

# Fine-Grained Location Extraction from Tweets with Temporal Awareness

Chenliang Li

State Key Lab of Software Engineering  
School of Computer, Wuhan University, China  
cllee@whu.edu.cn

Aixin Sun

School of Computer Engineering  
Nanyang Technological University, Singapore  
axsun@ntu.edu.sg

## ABSTRACT

Twitter is a popular platform for sharing activities, plans, and opinions. Through tweets, users often reveal their location information and short term visiting plans. In this paper, we are interested in extracting fine-grained locations mentioned in tweets with temporal awareness. More specifically, we like to extract each point-of-interest (POI) mention in a tweet and predict whether the user has visited, is currently at, or will soon visit this POI. Our proposed solution, named PETAR, consists of two main components: a *POI inventory* and a *time-aware POI tagger*. The POI inventory is built by exploiting the crowd wisdom of Foursquare community. It contains not only the formal names of POIs but also the informal abbreviations. The POI tagger, based on Conditional Random Field (CRF) model, is designed to simultaneously identify the POIs and resolve their associated temporal awareness. In our experiments, we investigated four types of features (*i.e.*, lexical, grammatical, geographical, and BILOU schema features) for time-aware POI extraction. With the four types of features, PETAR achieves promising extraction accuracy and outperforms all baseline methods.

## Categories and Subject Descriptors

H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing—*Linguistic processing*

## Keywords

Twitter, Tweet, Location extraction, Temporal awareness, POI, CRF

## 1. INTRODUCTION

As a real-time communication platform, Twitter has attracted more than 200 million active users around the world every month. Users share about their mood, activities, and opinions through short messages limited to 140 characters. Users also use Twitter as a communication channel to coordinate meetings or other activities.

Through tweets, users often casually or implicitly reveal their current locations and short term plans where to visit next, at fine-grained granularity. Such information enables tremendous opportunities for personalization and location-based services/marketing.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGIR'14, July 6–11, 2014, Gold Coast, QLD, Australia.

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-2257-7/14/07 ...\$15.00.

<http://dx.doi.org/10.1145/2600428.2609582>.

For example, a user from New York city posts a tweet: “heading off to watch G.I. Joe at sunshine”. From this tweet, we can infer that the user is soon to visit Landmark’s Sunshine Cinema.<sup>1</sup> In this context, promotions related to the cinema and recommendations of nearby restaurants become much relevant to the user. Another example, a user may like to view advertisement about *The Smile* (but not *L’Artusi*) after her tweet: “just back from L’Artusi, wonderful dinner :> like to try the smile tmr for lunch”.<sup>2</sup> Both examples highlight the importance of recognizing fine-grained locations (*e.g.*, cinema and restaurant) and their associated temporal awareness (visited or to visit) to support more effective location-based services/marketing.

Recently, geo-locating tweets and inferring users’ locations have become a hot research topic for location-based services, advertisement, personalization and others [4, 7, 9, 15, 19]. However, most existing studies largely rely on GPS/human-annotated tweets to infer the location of a user or a tweet at *coarse level* of granularity, ranging from country, state, to city levels. Some studies further investigate the interplay between the geographic locations and user interests [5, 6]. However, these techniques are far from sufficiency for precise location-based services/marketing. Either the granularity is not fine enough to identify potential customers, or the extracted information/topic is too coarse for targeted marketing.

In this paper, we are interested in extracting fine-grained locations mentioned in tweets with temporal awareness. More specifically, if a user mentions a point-of-interest (POI) (*e.g.*, restaurant, shopping mall, bookstore or any other fine-grained locations) in her tweet, we are interested in extracting *the name of the POI*, and at the same time, predicting *whether the user has visited, is currently at, or will soon visit this POI* (*i.e.*, the temporal awareness of the POI in this tweet).<sup>3</sup> We believe such information greatly facilitates fine-grained location-based services/marketing and personalization. However, both subtasks of extracting POI names and predicting the associated temporal awareness are challenging.

- First, tweets are written in free writing styles and are limited to 140 characters, leading to the predominant usage of colloquial language, misspellings and grammatical errors. Users often use short names or informal abbreviations to mention POIs. Existing studies have demonstrated significant performance degradation in Named Entity Recognition (NER) from tweets, where “location” is commonly considered an important type of named entity [18, 23]. For the same reason, capturing temporal awareness of POIs based on either existing work [27] or temporal expression extraction tools (*e.g.*, SUTIME [2] TARSQI [28]) becomes less practical.

<sup>1</sup><http://www.landmarktheatres.com/market/NewYork/SunshineCinema.htm>

<sup>2</sup>*L’Artusi* and *The Smile* are two restaurants located at New York city.

<sup>3</sup>In our following discussion, we use the term POI to refer fine-grained location.

- Second, many POI names are ambiguous and may refer to different concepts in different contexts (*i.e.*, location name ambiguity). The aforementioned restaurant *The Smile* is one example where the word *smile* is a commonly used English word and does not refer to location names in most cases. The word *mac* may refer to Apple’s products and McDonald’s chain restaurant or product. In this sense, simply matching a tweet against a pre-built gazetteer leads to an ineffective solution. The situation becomes even more complicated by considering the noisy and informal nature of tweets.

To slightly simplify the problem, in this study, we only focus on the tweets posted by users from a predefined geographical region (*e.g.*, a city). This simplification enables us to utilize rich background information about the region. In our data analysis and experiments, we used tweets published by Singaporean users. A user is considered Singapore-based if she specifies Singapore in the location field of her Twitter profile.

Our proposed solution to the above problem, named PETAR (POI Extractor with Temporal Awareness), consists of two main components: a *POI inventory* and a *time-aware POI tagger*. The POI inventory is a collection of words and phrases, each of which is either a POI name or a part of a POI name. To ensure that our POI inventory contains not only formal names of POIs but also informal abbreviations, we construct the inventory by exploiting the Foursquare check-ins, collaboratively contributed by users from the same geographical region.<sup>4</sup> Each entry in the POI inventory is a *candidate POI name* which may be used to refer a POI. Note that these candidate POI names are very likely to be ambiguous and many of them are incomplete names. The aforementioned *mac* and *smile* are both candidate POI names. Another example is *popular* which is a commonly used word in English but may refer to the *Popular Bookstore* in Singapore. To disambiguate a candidate POI name mentioned in a tweet and at the same time to resolve its temporal awareness, we develop a *time-aware POI tagger*. The time-aware POI tagger is based on Conditional Random Field (CRF), a widely used model for sequence labeling. The tagger takes POI inventory as a knowledge base and utilizes four types of features (*i.e.*, lexical, grammatical, geographical, and BIOES schema features) to label POI names and their temporal awareness. In this sense, the POI inventory can be considered as a noisy version of a gazetteer to the CRF classifier. However, we demonstrate in our experiments that the POI inventory is critical for our task because the linguistic features in tweets are much less reliable compared with formal text. To summarize, the main contributions of this paper are:

1. We propose and formalize the problem of fine-grained location extraction from tweets with temporal awareness. We conduct data analysis and make four observations on Twitter user sharing fine-grained locations and short-term visiting plans.
2. We propose a mechanism to build a POI inventory without human efforts by exploiting the crowd wisdom of Foursquare community. The POI inventory includes not only the formal names of POIs but also their informal short forms and abbreviations.
3. We propose and investigate four types of features, lexical, grammatical, geographical, and BIOES schema features, for learning the time-aware POI tagger. This CRF-based classifier is able to simultaneously extract POI names in tweets and resolve temporal awareness. All the four types of features are easy to derive, enabling real-time tweet processing.
4. We conducted extensive experiments to evaluate PETAR and compare it with baseline methods. PETAR achieves promising accu-

<sup>4</sup><https://foursquare.com/>.

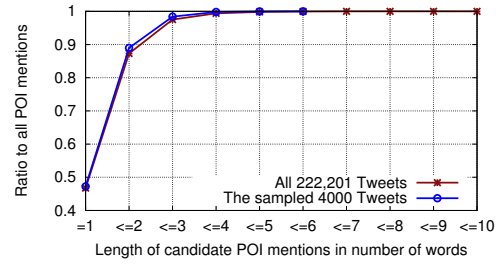


Figure 1: Length distribution of the candidate POI mentions

racy and outperforms all baseline methods. We also evaluated the effectiveness of the proposed features.

