, 3, 52]. Next, we describe these applications.

3.1.1 Mobility modeling

Mobility modeling [118, 70, 7, 121] using spatio-temporal text data allows us to know not only the geometric aspects of human mobility data but also the semantics: i.e., going from point A at time t_0 to point B at time t_1 is not as informative as going from *home* at time t_0 to *work* at time t_1 or from *work* at time t_2 to *a restaurant* at time t_3. Studying human mobility patterns has applications like place prediction/recommendation [70, 7] for individual users and trajectory pattern mining for mobility understanding in urban areas [121].

Works in mobility modeling have focused mainly on two problems: next-place prediction/recommendation and trajectory modeling. A trajectory is a sequence

$T = (p_1, \ldots, p_n)$, of places a user has visited with time and space constraints to guarantee semantic consistency. If the time or geographic distance between two consecutive places p_i, p_{i+1} is too big, then they should not belong to the same trajectory. Knowing the sequence of places $(p_1, \ldots, p_{n-1})$ a user u has visited, the objective of next place prediction/recommendation [118, 70, 7] is to select p_n as the next place u will visit from a set of candidates $C = \{c_1, \ldots, c_m\}$. Trajectory modeling [118, 121] studies trajectories at individual and global levels. At an individual level, it studies specific trajectory patterns for individual users; at a global level, it studies aggregated mobility patterns. This information can lead to grasping the reasons that motivate people's mobility behaviors, understanding the nuances of mobility problems in urban environments, and then taking effective actions to solve them.

3.1.2 Event detection

Event detection methods [118, 83, 84, 71, 75, 101] applied to the streaming of spatio-temporal text data from location-based social networks allow us to detect; in real-time, geo-localized events from first-hand reporters. As defined by Allan et al. [4], an event is something that happens at a specific time and place and impacts people's lives, e.g., protests, disasters, sports games, or concerts. Some events reflected in location-based social networks and can be detected are earthquakes [83, 84, 71], or traffic congestion [75, 101].

Event detection techniques can be classified into two approaches: document-pivot [94, 31] and feature-pivot [2]. In the document pivot approach, events are represented as clusters of documents. The feature-pivot approach detects bursty terms, which are then clustered to form events. A common approach is to use spatial information to localize the unusual activity, while temporal information is used to detect it. Once the data is localized by the spatial variable, the text is used as the clustering unit, either document-based for the document-pivot approach or term-based for the feature-pivot approach.

3.1.3 Event forecasting

Event forecasting methods [118, 103, 33, 19, 123], unlike event detection, which typically discovers events while they are occurring, predict the incidence of events in the future. The common approach is using social network data in conjunction with external sources to build prediction models. For some events like criminal incidents [103, 33, 19] or civil unrest [123], predicting the exact location as far in advance as possible is paramount.

A common approach is to define features as indicators and train prediction models for spatial regions. For civil unrest, the prediction is usually at the city level or smaller administrative regions, while for crimes and traffic events, the prediction is at a finer-grained level like neighborhoods or blocks. The temporal variable is used to identify the changing patterns that indicate the occurrence of an event in the future.

3.1.4 Activity modeling

Activity modeling studies human activities in urban environments using spatio-temporal text data related to human activities. As people share information about activities they do in everyday life, spatio-temporal text data from social networks provides useful information about spatial and temporal patterns of human activities. Unlike static analysis of spatial data, spatio-temporal text data can discover the purpose of a visit to a point of interest that hosts multiple kinds of events. For instance, the STAPLES Center, a multi-purpose arena in Los Angeles, California, holds sports events such as basketball matches but also can hold others, such as concerts. People may visit the STAPLES Center for different purposes. Using "STAPLES Center" to annotate a location record could fail to reveal the complete purpose of the location.

Works in activity modeling focus on place labeling and models that jointly represent text, time, and space. Both approaches characterize urban areas using data collected from location-based social networks. Given a set $R = \{r_1, ..., r_m\}$ of spatio-temporal text data records, place labeling finds labels that best describe points of interest, either static or at different time periods. Models that combine text, time, and space in a joint representation are the closest to the subject of this thesis. Next, we provide an in-depth description and analysis of these works that jointly represent text, time, and space for activity modeling.

Activity modeling: jointly modeling text, time, and space

Analyzing the former applications, the joint modeling of text, time, and space in activity modeling [118] can be considered the basic task. It allows to answer $\langle what \rangle$ happens, $\langle when \rangle$ it happens, and $\langle where \rangle$ it happens, and the remaining applications can benefit from an activity modeling model. For example, spatial and temporal activity patterns can be used to define transition points in trajectories for mobility models, spatial and temporal activity patterns are used as features for event forecasting models, and unusual localized bursty activity is used to detect events. Next, we focus on specialized models for activity modeling; first, we describe models that detect geographical topics, and after that, we describe feature embedding models for spatio-temporal text data.

Spatio-temporal topic modeling discovers topics related to geographical areas [64, 27, 100, 90, 47, 3, 52]. Mei et al. [64] proposed a generalization of the Probabilistic Latent Semantic Indexing [45] model, where topics can be generated by *text* or by the combination of *timestamp* and *location*. Eisenstein et al. [27] proposed a cascading topic modeling. Words are generated by a multinomial distribution that is the mean of a latent topic model and a region topic model. Regions are latent variables that also generate coordinates. Topics are generated by a Dirichlet distribution, regions are generated by a multinomial distribution, and coordinates are generated by a bivariate Gaussian distribution. Each region has a multinomial distribution over topics, and each topic has a multinomial distribution over keywords. Wang et al. proposed LATM [100], which is an extension of Latent Dirichlet Allocation [11], 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 the location, in both cases with a multinomial distribution depending on a topic generated by a Dirichlet distribution. Additionally, Sizov et al. [90] developed a model similar to the work of Wang et al. [100]. Rather than using a multinomial distribution to generate locations, they replace it with two Gaussian distributions that generate latitudes and longitudes. Yin et al. [113] studied a generative model where there are latent regions that are geographically distributed by a Gaussian. Hong et al. [47] use a base language model, a region-dependent language model, and a topic language model. Geo-coordinates are discretized into regions using clustering algorithms. Regions are generated by a multinomial distribution depending on the user and a global region distribution. Geo-coordinates are generated by the regions using multivariate Gaussian distributions. Words are generated by topics depending on the global topic distribution, the user, and the region. Ahmed et al. [3] developed a hierarchical topic model which models document and region-specific topic distributions and also models regional variations of topics. Relations between the Gaussian distributed geographical regions are modeled by assuming a strict hierarchical relation between regions that are learned during inference. Finally, Kling et al. proposed MGTM [52], a model based on multi-Dirichlet processes. The authors used a three-level hierarchical Dirichlet process with a Fischer distribution for detecting geographical clusters, a Dirichlet-multinomial document-topic distribution, and a Dirichlet-multinomial topic-word distribution.

Feature embedding methods find distributed learned representations for discrete variables. Learned embedded representations are very popular in natural language processing [65, 76] and graph node representation [49]. For spatio-temporal textual data, embedded representations learn a joint representation for the elements of the tuple $\langle time, location, text \rangle$. Zhang et al. proposed CrossMap [119]. In CrossMap, the first step is to discretize timestamps and coordinates using Kernel Density Estimation techniques. After that, CrossMap uses two different strategies to learn the embedded representations: Recon and Graph. In Recon, the problem is modeled as a relation reconstruction task between the elements of the tuple $\langle time, location, text \rangle$, while in Graph; the goal is to learn representations such that the structure of a graph built from the tuples $\langle time, location, text \rangle$ is preserved. Zhang et al. extended CrossMap [120] to learn the embedded representation in a stream. The authors propose two strategies based on life-decay learning and constrained learning to find representations from the streaming data. Unlike Crossmap, timestamps and geo-coordinates are discretized into hand-crafted spatial windows and temporal cells instead of Kernel Density Estimation. Zhang et al. proposed another extension

[117] to Crossmap to learn representations from multiple sources. The main dataset is the set of tuples $\langle time, location, text \rangle$. Each dataset defines a graph, and the representations are learned to preserve the graph structure. Nodes representing the same entity are shared between the main graph and secondary graphs. During training, the learning process alternates between learning the main graph's embeddings and learning the embeddings for the secondary datasets.

3.2 Discussion

In Table 3.1, we present a summary of works that study the joint representation of text, time, and space for activity modeling. Existing approaches are based on topic modeling or feature embedding. Works following the topic modeling approach are based on topic models such as Probabilistic Latent Semantic Analysis [10] or Latent Dirichlet Allocation [11] and extend these models by assigning distributions over locations to topics or by introducing latent geographical regions.

Both topic models and feature embedding models assume a bag-of-words approach for text modeling, ignoring the text's sequential structure. These models discretize coordinates and timestamps using clustering algorithms or hand-crafted spatial cells and temporal windows.

When considering time and space modeling, each work models timestamps and coordinates at a single level of granularity using hand-crafted discretizations or clustering algorithms. Only Ahmed et al. [3] model hierarchy, but only for space. To the best of our knowledge, there are no studies on how representing time and space at different levels of granularity impact the modeling of text generation under spatio-temporal conditions. Also, no works model the sequential structure of texts.

An additional problem with modeling spatio-temporal text data which is important to mention is the evaluation framework. On the one hand, there is no consensus about the evaluation metric. On the other hand, building a reference dataset in this field is complex. First, a temporal variable is involved: this means that data should be collected for a long time. Second, data is related to a specific region: this means that using models in a new region would require collecting data from that region.

Overall, we can conclude that existing approaches ignore two dimensions of the problem:

1. the sequential structure of language.
2. a unified model that leverages time and space at different granularities as context for language generation.

In the next two chapters, we present models to tackle the drawbacks of current approaches for modeling spatio-temporal annotated textual data presented in this

Table 3.1: Spatio-temporal text data modeling.

Work	Time Modeling	Space Modeling	Text Modeling	Time-Space-Text Integration	Evaluation Metric
[64]	Days in a week	City	Multinomial distribution	Topic modeling	-
[27]	-	User aggregation + Gaussian	Multinomial distribution	Topic modeling	Accuracy and Mean Distance
[90]	-	Two Gaussian	Multinomial	Topic modeling	Accuracy
[113]	-	Region + Gaussian	Multinomial	Topic modeling	Perplexity
[100]	-	Multinomial	Multinomial	Topic modeling	Perplexity
[47]	-	Clustering + Gaussian	Multinomial	Topic modeling	Mean Distance
[3]	-	Hierarchical Gaussian	Hierarchical multinomial	Hierarchical Topic modeling	Accuracy and Mean Distance
[52]	-	Fisher distribution	Multinomial	Multi-Dirichlet process	Perplexity
[119]	Clustering over seconds in a day	Clustering	Embedding	Multi-modal embedding	Mean Reciprocal Rank
[120]	Hours in a day	Equal-sized grids	Embedding	Online multi-modal embedding	Mean Reciprocal Rank
[117]	Hours in a day	Equal-sized grids	Embedding	Cross-modal embedding	Mean Reciprocal Rank

section. First, in Chapter 4, we present a multi-modal retrieval model which captures the sequential structure of language and allows us to query the model with pairs of elements of space, time, and text to retrieve the third one. Second, in Chapter 5, we present a spatio-temporal conditioned neural language model that allows us to represent time and space as a context for text generation at different granularities.

