TWEETS2CUBE: INTERACTIVE SPATIOTEMPORAL KNOWLEDGE ACQUISITION FROM MASSIVE SOCIAL MEDIA

BY

LUNAN LI

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2018

Urbana, Illinois

Adviser:

Jiawei Han

Abstract

We present the TWEETS2CUBE system that uncovers the patterns underlying people's spatiotemporal activities from massive online social media. TWEETS2CUBE organizes unstructured social media records into a multi-dimensional data cube along three dimensions: (1) *what* is the user's activity; (2) *where* does that activity occur; and (3) *when* does that activity occur. As such, the end users can use simple queries to retrieve task-relevant sub-corpus from the data cube in a flexible way. Moreover, TWEETS2CUBE consists of a set of spatiotemporal modeling algorithms, which can be readily applied to the retrieved data for extracting knowledge about people's activities in the physical world. Such algorithms jointly model location, time, and text and are capable of discovering a variety of patterns, such as routine spatiotemporal activities, unusual events, and mobility patterns. With TWEETS2CUBE, the end users can interactively retrieve task-relevant social media and choose appropriate spatiotemporal modeling algorithms for knowledge acquisition, which makes TWEETS2CUBE highly useful for downstream tasks like disaster relief, targeted advertising, and location-based recommendation.

To my parents, for their love and support.

Acknowledgments

I want to thank my parents for their support in my whole life and education. I want to thank Chao, Zhang and Dr. Han for the guidance in my master thesis. I want to thank Kangyan Zhou and Hong Cheng for their contribution in the user mobility discovery model.

Table of Contents

Chapter 1: Introduction

Today's world is being explored in a human-centric and digitalized manner. Every day, billions of people go to different places in the world and broadcast their and others' activities on social media platforms (*e.g.*, Facebook, Twitter, Instagram). The confluence of people's offline activities and online interactions sheds light on utilizing massive social media for modeling people's activities in the physical world. Nevertheless, while people are good at producing such massive social media, they often struggle to digest it and gain useful knowledge due to its highly unstructured and multifaceted nature.

We present Tweets2Cube, a system that allows users to interactively extract actionable knowledge about people's spatiotemporal activities from massive social media. With Tweets2Cube, one is able to answer questions like: (1) What are the typical activities at a given location and time? (2) Can we detect emergent events (*e.g.*, disasters, social unrest) at their onsets? and (3) What are the patterns underlying people's daily movements?

To acquire actionable spatiotemporal knowledge from social media, Tweets2Cube features a structuring-and-mining framework. First, it contains a multi-dimensional structuring module, which discovers the latent structures of social media and organizes the massive social media into a multi-dimensional data cube along three dimensions: (1) *topic* — what is the user's activity; (2) *location* — where does that activity occur; and (3) *time* — when does that activity occur. From the data cube, the end users can easily retrieve task-relevant social media data (*e.g.*, *protest*-related tweets in *Chicago* in *2017*) with simple queries. The second module of Tweets2Cube consists of a collection of spatiotemporal mining algorithms [1, 2, 3]. By jointly modeling text, location, and time, these algorithms extract a variety of patterns underlying people's activities, including: (1) routine spatiotemporal activities; (2) unusual events; and (3) mobility patterns.

With a user-friendly Web interface, the Tweets2Cube system is versatile and easy to use. An end user can use simple queries to retrieve relevant sub-corpus in the data cube that meet his/her information needs. The user can further select different mining algorithms and apply them on the retrieved data to build corresponding spatiotemporal models. Based on the map-based visualization, the user can further refine his/her queries and explore the spatiotemporal patterns underlying people's activities in an interactive way.

The contributions of Tweets2Cube are: (1) an interactive system that uncovers the patterns underlying people's spatiotemporal activities from massive social media; (2) a multi-dimensional structuring module that extracts structured information from social media, and (3) a collection of spatiotemporal models that allows end users to explore different kinds of

spatiotemporal patterns for downstream tasks. (4) a new event detection module based on deep learning.

The rest of this thesis is organized as following: (1) In section 2, we will briefly go over the system design. (2) In section 3, we will introduce related work of automatically taxonomy construction and phrase mining. (3) In section 4, the detailed design of the Web interface is explained. (4) In section 5, we will explore how TaxonGen[4] is adopted on tweets stream and some modifications we have made to improve its performance. (5) In section 6, we will introduce our new deep learning model on phrase mining. In section 7, we will show some experiments that are realted to TaxonGen and the deep learning model. (6) Section 8 is Conclusion.

Chapter 2: System overview

2.1 ARCHITECTURE.

Figure 2.1 shows the architecture of the Tweets2Cube system. In Tweets2Cube, we monitor the Twitter Streaming API[1] and collect publicly streaming tweets. To extract spatiotemporal knowledge from Twitter data, Tweets2Cube consists of two major components: a multi-dimensional structuring module and a multi-dimensional mining module. The former discovers the latent structures of social media and organize all the social media records into a three-dimensional (location, time, topic) cube structure. The latter includes a collection of spatiotemporal models for: (1) activity discovery; (2) event detection; and (3) mobility modeling. We introduce the details of the two components in the following.

2.2 MULTI-DIMENSIONAL STRUCTURING.

The multi-dimensional structuring module is designed to automatically organize massive social media into a three-dimensional (topic-location-time) data cube. The first key step is to define the structure of such a data cube. While the location and time dimensions have natural structures (*e.g.*, country-state-city), the structure for the topic dimension is hidden. To address this issue, a general solution is to automatically construct a taxonomy from the corpus. A taxonomy organizes information in a general to specific pattern. By constructing such a hierarchical structure, users can easily search for their interesting topics from top to bottom. In our system, we have adopted TaxonGen[4], a topical concept taxonomy algorithm, on tweets. The algorithm is made up of two steps. (1). We first extract structured information (entities, noun phrases, *etc.*) from unstructured social media with an existing NLP tool [5]. (2). Based on the distributed representations [6] of such phrases, we apply hierarchical spherical clustering [7] on them in a top-down manner to generate a topic hierarchy, and the center phrase of each node is selected as the label for the corresponding topic.

However, there are several limitations of TaxonGen if we apply it directly on tweets stream. First, tweets steam, unlike traditional large text corpora, are more sparse and unstructured. People tend to include more specific information in tweets and ignore those general information, which makes the general-specific topic hierarchy hard to build. Secondly, TaxonGen picks the closest phrase to the center of the cluster as representative. This method works

[1] https://dev.twitter.com/rest/public

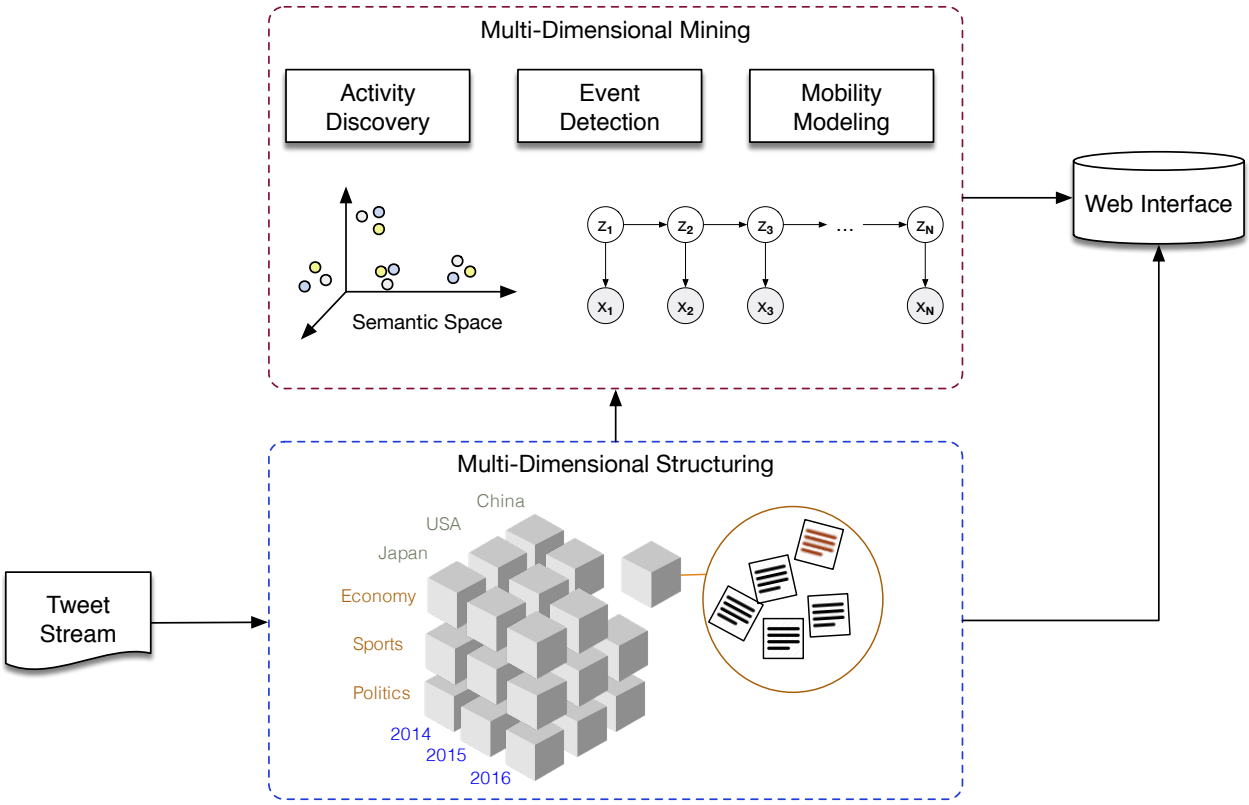

Figure 2.1: The architecture of the TWEETS2CUBE system.

on academic publications but not on tweets because word distributions of tweets are unbalanced as stated previously. Thirdly, TaxonGen uses word2vec [6] to determine the syntactic similarity of words. It's also worth to explore how the algorithm performs with semantic similarity. To address these issues, we proposed following solutions: 1) A new encoder-decoder model to predict non-existing keyphrases from tweets based on existing hashtags.The idea of mining non-existing keyphrases was first porposed by meng *et al.* in 2017 [8], which is explained in detail in section 3.2. 2) A new center picking algorithm that relies on both the distance to the center and the weighting of words in the corpus. The algorithm shows significant improvement in center picking. 3) A new graph embedding algorithm [9] that brings us more topical clusters.