The rest of this paper is organized as follows. We start with a data analysis in Section 2 to illustrate that many users reveal their locations and short-term plans in tweets. In Section 3, we define our problem and give an overview of PETAR. The POI inventory and time-aware POI tagger are detailed in Sections 4 and 5 respectively. Section 6 presents the experiments. After related work in Section 7, we conclude this paper in Section 8.

## 2. DATA ANALYSIS AND OBSERVATIONS

Our data analysis is conducted on 4,331,937 tweets published by 19,256 unique Singaporean users during June 2010. A Singaporean user means that the user specifies Singapore in the location field of her Twitter profile. To be detailed in Section 4, the POI inventory used in this analysis consists of 36,201 candidate POI names extracted from 239,499 Foursquare check-in tweets made by Singaporean users.

All tweets are cleaned by removing HTML special characters (*e.g.*, “&gt;” is replaced with “>”). Each tweet is then matched against the candidate POI names in the POI inventory. If a span of words in a tweet matches more than one candidate POI name, then the longest match is preferred. For example, the phrase “popular bookstore” in a tweet has three matches “popular”, “bookstore”, and “popular bookstore”, then the last match is taken. In some rare cases, if there is more than one longest match, then the first match is taken based on the word sequence. For example, if three words  $w_1w_2w_3$  in a tweet match two candidate POI names  $w_1w_2$  and  $w_2w_3$ , then the first match  $w_1w_2$  is taken. Each candidate POI name matched in a tweet is also known as a *candidate POI mention* in the tweet.

**OBSERVATION 1.** *Many users reveal their fine-grained locations in their tweets.*

After the matching process, there are 222,201 tweets (or 5.1% of the 4.3 million tweets in our dataset) that each contains at least one candidate POI mention. Although 5.1% is not a very high percentage, these 222,201 tweets were published by 13,758 unique users, or 71.4% of all users in our dataset. This percentage rises to 91.3% if we only consider the users who had published at least 20 tweets. The high percentage suggests that many users casually or implicitly reveal their locations (in the form of fine-grained POIs like restaurant or shopping mall names) in tweets. Based on our manual annotation, to be reported shortly, about half of the candidate POI mentions indeed refer to fine-grained locations.

**OBSERVATION 2.** *The candidate POI mentions are mostly very short with one or two words. Many of the mentions are partial location names.*

**Table 1: Example POI labels in tweets. The location names are in boldface, followed by their labels in brackets.**

$t_1$	Soccer fever at <b>mac</b> [ $POI_z$ ] now.!
$t_2$	@username yes i will msg u. do u mind eating at <b>bukit panjang plaza</b> [ $POI_f$ ]? cos i've got stuff to collect at <b>popular</b> [ $POI_f$ ] at night. :(
$t_3$	We're all for Asian delights! <b>Thai express</b> [ $POI_z$ ] today, <b>suki sushi</b> [ $POI_f$ ] tomorrow

Figure 1 plots the length distribution of the candidate POI mentions in the 222,201 tweets. Observe that nearly half or 46.7% of the candidate POI mentions are unigrams (*i.e.*, a single word), leading to very high chance of ambiguity. The most frequent candidate POI mention is *mac* which is often used to refer both Apple products and McDonald’s, the chain of fast food restaurants. Longer candidate POI mentions with 3 or more words are very rare, about 2.5%. Moreover, about 41.6% of the candidate POI mentions are partial POI names. Note that the POI inventory captures both full and partial names of POIs (*e.g.*, “popular” is a partial name of “popular bookstore”). The short, ambiguous, and partial names make the problem of POI name extraction extremely challenging. On the other hand, our observation is consistent with the nature of the tweet language.

**OBSERVATION 3.** *About half of the candidate POI mentions indeed refer to locations and their associated temporal awareness can be determined.*

To investigate whether a candidate POI mention truly refers to a location and to determine its possible temporal awareness, we randomly sampled 4,000 tweets for manual annotation, from the 222,201 tweets. Plotted in Figure 1, the length distribution of the candidate POI mentions in the sampled 4000 tweets is the same as the 222,201 tweets. For each candidate POI mention in the sampled tweets, a human annotator is asked to assign one of the 5 labels:  $POI_p$ ,  $POI_z$ ,  $POI_f$ ,  $NPOI$ ,  $Unknown$ .

The first three labels indicate that a candidate POI mention indeed refers to a location. The three subscripts *p* (past), *z* (present), and *f* (future) indicate the temporal awareness of the POI, *i.e.*, the user has visited ( $POI_p$ ), is currently at ( $POI_z$ ) or will be visiting ( $POI_f$ ) the POI. The label  $NPOI$  means that the mention does not refer to a location, and the last label  $Unknown$  is assigned if the annotator cannot determine whether the mention is a location or the annotator cannot resolve the temporal awareness.

Table 1 lists three example tweets with their assigned labels.<sup>5</sup> The POI names are in boldface followed by their labels in brackets. In the first tweet  $t_1$ , *mac* is assigned  $POI_z$ , where the user is reporting an ongoing event (watching soccer games) at a McDonald’s chain restaurant. Note that some of the labels may not be purely determined based on the single tweet alone. For example, it seems also reasonable to label “Thai express” in  $t_3$  with  $POI_p$  or  $POI_f$  based on this single tweet. To facilitate the annotation process, for each tweet to be labeled, we provide the previous and the following two tweets published by the same user. These five tweets and their timestamps together provide the context for the annotation. Moreover, all our human annotators have stayed in Singapore for more than 10 years with good knowledge about the city. They are also encouraged to use search engines to refine their annotations.

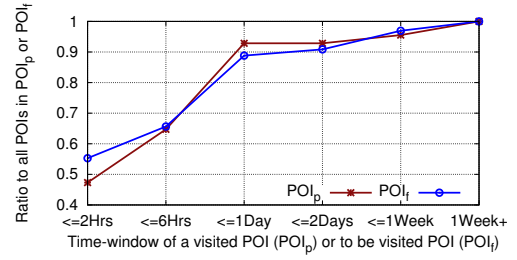
From the 4,000 sampled tweets, 320 tweets are filtered away for containing mostly words in other languages than English.<sup>6</sup> In

<sup>5</sup> All Twitter usernames are replaced by @username in this paper.

<sup>6</sup> Singapore is a multi-cultural country. Some tweets are written in mixture of English, Chinese, Malay, Bahasa or other languages.

**Table 2: Distribution of the candidate POI mention labels**

# $POI_p$	# $POI_z$	# $POI_f$	# $NPOI$	# $Unknown$	Total
307	1,202	547	1,801	120	3,977
Total #POIs: 2,056			-	-	



**Figure 2: Time-window of the visits for  $POI_p$  and  $POI_f$**

the remaining 3,680 tweets, there are 110 tweets within which all candidate POI mentions are labeled  $Unknown$ . In the following, we report the annotation results of the remaining 3,570 tweets.

In these 3,570 tweets, there are 3,977 candidate POI mentions which involve 906 distinct candidate POI names. Table 2 reports the distribution of the labels assigned to the 3,977 candidate POI mentions. Observe that 51.7% of the candidate POI mentions are truly locations. Among them, the numbers of  $POI_p$ ,  $POI_z$ , and  $POI_f$  are 14.9%, 58.5%, and 26.6% respectively. That is, slightly more than half of POI mentions are indications of users being at the current locations (*i.e.*,  $POI_z$ ). This observation is consistent with the earlier finding that Twitter is an individual news media [10, 25, 29]. There are 26.6% of POI mentions are for future visit plans (*i.e.*,  $POI_f$ ). The high percentage of current and future location mentions makes Twitter an ideal source for POI-targeted advertisement and marketing.

**OBSERVATION 4.** *Among all POIs that were visited (labeled  $POI_p$ ) or to be visited (labeled  $POI_f$ ), about 90% of the visits to these POIs happen within a day.*

To better understand the temporal awareness expressed by users, if a POI mention is labeled either  $POI_p$  or  $POI_f$ , the human assessor is asked to further determine the time-window of the visit using the 5 tweets as context, *i.e.*, the time-window within which the user has visited or will be visiting the POI. We use 6 time-windows: 2Hrs, 6Hrs, 1Day, 2Days, 1Week, 1Week+, and  $NT$ . For example, from tweet: “@username heading to gucci at paragon now!”, we infer that the user is going to visit “paragon” within 2 hours (2Hrs) because traveling from one point to another within Singapore usually takes shorter than 2 hours.  $NT$  is used if the time-window cannot be determined from the context. For example, we cannot determine the time-window in tweet: “I wanna go Sentosa, VivoCity, Clarke Quay, and Overseas!! : (“ by considering this tweet with the other four temporally related tweets.