Chapter 4

Multi-modal retrieval model

As mentioned in Section 3, previous works that model the joint generation of text, time, and space use topic models [64, 27, 100, 90, 47, 3, 52] or feature embedding [119, 117, 120] to represent the elements of the tuple $\langle time, location, text\rangle$. These approaches discard the order of words, which can limit the expressiveness power of a model. In this section, we present a multi-modal retrieval model that follows an *Acceptor* RNN usage pattern. We refer to this model as STT-RNN. STT-RNN jointly represents $\langle time, location, text\rangle$ and captures the sequential structure of texts.

4.1 Model description

Figure 4.1 shows STT-RNN's architecture. As a first step, we build a text indexer and discretize timestamps and geo-coordinates (Equation 4.1). 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). After discretization, each word, temporal window, and spatial cell is assigned an index in a look-up embedding matrix shared by the three components. Each component occupies a segment of the embedding matrix (Equation 4.2). The embedding layer is similar to the embedding layer used for tokens (e.g. words) representation in natural language processing tasks, where each token is assigned an index between zero (0) and the size of the vocabulary of tokens ($vocab\_size$). In the case of timestamps and geo-coordinates, it is the same principle, just that the vocabulary instead of a set of words is a set of temporal windows or spatial cells. Each temporal window and spatial cell has an index in the respective vocabulary in the same way each word has an index in the vocabulary of words. Next, the *query-context* elements are concatenated as a sequence input to the GRU-RNN (Equations 4.3, 4.4, and 4.5). In this way, the RNN processes the input as a sequence of tokens formed by words, times, or locations.

$$\begin{gathered} IDTime = \text{DiscTime}(\langle timestamp \rangle) \\ IDPlace = \text{DiscCoordinates}(\langle latitude, longitude \rangle) \\ IDWord_1, \ldots, IDWord_s = \text{TextIndexer}(\langle text \rangle) \end{gathered} \tag{4.1}$$

$$\begin{gathered} EmbTime^{1,d} = \text{EmbLayer}(IDTime) \\ EmbPlace^{1,d} = \text{EmbLayer}(IDPlace) \\ EmbWord_1^{1,d}, \ldots, EmbWord_s^{1,d} = \text{EmbLayer}(IDWord_1, \ldots, IDWord_s) \end{gathered} \tag{4.2}$$

$$\begin{gathered} Context2Text^{2,d} = [EmbTime^{1,d}, EmbPlace^{1,d}] \\ EmbContext2Text^{1,d} = \text{RNN}(SeqContext^{2,d}) \end{gathered} \tag{4.3}$$

$$\begin{gathered} Context2Place^{s+1,d} = [EmbTime^{1,d}, EmbWord_1^{1,d}, \ldots, EmbWord_s^{1,d}] \\ EmbContext2Place^{1,d} = \text{RNN}(SeqContext^{n+p,d}) \end{gathered} \tag{4.4}$$

$$\begin{gathered} Context2Time^{s+1,d} = [EmbPlace^{1,d}, EmbWord_1^{1,d}, \ldots, EmbWord_s^{1,d}] \\ EmbContext2Time^{1,d} = \text{RNN}(SeqContext^{s+1,d}) \end{gathered} \tag{4.5}$$

$$\begin{gathered} PredictedWords^{s,vocab\_size} = \text{PredictorWord}(EmbContext2Text^{1,d}) \\ PredictedPlace^{1,locs\_size} = \text{PredictorPlace}(EmbContext2Place^{1,d}) \\ PredictedTime^{1,times\_size} = \text{PredictorTime}(EmbContext2Time^{1,d}) \end{gathered} \tag{4.6}$$

$$\begin{gathered} LossText = \text{CrossEntropy}(PredictedWords, Words) \\ LossPlace = \text{CrossEntropy}(PredictedPlace, Place) \\ LossTime = \text{CrossEntropy}(PredictedTime, Time) \\ Loss = LossText + LossPlace + LossTime \end{gathered} \tag{4.7}$$

The output of the GRU-RNN is passed as input to the *retrieval-predictor* component. The *retrieval-predictor* component is formed by three different fully connected layers to predict either *time*, *place*, or *words*; depending on the target. We call them *PredictorTime*, *PredictorLoc*, and *PredictorWord* (Equation 4.6). 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 retrieval task: words for *PredictorWord*, locations for *PredictorLoc*, and time for *PredictorTime*. The loss function is computed as the addition of the three losses (Equation 4.7). At retrieval time, the probability of a *text* passage is computed as the average probability of the predictions made by *PredictorWord* for each of its *words*.

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 the others and allows us to query the model with any combination of $\langle time, location, text\rangle$ and to retrieve any of the three variables as output. Any combination of $\langle time, location, text\rangle$ can be processed as an 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 retrieve any of the three variables.

Although the proposed model is evaluated following an information retrieval approach (see Section 2.4.1), the training is carried out in a classification environment. Evaluating the model in the form of information retrieval allows, firstly, to keep the evaluation process consistent with previous works [119] and, secondly, to provide a view of the model closer to practical applications, since information retrieval allows to perform tasks related to people's information needs.

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 worse or did not exhibit any improvement over our model while adding complexity in some cases. We tested using LSTM-RNN [43] instead of GRU-RNN and also experimented with attention mechanisms [6]. 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-RNN, our results were consistent with previous results [23]. Regarding the attention mechanism, since our sentences were truncated to 15 tokens for the social media datasets and 10 tokens for the crime incidents description dataset, we believe that the 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-RNN having also the GRU-RNN as the encoder. We got poor results with this approach when compared to generating each word independently.

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

4.1.1 Timestamps and geo-coordinates discretization

As mentioned before, texts, timestamps, and geo-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 geo-coordinates are continuous variables. To jointly model the three variables, the approach we follow is to discretize timestamps and geo-coordinates. We use two different discretization approaches: 1) density-based and 2) equal-width binning. The density-based approach was proposed by Zhang et al. [119], we use it to make our results comparable to

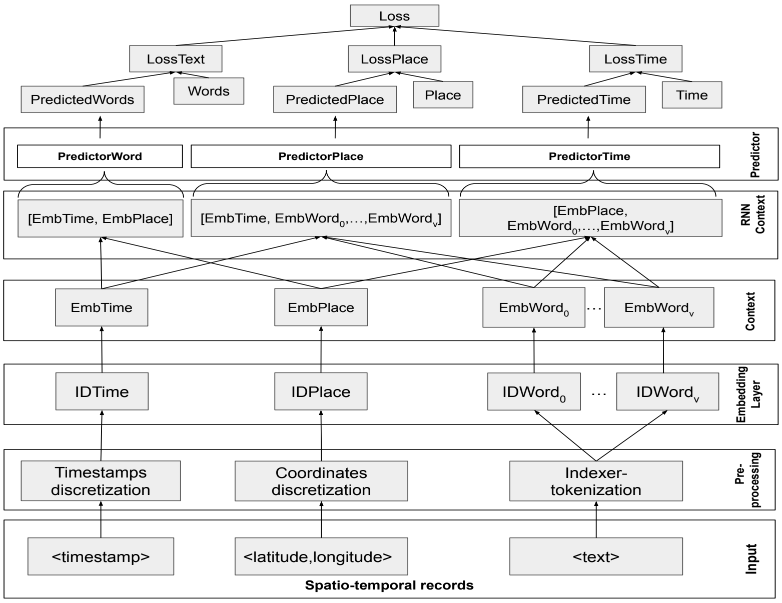


Figure 4.1: Multi-modal retrieval neural network architecture.

previous work. Timestamps are converted to numbers in the range [0-86,400][1] by calculating their offset in seconds with respect to 12:00 am. Then, a density-based automatic discretization technique is applied to both the transformed temporal variables and geo-coordinates. This leads to high-density temporal windows and spatial cells (for more details, refer to [119]).

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 impact of the discretization granularity on 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 4.1 shows an example of the discretization of a tweet.

[1] 86,400 is the number of seconds in a day.

Table 4.1: Example of discretization of a geo-tagged tweet using a one-hour time window size and a 0.02 spatial cell size.

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

4.1.2 Training, parameters, and evaluation

Algorithm 1 shows the training process for STT-RNN. First, we split the dataset into training (60%), validation (20%), and testing (20%). The model is trained for a number of *epochs* (30) using mini-batch gradient descent with Adam optimizer [51]. At each step, we batchify the dataset, and for each batch, we compute the loss as the addition of the losses (three) associated with each task. The goal is to train the model to take steps in the gradient descent toward solving the three tasks at the same time. After each epoch, we store the model's weights as well as the results of evaluating the model on the held-out validation set. The returned model is the one that performed best on the validation set across the various training steps.

We use 64-dimensional feature embedding representation for $timestamp$, $location$, and $words$. The GRU-RNN representation uses a single layer with a hidden layer size of 128 (see Section 2.2.2).

To evaluate the model, for each tuple in the test we want to retrieve an element, given the others of the tuple as a query. For each test example, we randomly selected $k = 10$ negative examples. We ranked the negative examples and the target according to the model using the query elements as input. We used the mean reciprocal rank (MRR) to evaluate the quality of the ranks produced by the model; MRR is described in Section 2.4.1.

It is worth mentioning that we chose this evaluation setting to keep the evaluation methodology consistent with the evaluation setting of previous works [119]. We used the same discretization techniques with the same parameters and the same methodology described in [119].

4.2 Experiments

Our experiments aim to answer our research question: can the joint representation of text, time (timestamp), and space (geographic coordinates), be better modeled; by capturing the sequential structure of texts? In that sense, we compare the proposed multi-modal retrieval model with previous approaches for spatio-temporal textual data modeling.

First, we describe the datasets used in our evaluation. After that, we present the results of comparing STT-RNN with previous approaches. We describe the