After we build the hierarchical taxonomy for text, the second key step for TWEETS2CUBE is to allocate all the social media records into the multi-dimensional cube. To achieve this, we embed all the labels, terms, and records into the same latent space, by constructing a co-occurrence graph among them and applying graph embedding techniques [9]. With the generated embeddings, we choose the closest label for each record based on directional similarities. Since this section is not part of my work, I won't go in detail for it.

2.3 MULTI-DIMENSIONAL MINING.

The multi-dimensional mining module includes a set of spatiotemporal models for discovering different kinds of spatiotemporal patterns. These models integrate location, time, and text in the modeling process, thereby enabling the end users to obtain a comprehensive *where-when-what* view about people's activities. Specifically, the multi-dimensional mining module consists of four models: (1) The first is a *activity discovery* model that allows users to retrieve routine activities at different locations and time. The discovery model is underpinned by a cross-modal representation learning procedure, which discovers the spatial and temporal hotspots underlying people's activities and embeds location, time, and text into the latent space [1]. (2) The second is an event detection model, which accepts a query time window from the user and extracts all the unusual events occurring in that window based on temporal analysis [2]. (3) The third is a mobility pattern mining model, which uncovers the sequential regularities behind people's daily movements. The mobility model is developed by discovering different users groups and training group-level hidden Markov model [3]. The mobility model can also support predicting the next location a user tends to visit based on his/her current trajectory. (4) The last is a deep learning model that emphasizes on phrase mining and event detection. The model accepts a query, which is similar to [2], predicts phrases by a encoder-decoder model, and group text corpus by the predicted phrases. Unlike tradition phrase mining model, which typically mining quality words in the corpus and reorganized them to form quality phrases, the encoder-decoder model is not only able to mine quality phrases from the text, but also can predicts phrases that are not existing in the corpus.

Chapter 3: Related Work

3.1 AUTOMATICALLY TAXONOMY CONSTRUCTION

3.1.1 Pattern-based methods

Hearst proposed a method to automatically extract the hyponymy lexical relation, from unrestricted text in 1992[10]. Numerous lexical-pattern-based methods were developed to extract relations from world wide web corpus [11, 12, 13] or knowledge base, such as Wikipedia[14, 15]. Agichtein *et al.* proposed the Snowball framework to generate patterns and extract tuples from plain-text documents in 2000 [16] . Zhu *et al.* improved the performance of Snowball framework by add statistic techniques, such as $l1$-regularized feature selection [17]. Carlson *et al.* built a never-ending language learner that could extract, or read, information from the web to populate a growing structured knowledge base in 2010 [18]. Nakashole *et al.* built PATTY, a system that organized relational patterns into taxonomy by parsing leveraging semantic types in 2012 [19]. Jiang *et al.* proposed [20] a context-aware segmentation method to discover high-quality typed textual patterns from large corpus effectively [20].

Although there are many success in patten-based methods, the low recall and the requirement of pre-defined rules or patterns make them not scalable in tweets stream.

3.1.2 Clustering-based methods

Generally speaking, cluster-based methods first learning words or phrases representation and then group words or phrases that have high similarity. There have been various methods on word representation learnings. Bansal *et al.* build a inducing hypernym taxonomies by using a probabilistic graphical model formulation [21]. Fu *et al.* developed a method to identify whether candidate word pair has hypernymhyponym relation by the word-embedding-based semantic projections between words and their hypernyms in 2014 [22]. Luu invented a dynamic weighting neural network to learn term embeddings based on not the contextual information between hypernyms and hyponyms [23]. However, those methods are not good enough for learning word representation of tweets stream because words in tweets don't have hypernym-hyponym relations. In addition, there are always new words, which are related to specific events in a certain period, that would not occur previously. Thus, previous word representation learning cannot be directly applied on our task.

For clustering techique, Davis *et al.* proposed a cluster separation measure in 1979 [24]. Yang *et al.* applied an ontology metric, a score indicating semantic distance, to automatic taxonomy induction tasks in 2009 [25]. Liu *et al.* hierarchically cluster keywords into a taxonomy by Bayesian rose tree [26]. Wang *et al.* proposed a phrase-centric framework for topical hierarchy generation via recursive clustering and ranking [27]. Blei *et al.* developed a generative probabilistic model for hierarchical structures and adopted Chinese restaurant process to hierarchy partitions. [28]. Downey *et al.* invented Sparse Backoff Tree to effectively infer accurate topic spaces of over a million topics [29]. All those methods have shown great effectiveness on clustering, but none of them really handles the hierarchical topic clustering, which separates phrases of detailed topics from general topics.

3.1.3 Supervised methods

There are also supervised methods for taxonomy constructions. General supervised methods first train a model to classify whether pairs of words into relation and non-relation categories based on either human-annotated data or other knowledge base. However, there are such knowledge in raw tweets stream. In section 6, I will present a new method that uses hashtags in tweets to build pairs of words, which have relation, without much human effort.

3.2 AUTOMATICALLY KEYPHRASE EXTRACTION

Automatically keyphrase extraction usually consists of two steps. The first step is to generate keyphrases from text corpus and the second step is to rank quality keyphrases by certain mertric.

There are several methods to extract keyphrases based on some lexical patterns. Turney *et al.* developed a topical keyphase extraction by decision tree in 2000 [30]. Liu *et al.* mined phrases by selecting word sequences which matched certain POS tagger patterns [31]. El-Kishky *et al.* adopted frequent pattern mining to select phrases from corpus [32]. However, all those methods are limited on our task because of the undetermined number of topics and the low frequency of repeated phrases in tweets stream. Another popular keyphrase extraction method based on knowledge base, such as Wikipedia. Shang *et al.* proposed AutoPhrase, which is a framework that mine quality phrases based on existing knowledge based. AutoPhrase recognizes titles from Wikipedia as quality keyphrases, and mines phrases that are similar to them [33]. Our deep learning model follows the idea of AutoPhrase, and converts hashtags in tweets to keyphrases.

The second step is to rank keyphrases by quality. There are various measures of quality. Tomokiyo *et al.* presented a new approach to extracting phrases based on statistical language model [34]. The method used pointwise KL-divergence between multiple language models for scoring both *phraseness* and *informativeness*, which can be unified into a single score to rank extracted phrases [34]. Liu *et al.* proposed a framework to integrate phrase extraction and phrasal segmentation, which mutually enhance each other [35]. In our task, the quality of keyphrases is related to the TaxonGen algorithm that focuses on the popularity and concentration, which is explained in detail in section 5.

3.3 ENCODER-DECODER MODEL

Sutskever first brought deep learning to sequence-to-sequence learning tasks in 2014 [36]. Later, Cho *et al.* proposed the Encoder-Decoder model to solve neural machine translation problems in 2014 [37]. The method solves variable-length translation problems in an end-to-end fashion, and achieved great success in many baseline experiments. Due to the effectiveness and flexibility of the Encoder-Decoder model, it has been applied to various NLP taks. Rush *et al.*, Nallapati *et al.* and Abigail *et al.* have brought it to text summarization [38, 39, 40]. Researchers have also explored different algorithms to improve its performance. Bahdanau *et al.* proposed the attention mechanism that allows the decoder to automatically learn weighting of inputs from each time step of the encoder rather than simply decode everything from the last time step.

The Encoder-Decoder model is first brought to automatically keyphrase extraction by Meng *et al.* in 2017 [8]. They combined the Encoder-Decoder model with copy mechanism, which enables the decoder to extract phrases with out of vocabulary words by selecting words directly from input [41]. Although the copy mechanism relied on the term frequency in long text, the encoder-decoder model itself is a good model to predict non-existing keyphrases from short text.

Chapter 4: Web Interface

In this section, I described my user-friendly web interface. I first give an overview of it in section 4.1. In section 4.2, I introduce some spatiotemporal models that are supported in our interface and demonstrate some example usages of our web interface.

4.1 DESIGN OVERVIEW

The web interface is composed of two parts, which are the front-end and the back-end. The front-end handles all the visualization and user inputs. Figure 4.1 shows the homepage of our web interface. Before trying any spatiotemporal model, users need to first set up the three dimensions (Time, Location and Topic) of Tweets2Cube by clicking the blue button at the top-right corner. Figure 4.2 and figure 4.3 shows the form of those three dimensions. The form is made up of three parts. The first part asks users to select a specific range for the dataset. The second part allows user to define a country-state-city-area tuple that reflects users' preference. The last part allows user to select topics that they are interested in.

Once the dimension of Tweets2Cube is set up, the back-end automatically filters tweets and train the corresponding model. Then, all available models can be select from the drop-down menu at the top.

4.2 SPATIOTEMPORAL MODEL INTRODUCTION

4.2.1 Activity Discovery

The activity model is developed by Zhang *et al.* in 2017, which can models the spatial and temporal hotspots underlying people's activities [1]. Figure 4.1 shows the input to this model, where users are allowed to choose any combination of one or two inputs. The model is made up of two major components. The first component detects both spatial and temporal hotspots based on kernel density. The second component combines location, time and text to form a heterogeneous information network and learns the joint embedding from the network.