Out of the 854 POIs with labels  $POI_p$  and  $POI_f$ , 144 are labeled as  $NT$ . The distribution of the time-windows for the remaining 710 POIs is plotted in Figure 2. It shows that both  $POI_p$  and  $POI_f$  demonstrate very similar patterns. About 50% of the visits happen within 2Hrs and more than 90% of the visits happen within a day. That is, Twitter users reveal very short-term visiting history or plan, mostly within a day. This observation suggests that efficiency is also an important factor to support targeted fine-grained location-based services/marketing given the large number of tweets.

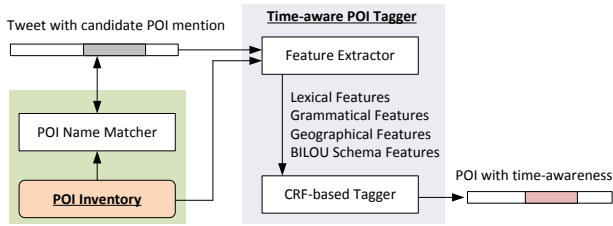


Figure 3: Overview of PETAR

### 3. TIME-AWARE POI EXTRACTION

We first define the problem of *POI extraction with temporal awareness* and then give an overview of PETAR, the proposed solution.

#### 3.1 Problem Definition

Given a tweet  $t$  published by a user from a predefined geographical region, the task of *POI extraction with temporal awareness* is to identify all locations or POIs mentioned in  $t$  and to associate each POI mention with a temporal awareness label  $c_i \in \{c_1, c_2, \dots, c_k\}$ . In other words, let  $\ell$  be a POI mentioned in tweet  $t$ , we aim to extract all POI and temporal awareness label pairs,  $\{(\ell, c)\}$  from  $t$ . Following [17, 21], we define a POI as a focused geographic entity such as district, area, street, road, or a specific point location such as hotels, landmarks, schools, shopping centers and restaurants etc. The temporal awareness labels can be defined in a task-dependent manner, for example  $\{last-six-hours, present, next-six-hours\}$ .

From Observation 4, more than 90% of the visits to POIs happen within a day. In this study, we therefore simply use three temporal awareness labels  $\{past, present, future\}$  and do not use more fine-grained time-windows. That is, we use  $POI_p$ ,  $POI_z$ , and  $POI_f$  to indicate the temporal awareness of the extracted POIs.

#### 3.2 Overview of Petar

Given the problem definition, at first glance, it may seem that the problem can be easily addressed by using an existing off-the-shelf NER package to extract locations and then to label these locations using temporal awareness classifier. However, as discussed in Section 1, given the short and noisy nature of tweets, named entity extraction (including locations) from tweets remains challenging. Moreover, a gazetteer with formal names of POIs does not necessarily help much because most Twitter users use short forms or abbreviations to mention POIs in tweets (see Observation 2).

In this paper, we propose to use a POI inventory and a time-aware POI tagger to extract POIs and assign temporal awareness labels. The solution is named PETAR and Figure 3 gives an overview. The construction of the POI inventory exploits the crowdsourcing knowledge embedded in the tweets associated with Foursquare check-ins. Therefore, the POI inventory contains words or phrases that are commonly used by Twitter users to mention POIs. With such a “noisy version of a gazetteer”, the candidate POI mentions in a tweet are pre-labeled (*i.e.*, the BILOU features in Figure 3) and are then passed to a time-aware POI tagger for prediction. The POI tagger, based on the widely used linear-chain CRF model, also takes in three other types of features: lexical, grammatical, and geographical features derived from the tweet and also the POI inventory. Next, we detail the construction of POI inventory in Section 4 and the time-aware POI tagger in Section 5.

### 4. POI INVENTORY

The POI inventory is constructed by extracting the POI names mentioned in tweets that are associated with Foursquare check-ins. Foursquare is a popular location-based social networking (LBSN)



Figure 4: POIs covered by 1K sample Foursquare check-ins

Table 3: Example tweets associated with Foursquare check-ins

$t_1$	I'm at Mac @ Bukit Panjang Plaza
$t_2$	I'm at ITE College Central MacPherson Campus Main (201 Circuit Road)
$t_3$	Birthday dinner (@ Ambush @ JP w/ 2 others)
$t_4$	Watching "Hello Stranger" (@ Golden Village Cinema 9 @ Plaza Singapura)

platform. It has attracted more than 45M people worldwide with billions of check-ins. A check-in may be associated with a “*check-in tweet*” which contains formal or informal POI names. Because of the large user base and large number of check-ins, it is expected that the POI coverage for a given geographical region is broad or even exhaustive in a fine-grained scale. For illustration purpose, Figure 4 plots the locations covered by 1,000 randomly sampled Foursquare check-ins in Singapore. The region with the highest density of check-ins, *i.e.*, the central part of the map, is the downtown area, which attracts a large number of visits everyday.

Next, we report the details of POI inventory construction. Note that, the technique presented here is not restricted to Foursquare. Check-in data from other LBSN services like Facebook Places, Gowalla can be easily adopted.

#### 4.1 Foursquare Check-in Dataset

We collected 259,204 check-ins from Foursquare, which were made by Twitter users in Singapore between August 2010 and July 2011. Each check-in in this collection is associated with a tweet (called check-in tweet) and a latitude/longitude coordinate. After removing the check-in tweets with non-Latin characters, we have 239,499 tweets left. Table 3 demonstrates the two kinds of check-in tweets observed in the collection. The first two tweets  $t_1$  and  $t_2$  simply report the users’ current locations, while the other two tweets  $t_3$  and  $t_4$  report users’ activities at the locations. The locations like “(@ Golden Village Cinema 9 @ Plaza Singapura)” are specified by users, and automatically formatted by Foursquare. The location names may also appear in its abbreviated form like *JP* in  $t_3$ , which refers to *Jurong Point*, a shopping mall in the western part of Singapore.<sup>7</sup> Note that, the check-in tweets are solely used for constructing the POI inventory and not used for evaluating PETAR.

#### 4.2 POI Inventory Construction

Because check-in tweets are relatively well formatted, the POI names can be reliably extracted by applying handcrafted rules with regular expressions. For example, from tweet  $t_1$  in Table 3, we obtain two POI names: *mac* and *bukit panjang plaza*; from  $t_4$  we obtain *golden village cinema 9* and *plaza singapura*. From all check-in tweets, we extracted 37,160 POI names. The average length of

<sup>7</sup><http://www.jurongpoint.com.sg/>



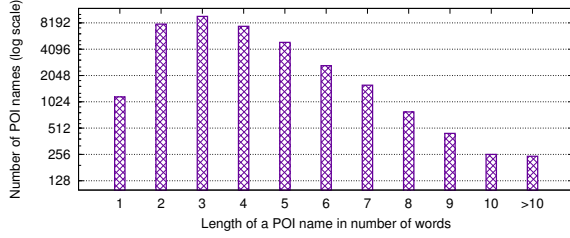


Figure 5: Length distribution of location names from check-ins

the POI names is 3.9 words. Plotted in Figure 5, most POI names are in the range of 2 to 5 words.

In Twitter, people often mention POIs with abbreviations or partial names, assuming the audience’s context-awareness [16]. For example, *popular* is used in tweets for *Popular Bookstore*. To ensure our POI inventory captures POI names mentioned in tweets, we further augment the POI inventory with partial names. More specifically, for each of the extracted 37,160 POI names, if a name consists of 2 or more words, we extract its partial names by taking all the sub-sequences of the name (up to 5 words). The length of a partial name is limited to 5 words because longer names are less likely to appear in tweets (see Observation 2). The stopwords (e.g., *the*, *at*, *of*) are ignored and employed as separators in this process. For example, partial names *frolick*, *bukit*, *batok*, *bukit batok* are extracted from the POI name *frolick at bukit batok* where “at” is a stopword and a separator. The major issue of this process, however, is generating many invalid location names. For example, partial name *east bus* extracted from *jurong east bus interchange* is invalid. To partially address this issue, entries that appear in fewer than 5 check-in tweets are removed from the POI inventory. This filtering process removes not only most invalid partial names but also some noisy single-word POI names like *office* and *somewhere*. After filtering, we have 36,201 entries in the POI inventory including POI names directly extracted from Foursquare check-in tweets and their partial names. Each entry is known as a candidate POI name.