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.
Split H in Train(60%), Validation(20%), Test(20%).
$//TrainingStep$
Initialize Parameters $\theta$
Initialize $EPOCHS = 30$
for $epoch \in 1, 2, ..., EPOCHS$ do
    Batches = Batchify(Train)
    for $batch \in Batches$ do
        Loss = 0
        for $target \in \{t, l, e\}$ do
            $context$ is $\{t, l, e\}$ - $target$
            ForwardPass(batch(context, target))
            Loss = Loss + Loss(context, target)
        BackwardPass(batch)
        Update $\theta$ using the three batches of $\langle context, target \rangle$ and the
         optimization algorithm
    $EvalVal_{epoch}$ = ObjectiveFunction(Validation)
    Save $\theta$ and $EvalVal_{epoch}$ at this step.
Output trained model T with weights $\theta$ at step with best results over
the Validation set
```

baselines, the evaluation setting, and the experimental results. Then, we present a study of the sensitivity of the model to the granular representation of times and places. We conclude our analysis with a real-world crime description dataset case study.

4.2.1 Datasets

We evaluate the retrieval 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 the proposed model and previous works (using the same settings used in previous works). 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. All datasets are described in Section 2.5. Table 5.1 shows a summary of the datasets employed in this section.

4.2.2 Comparison to previous works

In this section, we show a comparison with previous works. First, we describe the baselines against which we compare the proposed model; after that, we describe the

Table 4.2: Datasets employed to evaluate the multi-modal retrieval model.

	Records	City	Start Date	End Date
LA-TW	1,584,307	Los Angeles	2014.08.01	2014.11.30
NY-FS	479,297	New York	2010.02.25	2012.08.16
NY-Crime	1,016,008	New York	2000.01.01	2015.12.31

evaluation methodology, and at the end, we present the experimental results.

Baselines

This section describes the baseline methods we use to validate our approach. As baselines, we use previous works for modeling spatio-temporal textual data based on feature embedding as well as existing approaches for geographic topic modeling. Next, we detail these baselines.

- **LGTA [113]** is a generative model where latent regions 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 [52]** is a generative model based on the 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 a Dirichlet-multinomial topic-word distribution.
- **SVD** performs Singular Value Decomposition on the co-occurrence matrix of *timestamps*, *location*, and *words.*
- **Recon [119]** 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 [119]** 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.

Results

In Table 4.3, we show the results for the social media datasets. We can see that STT-RNN outperformed all the models for location retrieval and time retrieval. This is consistent with our idea that RNNs will produce a better representation than the average of word embeddings for texts. Since *text* is only considered as input for *location* retrieval and *time* retrieval, these are the tasks that STT-RNN performed the best. Consistent with previous works, it showed better results for NY-FS (Foursquare) than for LA-TW (Twitter). Also, *time* prediction proved to be the hardest task. These results validate our exploratory analysis in Section 2.6.1, where we discovered that words are more correlated to spatial cells than to temporal windows.

Table 4.3: Mean Reciprocal Rank for spatio-temporal textual data modeling. The three tasks evaluated are to retrieve each one of the elements of the tuple $\langle time, location, text \rangle$ knowing the other two. This table shows results for the social media datasets of STT-RNN and baseline methods.

	Text		Location		Time	
Method	LA-TW	NY-FS	LA-TW	NY-FS	LA-TW	NY-FS
LGTA	0.3760	0.6107	0.3792	0.6083	-	-
MGTM	0.3874	0.5974	0.4474	0.5753	-	-
SVD	0.4475	0.4475	0.3953	0.6460	0.3256	0.3187
STT-RNN	0.4947	0.7227	**0.7175**	**0.9547**	**0.3939**	**0.4505**
Recon	0.6870	0.9219	0.6526	0.9044	0.3582	0.3612
Graph	**0.7011**	**0.9449**	0.6758	0.9168	0.3895	0.3716

In the case of *text* retrieval, the proposed model ranked third behind the two variants of the feature embedding models proposed by Zhang et al. [119]; though outperforming three of the previous works. A relevant aspect of the models proposed by Zhang et al. that can help to explain the results in Table 4.3 is that these works encode neighborhood relationships. These works capture neighborhood relationships by computing Gaussian kernel strengths between temporal windows or spatial cells. Later, these kernel strengths are introduced as weighting factors over related spatial cells and temporal windows in the feature embedding algorithms. Since modeling neighborhood properties allows for better generalization on the test dataset, this representation property, missing in our model, is an important aspect influencing the results of the previous feature embedding works when modeling the spatio-temporal context for text retrieval. Modeling the sequential structure of the text in our work is enough to overcome not modeling neighborhood properties in the case where text is part of the context representation (time retrieval and place retrieval), but for text retrieval, there is no sequential structure in the spatio-temporal context, hence the modeling of neighborhood relationships is a relevant factor in the spatio-temporal context that we lack to model. Later, in Section 5, we present a spatio-temporal conditioned language model where we address the weakness of the model presented in this section; text generation under spatio-temporal conditions. We focus on the modeling of spatio-temporal contexts for language generation by representing time and space at different granularities as contexts for language generation.

4.2.3 Sensitivity analysis of spatial and temporal granularities

In this section, we show how the spatial and temporal granularities influence 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 4.4, we show results while 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 24×7=168 hour ranges. In our experiments, we used temporal window sizes 1,2,4,8,12, and 24 (see Section 4.1.1). In Table 4.4, we can see that location retrieval is not affected by changes in the temporal variable, while text retrieval shows a small drop. For time prediction, the clear tendency is to decrease the MRR while increasing the window 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 has a low influence on the prediction of places and texts.

Table 4.4: Mean Reciprocal Rank for spatio-temporal textual data modeling. The three tasks evaluated are to retrieve each one of the elements of the tuple $\langle time, location, text \rangle$ knowing the other two. This table shows how STT-RNN performs while changing the temporal window, here we evaluate using the Foursquare dataset.

	Text	Location	Time
Window-Size	NY-FS	NY-FS	NY-FS
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	0.9530	0.2966

Spatial-granularity analysis. In Table 4.5, we show the results by changing the spatial cell size. We use squared equal-size spatial cells by manipulating the continuous values representing the latitudes and longitudes (see Section 4.1.1). We experimented with cells size 0.01, 0.02, 0.03, 0.04, 0.05, and 0.06. Retrieving 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 retrieval when expanding the temporal window, expanding the spatial cell makes it harder to retrieve the spatial cell correctly. Also, expanding the spatial cell makes the task of text retrieval harder, confirming a strong correlation between places and texts.

4.2.4 Qualitative 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-

Table 4.5: Mean Reciprocal Rank for spatio-temporal textual data modeling. The three tasks evaluated are to retrieve each one of the elements of the tuple $\langle time, location, text \rangle$ knowing the other two. This table shows how STT-RNN performs while changing the spatial granularity. Here we evaluate using the Foursquare dataset.

	Text	Location	Time
Cell-Size	NY-FS	NY-FS	NY-FS
0.01	0.6373	0.9524	0.4489
0.02	0.5612	0.9410	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

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 Section 2.5) which contains texts used by police agents to describe the incident, timestamps of *when* the crime took place and geo-coordinates of *where.*

First, we compared STT-RNN to previous work [119] following the same methodology described in Section 4.2.2. Similar to the results using the social media datasets, in Table 4.6 we can see that STT-RNN shows the best results for retrieving times and places. Given that STT-RNN is at its best for retrieving places and times, in Table 4.7 and Table 4.8 we show results querying STT-RNN when trained with the crime dataset. As mentioned in Section 4.1, we could query the model with any of the sequences that the model is trained to represent ⟨time, loc⟩, ⟨time, text⟩ and ⟨loc, text⟩, but also with sequences like just text ⟨text⟩, or just a token of time ⟨time⟩ or a token of location ⟨loc⟩ and ask the model to retrieve any of the three variables. 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"*; and 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.

Table 4.6: Mean Reciprocal Rank for spatio-temporal textual data modeling. This table shows results for the crime incident dataset of STT-RNN and the previous work Graph.

	Text	Location	Time
Method	NY-Crime	NY-Crime	NY-Crime
Graph	**0.370**	0.385	0.319
STT-RNN	0.311	**0.559**	**0.368**