Figure 4.4 gives an example query with keywords *"beach"*. The model returns a combination of location, keywords and time that are closely related to the query. As shown in the figure 4.4, most location points are near the pacific ocean, which are famous beaches in Los Angeles. Top keywords either are famous beach names, such as "Redondo beach", or they represent some activities around beach like cruisin. The top time is around later afternoon,

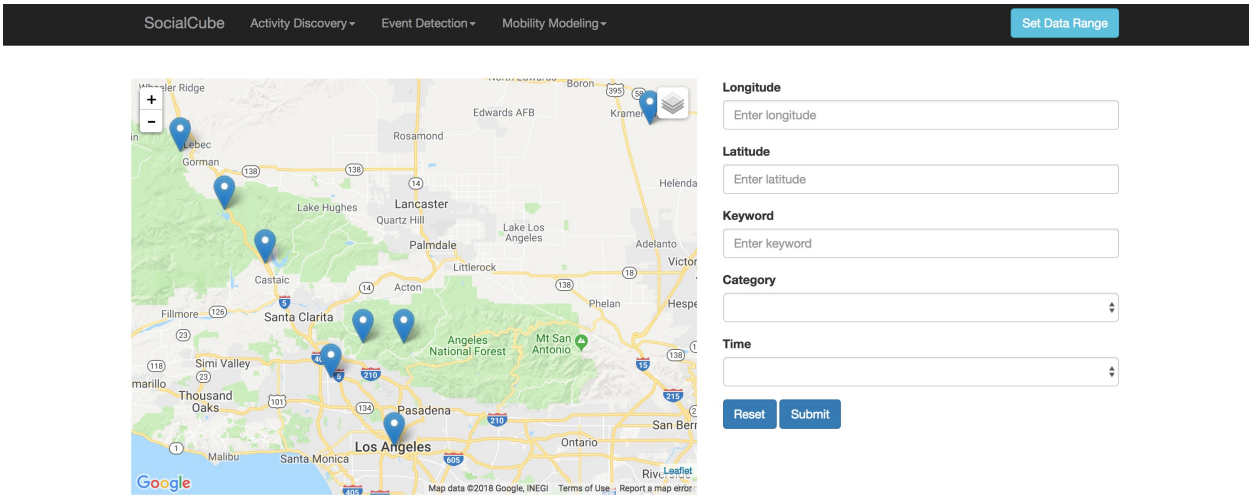

Figure 4.1: Tweets2Cube Web Interface Homepage

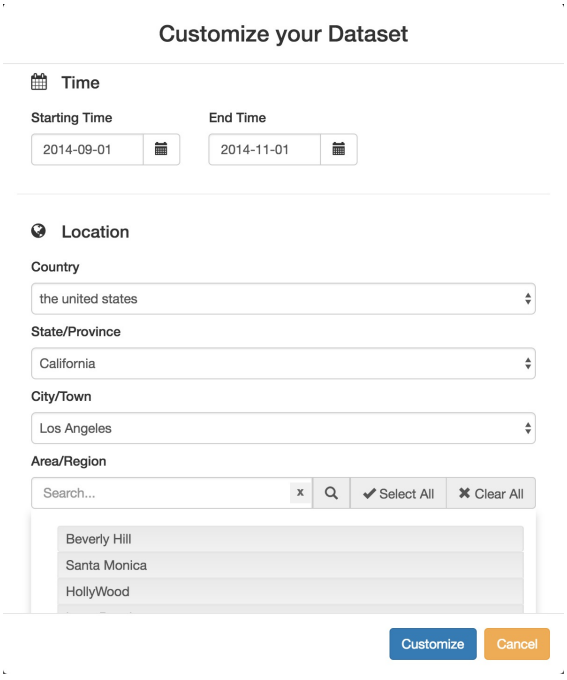

Figure 4.2: Tweets2Cube Time and Location Setup

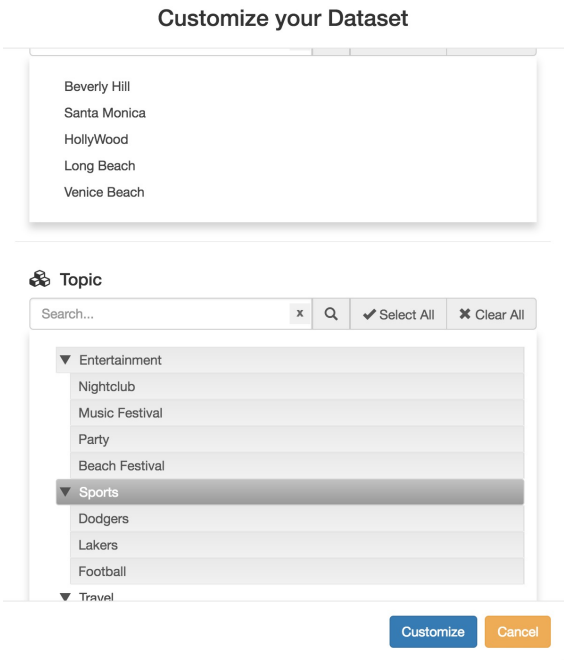

Figure 4.3: Tweets2Cube Topic Setup

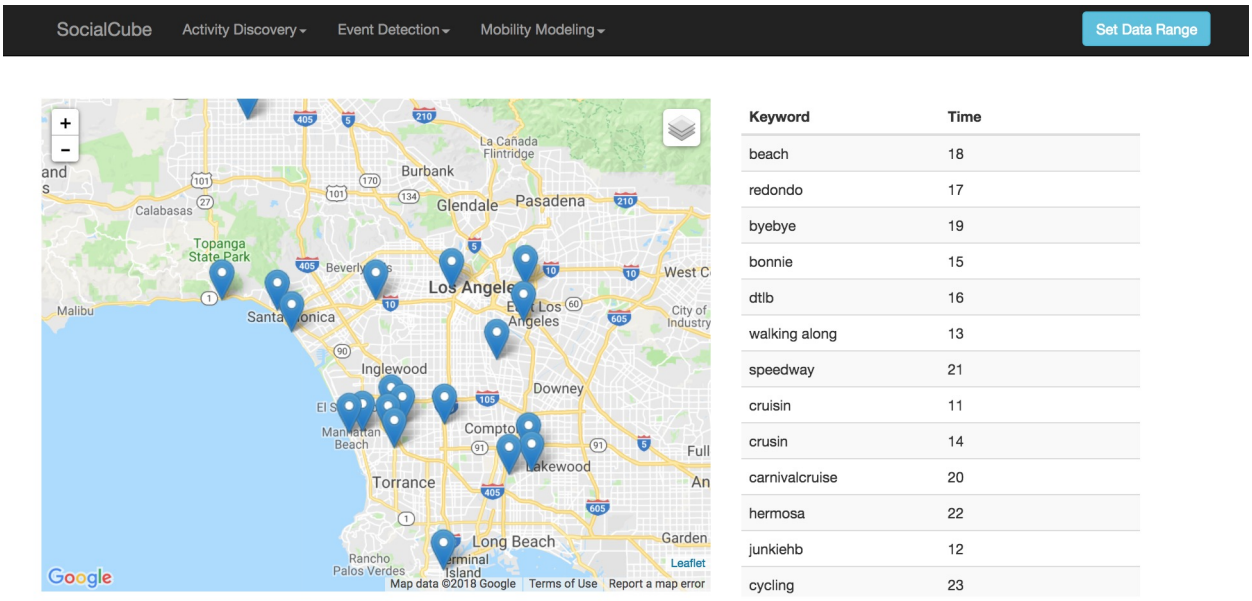

Figure 4.4: TWEETS2CUBE Activity Discovery

which is also the most common time people go to beach and enjoy the good time there.

4.2.2 Event Detection Model

The event detection model is developed by Zhang *et al.* in 2016, which provides effective and real-time local event detection from geo-tagged tweet streams [2]. The model uses a novel authority measure to capture geo-topic relation among tweets [2]. The model first detects *pivots*, which are representative tweets for certain events, and group similar tweets around pivots to form a cluster. An updating module monitors new tweets stream and replace old piovts by new events.

The model requires a time window as the input. Figure 4.5 gives an example of event detection in our web interface. Users first specify a time window by using the date selector at the top, and the model summarizes all special events in that time window and display them at the front end. In figure 4.5, we set the time window between Sep 18th 2014 and Sep 20th 2018. A Dodger's baseball game have been detected. The place is at Dodger's stadium, which is exactly the home court of Los Angeles Dodgers. From the tweet, we can infer that people come to support the Dodgers' team with their friends.

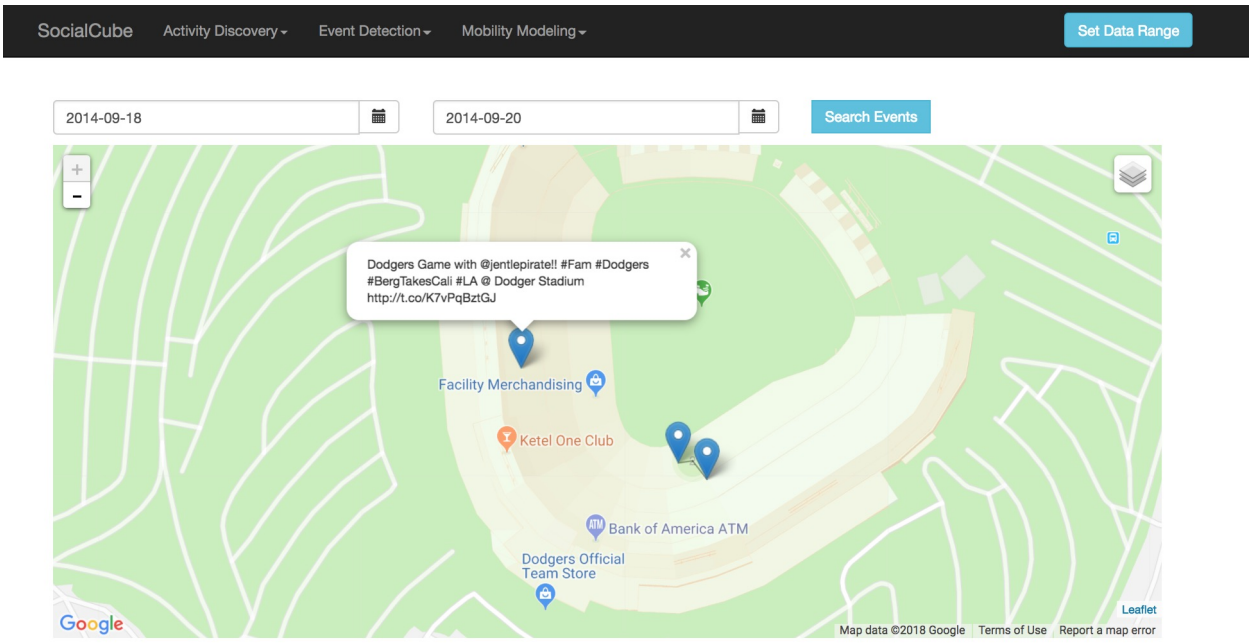

Figure 4.5: Tweets2Cube Event Detection

4.2.3 Mobility Pattern Discovery

In this section, we will introduce two mobility pattern discovery model. The first one is a mobility pattern based on the activity discovery model above, which detects mobility patterns among different social groups [42]. The second one is a mobility pattern based RNN sequence prediction that can be used in many recommendation system.