Unfortunately, although the check-in tweets are well formatted by Foursquare, being a crowdsourcing knowledge base, many candidate POI names are directly contributed by users and are noisy. For example, we get *my home* as a candidate POI name from check-in tweet: “I’m at My Home @ Serangoon Ave 3 (Serangoon Avenue 3)”; similarly, we get *my room*, *my work place*, and *my bed* as candidate names. Very unlikely such names are POIs. Moreover, many candidate names are ambiguous, such as *mac* and *popular*. Even a tweet mentions a candidate POI name, the mention may not be a true POI. Reported in Observation 3, about half of the mentions truly refer to POIs with determinable temporal awareness labels. In the following, we develop a time-aware POI tagger that simultaneously disambiguates the candidate POI mentions and resolves the temporal awareness.

## 5. TIME-AWARE POI TAGGER

Prediction of whether a candidate POI mention is truly a POI and its temporal awareness largely relies on the context expressed in the tweet. For example, given a tweet “Off to jp now! Hope it DOESN’T rain”, the contextual cues like “*off to*” and “*now!*” are important information for the prediction of candidate POI mention *jp*. Conditional Random Fields (CRF) therefore becomes a natural choice for our task. CRF takes context into account by allowing arbitrary complex dependencies among class variables [11]. Also, it makes independence assumption for observation variables (i.e., features), enabling it for fast learning and inference.

The proposed time-aware POI tagger is based on the widely applied linear-chain CRF model, which models the output classes as a sequence. To encode the contextual knowledge for candidate POI mention disambiguation and temporal awareness classification, we investigate four classes of features: *lexical*, *grammatical*, *geographic* features, and BILOU schema features. In the following, we use  $w_i$  to denote the  $i$ -th word in tweet  $t$ ,  $x_i$  to denote its lowercased form, and  $\ell$  to denote a candidate POI name or mention.

### 5.1 Lexical Feature

Lexical features are widely used in NER tasks and proven to be effective [22, 24, 31]. In our implementation, we utilize 4 basic lexical features for a word  $w_i$ , and 3 contextual features derived from the surrounding words of  $w_i$ .

**Basic lexical features of a word.** The 4 lexical features of a word  $w_i$  are: 1) the word  $w_i$  itself and its lowercased form  $x_i$ ; 2) the word shape of  $w_i$ : all-capitalized, is-capitalized, all-numeric, alphanumeric; 3) the prefixes and suffixes of  $x_i$ , from 1 to 3 characters; 4) the prior probabilities of  $x_i$  being in capitalization and in all-capitalization forms respectively. The first three features are computed based on the surface form of the word in the given tweet. The 4th feature, i.e., the prior probabilities, are estimated from the tweet collection. In our implementation, a continuous value feature is discretized by applying a greater-than threshold test at each equal interval in its range. The prior probability in the range of  $[0, 1]$  is discretized into 5 binary features using 0.2 as the interval.

**Contextual features of a word.** Context window feature is often used in NER to identify the boundaries of named entities [22, 31]. We exploit three contextual features for a word: 5) bag-of-words of the context window up to 5 words:  $x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2}$ ; 6) bag-of-words of the preceding two words  $x_{i-2}, x_{i-1}$ ; 7) bag-of-words of the following two words  $x_{i+1}, x_{i+2}$ . Note that, the last two features are proposed to distinguish the contextual cues from the left-hand side window and the right-hand side window of a word. In tweets, we observe that the left-hand side words are likely used to convey the activity associated with a POI (e.g., *off to*, *eating at*), while the right-hand side words often reflect the temporal awareness (e.g., *now*, *today*, *soon*). However, in some cases, either the left-hand or the right-hand side words are informative, while the counterpart is misleading. The POI “suki sushi” in  $t_3$  in Table 1 is an example. The right-hand side word *tomorrow* is informative for temporal awareness resolution, while its left-hand side word *today* is misleading.

### 5.2 Grammatical Feature

**Part-of-speech (POS) tag.** We use TwitterNLP, a tweet-specific NLP tool to tag each tweet.<sup>8</sup> TwitterNLP incorporates tweet-specific features and outperforms conventional POS taggers in tweet domain [23]. Based on the Penn TreeBank tagset [20], TwitterNLP provides different tag for each verb tense, providing us with grammatical cues to infer the temporal awareness of the POIs. We consider the POS tags of the current word  $w_i$  and its surrounding two words  $w_{i-1}$  and  $w_{i+1}$ .

**Word group by Brown clustering.** To tackle the issue of Out of Vocabulary (OOV) words such as informal abbreviations and misspellings, we apply Brown clustering to capture the distributional similarity of words. For example, lexical variations like *shd*, *ishould*, *shudd*, *shuld*, *shoud*, *shud*, *shld*, *sould*, *shouldd* are clustered together with the modal verb *should*. Brown clustering is an

<sup>8</sup>[http://github.com/aritter/twitter\\_nlp](http://github.com/aritter/twitter_nlp)

algorithm that groups words that appear in similar contexts in a hierarchy [1]. Each word is uniquely represented by a bit string based on its path from the root of the hierarchy. The depths of a path offer different levels of word group abstraction. For a given word, we use the 4th, 8th and 12th bits of its path to abstract its lexical variations, resulting in three features.

**Time-trend score of tweet.** To capture the temporal awareness of POIs, we manually constructed a dictionary of time-trend words as reference. The dictionary, denoted by  $\mathcal{D}$ , contains 36 commonly used words in English with manually assigned time-trend scores: 1, 0, and -1 for future-, present-, and past-trend respectively.<sup>9</sup> Example time-trend words in  $\mathcal{D}$  include modal verbs like *will*, *would*, auxiliary verbs like *was*, *be*, *is*, *am*, and adjectives or adverbs like *tomorrow*, *future*, *yesterday* etc.

Next, we compute a time-trend score for a tweet  $t$  by assigning time-trend scores to some of  $t$ 's words and then take the average of the scores assigned. The time-trend score of word  $w \in t$  is assigned through three steps.

1. If a word  $w \in t$  matches an entry in  $\mathcal{D}$ , then its time-trend score is assigned accordingly with reference to  $\mathcal{D}$ .
2. If  $w$  does not match any entry in  $\mathcal{D}$ , then we get all the words that appear in the same cluster as  $w$  based on the Brown clustering results using the path of 12 bits (*i.e.*, the finest-granular level). Let  $B_w$  be the word cluster where  $w \in B_w$ . If all the words in the intersection  $B_w \cap \mathcal{D}$  have the same time-trend score, then  $w$  is assigned with the score. Otherwise, if  $B_w \cap \mathcal{D}$  is an empty set, or the words in  $B_w \cap \mathcal{D}$  have different time-trend scores, we consider the word cluster less reliable, and  $w$  will not be assigned a time-trend score. We use  $D_T$  to denote the set of words that have been assigned time-trend scores in  $t$  in the first two steps ( $D_T \subset t$ ), because the assignment involves either direct or indirect match with  $\mathcal{D}$ .
3. Next, we exploit POS tags to assign time-trend scores to all the verbs that are in tweet  $t$  but not in  $D_T$ . Verbs tagged with VBN and VBD are assigned score -1; VBZ, VBP, VBG and VB assigned with score 0.<sup>10</sup>

The overall time-trend score  $T(t)$  of tweet  $t$  is computed as the average of the time-trend scores that have been assigned to its words. If no word in  $t$  has been assigned a score, then  $T(t) = 0$ .

**The closest verb.** While  $T(t)$  implies the overall temporal awareness of tweet  $t$ , the tweet may mention multiple POIs which are associated with different temporal awareness (*e.g.*, tweet  $t_3$  in Table 1). In this sense, for each candidate POI mention in a tweet, we further consider the closest verb to the POI mention, the tense of the verb, and the distance between the verb and the POI mention. More specifically, a tense label is assigned to the closest verb based on its POS tag. Verb with POS tags VBN or VBD is assigned the label “*pst*”; VBZ, VBP or VB the label “*pre*”; and VBG the label “*pre-p*”. Here we distinguish VBG from other verbs in present tense because present participle could indicate futurity in some context (*e.g.*, tweet “*heading to jp for dinner!*”). The distance between the closest verb to a POI mention is encoded in 11 binary features. The first 10 binary features indicate the number of words in between and the last binary feature indicates the distance is more than 10 words. An additional binary feature is used to indicate whether the closest verb is on the left-hand or right-hand side of the POI mention. If a tweet contains no verb, the aforementioned 12 features are set to “NULL”.

