Extracting fine-grained location with temporal awareness in tweets: A two-stage approach

Li, Chenliang; Sun, Aixin

2017

Li, C., & Sun, A. (2017). Extracting fine-grained location with temporal awareness in tweets: A two-stage approach. Journal of the Association for Information Science and Technology, 68(7), 1652-1670.

https://hdl.handle.net/10356/85144

https://doi.org/10.1002/asi.23816

Extracting Fine-grained Location with Temporal Awareness in Tweets: A Two-Stage Approach*

Chenliang Li
State Key Lab of Software Engineering, Wuhan University, China 430079
E-mail: cllee@whu.edu.cn

Aixin Sun
School of Computer Engineering, Nanyang Technological University, Singapore 639798
E-mail: axsun@ntu.edu.sg

Abstract

Twitter has attracted billions of users for life logging and sharing activities and opinions. In their tweets, users often reveal their location information and short term visiting histories or plans. Capturing user's short term activities could benefit many applications for providing the right context at the right time and location. In this paper, we are interested in extracting locations mentioned in tweets at fine-grained granularity, with temporal awareness. More specifically, we like to recognise the point-of-interests (POI) mentioned in a tweet and predict whether the user has visited, is currently at, or will soon visit the mentioned POIs. A POI can be a restaurant, a shopping mall, a bookstore or any other fine-grained location. Our proposed framework, named TS-Petar (**T**wo-**S**tage **POI** **E**xtractor with **T**emporal **A**wa**r**eness), consists of two main components: a *POI inventory* and a *two-stage time-aware POI tagger*. The POI inventory is built by exploiting the crowd wisdom of Foursquare community. It contains both POIs' formal names and their informal abbreviations, commonly observed in Foursquare check-ins. The time-aware POI tagger, based on the Conditional Random Field (CRF) model, is devised to disambiguate the POI mentions and to resolve their associated temporal awareness accordingly. Three sets of contextual features (linguistic, temporal, and inventory features) and two labeling schema features (OP and BILOU schemas) are explored for the time-aware POI extraction task. Our empirical study shows that the subtask of POI disambiguation and the subtask of temporal awareness resolution call for different feature settings for best performance. We have also evaluated the proposed TS-Petar against several strong baseline methods. The experimental results demonstrate that the two-stage approach achieves the best accuracy and outperforms all baseline methods in terms of both effectiveness and efficiency.

1 Introduction

As a real-time social communication platform, Twitter has attracted more than 302 million active users around the world as of May 2015 according to Wikipedia. Users share information about their mood, activities, and opinions

*This paper is an extended version of the SIGIR 2014 paper "Fine-grained location extraction from tweets with temporal awareness" (C. Li & Sun, 2014).

through short messages limited to 140 characters, called tweets. Powered by the recent advancement of wireless Internet access, users often use Twitter as a communication channel to constantly update their activities through their mobile devices.

In their tweets, users often casually or implicitly reveal their current locations and short term travel histories or plans (*i.e.,* places visited just now and places to visit shortly), at fine-grained granularity. Acquiring this kind of information enables tremendous opportunities for personalization and location-based services/marketing. 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.[1] 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 advertisements about *The Smile* (but not *L'Artusi*) for her tweet: "just back from L'Artusi, wonderful dinner :> like to try the smile tmr for lunch".[2] Both examples highlight the importance of recognizing fine-grained locations (*e.g.,* the cinema and restaurants in the two examples) and their associated temporal awareness (*e.g.,* 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 (Cheng, Caverlee, & Lee, 2010; Mahmud, Nichols, & Drews, 2012; Kinsella, Murdock, & O'Hare, 2011; W. Li, Serdyukov, de Vries, Eickhoff, & Larson, 2011; Ikawa, Enoki, & Tatsubori, 2012). 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 (Hong, Ahmed, Gurumurthy, Smola, & Tsioutsiouliklis, 2012; Eisenstein, O'Connor, Smith, & Xing, 2010). However, the extracted information/topic is too coarse for targeted marketing. In a nutshell, these techniques are far from adequate for precise location-based services/marketing. Although the city-level geolocation is helpful for location-based personalization, further fine-grained POI extraction and the related temporal awareness identification could be more useful to identify potential customers.

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, landmark building or 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).[3] 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 (Ritter, Clark, Mausam, & Etzioni, 2011; Liu, Zhang, Wei, & Zhou, 2011). For the same reason, capturing temporal awareness

[1] http://www.landmarktheatres.com/market/NewYork/SunshineCinema.htm

[2] *L'Artusi* and *The Smile* are two restaurants located at New York city.