User Specific Mobility Pattern Discovery

The user specific mobility pattern model discovers two mobility patterns for different social groups [42]. The first one is a sequential pattern that indicates the mobility flow of a specific group. For example, a general mobility flow for UIUC CS students may be from Grainger Library to Siebel building. The second mobility pattern is a frequent triplet, which indicates life style for a specific group such as college students loves go to bar during Friday evening. The model consists of three steps. The first step is based on the heterogeneous information network in section 4.2.1, however, the network is made up of four different types of nodes, which are user, time, location and text. After learning the embedding for each type of node from the network, users will be clustered into different groups and keyphrases will be extracted from each group as the semantic analysis. The last step is to detect the hotspots for each user group. Embeddings of location and text will be combined linearly and then

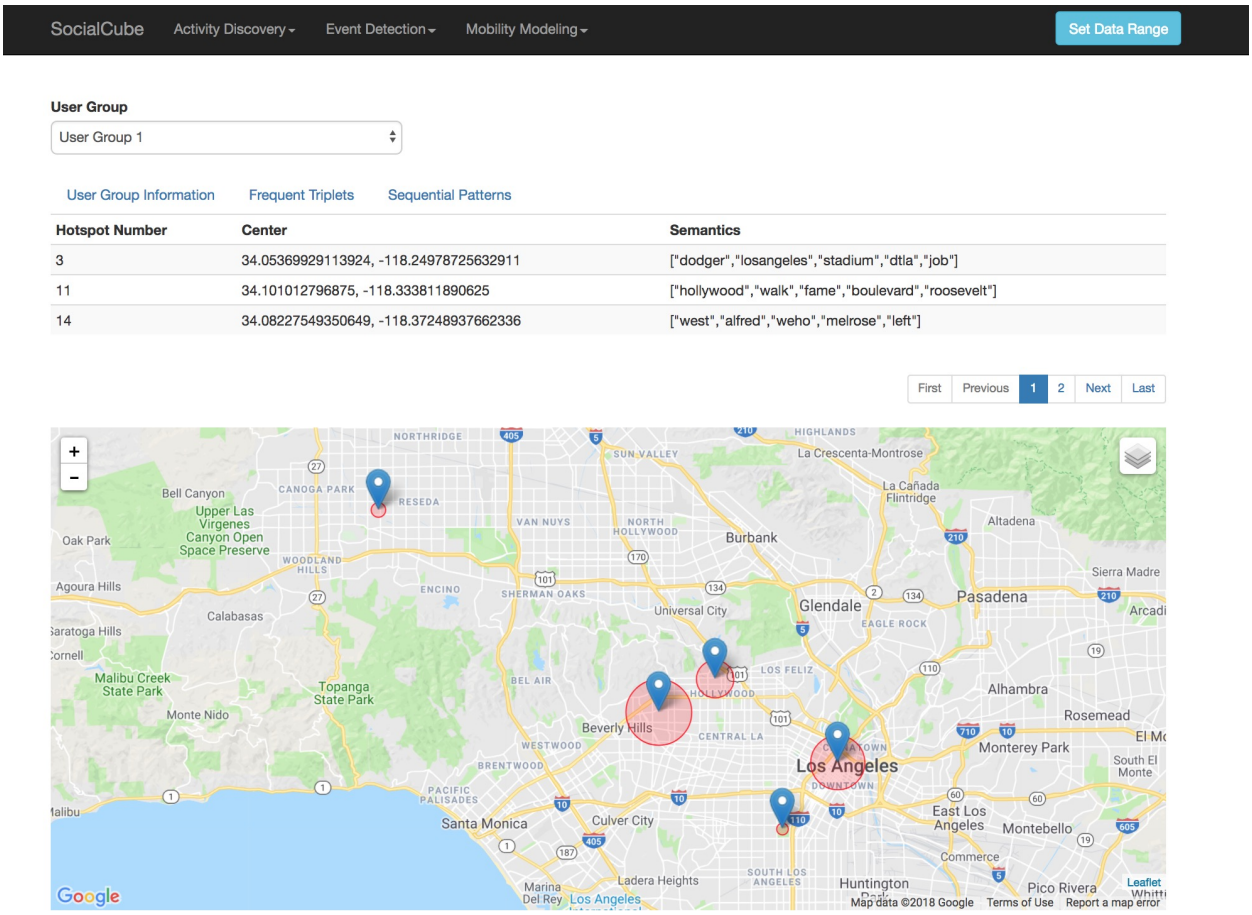

Figure 4.6: Tweets2Cube User Mobility Pattern - Group Information

hierarchically clustered. Each hotspot should not only be close in physical dimension but should also share similar interests.

In figure 4.6, we show a group who are interested in sports and Hollywood. As we can see from the table, repesentative words of hotspot 3 are Dodgers, and stadium, which are typically sport related. Figure 4.7 reflects a mobility flow from stadium to nightclub. A possible explaination for this mobility flow could be some sports fan go to night club to celebrate the winning of their team. Figure 4.8 shows some frequent life style for this group. As we can see, in addition to game, this group also loves night club lives.

Sequence Prediction

The sequence prediction model is developed by Yao *et al.* in 2017, which predicts the next location that users want to visit based on the previous information trajectory. The model combines embedding of location, time, and text information from users' previous trajectory