<sup>9</sup>The list is available at <https://sites.google.com/site/lichenliangpage/>

<sup>10</sup>VBN: past participle; VBD: past tense; VBZ: 3rd person singular present; VBP: non-3rd person singular present; VBG: gerund/present participle; VB: verb base form

**The closest time-trend word.** Besides the closest verb to a candidate POI mention, we also consider the closest word that appear in  $D_T$  (*i.e.*, time-trend words by time-trend dictionary matching) to the POI mention. Similarly, the word itself, its time-trend score, its distance to the POI mention and the indicator of being on the left-hand or right-hand side of the POI mention are used as features. If no such word exist, then the features are set to “NULL”.

### 5.3 Geographical Feature

**Spatial randomness.** Because the POI inventory is built from the check-in tweets from Foursquare. Each candidate POI name  $\ell$  is mentioned by at least one check-in tweet. Recall that check-in tweets are associated with latitude/longitude coordinates. We compute the spatial randomness of a candidate POI name  $\ell$ , denoted by  $R(\ell)$ , using spatial distribution of the check-in tweets which mention  $\ell$ . Specifically, we divide the map of Singapore into lattices with a size of 1KM  $\times$  1KM. There are 608 lattices (denoted as  $S$ ), each contains at least one check-in tweet. Let  $k_\ell$  be the total number of check-in tweets mentioning  $\ell$ , and  $k_{\ell,s}$  be the number of check-in tweets that mention  $\ell$  and fall in lattice  $s$ , then the probability of  $\ell$  being associated with  $s$  is  $P(\ell, s) = k_{\ell,s}/k_\ell$ . The spatial randomness  $R(\ell)$  of  $\ell$  is the normalized entropy:

$$R(\ell) = -\frac{1}{Z} \sum_{s \in S} P(\ell, s) \log P(\ell, s) \quad (1)$$

In Equation 1,  $Z = \log(|S|)$  is the maximum entropy value assuming uniform distribution.  $R(\ell)$  ranges from 0 to 1. The location names that appear in a single lattice have  $R(\ell) = 0$ . Chain restaurants like McDonald’s and Starbucks have much larger  $R(\ell)$  values.

**Location name confidence.** The spatial randomness measure alone can not fully describe a POI name. For example, if a POI name is mentioned by very few check-in tweets, then  $R(\ell)$  is small. On the other hand, POI names mentioned by many check-in tweets in many lattices may not necessarily names of chain restaurant/store, but common words like *home*, *room*, *bus*, *center*. We therefore propose *location name confidence* measure.

Because longer names are more likely true POIs, we measure the confidence of a candidate POI name with respect to the length of its name in number of words. Let  $\mu_i$  and  $\sigma_i$  be the average and the standard deviation of all  $k_\ell$ 's of length  $i$ , the confidence of POI name  $\ell$  of length  $i$ , denoted by  $F(\ell)$ , is defined in Equation 2, where 5 is a scaling constant.

$$F(\ell) = \frac{1}{1 + e^{-5(k_\ell - \mu_i)/\sigma_i}} \quad (2)$$

**Multiple candidate POI mention.** We observe that when multiple candidate POI names are mentioned in one tweet, all the mentions are more likely true POIs. For example, both tweets  $t_2$  and  $t_3$  in Table 1 mention two POIs. Thus, a binary feature is added to indicate whether a given tweet mentions multiple candidate POI names.

### 5.4 BILOU Schema Feature

We adopt the linear-chain CRF model with BILOU schema in our implementation [18]. BILOU schema identifies **B**eginning, **I**nside and **L**ast word of a multi-word POI name, and **U**nit-length POI name. The words that do not appear in any POI names are identified by the **O**utside label. For example, tweet  $t_3$  in Table 1 is labeled below for *training* the model where subscripts  $z$  and  $f$  indicate the temporal awareness *present* and *future* respectively.

---

We’re\O all\O for\O Asian\O delights\O !\O Thai\B<sub>z</sub>  
express\L<sub>z</sub> today\O ,\O suki\B<sub>f</sub> sushi\L<sub>f</sub> tomorrow\O

---

**Table 4: Summary of features used in PETAR. All features are applicable to words that appear in candidate POI mention ( $w_i \in \ell$ ); a subset of features are applicable to words that do not ( $w_i \notin \ell$ ), indicated by  $\checkmark$**

Lexical Features	$w_i \notin \ell$	$w_i \in \ell$
1. The word $w_i$ itself and its lowercased form $x_i$ ; The word shape of $w_i$ and the prefixes and suffixes of $x_i$ ; The prior probabilities of $x_i$ being in capitalization and in all-capitalization forms	$\checkmark$	$\checkmark$
2. Bag-of-words of the 5-word context window, the preceding two words, and the following two words, respectively	$\checkmark$	$\checkmark$
Grammatical Features		
1. POS tags of the preceding word $w_{i-1}$ , the current word $w_i$ , and the following word $w_{i+1}$ based on TwitterNLP	$\checkmark$	$\checkmark$
2. Word group features by Brown clustering based on the 4th, 8th, and 12th bits of the path	$\checkmark$	$\checkmark$
3. The time-trend score $T(t)$ of the tweet in range of $[-1, 1]$ , discretized into 20 binary features with interval of 0.1	-	$\checkmark$
4. The closest verb to a candidate location name $\ell$ based on TwitterNLP POS tagging; The tense label of the verb, the distance of the verb to $\ell$ , and whether the verb is to the left of $\ell$ . "NULL" is used if no verb is detected.	-	$\checkmark$
5. The closest time-trend word $w \in D_T$ matched directly with the time-trend dictionary or indirectly through Brown clustering; the distance between $w$ and $\ell$ , and whether $w$ appears to the left of $\ell$ . "NULL" is used if no such word.	-	$\checkmark$
Geographical Features		
1. Spatial randomness of the location name $\ell$ , $R(\ell)$	-	$\checkmark$
2. Location name confidence $F(\ell)$	-	$\checkmark$
3. Indicator of multiple candidate POI mentions	-	$\checkmark$
BILOU Schema Feature: Pre-labels of the current, the proceeding and the following words	$\checkmark$	$\checkmark$

Because of the POI inventory, the candidate POI mentions in a tweet can be pre-labeled with BILOU schema. Note that, the pre-labels do not contain the temporal awareness information (e.g., subscripts  $z$  and  $f$  in the above example) because such information cannot be determined from the name match with POI inventory. The pre-labels are passed to the CRF classifier as BILOU schema features in both training and testing phases, shown in Figure 3. Pre-labels are expected to enhance the model by explicitly encoding the label dependencies. A similar strategy was used for NER in [8]. More specifically, the BILOU schema feature for a word  $w_i$  include the pre-label of  $w_i$  itself and the pre-labels of its surrounding words  $w_{i-1}$  and  $w_{i+1}$ .

In the CRF model, each word is represented as a feature vector. If a feature is computed for the word (e.g., POS tag of the word), then the corresponding value is assigned in the feature vector. If a feature is computed for the whole tweet (e.g., tweet time-trend score), then all the words in the tweet are assigned the same value. If a feature is computed for a candidate POI mention (e.g., location name confidence), the same value is assigned to all the words contained in the candidate POI name. The corresponding feature is set to "NA" if a word does not appear in a candidate POI name. Table 4 summarizes all the features used in PETAR and indicates which features are applicable for words appearing in candidate POI names.

## 5.5 Efficiency

Given a tweet, there are three steps in PETAR as shown in Figure 3. The first step involves identifying all candidate POI mentions from the tweet. Here, we adopt a prefix tree algorithm to identify all candidate POI mentions with preference of longer matches with entries in POI inventory [12]. The algorithm has a linear complexity  $O(n)$  of tweet length in number of words  $n$ , regardless of the size of the POI inventory. The second step is feature extraction. Most of the features presented in the earlier sections are simple to derive. Specifically, the prior probabilities of  $x_i$  being in capitalization and in all-capitalization forms, Brown clustering, and geographical features ( $R(\ell)$  and  $F(\ell)$ ) are pre-computed. The most costly part of the feature extraction is POS tagging by using TwitterNLP. TwitterNLP is implemented using linear-chain CRF which is fast in inference. The last step is the inference by the CRF-based tagger. On a workstation with a 1.86GHz Xeon quad-core CPU and 12GB of RAM, PETAR processes about 400 raw tweets in a second (i.e., 1.44M/Hr)

by using a single CPU core. That is, PETAR can be easily applied in large-scale real-time setting with parallel computing technique.