Table 4.7: Spatial and temporal results for textual queries. **Query="alcoholic beverage control law"**.

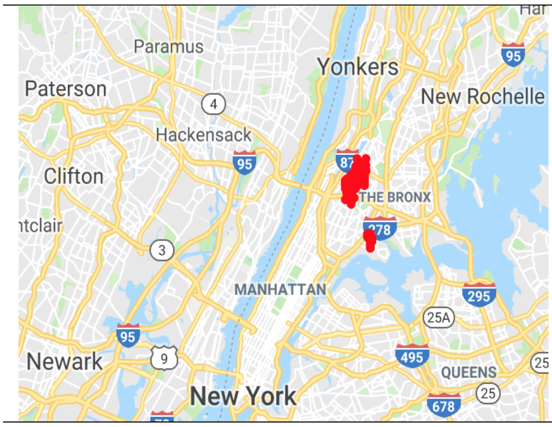
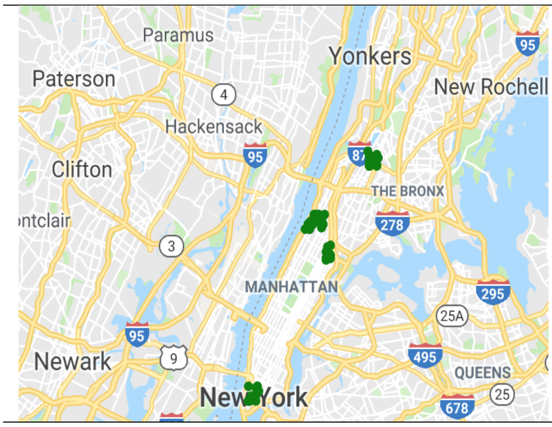
Coordinates	Time-Day	Time-Hour
	Friday	11pm
	Sunday	1am
	Saturday	1am
	Sunday	12am
	Thursday	11pm
	Saturday	10pm
	Saturday	12am
	Tuesday	10pm
	Wednesday	1am
	Thursday	1am

Table 4.8: Spatial and temporal results for textual queries. **Query="state laws non penal"**.

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

4.3 Chapter conclusions

In this chapter, we presented a multi-modal retrieval model for spatio-temporal textual data. We described the model and the results of the quantitative and qualitative experiments that we conducted. The proposed retrieval model outperformed previous works in our experiments in two of the three evaluated tasks and ranked third for the other task. Our qualitative experiments proved how the proposed model can be used in a crime-incident domain to gain insights into spatio-temporal patterns of crime incidents from their description and the information about *when* and *where* those crimes took place.

Of the three retrieval tasks, retrieving text given time and location was the task where the model performed the worst in comparison to the other methods. 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. Also, representing time and space with one level of granularity ignores properties of time and space like neighborhood and hierarchy (see Section 2.1). Motivated by this, we present additional studies in the next chapter on how text is generated under spatio-temporal conditions.

Chapter 5

Spatio-temporal conditioned language model

As mentioned in Section 3, ignoring properties of space and time like neighborhood and hierarchy can limit the capacity of a model for pattern learning in the joint representation of text, time, and space. Capturing such properties in the modeling of the spatio-temporal context should have a positive influence on spatio-temporal conditioned language models. In this section, we present an end-to-end neural network for encoding spatial and temporal contexts and decoding/generating text; we refer to this model as STT-LM. The neural network architecture design aims to model the spatio-temporal context at different granularities and to make the decoding/generating component agnostic to how the encoding of the spatial and temporal contexts are instantiated.

5.1 Model description

Figure 5.1 shows STT-LM's architecture. In order to feed the model with spatio-temporal textual data, some pre-processing steps are required: first, the text is tokenized, timestamps are discretized into temporal windows, and geo-coordinates are discretized into spatial cells (Equation 5.1). We set the vocabulary to the 12,288 most common words in the training set. The number of spatial cells and temporal windows is variable depending on the experiment. We filter out tuples where the number of words in the vocabulary is ten or less and reduce all URLs to the token '*http*'. After pre-processing, discretized timestamps and discretized geo-coordinates are passed through embedding layers (Equation 5.2). The embedding layer projects words, temporal windows, and spatial cells into a dense representation. Each item is embedded using a look-up table and there is a look-up table for each type of item: *temporal windows*, *spatial cells*, and *words*. Each item is associated with an integer that is used as an index in the correspondent look-up table.

After the discretization step, the next step is building the spatio-temporal con-

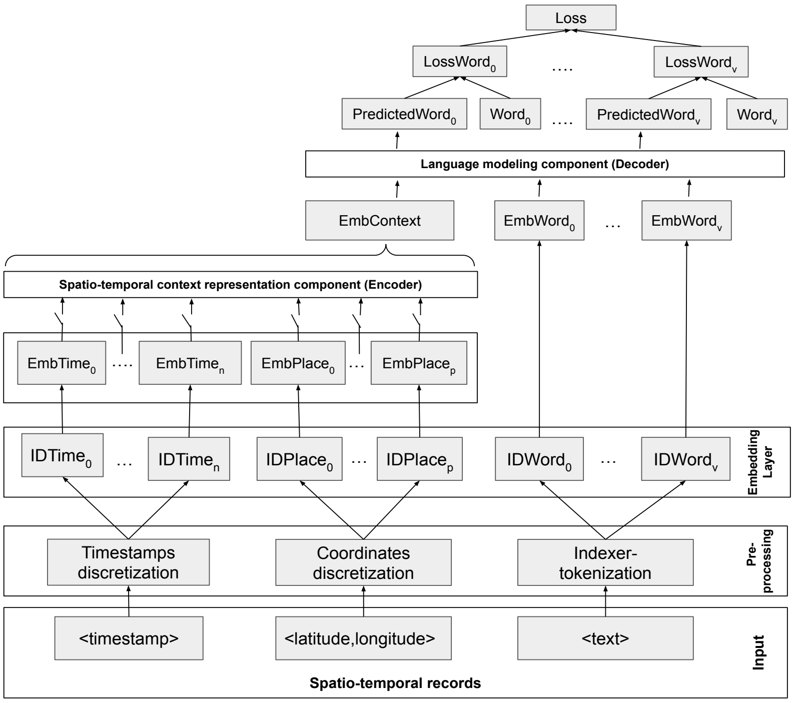


Figure 5.1: Spatio-temporal conditioned neural language model architecture.

text (Equation 5.3). Each timestamp can be discretized into n temporal windows and each coordinate can be discretized into p spatial cells. For each one of the $n+p$ temporal windows and spatial cells, there is a gate to select whether it is included in the context. Afterward, the context is passed through an encoder layer that results in a context-representation tensor (EmbContext). This context-representation tensor is of invariant/fixed dimensions ($\langle 1, d \rangle$ where d is the representation dimension) no matter how the context is selected. The EmbContext tensor is concatenated as the first element in the sequence of word embeddings (Equation 5.4); this sequence [EmbContext, EmbWords] is passed through a decoder that represents the language model. Finally, we compute the loss function as the cross-entropy between the predicted sequence of words and the observed sequence of words in the training examples (Equation 5.5). This is the general architecture that we propose. The main building blocks of our encoder-decoder architecture can be implemented using different approaches, such as recurrent neural networks or self-attention transformer blocks. We experiment with them in Section 5.2.2.

A salient property of this architecture is that it allows for representing time and space at different levels of granularity. This is achieved by modeling the spatio-temporal context as a sequence of discrete tokens representing each context type's particular semantics. For example, we could represent the temporal context by the hour of the day (0-23), day of the week (Sunday to Monday), week of the month, and month of the year (January to December), and the spatial context by block, neighborhood, district, etc.

$$\begin{gathered} IDTime_1, \ldots, IDTime_n = \text{DiscTime}(\langle timestamp \rangle) \\ IDPlace_1, \ldots, IDPlace_p = \text{DiscCoordinates}(\langle latitude, longitude \rangle) \\ IDWord_1, \ldots, IDWord_s = \text{TextIndexer}(\langle text \rangle) \end{gathered} \tag{5.1}$$

$$\begin{gathered} EmbTime_1^{1,d}, \ldots, EmbTime_n^{1,d} = \text{EmbLayer}(IDTime_1, \ldots, IDTime_n) \\ EmbPlace_1^{1,d}, \ldots, EmbPlace_p^{1,d} = \text{EmbLayer}(IDPlace_1, \ldots, IDPlace_p) \\ EmbWord_1^{1,d}, \ldots, EmbWord_p^{1,d} = \text{EmbLayer}(IDWord_1, \ldots, IDWord_s) \end{gathered} \tag{5.2}$$

$$\begin{gathered} SeqContext^{n+p,d} = [EmbTime_1^{1,d}, \ldots, EmbTime_n^{1,d}, EmbPlace_1^{1,d}, \ldots, EmbPlace_p^{1,d}] \\ EmbContext^{1,d} = \text{Encoder}(SeqContext^{n+p,d}) \end{gathered} \tag{5.3}$$

$$\begin{gathered} SeqPred^{s+1,d} = [EmbContext^{1,d}, EmbWord_1^{1,d}, \ldots, EmbWord_s^{1,d}] \\ PredictedWords^{s,vocabsize} = \text{Decoder}(SeqContext^{s+1,d}) \end{gathered} \tag{5.4}$$

$$Loss = \text{CrossEntropy}(PredictedWords^{s,vocabsize}, CorrectWords^{s,vocabsize}) \tag{5.5}$$

5.1.1 Timestamps and geo-coordinates discretization

To discretize geo-coordinates and timestamps we use equal-size squared cells in the case of the geo-coordinates and hand-crafted temporal windows in the case of the timestamps. For timestamp discretizations, we use human semantic arrangements of time, in particular: the hour of the day (0-23), day of the week (Sunday to Monday), week of the month (first week to the fifth week), and month of the year (January to December). Figure 5.2 shows a hierarchy describing these discretizations. For spatial discretization, we use equal-size spatial cells using the coordinates as metric units to define the spatial cells. Figure 5.3 shows a hierarchy describing the squared-cell discretizations.

It is important to remark that our approach to representing contexts as discrete sequences allows for working at different levels of granularity. For example, a coarse representation could represent time by a single token corresponding to the month, whereas a more fine-grained approach could encode time as a sequence containing month, day, hour, etc. We argue that this is a core property of our architecture as it allows us to adapt the spatio-temporal context representation depending on the application. For example, granularities at the hour level should be more efficient for events related to daily activities (e.g., going to work, and having lunch). On the other hand, for seasonal events (e.g., Christmas, Holidays) month-level granularities should work better.

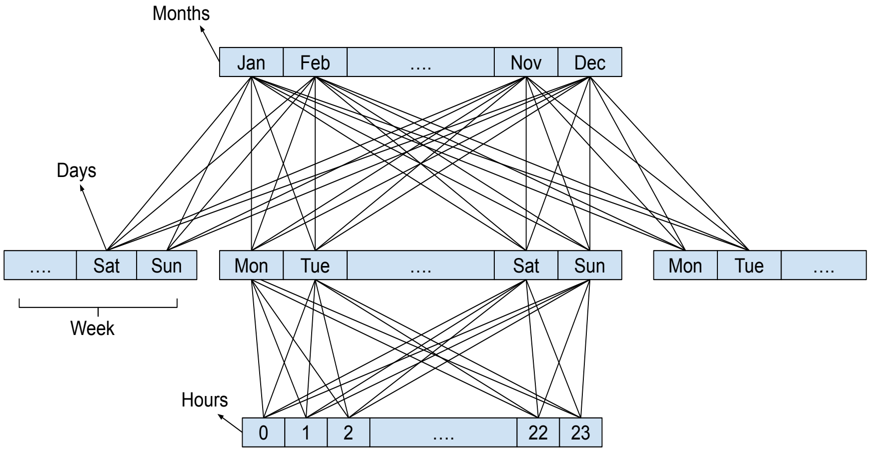


Figure 5.2: Hierarchy of timestamps discretization.

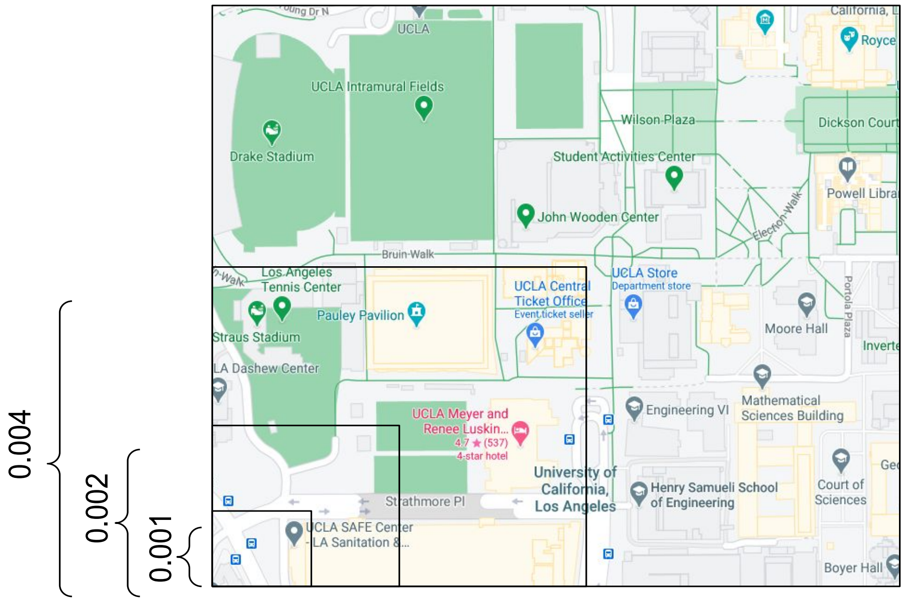


Figure 5.3: Hierarchy of coordinates discretization.

5.1.2 Training, parameters, and evaluation

Algorithm 2 shows the training process for STT-LM. First, we split the dataset into training-validation-testing, keeping 10% of each dataset for testing, 10% for validation, and 80% for training. The model is trained for E *epochs*, where E is computed following an early stopping approach. At each step, we batchify the dataset and train using mini-batch gradient descent with Adam optimizer [51]. After each epoch, we store the model's weights 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 (early stopping).

In all our experiments we use 128-dimensional feature embedding representation for *timestamp*, *location*, and *words* 128 examples as batch-size. We develop experiments with the encoder and the decoder self-attention architectures proposed in [99]. Also, we test a multi-layer GRU-RNN [23] for language modeling. The GRU-RNN has two layers with a hidden layer size of 128. The self-attention architectures are used in all cases with two self-attention layers, four heads, and 128 as vector sizes for queries, keys, and values (see Section 2.2.2 for additional details).

For our quantitative experiments, we use the traditional intrinsic evaluation metric for language modeling: Perplexity [16]. Perplexity measures how well a language model predicts a test sample, in our case: text generated given a spatio-temporal context. For additional details and the Perplexity formula, see Section 2.4.2.

Algorithm 2: Training algorithm for STT-LM