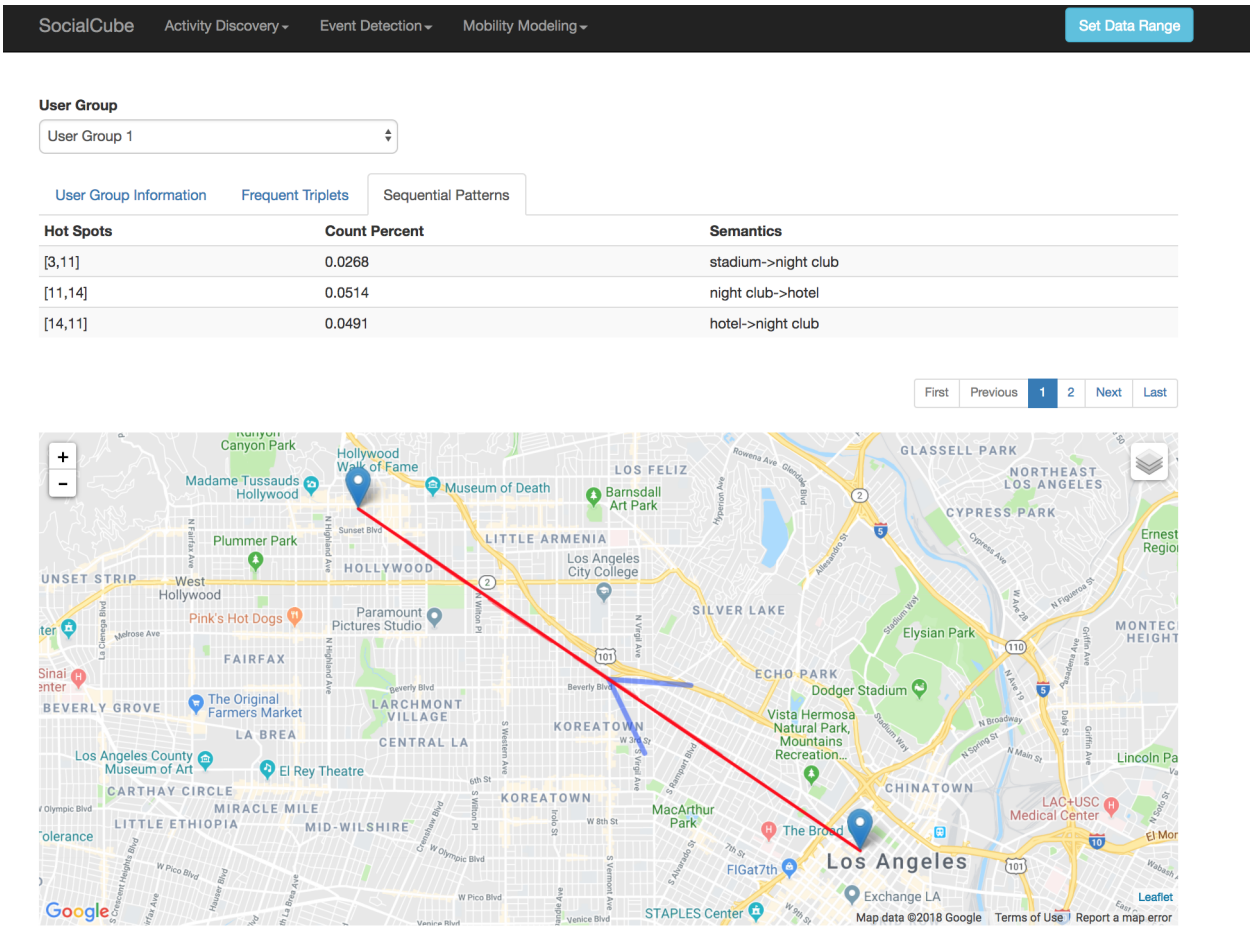

Figure 4.7: Tweets2Cube User Mobility Pattern - Sequential Pattern

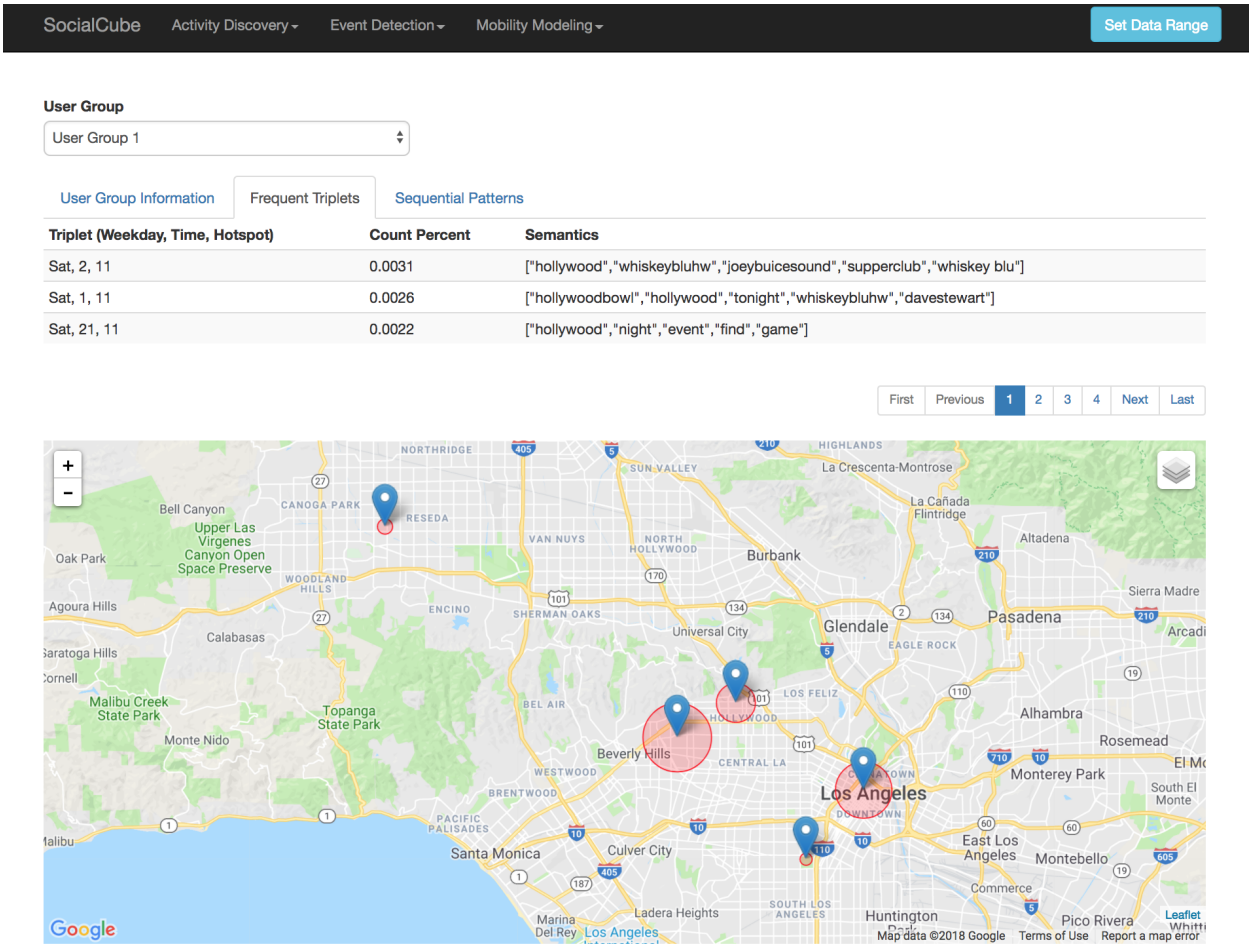

Figure 4.8: Tweets2Cube User Mobility Pattern - Frequent Triplet

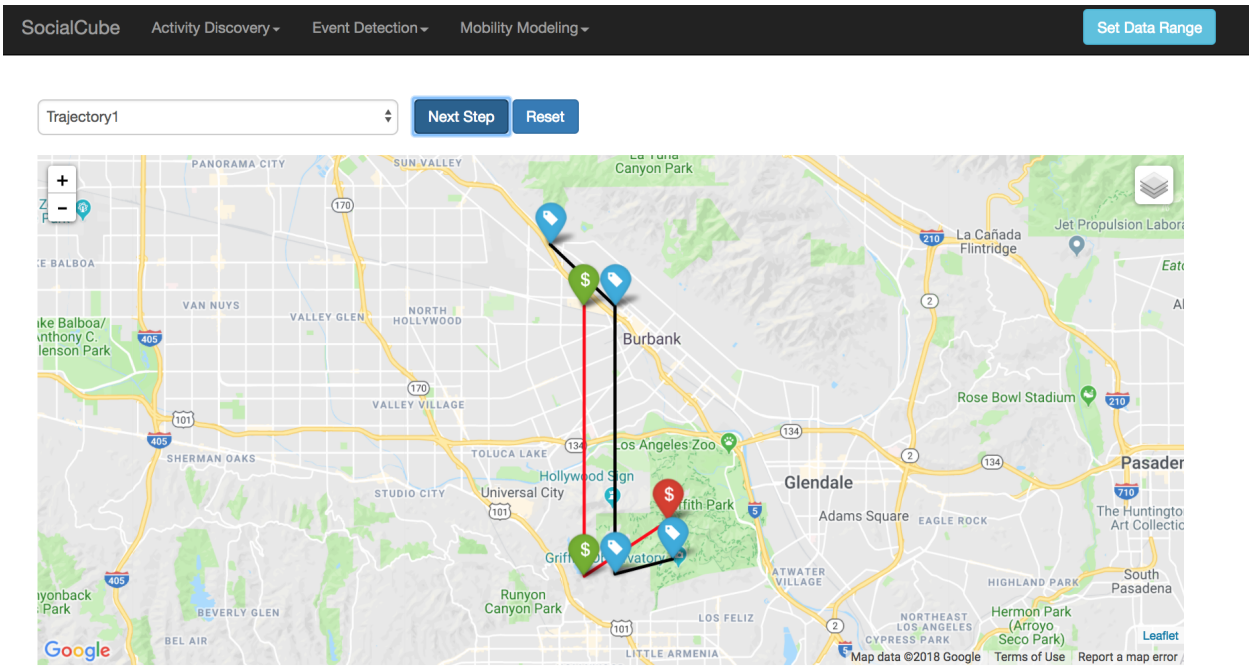

Figure 4.9: TWEETS2CUBE Sequence Prediction

and predict the next location. Figure 4.9 shows an animation of this model. In the figure, blue markers show the original trajectory. Green trajectory are locations that have already been predicted in the previous time step. The red location is the place predicted at the current time step based on previous trajectory.

Chapter 5: Automatically Taxonomy Construction

In this section, I introduce the TaxonGen algorithm and how to apply it to tweets stream. I start with reviewing the detail of the algorithm. I will then go over some existing issue in TaxonGen and the solution to them.

5.1 TAXONGEN DETAIL

The input to TaxonGen has two parts: 1) A corpus Document, which is a tweet stream T; and 2) a set of seed terms V. To generate V, venue categories from *FourSquare* are first transformed into keyphrases. Next, phrases and words of T, which are similar to keyphrases from *FourSquare*, are picked and form the initial V.

Algorithm 1 shows the process of TaxonGen. Starting from the whole tweet stream T, and seed terms V, seed terms are clustered into k sub topics by the Spherical KMeans algorithm. For each subtopic, CaseOLAP filters out general terms that should not be included in the subtopic, picks a center word c_i, which is the most representative word for this subtopic, and selects a new set of seed terms that are close to the center word c_i. The process will be repeated until no more general terms are removed from all subtopics. Then, TaxonGen is applied on every subtopic to form a hierarchical structure. The local embedding module calculates the new representation for seed terms in every subtopic, which makes them more distinguishable. Currently, SkipGram is used in the local embedding moduel.

However, there are two issue if TaxonGen is directly applied on the tweets stream. The first one is the embedding algorithm. While SkipGram measures syntactic similarity, tweets are really unstructured and may not contain that much syntactic information. Instead, it's worth trying graph embedding, which measures semantic similarity among words. The second issue is the center pick algorithm. Unlike normal text corpora, where general words occurs more often than specific words, tweets are more sparse and specific. The current center pick algorithm picks the phrase that is closed to the center of the cluster, which doesn't works well on tweets. In the next two section, solution to these two issues are discussed.

5.2 GRAPH EMBEDDING IN TAXONGEN

As discussed in last section, tweets are unstructured and it's worth trying semantic embedding in TaxonGen. Line, which focuses on the co-occurrence of words or phrases, is

Algorithm 5.1 TaxonGen algorithm

Input: Tweet stream T, and Seed Term V
Output: A hierarchical k-ary tree that each node in the tree represents a topical cluster of phrases
procedure TaxonGen(T, V, k)
    while True **do**
        $V_1, V_2, \ldots, V_k \leftarrow$ Spherical-KMeans(V, K)
        for *i from 1 to k* **do**
            $T_i, v_i, c_i \leftarrow$ CaseOLAP(V_i)
        $T^{'} \leftarrow T_1 \cup T_2 \cup \ldots\ T_K$
        if T $= T^{'}$ **then**
            break
        T $= T^{'}$
    for *i from 1 to k* **do**
        $v_i =$ Local-Embedding(T_i, v_i)
        c_i.children = TaxonGen(T_i, v_i, k)
    $C \leftarrow c_1, c_2, \ldots, c_k$
    return C

used in the local embedding module [9]. A phrase-phrase co-occurrence graph is constructed by simple counting the co-occurrence of every pair of phrases in every tweet. The first order proximity, which indicates the similarity of nodes that have direct edges among them, is used because some adjectives or general terms can co-occur with many different nouns which causes many noises in the second order proximity [9].

5.3 NEW CENTER PICK ALGORITHM

Figure 5.1 shows a food related topic cluster that is built by the TaxonGen algorithm with graph embedding. The hierarchy doesn't show a general-specific pattern. Specific nouns, like salad, pasta, and food, and adjectives, like spicy, are at the root level. However, general nouns, like food, restaurant, and cuisine are in child cluster of the root. As discussed in section 5.1, the issue is caused because the content of tweets are more specific, which means the number of more specific terms are more than the number of general terms. Thus, specific words take the dominant position inside a cluster, which makes the center is surrounded with specific terms, and general terms are further to the center. Table 5.1 gives an example of the distance of some terms to the center. The distance between specific terms, like salmon and salad, and center are around 0.1, while the distance between general terms and center, like food and brunch, are around 0.3 to 0.4.