## 6. EXPERIMENTS

We conduct extensive experiments to evaluate the proposed PETAR method against baselines. We then investigate the usefulness of the features in detail.

### 6.1 Experiment Setup

Recall that in Section 2, we have manually annotated 4,000 tweets and obtained 2,056 true POIs and 1,801 NPOIs. Among the 2,056 POIs, the number of POIs belonging to  $POI_p$ ,  $POI_z$ , and  $POI_f$  are 307, 1,202, and 547 respectively (see Table 2). In our experiments, we use this manually annotated data as groundtruth and evaluate the methods with 5-fold cross validation. That is, the annotated tweets are randomly split into 5 subsets: 4 subsets are used to train the classifier and the remaining subset is used as test set. The final results are averaged over 5 runs so that each subset is used as test set once. We use 4 category labels in training and testing:  $POI_p$ ,  $POI_z$ ,  $POI_f$ , and  $NPOI$ . In our evaluation, we also treat  $POI$  as a special category label. Instances of  $POI_p$ ,  $POI_z$ , and  $POI_f$  categories all belong to  $POI$ . That is, if an extracted location name is indeed a POI name, then it is a positive instance of  $POI$  category, regardless of its temporal awareness label.

The proposed PETAR method is implemented with the linear-chain CRF model by CRF++ toolkit with default settings for the system parameters.<sup>11</sup> We compare PETAR with the following three methods.

**Random Annotation (RA):** This is a weak baseline. Each candidate POI mention is randomly assigned one of four labels:  $POI_p$ ,  $POI_z$ ,  $POI_f$ , and  $NPOI$ . The reported results are averaged over 10 runs. The purpose of including this weak baseline is to show the accuracy of simple dictionary match, as a reference.

**K-Nearest Neighbor.** KNN is non-parametric method that has achieved good accuracy in many classification tasks. Here, each candidate POI mention  $\ell$  is represented by its surrounding 4 words (i.e., the context words), denoted by  $\mathcal{W}$ . The similarity between two POI names  $\ell_a$  and  $\ell_b$  is calculated by Jaccard coefficient. Note that, we do not weigh the words using TFIDF because many high-

<sup>11</sup><https://code.google.com/p/crfpp/>

frequent words (e.g., off, to, at) are important words in our task. The number of nearest neighbors was set to 10 in our experiments (i.e.,  $k = 10$ ).

$$\text{sim}(\ell_a, \ell_b) = \frac{|\mathcal{W}_a \cap \mathcal{W}_b|}{|\mathcal{W}_a \cup \mathcal{W}_b|}$$

**StanfordNER:** Also known as CRFClassifier, it is a state-of-the-art sequence labeling system which achieves robust performance across different domains.<sup>12</sup> We provide the labeled tweets as training data to build the classifier with default parameter settings. For a fair comparison, the POI inventory is provided to StanfordNER as an external gazetteer.<sup>13</sup>

In the evaluation, we adopt three widely used metrics: Precision ( $Pr$ ), Recall ( $Re$ ), and  $F_1$ .  $Pr$  of a category is the ratio of the correctly classified instances in that category.  $Re$  is the ratio of the instances that should be classified in the category.  $F_1$  is the harmonic mean of  $Pr$  and  $Re$ .

## 6.2 Experimental Results

Table 5 reports the experimental results of the four methods. The results of *PETAR* are achieved by using the lexical, grammatical, and BILOU schema features (See Section 6.3 for detailed feature analysis). We make the following observations:

1. *PETAR* achieves the best performance for POI extraction with temporal awareness labels ( $POI_f$ ,  $POI_z$ , and  $POI_p$ ), followed by StanfordNER. In terms of  $F_1$ , *PETAR* consistently outperforms StanfordNER by 21.4%, 24.6% and 111.9% for  $POI_f$ ,  $POI_z$ , and  $POI_p$  respectively. This result suggests that the conventional lexical features alone are not discriminative enough for this task. The much poorer performance by KNN and RA shows that exploiting the sequence dependencies among the class variables and feature observations greatly benefits candidate POI name disambiguation and temporal awareness resolution in tweets.
2. All the four methods deliver poorer performance for  $POI_p$  compared with  $POI_f$  and  $POI_z$ . One possible reason is the smaller number of training instances in  $POI_p$  compared with the other two categories. Another reason is that it is relatively harder to detect *past*-trend from a single tweet, particularly when the tweet is not composed in proper English. On the other hand, we argue that detection of  $POI_f$  and  $POI_z$  is more meaningful for the downstream applications.
3. If we ignore the temporal awareness and simply evaluate the methods on extracting POI names (i.e., category *POI* in Table 5), *PETAR* achieves the highest  $F_1$  of 0.8753 with slightly poorer precision (0.9094 vs 0.9390) but much better recall (0.8436 vs 0.6931) than StanfordNER. We believe the higher recall attributes to the BILOU pre-label of candidate POI mentions. However, the pre-labels may bring noise, resulting in the slight degradation of precision. Regarding *NPOI*, StanfordNER obtains slightly better recall than *PETAR* but much poorer precision.

In summary, our experimental results show that the proposed *PETAR* method achieves best results in disambiguating candidate POI names and resolving temporal awareness from tweets. Next, we investigate the impact of the three types of features in *PETAR*.

## 6.3 Feature Analysis

Other than the BILOU schema features used for encoding candidate POI mention in tweets, we present three types of features

**Table 7: Impact of individual feature within *Lex+Gra* for  $POI_f$**

Features	$Pr$	$Re$	$F_1$
<i>Lex+Gra</i>	<b>0.6895</b>	<b>0.5511</b>	<b>0.6126</b>
<i>Lex+Gra</i> - ContextWindow	0.6360	0.5420	0.5852
<i>Lex+Gra</i> - LRContextWindow	0.6520	0.5401	0.5908
<i>Lex+Gra</i> - TimeTrend	0.6736	0.5310	0.5939
<i>Lex+Gra</i> - ClosestVerb	0.6628	0.5237	0.5851
<i>Lex+Gra</i> - ClosestTrend	0.6590	0.5255	0.5848
<i>Lex+Gra</i> - BILOU	0.6522	0.5201	0.5787

in Section 3: *Lexical*, *Grammatical*, and *Geographical* features. In this section, we conduct experiments and evaluate their effectiveness on time-aware POI extraction. The results are reported in Table 6. Note that BILOU schema features are used in this set of evaluation, unless stated otherwise.

**Effectiveness of *Gra*, *Lex*, and *Geo* features.** If each of the three types of features is used alone, then *Gra* achieves best performance for  $POI_f$ ,  $POI_z$ , and  $POI_p$  followed by *Lex*. *Geo* performs the worst for the temporal awareness resolution. However, if the temporal awareness is ignored, then *Lex* becomes the winner in classifying *POI* and *NPOI*. This result is expected as grammatical features are designed to capture the temporal awareness while lexical features are more for capturing the context for location name disambiguation. Geographical feature, on the other hand, provides extra information about a candidate POI name but does not help in capturing additional contextual information for POI name extraction or temporal awareness resolution.

With two or three types of features, *Lex+Gra* outperforms other feature combinations for *POI*,  $POI_f$ ,  $POI_z$  and *NPOI* in terms of  $F_1$ . In terms of precision and recall, *Lex+Gra* either achieves the best results or is slightly worse than the best results. Inclusion of geographical feature, i.e., *Lex+Gra+Geo* improves the recall of *POI* and precision of *NPOI* with a small margin. That is, the additional information about each candidate POI name does help in distinguishing true POIs from all candidate POI mentions, but hurt temporal awareness resolution.

In summary, our feature analysis shows lexical and grammatical features together achieve the best results for candidate POI name disambiguation and temporal awareness resolution. Geographic features contribute to better recognition of location names but degrades the accuracy of temporal awareness resolution.

**Effectiveness of individual features in *Gra+Lex*.** Next, we evaluate the individual feature(s) within the *Lex+Gra* combination that we have specifically proposed in *PETAR*. Specifically, in each evaluation, one of the following feature(s) is removed from the *Lex+Gra* combination:

- ContextWindow: The bag-of-words of the 5-word context window, the preceding two words, and the following two words, respectively.
- LRContextWindow: The bag-of-words of the preceding and the following two words, respectively.
- TimeTrend: The overall time-trend score of the whole tweet  $T(t)$ .
- ClosestVerb: The closest verb, its time-trend score, the distance, and the left/right-hand side indicator.
- ClosestTrend: The closest time-trend word  $w \in D_t$ , its time-trend score, the distance and the left/right-hand side indicator.