```
Input: Set H of spatio-temporal records tuples of the form $\langle t_i, l_i, e_i \rangle$.
Output: Trained model T.
Split H in Train(80%), Validation(10%), Test(10%).
$//TrainingStep$
Initialize Parameters $\theta$
Initialize $EarlyStop$ = False, $i = 0$
while not $EarlyStop$ do
    Batches = Batchify(Train)
    for $batch \in Batches$ do
        ForwardPass(batch)
        Loss(batch)
        BackwardPass(batch)
        Update $\theta$ using the optimization algorithm
    Save $\theta$ at this step
    $i = i + 1$
    $EvalVal_i$ = ObjectiveFunction(Validation)
    if $EvalVal_i$ is worse than $EvalVal_{i-1}$ then
        $EarlyStop$ = True
Output trained model T with weights $\theta$ at step with best results over
the Validation set
```

5.2 Experiments

Our experiments aim to answer our research question: can the joint representation of text, time (timestamp), and space (geographic coordinates), be better modeled by representing the spatio-temporal variables at multiple levels of granularity? In that sense, we conduct extensive experiments where we study how the modeling of time and space at different granularities influences the quality of the produced spatio-temporal conditioned language models.

Next, we describe the datasets used in our evaluation. We then present an analysis of different variants for the spatio-temporal context representation component (Encoder) and the language modeling component (Decoder). After that, we study how modeling time and space at different granularities influence the results of the spatio-temporal conditioned language model over the two social media datasets. We conclude the experiments with case studies where we show how the proposed model can be used to characterize urban places from the perspective of social media and how an attention-based neural network can be used to visualize relations between texts and the spatio-temporal context where it is generated.

5.2.1 Datasets

We conduct experiments over two social media datasets from Twitter and Foursquare. These datasets are described in Section 2.5. Both datasets are used in our quantitative analyses. Given the diversity and variety of language generated on Twitter compared to Foursquare, Los Angeles's Twitter dataset is used in our qualitative experiments. Table 5.1 shows a summary of the datasets employed in this section.

Table 5.1: Datasets employed to evaluate the spatio-temporal conditioned language model.

	Records	City	Start Date	End Date
LA-TW	1,584,307	Los Angeles	2014.08.01	2014.11.30
NY-FS	479,297	New York	2010.02.25	2012.08.16

5.2.2 Encoder-decoder analysis

In our first set of experiments, we evaluate different options for the spatio-temporal context representation component (Encoder) and the language modeling component (Decoder) (see Section 5.1). In each case, we test two variants. For the encoder, we test: 1) projecting the feature embedding output of the embedding layer with a fully-connected layer on top and 2) the self-attention encoder representation proposed in [99] (without the positional encoding since the order is irrelevant in the sequence of tokens representing the spatio-temporal context) also with a fully-connected layer on top. For the decoder, we test 1) a two-layer GRU-RNN [23] and 2) a transformer-based two-layer decoder representation proposed in [99].

In Table 5.2, we show the results for the dataset from Twitter, and in Table 5.3 for the dataset from Foursquare. For both datasets, we test two different options for times and places in the encoder: all times (alltimes), all places (allplaces), and all times-places (all). We can see that for both datasets and each option of times and places, using only the embeddings in the encoder performed better than using the self-attention component, while for the decoder, the self-attention component performed better than the GRU-RNN in the same analysis. The combination encoder(Embeddings)-decoder(Self-Attention) got the best results in all cases. Our interpretation of these results is that the self-attention mechanism in the spatio-temporal context introduces noise between the units in the spatio-temporal context; using only the embeddings keeps the representations of the spatio-temporal units independent from each other. In the case of the decoder, there is no such issue; what we are modeling is the sequential structure of the text, which can be captured with the self-attention decoder. In the next section, where we analyze different granularities for time and space, we use this setting for the encoder (Embeddings) and the decoder (Self-Attention) as the evaluation framework.

Table 5.2: Perplexity results for the Twitter dataset from Los Angeles. Testing only embeddings and self-attention for the encoder component and GRU-RNN or self-attention for the decoder. In the *Context* column: h means hour, d means day in the week, w means week in the month, and m means month in the year. Also: p1, p2, p4, and p8 mean squared cells of side: 0.001, 0.002, 0.004, and 0.008.

Context	Encoder	Decoder	Dataset	Perplexity
[]	-	GRU-RNN	LA-TW	63.03
[]	-	Self-Attn	LA-TW	57.35
[hdwm]-alltimes	Embeddings	GRU-RNN	LA-TW	61.90
[hdwm]-alltimes	Embeddings	Self-Attn	LA-TW	56.67
[hdwm]-alltimes	Self-Attn	GRU-RNN	LA-TW	63.02
[hdwm]-alltimes	Self-Attn	Self-Attn	LA-TW	193.77
[p1p2p4p8]-allplaces	Embeddings	GRU-RNN	LA-TW	61.13
[p1p2p4p8]-allplaces	Embeddings	Self-Attn	LA-TW	54.30
[p1p2p4p8]-allplaces	Self-Attn	GRU-RNN	LA-TW	62.42
[p1p2p4p8]-allplaces	Self-Attn	Self-Attn	LA-TW	161.14
[hdwm p1p2p4p8]-all	Embeddings	GRU-RNN	LA-TW	58.88
[hdwm p1p2p4p8]-all	Embeddings	Self-Attn	LA-TW	53.85
[hdwm p1p2p4p8]-all	Self-Attn	GRU-RNN	LA-TW	63.06
[hdwm p1p2p4p8]-all	Self-Attn	Self-Attn	LA-TW	72.80

5.2.3 Spatio-temporal granularities analysis

In this section, we study how modeling time and space at different granularities impacts the spatio-temporal conditioned language models. In Table 5.4, we show

Table 5.3: Perplexity results for the Foursquare dataset from New York. Testing only embeddings and self-attention for the encoder component and GRU-RNN or self-attention for the decoder. In the *Context* column: h means hour, d means day in the week, w means week in the month, and m means month in the year. Also: p1, p2, p4, and p8 mean squared cells of side: 0.001, 0.002, 0.004, 0.008.

Context	Encoder	Decoder	Dataset	Perplexity
[]	-	GRU-RNN	NY-FS	10.49
[]	-	Self-Attn	NY-FS	9.13
[hdwm]-alltimes	Embeddings	GRU-RNN	NY-FS	10.02
[hdwm]-alltimes	Embeddings	Self-Attn	NY-FS	9.00
[hdwm]-alltimes	Self-Attn	GRU-RNN	NY-FS	10.14
[hdwm]-alltimes	Self-Attn	Self-Attn	NY-FS	47.15
[p1p2p4p8]-allplaces	Embeddings	GRU-RNN	NY-FS	6.51
[p1p2p4p8]-allplaces	Embeddings	Self-Attn	NY-FS	5.45
[p1p2p4p8]-allplaces	Self-Attn	GRU-RNN	NY-FS	10.13
[p1p2p4p8]-allplaces	Self-Attn	Self-Attn	NY-FS	36.62
[hdwm p1p2p4p8]-all	Embeddings	GRU-RNN	NY-FS	6.38
[hdwm p1p2p4p8]-all	Embeddings	Self-Attn	NY-FS	5.34
[hdwm p1p2p4p8]-all	Self-Attn	GRU-RNN	NY-FS	10.14
[hdwm p1p2p4p8]-all	Self-Attn	Self-Attn	NY-FS	34.93

the results for the Twitter dataset from Los Angeles. We can see that for all cases, including a spatial or a temporal context, proved to be better than not including it at all (first row in the table). The temporal context improvements were also marginal compared to a language model that ignores the spatio-temporal context (first row in the table). The spatial contexts show notable improvements in all cases; the larger the spatial cell, the better the results.

As a complement to the results in Table 5.4, in Table 5.5, we show the results with bigger spatial cells. We can see that Perplexity gets worse instead of getting better results, indicating that the optimal is around 0.008 in geo-coordinates values for cell size.

In Table 5.6, we show the results for the Foursquare dataset from New York. The Perplexities for this dataset are lower than the Perplexities for the Twitter dataset from Los Angeles. This is due to most of the Foursquare reports being generic text generation suggested by the application. These texts differ in most cases on the checked-in place, while the Twitter dataset is mostly free texts. About the spatio-temporal modeling, we observe similar results to the Twitter dataset; in all cases, including the spatio-temporal context improves the Perplexity. The temporal contexts produce marginal improvements, while the spatial contexts show the biggest improvement margin. Contrary to the results over the Twitter dataset, with this dataset, smaller cell-size produced better results than the wider ones. We consider this is due to texts being correlated to places of interest where people report activities in Foursquare (restaurants and small businesses) with a fine granularity.

Table 5.4: Perplexity results for the Twitter dataset from Los Angeles. In this table, we show the results using squared cells as spatial discretizations.

Context	Cells	Dataset	Perplexity
[]	-	LA-TW	57.35
[h]-hour	24	LA-TW	57.07
[d]-day	7	LA-TW	57.17
[w]-week	5	LA-TW	57.13
[m]-month	12	LA-TW	56.95
[hdwm]-alltimes	48	LA-TW	56.67
[p1]-0.001	77,065	LA-TW	54.65
[p2]-0.002	34,284	LA-TW	52.91
[p4]-0.004	11,359	LA-TW	51.45
[p8]-0.008	3,283	LA-TW	51.30
[p1p2p4p8]-allplaces	125,992	LA-TW	54.30
[hdwm p1p2p4p8]-all	126,036	LA-TW	53.85

Table 5.5: Perplexity results for the Twitter dataset from Los Angeles. In this table, we show the results using bigger squared cells as spatial discretizations.

Context	Cells	Dataset	Perplexity
[]	-	LA-TW	57.35
[p]-0.016	1,253	LA-TW	52.39
[p]-0.024	460	LA-TW	52.81
[p]-0.032	197	LA-TW	53.32

As a complement to the results in Table 5.6, in Table 5.7, we show the results with smaller spatial cells. Perplexity gets lower as the cells get smaller, meaning the results improve. We could not continue the decrease the spatial cell size because of resources restriction. Also, to find a point where the Perplexity begins to deteriorate, we need to test spatial cells smaller than the regular size of popular places where activities are reported on Foursquare.

5.2.4 Qualitative analysis

In this section, we present qualitative studies of language generation for the proposed model. First, we show examples of texts generated by a trained spatio-temporal conditioned language model. After that, we present examples in Figures 5.4, 5.5, and 5.6 where we can see the attention weights that the text generation component gives to the elements in the spatio-temporal context. Attention weights in our model can be particularly useful for the GIS community since they relate words to spatial and temporal contexts and offer interpretability. We can see the direct relationship between individual words and different granularities of representation.

Table 5.6: Perplexity results for the Foursquare dataset from New York. In this table, we show the results using squared cells as spatial discretizations.

Context	Cells	Dataset	Perplexity
[]	-	NY-FS	9.13
[h]-hour	24	NY-FS	8.97
[d]-day	7	NY-FS	9.10
[w]-week	5	NY-FS	9.21
[m]-month	12	NY-FS	9.09
[hdwm]-alltimes	48	NY-FS	9.00
[p1]-0.001	17,929	NY-FS	5.40
[p2]-0.002	11,260	NY-FS	5.74
[p4]-0.004	6,060	NY-FS	6.10
[p8]-0.008	3,283	NY-FS	6.63
[p1p2p4p8]-allplaces	38,532	NY-FS	5.45
[hdwm p1p2p4p8]-all	38,580	NY-FS	5.34

Table 5.7: Perplexity results for the Foursquare dataset from New York. In this table, we show the results using smaller squared cells as spatial discretizations.

Context	Cells	Dataset	Perplexity
[]	-	NY-FS	8.31
[p]-0.00075	21,250	NY-FS	5.33
[p]-0.00050	26,431	NY-FS	5.22
[p]-0.00025	35,091	NY-FS	5.07

Language generation given a spatio-temporal context

In Table 5.8 we show examples of a language model trained with the Twitter dataset from Los Angeles with all granularities of time and space discretization (last row in Table 5.4). We selected two hubs for urban activities in Los Angeles: the Staples Center and Venice Beach. For the Staples Center, we selected a date for a concert by the British band Arctic Monkeys and a date for a basketball game between the Los Angeles Lakers and the Los Angeles Clippers. We can observe that even for the same location, the texts generated can be associated with different events. For the examples using Venice Beach as context, we can see that the generated texts are associated with beach activities.