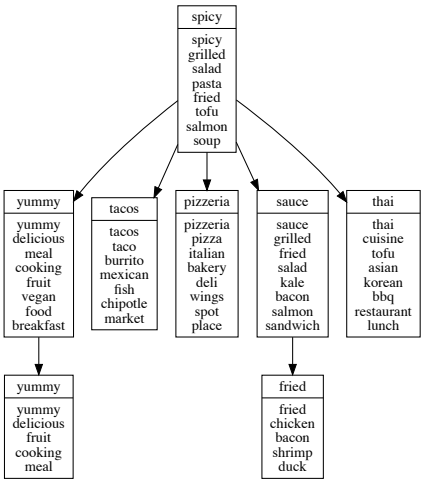

Figure 5.1: A Food Related Topic Cluster in Taxonomy

Table 5.1: Words distance to center - Food cluster

Word	Distance
salmon	0.094
soup	0.110
salad	0.113
pasta	0.114
food	0.276
cuisine	0.299
bbq	0.384
brunch	0.474

The insight for addressing this issue is that, a representative center should not only close to center but also be popular in this cluster. Hence, the popularity is measured by counting the number of co-occurrence a word and all other words in the cluster. Then, the word with most co-occurrence that are within a distance threshold, r, is picked as the new center of the cluster.

Chapter 6: Deep Event Detection

In this section, I introduce our deep learning model, which is basically an encoder-decoder model, on event detection. Unlike tradition event detection models relies on location clustering, the model first uses phrase mining to detect topics on tweets stream. Then, tweets in each topic will be grouped by location. In this thesis, I focused on the phrase mining part. First I give a simple introduction to the method.

6.1 METHOD OVERVIEW

The Encoder-Decoder model has widely been used in many NLP tasks, such as NMT, text summarization and etc. Meng *et al.* combined the Encoder-Decoder model with the copy mechanism to keyphrase extraction in 2017 [8, 41]. Based on Meng's work, I come up with the idea to use the Encoder-Decoder model to group tweets by events. The major challenge for this task is the lack of event tags in tweets. To resolve this issue, I come up a new method to build event tags of tweets based on some popular hashtags. The algorithm consists of two steps. The first step is an online process step that the Encoder-Decoder model predicts event tags for tweets stream. The second step is to group tweets with same event tags.

6.2 PROBLEM DEFINITION

Given a tweets steam of N tweets, the ith tweet contains $(t_i,\ h_i)$, where t_i represents the tweet text and $h_i = (h_{i1},\ h_{i2},\ \ldots,\ h_{in})$ contains n possible hashtags. Both t_i and $\forall\ h_{ij}$ are sequence of words:

$$t_i = t_{i1},\ t_{i2},\ \ldots,\ t_{il_{ti}}$$
$$h_{ij} = h_{ij1},\ h_{ij2},\ \ldots,\ h_{ijl_{hij}}$$

l_{t_i} and $l_{h_{ij}}$ are the length of the tweet text t_i and hashtag h_{ij}. Each pair of $(t_i,\ h_{ij})$ is fed into the Encoder-Decoder model, where t_i is the input to the encoder and h_{ij} is the target of the decoder output. For the purpose of simplicity, $(t,\ h)$ is used to denote every pair of inputs in the rest of this section.

6.3 ENCODER-DECODER MODEL AND ATTENTION MECHANISM

Both encoder and decoder are implemented with RNN. The basic idea of the Encoder-Decoder model is to train the vector representation for every word in the text sequence, and the decoder decodes the vector representation from every time step. The attention mechanism is firstly proposed by Bahdanau *et al.* in 2014 [43]. Traditional decoder only takes the vector representation from the last time step and decode the output sequence from it. Storing all information in a single vector representation is hard and Bahdanau proposed that the vector presentation from every time of the encoder should be considered. The attention mechanism gives an weight to every vector representation and automatically align the weight so that the model can locate relevant components [43].

Every encoder input t = $(t_1, t_2, \ldots, t_n)$, where n is the total number of time steps, is transformed into o = $(o_1, o_2, \ldots, o_n)$ and by h = $(h_1, h_2, \ldots, h_n)$ iterating through the following equation at every time step i:

$$o_i, h_i = f(h_{i-1}, t_i) \tag{6.1}$$

where f is a non-linear activation function, o_i is the output of the encoder and h_i is the hidden state.

Every decoder takes h_n, which is the last hidden state from encoder and denoted as l_0, as its initial hidden state. For every time step j, attention c_j is generated by the following equation:

$$c_j = g(s_{j-1}, l_{j-1}, o) \tag{6.2}$$

where s_{j-1} is output and l_{j-1} is the hidden state at the previous time step of the decoder. o is the summary of all vector representation from the encoder. g is a combination of several non-linear functions that will be explained in detail in the next section. After attention c_j is calculated, s_j is calculated as following:

$$s_j, l_j = f(l_{j-1}, c_j) \tag{6.3}$$

s = $(s_1, s_2, \ldots, s_m)$ is then transformed into a text sequence. Both encoder and decoder are trained in an end-to-end fashion. During evaluation phase, we applied beam search in decoder to select the top k phrases for each tweet text.

6.4 MODEL DETAILS

Our Encoder is implemented by a bi-directional GRU since it has better performance, which is proved by previous studies [37, 43]. The non-linear activation function using in encoder is ReLU.

A simple forward GRU is applied in our decoder model. The detail of the attention mechanism, which is c_j in equation 5.2, is implemented as following:

$$\begin{aligned} \alpha_{ji} &= \frac{\exp(s_{j-1}, o_i)}{\sum_{k=1}^{T} \exp(s_{j-1}, o_k)} \\ c_j &= \sum_{i=1}^{T} \alpha_{ji} \times l_{j-1} \end{aligned} \tag{6.4}$$

where o = $(o_1, o_2, \ldots, o_T)$ is the output from encoder, s_{j-1} is the output of the decoder at previous time step and l_{j-1} is the hidden state of the decoder from the previous time step.

Chapter 7: Experiments

7.1 TAXONGEN EXPERIMENTS

7.1.1 Experiment Setup

The dataset used in the experiment is: a tweets stream from LA. Venue categories from FourSquare are used to initialize seed terms for TaxonGen. For the SkipGram, TaxonGen is experimented with words, and hashtags. Then, we experimented TaxonGen with graph embedding and the new center pick algorithm.

7.1.2 TaxonGen with SkipGram

In this section, we demonstrate the topical taxonomies generated by TaxonGen with different types of terms. A 2-level topic taxonomy is built for words, and hashtags.

Figure 7.1 shows parts of the taxonomy generated by TaxonGen with words. As shown in Figure 7.1, given the tweet stream, TaxonGen detects 5 topics, which are website, game, accessories, bus, and education. Labels for all topics have good quality at this level and all of them are common topics that people would love to post on Twitter. Words in every cluster are also coherent and closely related to the representative term. Then, we can see how TaxonGen splits these two topics into more specific topic. Taking accessories as an example, TaxonGen successfully find 5 major topics in accessories, which are salon, dresses, leather, shoes, and photography. The only flaw is photography, which is not a directly related topic to accessories. However, since there are many pictures of accessories on fashion magazines, photography is still an acceptable topic.

Figure 7.2 shows another taxonomy generated by TaxonGen but in the term of hashtags. Compared to taxonomy with words, taxonomy of hashtags also have close coherent relations in every topic but the parent-child relationship doesn't strictly follow a general-specific pattern. Given the tweets stream, TaxonGen detects 7 topics at the first level, which are #beach, #itfdb, #beer, #vintage, #concert, #dessert, and #food. Taking #beach as an example, all other hashtags that are in this topic are obviously closely related to #beach, for example #beachday, and #venicebeach. However, the subtopic of #beach aren't all closely related to it, like #biking that has hashtags which are more closely related to sports not beach. The reason is probably because SkipGram focuses on syntactic similarity that makes #biking are more closely related to other sports hashtags.

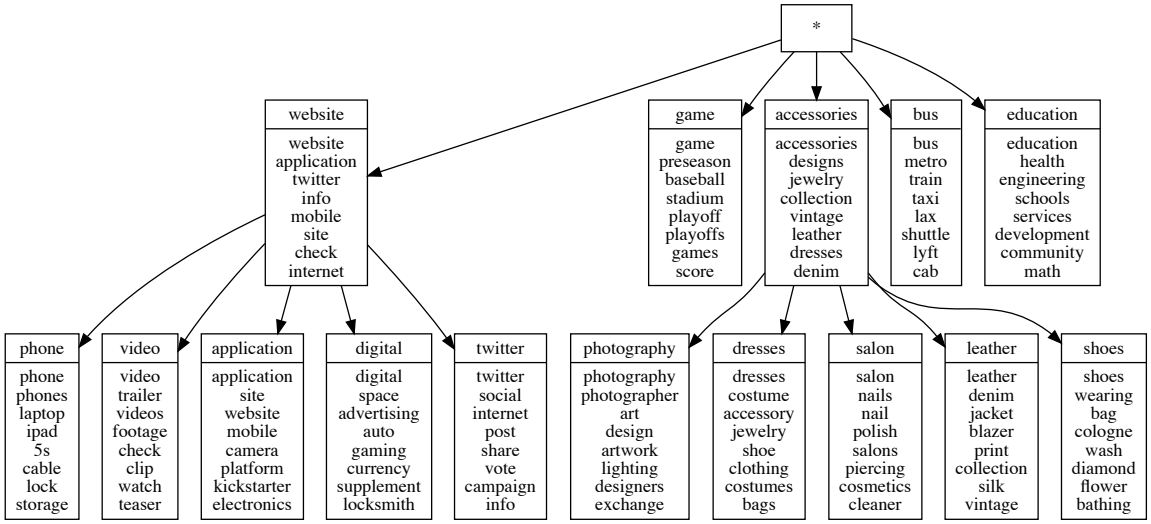

Figure 7.1: Tweet Taxonomy generated by TaxonGen under topics '*' (level 1), 'website' (level 2), and 'accessories' (level 2)