Table 7 reports the performance for  $POI_f$  after removing each aforementioned feature(s) from the *Lex+Gra* combination. Simi-

<sup>12</sup><http://nlp.stanford.edu/software/CRF-NER.shtml>

<sup>13</sup><http://nlp.stanford.edu/software/crf-faq.shtml#gazette>



**Table 5: The performance comparison of different methods. The best performance is highlighted in boldface**

Method	<i>POI</i>			<i>POI<sub>f</sub></i>			<i>POI<sub>z</sub></i>			<i>POI<sub>p</sub></i>			<i>NPOI</i>		
	<i>Pr</i>	<i>Re</i>	<i>F<sub>1</sub></i>	<i>Pr</i>	<i>Re</i>	<i>F<sub>1</sub></i>	<i>Pr</i>	<i>Re</i>	<i>F<sub>1</sub></i>	<i>Pr</i>	<i>Re</i>	<i>F<sub>1</sub></i>	<i>Pr</i>	<i>Re</i>	<i>F<sub>1</sub></i>
RA	0.5254	0.7419	0.6152	0.1438	0.2464	0.1816	0.3040	0.2582	0.2792	0.0795	0.2426	0.1197	0.4509	0.2428	0.3156
KNN	0.7761	0.4980	0.6067	0.4622	0.2792	0.3481	0.5685	0.4593	0.5081	0.1333	0.0066	0.0125	0.5948	0.8385	0.6959
StanfordNER	<b>0.9397</b>	0.6931	0.7977	0.5701	0.4526	0.5046	0.5886	0.5264	0.5558	0.3147	0.1475	0.2009	0.7308	<b>0.9493</b>	0.8259
PETAR	0.9094	<b>0.8436</b>	<b>0.8753</b>	<b>0.6895</b>	<b>0.5511</b>	<b>0.6126</b>	<b>0.6752</b>	<b>0.7108</b>	<b>0.6925</b>	<b>0.5266</b>	<b>0.3574</b>	<b>0.4258</b>	<b>0.8354</b>	0.9042	<b>0.8684</b>

**Table 6: Impact of features to PETAR. The best performance is highlighted in boldface.**

Feature	<i>POI</i>			<i>POI<sub>f</sub></i>			<i>POI<sub>z</sub></i>			<i>POI<sub>p</sub></i>			<i>NPOI</i>		
	<i>Pr</i>	<i>Re</i>	<i>F<sub>1</sub></i>	<i>Pr</i>	<i>Re</i>	<i>F<sub>1</sub></i>	<i>Pr</i>	<i>Re</i>	<i>F<sub>1</sub></i>	<i>Pr</i>	<i>Re</i>	<i>F<sub>1</sub></i>	<i>Pr</i>	<i>Re</i>	<i>F<sub>1</sub></i>
Lexical	0.9161	0.8109	0.8603	0.4727	0.2682	0.3423	0.5701	0.6915	0.6250	0.2264	0.0393	0.0670	0.8095	0.9154	0.8592
Grammatical	0.8688	0.8152	0.8411	0.6525	0.5310	0.5855	0.6425	0.6764	0.6590	0.4727	0.3410	0.3962	0.8033	0.8597	0.8306
Geographical	0.7787	0.5762	0.6624	0.1667	0.0055	0.0106	0.4519	0.5666	0.5028	0	0	0	0.6276	0.8135	0.7085
Gra+Geo	0.8741	0.8241	0.8484	0.6453	0.5191	0.5753	0.6480	0.6858	0.6663	0.5026	<b>0.3725</b>	0.4279	0.8121	0.8650	0.8377
Lex+Gra	0.9094	0.8436	<b>0.8753</b>	<b>0.6895</b>	<b>0.5511</b>	<b>0.6126</b>	<b>0.6752</b>	<b>0.7108</b>	<b>0.6925</b>	<b>0.5266</b>	0.3574	0.4258	0.8354	0.9042	<b>0.8684</b>
Lex+Geo	<b>0.9206</b>	0.8045	0.8586	0.4748	0.2755	0.3487	0.5811	0.6873	0.6298	0.2373	0.0459	0.0769	0.8053	<b>0.9209</b>	0.8592
Lex+Gra+Geo	0.8702	<b>0.8709</b>	0.8706	0.6788	0.5438	0.6039	0.6712	0.7083	0.6892	0.5211	0.3639	<b>0.4286</b>	<b>0.8361</b>	0.9003	0.8670

lar observations hold for *POI<sub>z</sub>*, *POI<sub>p</sub>*, *NPOI*, and *POI*. The results are not shown due to the page limit. Observe from Table 7, removing each individual feature(s) leads to degradation in precision, recall, and *F<sub>1</sub>*. That is, all these features contribute positively to PETAR. Relatively, based on the difference of *F<sub>1</sub>* values, larger performance degradation is observed for removal of ContextWindow, ClosestVerb and ClosestTrend, than the removal of LRContextWindow and TimeTrend. A possible reason is that TimeTrend is partially covered by the knowledge derived from ClosestVerb and ClosestTrend. The features in LRContextWindow are included in ContextWindow. However, the performance degradation by the exclusion of LRContextWindow indicates that explicitly discriminating left/right-hand side windows is beneficial. Similarly, TimeTrend feature complements other features and improves the accuracy of PETAR.

**Effectiveness of BILOU schema features.** So far, BILOU schema features have been used in all the analysis. In the last row of Table 7, we report the result of using *Lex+Gra* feature combination but excluding the BILOU schema features. Observe that, exclusion of BILOU schema features from the *Lex+Gra* combination results in big drop in precision, recall, and *F<sub>1</sub>*. This shows the importance of providing the boundary and constituents of each candidate POI mention in tweet. Furthermore, the BILOU schema features complement the ClosestVerb and ClosestTrend grammatical features because these features are computed based on candidate POI mentions.

## 7. RELATED WORK

To the best of our knowledge, this work is the first to address the task of fine-grained location extraction from tweets with temporal awareness. Accordingly, this section review related work on POI extraction, NER and Geolocalization for tweets respectively.

**POI Extraction.** The most relevant work to ours are the approaches presented in [17, 21]. Rae *et al.* proposed an approach to identify POI mentions in formal text [21]. Given the expensive manual annotation procedure, they proposed to build a training set by taking the abstract of the Wikipedia articles that are related to a POI, and the snippets returned by querying these Wikipedia article titles. However, POIs covered in Wikipedia are mainly landmarks and government buildings. To support fine-grained POI extraction, they obtained POIs from Foursquare and Gowalla and then used the POIs as queries to get web snippets as training samples. A linear-chain CRF model is trained for POI recognition using conventional linguistic features (*e.g.*, capitalization, POS tag). Our work is sig-

nificantly different from theirs in twofold: 1) While their approach was developed for formal text like news articles and web pages, we aim to recognize POI mentions in tweets. The brevity property and noise-prone nature of tweets introduce new challenges; 2) The temporal awareness of the POIs in tweets is a key consideration in our task. Lingad *et al.* tried to extract POI mentions from disaster-related tweets by retraining existing NER tools [17]. Several state-of-the-art NER tools, including StanfordNER, OpenNLP and TwitterNLP, were investigated. They took locations and organizations recognized by these tools as POIs. StanfordNER outperforms other alternatives in their experiments. Their experiments also show that the performance of extracting POIs of fine granularity remains inferior. In comparison, our work exploits check-in data from Foursquare for fine-grained POI extraction, leading to promising performance of POI recognition in tweets.

**Geolocalization for Tweets.** Recently, there have been many studies on estimating location of Twitter users or tweets. Cheng *et al.* proposed a probability framework to estimate city-level location of a Twitter user based on tweet content [4]. The spatial usage of each word is considered and a language model is built for each location. They reported that about half of the Twitter users can be placed within 100 miles of their true locations. Following this line, researchers propose to model the spatial usage of a word as a gaussian mixture model [3], or estimate location by using Kullback-Leiber divergence with Dirichlet smoothing [9]. Li *et al.* further considered the time dimension for location estimation in [15]. Mahmud *et al.* applied statistical learning approaches with an ensemble model to infer Twitter user’s home location [19]. A gazetteer containing references to US cities and states was used to build the training set. In summary, the estimation of user location is at coarse level of granularity, ranging from country, state, to city levels.