Attention weights given a spatio-temporal context

Figures 5.4, 5.5, and 5.6 present examples given the Staples Center as context. In Figure 5.4, we show the date from a Los Angeles Lakers game. We can see that the word *staples* is associated with the finer granularity of geo-coordinates discretization while the word *night* pays attention to the timestamp discretization as the hour of the day. In Figure 5.5, we show the date from a Katy Perry concert. We can

Table 5.8: Examples of text generation after training a spatio-temporal conditioned language model with the dataset of Twitter from Los Angeles. In this table, we show results for two points of interest: the Staples Center and Venice Beach. For the Staples Center, we selected a date for a concert and a date for a basketball game.

Context	Text Generated
(Staples Center) (34.043; -118.267) (Concert Date) '2014/08/07 22:00:00'	['<START>', 'taking', 'a', 'break', 'from', 'the', 'arctic', 'monkeys', 'concert', 'and', 'i', 'love', 'the', 'place', 'if', 'you', 'are', 'here', '#staples', 'staplescenter', 'http', '<END>' ['<START>', 'during', 'the', 'night', '#arcticmonkeys', 'http', '<END>'] ['<START>', 'arctic', 'monkeys', 'anthem', 'with', 'my', 'mom', 'at', 'staples', 'center', 'http', '<END>']
(Staples Center) (34.043; -118.267) (Game Date) '2014/10/31 22:00:00'	['<START>', 'just', 'posted', 'a', 'photo', '105', 'east', 'los', 'angeles', 'clippers', 'game', 'http', '<END>'] ['<START>', '#lakers', '#golakers', 'los', 'angeles', 'lakers', 'surprise', 'summer', '-', 'great', 'job', '-', 'lakers', 'nation', 'http', '#sportsroadhouse', '<END>'] ['<START>', 'who', 'wants', 'to', 'go', 'to', 'the', 'lakings', 'game', 'lmao', '<END>']
(Venice Beach) (33.985; -118.472) (Date) '2014/08/24 13:50:00'	['<START>', 'touched', 'down', 'venice', 'beach', '#venice', '#venicebeach', 'http', '<END>'] ['<START>', 'venice', 'beach', 'cali', '#nofilter', '#venice', '#venicebeach', 'is', 'rolling', 'great', '<END>'] ['<START>', 'who', 'wants', 'to', 'go', 'to', 'venice', 'beach', 'shot', 'on', 'the', 'beach', '<END>'] ['<START>', 'venice', 'beach', '#venicebeach', '#california', '#travel', 'venice', 'beach', 'ca', 'http', '<END>'] ['<START>', '#longbeach', '#venicebeach', '#venice', '#beach', '#sunset', '#venice', '#venicebeach', '#losangeles', '#california', 'http', '<END>']

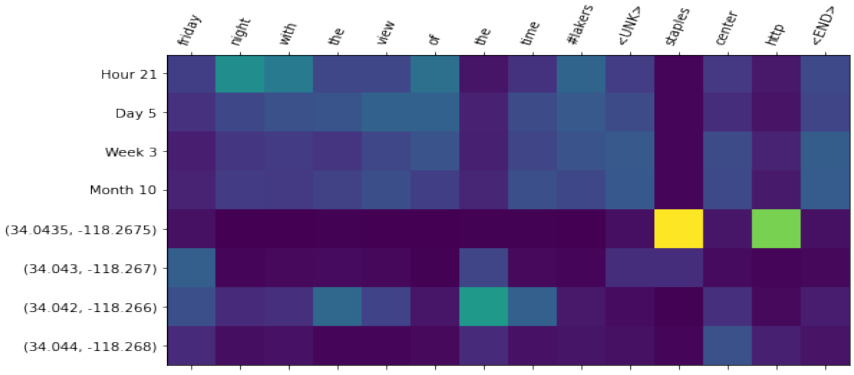


Figure 5.4: Example sentence attention to the spatio-temporal context from a Los Angeles Lakers basketball game at The Staples Center. Yellow means more attention, while blue means less attention.

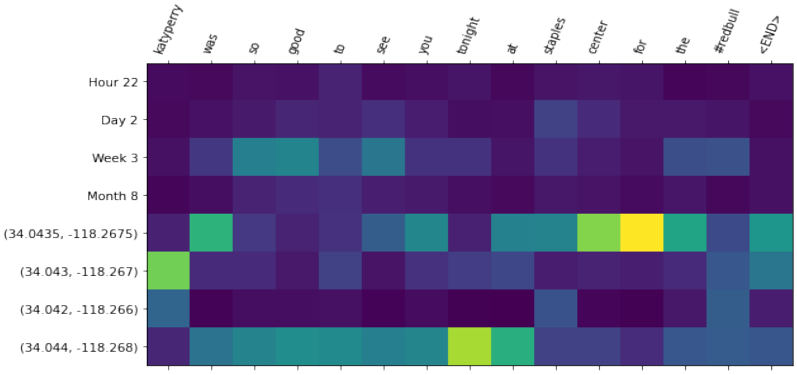


Figure 5.5: Example sentence attention to the spatio-temporal context from a Katy Perry concert at The Staples Center. Yellow means more attention, while blue means less attention.

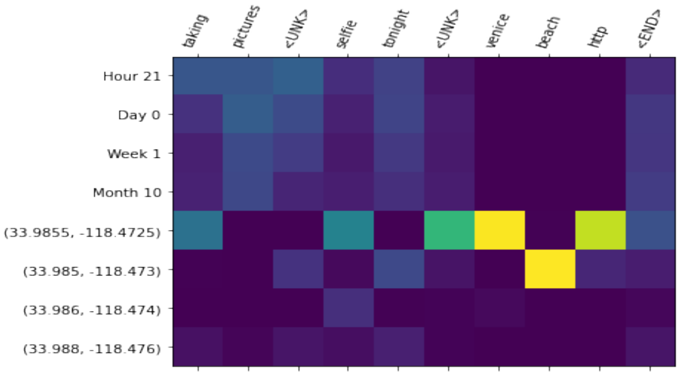


Figure 5.6: Example sentence attention to the spatio-temporal context from a beach day at Venice Beach. Yellow means more attention, while blue means less attention.

see how the words *katyperry* and *at the staples center* are associated with the finest granularities of geo-coordinates discretization, while the word *tonight*, a more general term, is associated with the coarsest granularity. In Figure 5.6, we show an example with the geo-coordinates of Venice Beach as spatial context. We can observe how the word *venice* is associated with the finest level of spatial discretization; while the word *beach* is associated with the second finest granularity, *beach* is a more general term than *venice*, but also is only associated with coastal regions in a city.

5.3 Chapter conclusions

In this chapter, we presented a neural network architecture for spatio-temporal conditioned language modeling. This model is adaptable to different granularities of time and space; which proved to be effective in changing patterns for language generation on two social media datasets.

A remarkable result of our experiments is how modeling space and time at different granularities influences language generation. The optimal when modeling spatial cells for the Twitter dataset is around 800m × 800m cells. For the Foursquare dataset, we observed the best results as the spatial cell got smaller. For the temporal context, the Twitter dataset showed small improvements but was not as important as the spatial context; for the Foursquare dataset, including the temporal context is better than not including it at all, but when combined with the spatial context it does not play a positive role.

We conducted qualitative evaluations, illustrating the potential of our model for spatio-temporal analyses. On the one hand, we demonstrate that our language models are able to generate sentences that efficiently and coherently describe a spatio-temporal context. This can be especially useful for researchers trying to describe or summarize an event using natural language from spatio-temporal contexts. Moreover, our attention weights provide an interpretable relationship between text, space, and time. To the best of our knowledge, this is the first work to use an attention mechanism for this purpose. These interpretations are valuable as they provide insights into how space and time influence what people say (whether on social networks or any other data source of this nature). Although neural networks are known to be difficult to interpret, attention weights are a well-known example of an interpretable component that has been widely used in machine translation and video captioning, among others. We hope that the results presented here will increase interest in the use of this mechanism in spatio-temporal domains.

Chapter 6

Conclusions

This thesis studies the problem of modeling spatio-temporal annotated textual data. We proposed two models that jointly represent *text*, *timestamps* and *geographical coordinates*: a multi-modal retrieval model and a spatio-temporal conditioned language model. These models aim to answer the research question stated in Section 1.3:

> Can the joint representation of text, time (timestamp), and space (geographic coordinates), be better modeled by capturing the sequential structure of texts and representing the spatio-temporal variables at multiple levels of granularity?

In this chapter, we present the main findings of this thesis. The rest of the chapter is structured as follows: first, we present the main results in Section 6.1; after that the main contributions are presented in Section 6.2, and finally, in Section 6.3, we point out future directions of work.

6.1 Main results

The main results of this thesis are listed next:

- In two social media datasets from Twitter and Foursquare, words are more related to places than to times; covering 91% and 86% of the maximum entropy with temporal windows while only 50% and 34% with spatial cells.
- The distribution of examples over temporal windows (hours of the day (0-23)) and spatial cells (0.001×0.001) show that for both datasets from Twitter and Foursquare; early morning hours are the least frequent, starting to increase in the afternoon until the night hours. Also, for spatial cells around 82%-83%

of the cells have less than the mean number of examples per cell for both datasets.

- The multi-modal retrieval model outperformed previous works based on feature embedding for place retrieval given text and time by 6% in the LA-TW dataset and 4% in the NY-FS dataset.
- The multi-modal retrieval model outperformed previous works based on feature embedding for time retrieval given text and place by 1% in the LA-TW dataset and 21% in the NY-FS dataset.
- The multi-modal retrieval model ranked third outperforming four of the baselines for text retrieval given time and place as querying elements.
- For the spatio-temporal conditioned language model, when considering the spatial context, we observed with the Twitter dataset that the optimal for spatial cells is around 800m × 800m cells (around 0.008 in geo-coordinates values for cell size); while for the Foursquare dataset, the observed pattern is that the lower the spatial cell, the better the modeling of the spatial context.
- For the spatio-temporal conditioned language model, when considering the temporal context, we observed with the Twitter dataset small improvements, but not as important as with the spatial context; for the Foursquare dataset, including the temporal context, it is better than not including it at all, but when combining it with the spatial context proved to be harmful.

The results listed above suggest that modeling the sequential structure of texts, as well as modeling time and space at different levels of granularity, have a positive impact on modeling text generated under spatio-temporal dimensions. Therefore, we can conclude that our research question is supported by the experimental results.

6.2 Main contributions

The main research contributions of this thesis are listed next:

- Proposed a multi-modal retrieval model which given a collection of spatio-temporal annotated texts allows for querying the model with any pair from $\langle time, location, text \rangle$ and retrieving the missing one.
- Demonstrated how the multi-modal retrieval model could be used to find patterns of crime incidents given a dataset of crime incident descriptions from the city of New York.
- Proposed a spatio-temporal conditioned language model which allows for representing times and places at different levels of granularities as context for language generation.

- Demonstrated how the spatio-temporal conditioned language model could be used to characterize urban locations from the perspective of social media with natural language.
- Demonstrated how an attention-based neural network could be used to visualize relations between texts and the spatio-temporal context where it is generated.

These contributions make us believe that the models developed in this thesis offer practical solutions for people looking to explore, analyze, characterize and find patterns of interest from spatio-temporal annotated textual datasets.

6.3 Future work

In this section, we discuss directions for further research for the models presented in this thesis. These research directions focus on pre-trained language models, discretization approaches, periodic patterns of timestamps and geo-coordinates, properties of spatial and temporal embeddings, transferability of the presented models, and studies over additional datasets. Next, we provide additional details on these future lines of research.

6.3.1 Pre-trained language models

Pre-trained language models are state-of-the-art for most natural language processing tasks. In the case of spatio-temporal conditioned language modeling, pre-trained language models can be fine-tuned to learn to generate language given the spatio-temporal context.

6.3.2 Discretization based on socioeconomic divisions

The two discretization approaches followed in this thesis are density-based automatic discretization and hand-crafted discretizations. Other discretization techniques may provide additional semantic information. For example, in the case of spatial discretization, geographical divisions based on socioeconomic criteria could be interesting to explore.

6.3.3 Periodic patterns of timestamps and geo-coordinates