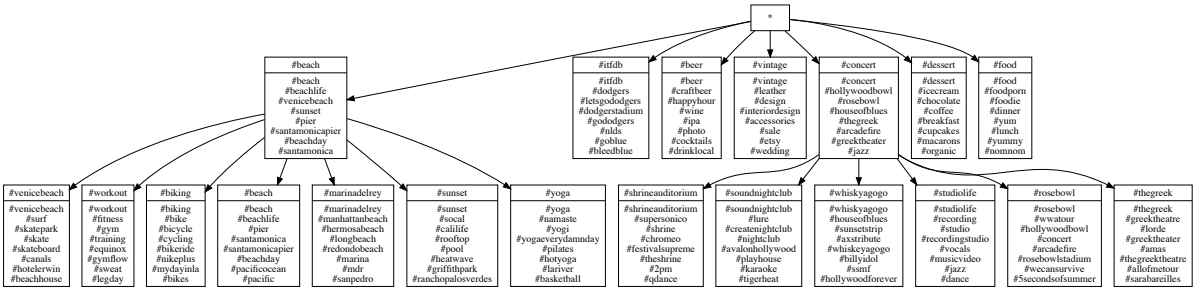

Figure 7.2: Tweet Taxonomy generated by TaxonGen under topics '*' (level 1), '#beach' (level 2), and '#concert' (level 2)

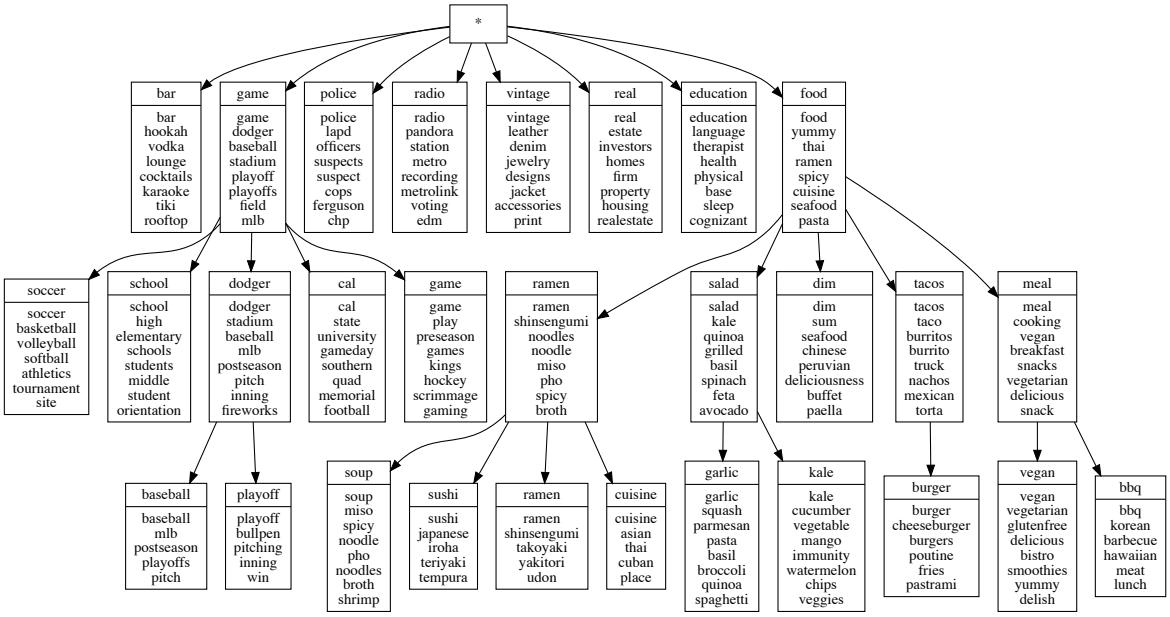

Figure 7.3: Tweet Taxonomy generated by TaxonGen with Line(Graph Embedding)

7.1.3 TaxonGen with Graph Embedding

Figure 7.3 shows a taxonomy generated by TaxonGen with graph embedding. Compared to SkipGram, graph embedding detects more topics in both width and depth. At level 1, eight topics are detected, which are bar, game, police, radio, vintage, real, education, and food. Compared to SkipGram, terms in every topic are more meaningful and specific. Taking the game topic as an example, graph embedding detects not only general terms like playoff and baseball but also more specific terms such as mlb. It is reasonable because we count the co-occurrence for every pair of terms in a tweet, however, SkipGram only considers terms in the context window.

Nonetheless, the incapability to distinguish between quality nouns and adjectives or phrases makes some noise in the taxonomy. Taking the food topic as an example, adjectives like spicy and yummy, which occurs frequently when people talk about food, have been selected in the topic. However, those adjectives don't bring much meaningful information because they co-occur with many different words. Another issue is the phrase. Taking the dim topic, which is a subtopic of food, as an example, it's quite straight forward the dim sum should be considered as a single phrase but not two separate terms. Thus, it's worth exploring how to identify phrases during the taxongen construction.

7.2 DEEP EVENT DETECTION EXPERIMENTS

7.2.1 Experiment Setup

The dataset in the experiment is a tweets stream from LA, which is generated by pick some popular hashtags and convert them to event tags. For the encoder, a GRU with 3 layers is used. Adam optimizer is used for both encoder and decoder during training phase.

7.2.2 Qualitative Analysis

Table 7.1 shows three events, which are MLB, NBA, and Foodie, that our model detects on the LA tweets stream. Within each event, 5 top-ranked tweets are picked. In the first topic, all 5 tweets have specifically contains hashtags, words, and phrases that are related to MLB. For example, all 5 tweets are related to LA Dodgers. Popular hashtags like $\#Dodgers$ occurs frequently. For the second topic, all tweets are relevant to NBA. The the first and the fourth tweets are about LA Clippers, and the second and the third are about LA Lakers. Both are NBA teams in LA. The last event is about Foodie, which refers to people who enjoying delicious food. As shown in the table, recommendations of different kinds of food have been proposed. For example, the fourth tweet recommends a Japanese food restaurant that offers yummy ramen.

Although our models detects events and top-ranked tweets in each event is coherent, our model suffers a low recall issue for more than 90% of tweets. Figure 7.4 shows the histogram of recall score vs. number of tweets. Since every tweet may have more than one event tag, the top 5 predicted event tags are picked to measure the recall score. As shown in the figure, 26,000 tweets have 0 recall, which means none event tags is correctly predicted . Only about 1,111 tweets have all event tags predicted correct. A possible reason for the low recall maybe caused by the imbalance distribution of the training datasets. The original dataset is randomly splited into a 7:3 ration, where 70% of data are used as training data. It is possible that event tags, like MLB and NBA, occurs more frequently in the training dataset, which makes the model overfit them. Another possible reason is the OOV issue in the testing dataset. As we discuss in section 5, tweets are unstructured and there are always new words or phrases coming every day. The model currently ignores all OOV words and only uses words that shown in the training dataset. If a tweet contains too much OOV words, then the truncated tweets may lose lots of information, which could lead to a bad prediction.