Schulz *et al.* proposed to use several geo-indicators together for more accurate location estimation [26]. Many external resources and tools were used to derive the indicators, including timezone mapping, Geonames, DBpedia Spotlight as well the links embedded in the tweets to Foursquare, etc.<sup>14</sup> Ikawa *et al.* exploited check-in tweets in Foursquare, and estimated location based on keyword match. These studies do not consider location name ambiguity or the temporal awareness of the locations.

The interplay between geographic locations, topics and Twitter user’s interests are mostly studied by using latent variable model. Eisentein *et al.* showed that each region has a specific topic distribution [5]. Hong *et al.* considered the user’s interest to model users’ geographic behavior [6]. Recently, Yuan *et al.* further inves-

<sup>14</sup><http://www.geonames.org/>

tingated spatial, temporal, and topical aspects to model users' geographic activities [30]. While these studies partially enable fine-grained location estimation, the specific POI information may be lost due to ambiguity, and the temporal awareness is still unknown.

**NER for Tweet.** NER has been extensively studied and reached promising performance on formal text corpus where linguistic features such as capitalization, POS tags are reliable and effective. However, significant performance degradation has been reported for NER from tweets [18, 23]. Liu *et al.* proposed a two-stage NER system for tweets [18]. In the first stage, a KNN classifier was used to pre-label each word based on the surrounding context. Then, the pre-label and the conventional linguistic features are incorporated into a CRF model for further refinement. Ritter *et al.* developed a pipelined NLP tool for tweets called TwitterNLP [23]. It consists of POS tagger, shallow parsing, capitalization classifier and named entity recognition. They reported superior performance compared to the retrained existing state-of-the-art systems. In our implementation, we applied TwitterNLP for POS tagging the tweets. Note that, the TwitterNLP works in supervised manner, which requires expensive human effort. Li *et al.* proposed an unsupervised NER solution by splitting tweet into non-overlapping segments [13, 14]. Then, they tried to identify named entities from these segments by using POS tagger or random walk algorithm. However, the proposed technique does not discriminate the type of the extracted entities, which make their solution less useful for POI extraction.

## 8. CONCLUSION

Market campaigning in Twitter is becoming very important in business world. In this paper, we attempt to facilitate the fine-grained location-based services/marketing and personalization by extracting POIs mentioned in tweets and predicting the temporal awareness of the POIs. The proposed solution, named PETAR, exploits the crowd wisdom of Foursquare community to enable fine-grained location extraction. The inclusion of partial location names largely tackles the problem of predominant usage of colloquial language in tweets. Then, a time-aware POI tagger is developed by using linear-chain CRF model, which conducts the location extraction and temporal awareness resolution in an effective and efficient way. Four types of features are extensively investigated. The comprehensive experiments show that PETAR achieves promising performance and is efficient for real-time applications. While many corporations just shout out their message in Twitter, PETAR, proposed in this work, could make the marketing in a way that is both enjoyable and profitable. As a part of our future work to further improve PETAR, we will investigate the context derived from historical tweets from a user as well as the social aspect of the users.

## 9. REFERENCES

- [1] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4):467–479, 1992.
- [2] A. X. Chang and C. D. Manning. SUTIME: A library for recognizing and normalizing time expressions. In *Proc. of LREC*, pages 3735–3740, 2012.
- [3] H.-W. Chang, D. Lee, M. Eltaher, and J. Lee. @phillies tweeting from philly? predicting twitter user locations with spatial word usage. In *Proc. of ASONAM*, pages 111–118, 2012.
- [4] Z. Cheng, J. Caverlee, and K. Lee. You are where you tweet: A content-based approach to geo-locating twitter users. In *Proc. of CIKM*, pages 759–768, 2010.
- [5] J. Eisenstein, B. O'Connor, N. A. Smith, and E. P. Xing. A latent variable model for geographic lexical variation. In *Proc. of EMNLP*, pages 1277–1287, 2010.
- [6] L. Hong, A. Ahmed, S. Gurumurthy, A. J. Smola, and K. Tsoutsoulouklis. Discovering geographical topics in the twitter stream. In *Proc. of WWW*, pages 769–778, 2012.
- [7] Y. Ikawa, M. Enoki, and M. Tsubori. Location inference using microblog messages. In *Proc. of WWW (Companion)*, pages 687–690, 2012.
- [8] J. Kazama and K. Torisawa. Exploiting wikipedia as external knowledge for named entity recognition. In *Proc. of EMNLP-CoNLL*, pages 698–707, 2007.
- [9] S. Kinsella, V. Murdock, and N. O'Hare. "i'm eating a sandwich in glasgow": Modeling locations with tweets. In *Proc. of SMUC*, pages 61–68, 2011.
- [10] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proc. of WWW*, pages 591–600, 2010.
- [11] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML*, pages 282–289, 2001.
- [12] C. Li, A. Sun, and A. Datta. TSDW: Two-stage word sense disambiguation using wikipedia. *JASIST*, 64(6):1203–1223, 2013.
- [13] C. Li, A. Sun, J. Weng, and Q. He. Exploiting hybrid contexts for tweet segmentation. In *Proc. of SIGIR*, pages 523–532, 2013.
- [14] C. Li, J. Weng, Q. He, Y. Yao, A. Datta, A. Sun, and B.-S. Lee. Twiner: Named entity recognition in targeted twitter stream. In *Proc. of SIGIR*, pages 721–730, 2012.
- [15] W. Li, P. Serdyukov, A. P. de Vries, C. Eickhoff, and M. Larson. The where in the tweet. In *Proc. of CIKM*, pages 2473–2476, 2011.
- [16] M. D. Lieberman, H. Samet, and J. Sankaranarayanan. Geotagging with local lexicons to build indexes for textually-specified spatial data. In *Proc. of ICDE*, pages 201–212, 2010.
- [17] J. Lingad, S. Karimi, and J. Yin. Location extraction from disaster-related microblogs. In *Proc. of WWW (Companion)*, pages 1017–1020, 2013.
- [18] X. Liu, S. Zhang, F. Wei, and M. Zhou. Recognizing named entities in tweets. In *Proc. of ACL*, pages 359–367, 2011.
- [19] J. Mahmud, J. Nichols, and C. Drews. Where is this tweet from? inferring home locations of twitter users. In *Proc. of ICWSM*, 2012.
- [20] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330, 1993.
- [21] A. Rae, V. Murdock, A. Popescu, and H. Bouchard. Mining the web for points of interest. In *Proc. of SIGIR*, pages 711–720, 2012.
- [22] L. Ratnov and D. Roth. Design challenges and misconceptions in named entity recognition. In *Proc. of CoNLL*, pages 147–155, 2009.
- [23] A. Ritter, S. Clark, Mausam, and O. Etzioni. Named entity recognition in tweets: An experimental study. In *Proc. of EMNLP*, pages 1524–1534, 2011.
- [24] S. Rüd, M. Ciaramita, J. Müller, and H. Schütze. Piggyback: using search engines for robust cross-domain named entity recognition. In *Proc. of ACL-HLT*, pages 965–975, 2011.
- [25] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proc. of WWW*, pages 851–860, 2010.
- [26] A. Schulz, A. Hadjakos, H. Paulheim, J. Nachtwey, and M. Mühlhäuser. A multi-indicator approach for geolocalization of tweets. In *Proc. of ICWSM*, 2013.
- [27] M. Verhagen, R. Gaizauskas, F. Schilder, M. Hepple, J. Moszkowicz, and J. Pustejovsky. The tempeval challenge: Identifying temporal relations in text. *Language Resources and Evaluation*, 43(2):161–179, 2009.
- [28] M. Verhagen, I. Mani, R. Sauri, R. Knippen, S. B. Jang, J. Littman, A. Rumshisky, J. Phillips, and J. Pustejovsky. Automating temporal annotation with tarsqi. In *Proc. of ACL*, pages 81–84, 2005.
- [29] J. Weng and B.-S. Lee. Event detection in twitter. In *Proc. of ICWSM*, pages 401–408, 2011.
- [30] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. Magnenat-Thalmann. Who, where, when and what: discover spatio-temporal topics for twitter users. In *Proc. of KDD*, pages 605–613, 2013.
- [31] T. Zhang and D. Johnson. A robust risk minimization based named entity recognition system. In *Proc. of CoNLL*, pages 204–207, 2003.