Humans follow patterns in their everyday life. Usually, in the morning, we go to work, in the afternoon we can go to the gym or get together with friends; while at night we are at home and go out on weekends. Mostly, these patterns are ruled by time and semantic meanings of time (hour, day, week, etc) that we arrange in circular patterns. For example hour 24 is near hour 1, and month 12 (December) is near month 1 (January); despite the numerical difference. It would be interesting to represent these periodic patterns of time in the joint modeling of text, time, and space. In the case of geo-coordinates, periodic patterns arise when a dataset covers the region of the world where the maximum longitude is near the minimum longitude due to the horizontal expansion of the world on maps.

6.3.4 Properties of spatial and temporal embeddings

Although previous works focus on feature embeddings and the main objective of the models presented in this thesis is to measure the influence of modeling the sequential structure of texts, as well as modeling time and space at different granularities, it would be insightful to explore the properties of the learned embeddings as a sub-product of the learning process. This analysis would provide a view into spatio-temporal representation patterns learned by the proposed models.

6.3.5 Transferability

The third option of further research is to study the transferability of the presented models by deploying spatio-temporal textual models trained on data from a source domain to a target domain. The source domain can be Twitter and the target domain can be crime incident 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., time and space are shared in both domains); then a transfer learning approach can be employed. In this example, the source domain can enrich the available information the target domain. Neural network models are very suitable for transfer learning as one can pre-train a model from the source domain and adapt it to the target domain via further training. Since neural networks leverage statistical strengths from large datasets (the source domain), the transfer learning approach may help to improve performance on the target domain.

6.3.6 Contemporary datasets

And fourth, we foresee valuable future research opportunities by working with contemporary datasets. We conducted our experiments with the LA-TW (Twitter messages collected from Los Angeles, USA, 2014), NY-FS (Foursquare check-ins reported on Twitter by users in the city of New York, USA, 2012) and NY-Crime (crime reports from the city of New York, USA, 2015) datasets to keep the evaluation process consistent with previous works. An interesting research option would be to work with recent contemporary and to study how recent events like the COVID-19 pandemic are reported on social media from a spatio-temporal perspective.

Bibliography

[1] Hamed Abdelhaq, Michael Gertz, and Ayser Armiti. Efficient online extraction of keywords for localized events in twitter. *GeoInformatica*, 21(2):365–388, 2017.

[2] Hamed Abdelhaq, Christian Sengstock, and Michael Gertz. Eventweet: Online localized event detection from twitter. *Proceedings of the VLDB Endowment*, 6(12):1326–1329, 2013.

[3] Amr Ahmed, Liangjie Hong, and Alexander J Smola. Hierarchical geographical modeling of user locations from social media posts. In *Proceedings of the 22nd international conference on World Wide Web*, pages 25–36. ACM, 2013.

[4] James Allan, Jaime G Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. Topic detection and tracking pilot study final report. 1998.

[5] Pramod Anantharam, Payam Barnaghi, Krishnaprasad Thirunarayan, and Amit Sheth. Extracting city traffic events from social streams. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(4):43, 2015.

[6] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. End-to-end attention-based large vocabulary speech recognition. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4945–4949. IEEE, 2016.

[7] Ranieri Baraglia, Cristina Ioana Muntean, Franco Maria Nardini, and Fabrizio Silvestri. Learnext: learning to predict tourists movements. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 751–756. ACM, 2013.

[8] Emily M Bender. Linguistic fundamentals for natural language processing: 100 essentials from morphology and syntax. *Synthesis lectures on human language technologies*, 6(3):1–184, 2013.

[9] Emily M Bender and Alex Lascarides. Linguistic fundamentals for natural language processing ii: 100 essentials from semantics and pragmatics. *Synthesis Lectures on Human Language Technologies*, 12(3):1–268, 2019.

[10] David M Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

[11] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[12] Alexander Boettcher and Dongman Lee. Eventradar: A real-time local event detection scheme using twitter stream. In *Green Computing and Communications (GreenCom), 2012 IEEE International Conference on*, pages 358–367. IEEE, 2012.

[13] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[14] Margaret M Bradley and Peter J Lang. Affective norms for english words (anew): Instruction manual and affective ratings. Technical report, Technical report C-1, the center for research in psychophysiology, University of Florida, 1999.

[15] Dirk Brockmann, Lars Hufnagel, and Theo Geisel. The scaling laws of human travel. *Nature*, 439(7075):462–465, 2006.

[16] Peter F Brown, Stephen A Della Pietra, Vincent J Della Pietra, Jennifer C Lai, and Robert L Mercer. An estimate of an upper bound for the entropy of english. *Computational Linguistics*, 18(1):31–40, 1992.

[17] Sandro Cavallari, Vincent W. Zheng, Hongyun Cai, Kevin Chen-Chuan Chang, and Erik Cambria. Learning community embedding with community detection and node embedding on graphs. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, page 377–386, New York, NY, USA, 2017. Association for Computing Machinery.

[18] Ling Chen and Abhishek Roy. Event detection from flickr data through wavelet-based spatial analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 523–532. ACM, 2009.

[19] Xinyu Chen, Youngwoon Cho, and Suk Young Jang. Crime prediction using twitter sentiment and weather. In *Systems and Information Engineering Design Symposium (SIEDS), 2015*, pages 63–68. IEEE, 2015.

[20] Tao Cheng and Thomas Wicks. Event detection using twitter: a spatio-temporal approach. *PloS one*, 9(6):e97807, 2014.

[21] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar, October 2014. Association for Computational Linguistics.

[22] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua 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)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

[23] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

[24] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina 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)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[25] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 135–144. ACM, 2017.

[26] Susan T Dumais et al. Latent semantic analysis. *Annu. Rev. Inf. Sci. Technol.*, 38(1):188–230, 2004.

[27] Jacob Eisenstein, Brendan O'Connor, Noah A Smith, and Eric P Xing. A latent variable model for geographic lexical variation. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 1277–1287. Association for Computational Linguistics, 2010.

[28] Paahuni Khandelwal Ekta, Priya Bundela, and Richa Dewan. Tweet analysis for real-time event detection and earthquake reporting system development. 2017.

[29] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[30] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 1459–1468. International World Wide Web Conferences Steering Committee, 2018.

[31] Wei Feng, Chao Zhang, Wei Zhang, Jiawei Han, Jianyong Wang, Charu Aggarwal, and Jianbin Huang. Streamcube: hierarchical spatio-temporal hashtag clustering for event exploration over the twitter stream. In *Data Engineering (ICDE), 2015 IEEE 31st International Conference on*, pages 1561–1572. IEEE, 2015.

[32] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. *Advances in neural information processing systems*, 26, 2013.

[33] Matthew S Gerber. Predicting crime using twitter and kernel density estimation. *Decision Support Systems*, 61:115–125, 2014.

[34] Daniel Gildea and Daniel Jurafsky. Automatic labeling of semantic roles. *Computational linguistics*, 28(3):245–288, 2002.

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

[36] Yoav Goldberg and Graeme Hirst. *Neural Network Methods in Natural Language Processing*. Morgan amp; Claypool Publishers, 2017.

[37] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[38] Alex Graves. Supervised sequence labelling. In *Supervised sequence labelling with recurrent neural networks*, pages 5–13. Springer, 2012.

[39] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *International Conference on Machine Learning*, pages 1764–1772, 2014.

[40] Aditya Grover and Jure Leskovec. Node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.

[41] William L. Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *IEEE Data Eng. Bull.*, 40(3):52–74, 2017.

[42] Jeff Heaton. Ian goodfellow, yoshua bengio, and aaron courville: Deep learning, 2018.

[43] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[44] Andrew Hoegh and Marco AR Ferreira. Spatiotemporal model fusion: multiscale modelling of civil unrest. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 65(4):529–545, 2016.

[45] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, 1999.

[46] Thomas Hofmann. Probabilistic latent semantic indexing. In *ACM SIGIR Forum*, volume 51, pages 211–218. ACM, 2017.

[47] Liangjie Hong, Amr Ahmed, Siva Gurumurthy, Alexander J Smola, and Kostas Tsioutsiouliklis. Discovering geographical topics in the twitter stream. In *Proceedings of the 21st international conference on World Wide Web*, pages 769–778. ACM, 2012.

[48] Ting Hua, Feng Chen, Liang Zhao, Chang-Tien Lu, and Naren Ramakrishnan. Sted: semi-supervised targeted-interest event detectionin in twitter. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1466–1469. ACM, 2013.

[49] Xiao Huang, Jundong Li, and Xia Hu. Label informed attributed network embedding. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 731–739. ACM, 2017.

[50] Daniel Jurafsky and James H. Martin. *Speech and Language Processing (2Nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2009.

[51] D Kinga and J Ba Adam. A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, volume 5, page 12, 2015.

[52] Christoph Carl Kling, Jérôme Kunegis, Sergej Sizov, and Steffen Staab. Detecting non-gaussian geographical topics in tagged photo collections. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 603–612. ACM, 2014.

[53] Gizem Korkmaz, Jose Cadena, Chris J Kuhlman, Achla Marathe, Anil Vullikanti, and Naren Ramakrishnan. Combining heterogeneous data sources for civil unrest forecasting. In *Advances in Social Networks Analysis and Mining (ASONAM), 2015 IEEE/ACM International Conference on*, pages 258–265. IEEE, 2015.

[54] John Krumm and Eric Horvitz. Eyewitness: Identifying local events via space-time signals in twitter feeds. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, page 20. ACM, 2015.

[55] John Krumm and Dany Rouhana. Placer: semantic place labels from diary data. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 163–172. ACM, 2013.

[56] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196, 2014.

[57] Ryong Lee and Kazutoshi Sumiya. Measuring geographical regularities of crowd behaviors for twitter-based geo-social event detection. In *Proceedings of the 2nd ACM SIGSPATIAL international workshop on location based social networks*, pages 1–10. ACM, 2010.

[58] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. Predicting the next location: A recurrent model with spatial and temporal contexts. In *AAAI*, pages 194–200, 2016.

[59] Shujie Liu, Nan Yang, Mu Li, and Ming 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, pages 1491–1500, 2014.

[60] Yafeng Lu, Xia Hu, Feng Wang, Shamanth Kumar, Huan Liu, and Ross Maciejewski. Visualizing social media sentiment in disaster scenarios. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1211–1215. ACM, 2015.