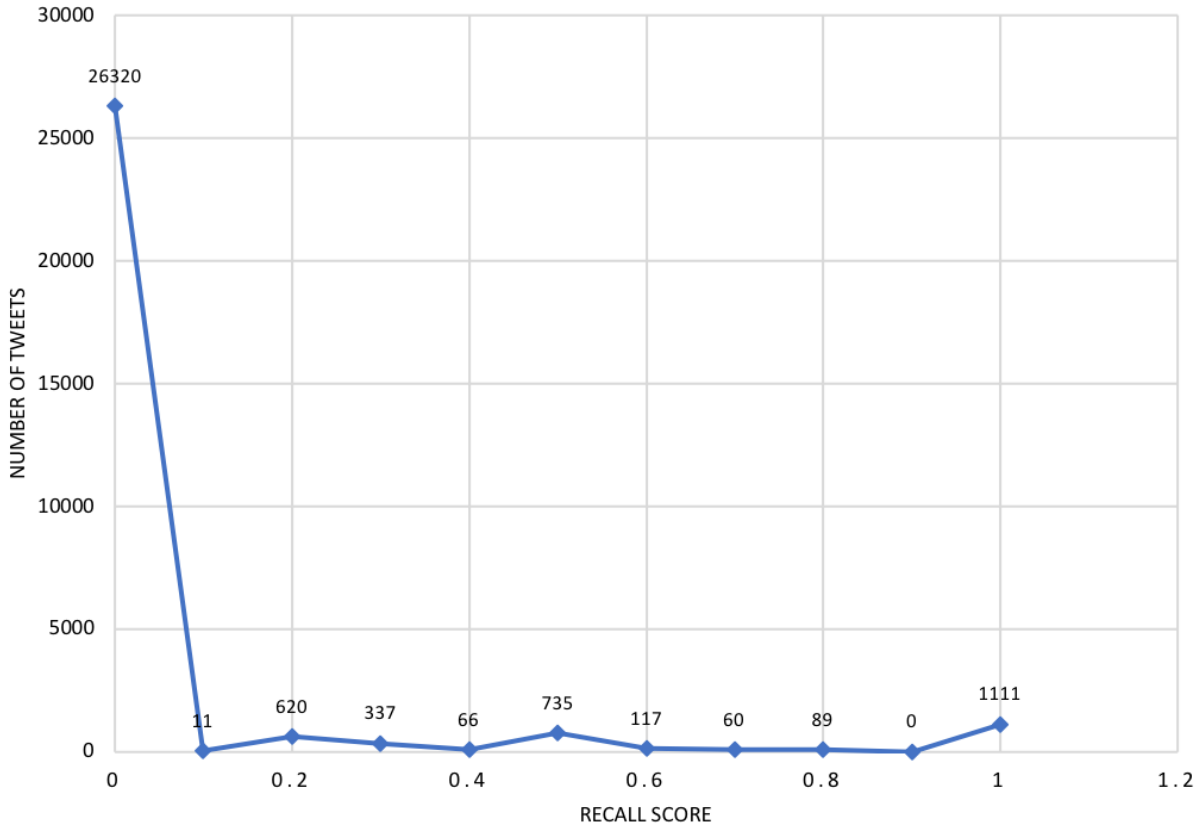

Figure 7.4: Recall Score vs. Number of Tweets on LA dataset

Table 7.1: Events Detection from LA tweets stream by the Encoder-Decoder Model

Events	Tweets
MLB	#WeLoveLA #Dodgers Los Angeles Dodgers plan to start Clayton Kershaw on short rest in NLDS Game #Dodgers baseball gods not kind today. Kershaw loses. Braun wins. Dodger fans were tested. But they will persevere. #Dodgers win or lose this umpire is pretty bad... equally stinky regardless of team. He also LOVES to throw the ball back to the pitcher. @Dodgers #Dodgers Crazy call at the plate and scoreboard has been wrong 3 times tonight re: strike/ball count. Hope we get a run out if it! @DodgersNation: #Dodgers lineup: 2B Gordon SS Ramirez 1B Gonzalez RF Kemp LF Crawford CF Ethier 3B Turner C Ellis P Haren
NBA	#WeLoveLA #Clippers Gameday Thread: Portland Trail Blazers vs. Los Angeles Clippers - Blazer's Edge #Lakers #GoLakers Lakers GM Mitch Kupchak says plan is to manage Kobe Bryant's minutes #Lakers #GoLakers Kobe Bryant To Operate Offense From Below Free Throw Line #WeLoveLA #Clippers Ray Allen rumors: Doc Rivers says Allen won't join the Los Angeles Clippers IF I PLAY OR COACH FOR THE HAWKS IM MAD AT THE AMOUNT OF LAKER FANS IN THE CROWD. #ESPN #nba #FanNight #LakersvsHawks #lakerfans
Foodie	A couple of good ways to use chefkate's homemade nut butters... On #glutenfree rice cakes with fresh #SantaMonica Bus. Park 12+ #foodtruck #lunch with parking! Serving our new #ShishitoPeppers bowl and wild hot #salmon Yellow curry with chard, bell pepper, onions and mushrooms on top of quinoa. #MeatlessMonday #vegan #curry Hippie Ramen comes w/wavy thicker noodles than the thin ones Tatsu usually serves. #food #ramen #sawtelle #losangeles Try our King Tut wrap filled with grilled chicken breast, sauted onion and tomato & falafel. #foodie #MiddleEastFood #LateNight #Hollywood

Chapter 8: Conclusion

In this thesis, I have demonstrated the Tweets2Cube system. Tweets2Cube helps users to organize unstructured social media resource into structred cube along time, location and time dimensions. It also offers users multiple spatiotemporal models to explore different mobility patterns and text summarization. As shown in our experiments, all models offers acceptable result.

However, there are still improvement that can be made. First of all, the taxonomy model mines more specific information due to the characteristic of tweets. The encoder-decoder model predicts non-existing phrases from tweets, which can be used to supplement general information for tweets. There can be an online tweets processing phase that uses the encoder-decoder model to predict general phrases, such as sports and food, for each tweet. The TaxonGen algorithm is then applied on pre-processed tweets to build the Taxonomy.

It's also worth exploring different algorithm for taxonomy construction. TaxonGen uses hierarchical clustering to build the k-ary tree structure. It may also be interesting to check how hierarchical graph partition algorithm performs on this problem.

References

[1] C. Zhang, K. Zhang, Q. Yuan, H. Peng, Y. Zheng, T. Hanratty, S. Wang, and J. Han, "Regions, periods, activities: Uncovering urban dynamics via cross-modal representation learning," in *WWW*, 2017, pp. 361–370.

[2] C. Zhang, G. Zhou, Q. Yuan, H. Zhuang, Y. Zheng, L. Kaplan, S. Wang, and J. Han, "Geoburst: Real-time local event detection in geo-tagged tweet streams," in *SIGIR*, 2016, pp. 513–522.

[3] C. Zhang, K. Zhang, Q. Yuan, L. Zhang, T. Hanratty, and J. Han, "Gmove: Group-level mobility modeling using geo-tagged social media," in *KDD*, 2016, pp. 1305–1314.

[4] C. Zhang, "Taxongen: Constructing topical concept taxonomy by adaptive term embedding and clustering."

[5] A. Ritter, S. Clark, Mausam, and O. Etzioni, "Named entity recognition in tweets: An experimental study," in *EMNLP*, 2011, pp. 1524–1534.

[6] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013, pp. 3111–3119.

[7] A. Banerjee, I. S. Dhillon, J. Ghosh, and S. Sra, "Clustering on the unit hypersphere using von mises-fisher distributions," *Journal of Machine Learning Research*, vol. 6, pp. 1345–1382, 2005.

[8] R. Meng, S. Zhao, S. Han, D. He, P. Brusilovsky, and Y. Chi, "Deep keyphrase generation," *arXiv preprint arXiv:1704.06879*, 2017.

[9] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *WWW*, 2015, pp. 1067–1077.

[10] M. A. Hearst, "Automatic acquisition of hyponyms from large text corpora," in *Proceedings of the 14th conference on Computational linguistics-Volume 2*. Association for Computational Linguistics, 1992, pp. 539–545.

[11] S. Brin, "Extracting patterns and relations from the world wide web," in *Selected Papers from the International Workshop on The World Wide Web and Databases*, ser. WebDB '98. London, UK, UK: Springer-Verlag, 1999. [Online]. Available: http://dl.acm.org/citation.cfm?id=646543.696220 pp. 172–183.

[12] A. Panchenko, S. Faralli, E. Ruppert, S. Remus, H. Naets, C. Fairon, S. P. Ponzetto, and C. Biemann, "Taxi at semeval-2016 task 13: a taxonomy induction method based on lexico-syntactic patterns, substrings and focused crawling," in *Proceedings of the 10th International Workshop on Semantic Evaluation*. San Diego, CA, USA: Association for Computational Linguistics, 2016.

[13] J. Seitner, C. Bizer, K. Eckert, S. Faralli, R. Meusel, H. Paulheim, and S. P. Ponzetto, "A large database of hypernymy relations extracted from the web." in *LREC*, 2016.

[14] G. Grefenstette, "Inriasac: Simple hypernym extraction methods," *arXiv preprint arXiv:1502.01271*, 2015.

[15] S. P. Ponzetto and M. Strube, "Deriving a large scale taxonomy from wikipedia," in *AAAI*, vol. 7, 2007, pp. 1440–1445.

[16] E. Agichtein and L. Gravano, "Snowball: Extracting relations from large plain-text collections," in *Proceedings of the Fifth ACM Conference on Digital Libraries*, ser. DL '00. New York, NY, USA: ACM, 2000. [Online]. Available: http://doi.acm.org/10.1145/336597.336644 pp. 85–94.

[17] J. Zhu, Z. Nie, X. Liu, B. Zhang, and J.-R. Wen, "Statsnowball: a statistical approach to extracting entity relationships," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 101–110.

[18] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr, and T. M. Mitchell, "Toward an architecture for never-ending language learning." in *AAAI*, vol. 5. Atlanta, 2010, p. 3.

[19] N. Nakashole, G. Weikum, and F. Suchanek, "Patty: a taxonomy of relational patterns with semantic types," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 2012, pp. 1135–1145.

[20] M. Jiang, J. Shang, T. Cassidy, X. Ren, L. M. Kaplan, T. P. Hanratty, and J. Han, "Metapad: Meta pattern discovery from massive text corpora," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 877–886.

[21] M. Bansal, D. Burkett, G. De Melo, and D. Klein, "Structured learning for taxonomy induction with belief propagation," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2014, pp. 1041–1051.

[22] R. Fu, J. Guo, B. Qin, W. Che, H. Wang, and T. Liu, "Learning semantic hierarchies via word embeddings," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2014, pp. 1199–1209.

[23] T. L. Anh, Y. Tay, S. C. Hui, and S. K. Ng, "Learning term embeddings for taxonomic relation identification using dynamic weighting neural network," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 403–413.

[24] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 224–227, 1979.

[25] H. Yang and J. Callan, "A metric-based framework for automatic taxonomy induction," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*. Association for Computational Linguistics, 2009, pp. 271–279.

[26] X. Liu, Y. Song, S. Liu, and H. Wang, "Automatic taxonomy construction from keywords," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 1433–1441.

[27] C. Wang, M. Danilevsky, N. Desai, Y. Zhang, P. Nguyen, T. Taula, and J. Han, "A phrase mining framework for recursive construction of a topical hierarchy," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 437–445.

[28] T. L. Griffiths, M. I. Jordan, J. B. Tenenbaum, and D. M. Blei, "Hierarchical topic models and the nested chinese restaurant process," in *Advances in neural information processing systems*, 2004, pp. 17–24.

[29] D. Downey, C. Bhagavatula, and Y. Yang, "Efficient methods for inferring large sparse topic hierarchies," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. 1, 2015, pp. 774–784.

[30] P. D. Turney, "Learning algorithms for keyphrase extraction," *Information retrieval*, vol. 2, no. 4, pp. 303–336, 2000.

[31] Z. Liu, X. Chen, Y. Zheng, and M. Sun, "Automatic keyphrase extraction by bridging vocabulary gap," in *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 2011, pp. 135–144.

[32] A. El-Kishky, Y. Song, C. Wang, C. R. Voss, and J. Han, "Scalable topical phrase mining from text corpora," *Proceedings of the VLDB Endowment*, vol. 8, no. 3, pp. 305–316, 2014.

[33] J. Shang, J. Liu, M. Jiang, X. Ren, C. R. Voss, and J. Han, "Automated phrase mining from massive text corpora," *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[34] T. Tomokiyo and M. Hurst, "A language model approach to keyphrase extraction," in *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment-Volume 18*. Association for Computational Linguistics, 2003, pp. 33–40.

[35] J. Liu, J. Shang, C. Wang, X. Ren, and J. Han, "Mining quality phrases from massive text corpora," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 2015, pp. 1729–1744.

[36] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[37] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[38] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," *arXiv preprint arXiv:1509.00685*, 2015.

[39] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang et al., "Abstractive text summarization using sequence-to-sequence rnns and beyond," *arXiv preprint arXiv:1602.06023*, 2016.

[40] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," *arXiv preprint arXiv:1704.04368*, 2017.

[41] J. Gu, Z. Lu, H. Li, and V. O. Li, "Incorporating copying mechanism in sequence-to-sequence learning," *arXiv preprint arXiv:1603.06393*, 2016.

[42] C. Zhang, L. Li, K. Zhou, and C. Hong, "Embedding-based user mobility pattern mining," 2017.

[43] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.