[61] Claudio Lucchese, Raffaele Perego, Fabrizio Silvestri, Hossein Vahabi, and Rossano Venturini. How random walks can help tourism. *Advances in Information Retrieval*, pages 195–206, 2012.

[62] Christopher D Manning. *Introduction to information retrieval.* Syngress Publishing,, 2008.

[63] Michael Mathioudakis and Nick Koudas. Twittermonitor: trend detection over the twitter stream. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 1155–1158. ACM, 2010.

[64] Qiaozhu Mei, Chao Liu, Hang Su, and ChengXiang Zhai. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *Proceedings of the 15th international conference on World Wide Web*, pages 533–542. ACM, 2006.

[65] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[66] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, volume 13, pages 746–751, 2013.

[67] Harvey J Miller. Tobler's first law and spatial analysis. *Annals of the Association of American Geographers*, 94(2):284–289, 2004.

[68] Anna Monreale, Fabio Pinelli, Roberto Trasarti, and Fosca Giannotti. Wherenext: a location predictor on trajectory pattern mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 637–646. ACM, 2009.

[69] Ruchit Nagar, Qingyu Yuan, Clark C Freifeld, Mauricio Santillana, Aaron Nojima, Rumi Chunara, and John S Brownstein. A case study of the new york city 2012-2013 influenza season with daily geocoded twitter data from temporal and spatiotemporal perspectives. *Journal of medical Internet research*, 16(10), 2014.

[70] Anastasios Noulas, Salvatore Scellato, Neal Lathia, and Cecilia Mascolo. Mining user mobility features for next place prediction in location-based services. In *Data mining (ICDM), 2012 IEEE 12th international conference on*, pages 1038–1043. IEEE, 2012.

[71] Ozer Ozdikis, Halit Oguztuzun, and Pinar Karagoz. Evidential location estimation for events detected in twitter. In *Proceedings of the 7th Workshop on Geographic Information Retrieval*, pages 9–16. ACM, 2013.

[72] Ozer Ozdikis, Halit Oğuztüzün, and Pinar Karagoz. Evidential estimation of event locations in microblogs using the dempster–shafer theory. *Information Processing & Management*, 52(6):1227–1246, 2016.

[73] Anand Padmanabhan, Shaowen Wang, Guofeng Cao, Myunghwa Hwang, Zhenhua Zhang, Yizhao Gao, Kiumars Soltani, and Yan Liu. Flumapper: A cybergis application for interactive analysis of massive location-based social media. *Concurrency and Computation: Practice and Experience*, 26(13):2253–2265, 2014.

[74] Martha Palmer, Daniel Gildea, and Nianwen Xue. Semantic role labeling. *Synthesis Lectures on Human Language Technologies*, 3(1):1–103, 2010.

[75] Bei Pan, Yu Zheng, David Wilkie, and Cyrus Shahabi. Crowd sensing of traffic anomalies based on human mobility and social media. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 344–353. ACM, 2013.

[76] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–43, 2014.

[77] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke 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)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[78] Vasin Punyakanok, Dan Roth, and Wen-tau Yih. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2):257–287, 2008.

[79] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 2019.

[80] Naren Ramakrishnan, Patrick Butler, Sathappan Muthiah, Nathan Self, Rupinder Khandpur, Parang Saraf, Wei Wang, Jose Cadena, Anil Vullikanti, Gizem Korkmaz, et al. 'beating the news' with embers: forecasting civil unrest using open source indicators. In *Proceedings of the 20th ACM SIGKDD*

international conference on Knowledge discovery and data mining, pages 1799–1808. ACM, 2014.

[81] Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. struc2vec: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 385–394. ACM, 2017.

[82] Tetsuhiro Sakai and Keiichi Tamura. Identifying bursty areas of emergency topics in geotagged tweets using density-based spatiotemporal clustering algorithm. In *Computational Intelligence and Applications (IWCIA), 2014 IEEE 7th International Workshop on*, pages 95–100. IEEE, 2014.

[83] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM, 2010.

[84] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Tweet analysis for real-time event detection and earthquake reporting system development. *IEEE Transactions on Knowledge and Data Engineering*, 25(4):919–931, 2013.

[85] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[86] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

[87] Christian Sengstock, Michael Gertz, Florian Flatow, and Hamed Abdelhaq. A probablistic model for spatio-temporal signal extraction from social media. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 274–283. ACM, 2013.

[88] Lidan Shou, Zhenhua Wang, Ke Chen, and Gang Chen. Sumblr: continuous summarization of evolving tweet streams. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 533–542. ACM, 2013.

[89] Ramaswamy Siddarth Shankar. Visualization of the sentiment of the tweets. *Master's Thesis, North Carolina State University, Raleigh, NC*, 2011.

[90] Sergej Sizov. Geofolk: latent spatial semantics in web 2.0 social media. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 281–290. ACM, 2010.

[91] Sergej Sizov. Latent geospatial semantics of social media. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(4):64, 2012.

[92] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[93] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[94] Keiichi Tamura and Takumi Ichimura. Density-based spatiotemporal clustering algorithm for extracting bursty areas from georeferenced documents. In *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, pages 2079–2084. IEEE, 2013.

[95] Keiichi Tamura and Hajime Kitakami. Detecting location-based enumerating bursts in georeferenced micro-posts. In *Advanced Applied Informatics (IIA-IAAI), 2013 IIAI International Conference on*, pages 389–394. IEEE, 2013.

[96] Waldo R Tobler. A computer movie simulating urban growth in the detroit region. *Economic geography*, 46(sup1):234–240, 1970.

[97] Oren Tsur, Adi Littman, and Ari Rappoport. Efficient clustering of short messages into general domains. In *ICWSM*, 2013.

[98] Berwin A Turlach et al. *Bandwidth selection in kernel density estimation: A review.* Université catholique de Louvain Louvain-la-Neuve, 1993.

[99] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[100] Chong Wang, Jinggang Wang, Xing Xie, and Wei-Ying Ma. Mining geographic knowledge using location aware topic model. In *Proceedings of the 4th ACM workshop on Geographical information retrieval*, pages 65–70. ACM, 2007.

[101] Senzhang Wang, Lifang He, Leon Stenneth, Philip S Yu, and Zhoujun Li. Citywide traffic congestion estimation with social media. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, page 34. ACM, 2015.

[102] Xiaofeng Wang and Donald E Brown. The spatio-temporal generalized additive model for criminal incidents. In *Intelligence and Security Informatics (ISI), 2011 IEEE International Conference on*, pages 42–47. IEEE, 2011.

[103] Xiaofeng Wang, Donald E Brown, and Matthew S Gerber. Spatio-temporal modeling of criminal incidents using geographic, demographic, and twitter-derived information. In *Intelligence and Security Informatics (ISI), 2012 IEEE International Conference on*, pages 36–41. IEEE, 2012.

[104] Xiaofeng Wang, Matthew S Gerber, and Donald E Brown. Automatic crime prediction using events extracted from twitter posts. In *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*, pages 231–238. Springer, 2012.

[105] Kazufumi Watanabe, Masanao Ochi, Makoto Okabe, and Rikio Onai. Jasmine: a real-time local-event detection system based on geolocation information propagated to microblogs. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2541–2544. ACM, 2011.

[106] Nigel Waters. Tobler's first law of geography. *International Encyclopedia of Geography: People, the Earth, Environment and Technology*, pages 1–15, 2016.

[107] Fei Wu, Zhenhui Li, Wang-Chien Lee, Hongjian Wang, and Zhuojie Huang. Semantic annotation of mobility data using social media. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1253–1263. International World Wide Web Conferences Steering Committee, 2015.

[108] Cheng Yang, Maosong Sun, Wayne Xin Zhao, Zhiyuan Liu, and Edward Y Chang. A neural network approach to jointly modeling social networks and mobile trajectories. *ACM Transactions on Information Systems (TOIS)*, 35(4):36, 2017.

[109] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. *XLNet: Generalized Autoregressive Pretraining for Language Understanding.* Curran Associates Inc., Red Hook, NY, USA, 2019.

[110] D. Yao, C. Zhang, Z. Zhu, J. Huang, and J. Bi. Trajectory clustering via deep representation learning. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 3880–3887, May 2017.

[111] Di Yao, Chao Zhang, Jianhui Huang, and Jingping 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*, pages 2411–2414. ACM, 2017.

[112] Mao Ye, Dong Shou, Wang-Chien Lee, Peifeng Yin, and Krzysztof 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*, pages 520–528. ACM, 2011.

[113] Zhijun Yin, Liangliang Cao, Jiawei Han, Chengxiang Zhai, and Thomas Huang. Geographical topic discovery and comparison. In *Proceedings of the 20th international conference on World wide web*, pages 247–256. ACM, 2011.

[114] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. Who, where, when and what: discover spatio-temporal topics for twitter users. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 605–613. ACM, 2013.

[115] Quan Yuan, Wei Zhang, Chao Zhang, Xinhe Geng, Gao Cong, and Jiawei 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*, pages 263–272. ACM, 2017.

[116] Chao Zhang, Liyuan Liu, Dongming Lei, Quan Yuan, Honglei Zhuang, Tim Hanratty, and Jiawei Han. Triovecevent: Embedding-based online local event detection in geo-tagged tweet streams. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 595–604. ACM, 2017.

[117] Chao Zhang, Mengxiong Liu, Zhengchao Liu, Carl Yang, Luming Zhang, and Jiawei Han. Spatiotemporal activity modeling under data scarcity: A graph-regularized cross-modal embedding approach. AAAI, 2018.

[118] Chao Zhang, Quan Yuan, and Jiawei Han. Bringing semantics to spatiotemporal data mining: Challenges, methods, and applications. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pages 1455–1458. IEEE, 2017.

[119] Chao Zhang, Keyang Zhang, Quan Yuan, Haoruo Peng, Yu Zheng, Tim Hanratty, Shaowen Wang, and Jiawei Han. Regions, periods, activities: Uncovering urban dynamics via cross-modal representation learning. In *Proceedings of the 26th International Conference on World Wide Web*, pages 361–370, 2017.

[120] Chao Zhang, Keyang Zhang, Quan Yuan, Fangbo Tao, Luming Zhang, Tim Hanratty, and Jiawei Han. React: Online multimodal embedding for recency-aware spatiotemporal activity modeling. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 245–254. ACM, 2017.

[121] Chao Zhang, Keyang Zhang, Quan Yuan, Luming Zhang, Tim Hanratty, and Jiawei Han. Gmove: Group-level mobility modeling using geo-tagged social media. In *KDD: proceedings. International Conference on Knowledge Discovery & Data Mining*, volume 2016, page 1305. NIH Public Access, 2016.

[122] Chao Zhang, Guangyu Zhou, Quan Yuan, Honglei Zhuang, Yu Zheng, Lance Kaplan, Shaowen Wang, and Jiawei Han. Geoburst: Real-time local event detection in geo-tagged tweet streams. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 513–522. ACM, 2016.

[123] Liang Zhao, Feng Chen, Chang-Tien Lu, and Naren Ramakrishnan. Spatiotemporal event forecasting in social media. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 963–971. SIAM, 2015.

[124] Liang Zhao, Jieping Ye, Feng Chen, Chang-Tien Lu, and Naren Ramakrishnan. Hierarchical incomplete multi-source feature learning for spatiotemporal event forecasting. In *Proceedings of the 22nd ACM SIGKDD Interna-*

tional Conference on Knowledge Discovery and Data Mining, pages 2085–2094. ACM, 2016.