[3] In our following discussion, we use the term POI to refer fine-grained location.

of POIs based on either existing work (Verhagen et al., 2009) or temporal expression extraction tools (*e.g.,* SUTIME (A. X. Chang & Manning, 2012),TARSQI (Verhagen et al., 2005)) becomes less practical. 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 this problem, in this study, we 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, such as exploiting check-ins in location-based social networks (LBSN) from the users of the same region. Note that the users from a specific region could post tweets talking about locations in the other regions. We argue that extracting the POIs located in the other geographical regions adds less value to the location-based services/marketing. By focusing on the tweets posted from a specific region, we restrict the target POIs to be extracted, and thus simplify the problem to be addressed. For example, we can construct an appropriate POI Inventory by exploiting the check-ins contributed by the users from the same geographical 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 profile.

Our proposed solution to the above problem, named TS-Petar (**T**wo-**S**tage **POI** **E**xtractor with **T**emporal **A**wa**r**eness), consists of two main components: a *POI inventory* and a *two-stage 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 their informal abbreviations, we construct the inventory by exploiting the Foursquare check-ins, collaboratively contributed by users from the same geographical region.[4] 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. For instance, 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 to infer a POI's temporal awareness, we develop a *two-stage time-aware POI tagger*. This tagger is based on Conditional Random Field (CRF), a widely used model for sequence labeling. We develop three sets of contextual features (*i.e.,* linguistic, temporal, inventory) and two labeling schema features (*i.e.,* BILOU and OP schemas) to learn the CRF model, for the labeling of POI names and predicting their temporal awareness. Previous work on this problem proposed to employ a single tagger to conduct the two subtasks simultaneously (*i.e.,* candidate POI mention disambiguation and POI temporal awareness resolution) (C. Li & Sun, 2014). However, a single tagger could lead to a large label space and a large number of feature weights to be learnt, given the predominate usage of out-of-vocabulary (OOV) words. In this sense, the resultant model may not generalize well to unseen instances. In this work, we argue that the two subtasks can work independently to each other. By configuring with the optimal contextual and labeling schema feature settings for each subtask, we can obtain further improvement in terms of effectiveness and efficiency. We perform intensive experiments to understand the impact of different feature combinations for the subtask of POI

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

recognition and the subtask of temporal awareness resolution. Our results show that BILOU and OP schemas lead to the best performance in the two subtasks respectively. Further, the two subtasks call for different feature settings to achieve the best accuracies. 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 revealing short-term visiting histories/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 three sets of contextual features (linguistic, temporal, and inventory) and two schema features (BILOU and OP), for learning the time-aware POI tagger. A two-stage tagging strategy is utilized to accommodate the requirement of different optimal feature settings for the two subtasks, POI recognition and temporal awareness resolution. All the three sets of contextual features are easy to derive, enabling real-time response.

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

2 Time-aware POI Extraction

2.1 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 $c_i \in \{c_1, c_2, \ldots c_k\}$. In other words, let ℓ be a POI mentioned in tweet t, we aim to extract all POI and temporal awareness label pairs, $\{\langle \ell, c \rangle\}$ from t.

Following (Rae, Murdock, Popescu, & Bouchard, 2012; Lingad, Karimi, & Yin, 2013), we define a POI to be a focused geographic entity (*e.g.,* district, area, street, road), or a specific point location (*e.g.,* hotel, landmark, school, shopping center and restaurant 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 our dataset. 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.

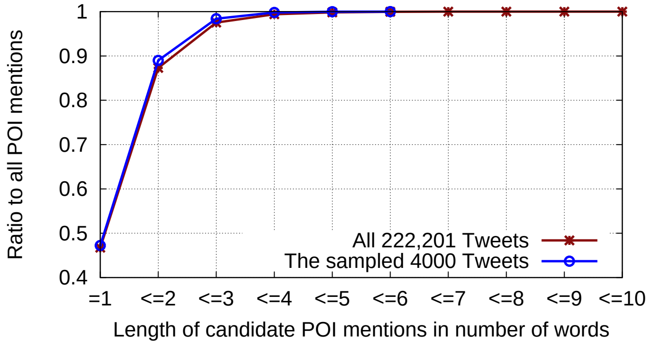

Figure 1: Length distribution of the candidate POI mentions

3 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 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 tweets, in the form of fine-grained POIs like restaurant or shopping mall names. 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.*

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,

Table 1: Example POI labels in tweets. The location names are in boldface, followed by their labels in brackets. All Twitter usernames are replaced by @username in this paper.

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

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, for example "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, two human annotators are 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. 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 5 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. The Cohen's kappa coefficient for the two annotators is 0.877, which indicates the almost perfect annotation agreement. As for the inconsistent annotations, the annotators further examined and discussed the candidate POI mentions and their contextual tweets together, and resolved the ambiguity. For example, the labels POI_z and POI_f were annotated to *mr bean* in tweet "@username from mr bean shop! it has 3 kinds, this wk is soccer and it's the cutest! Go buy!!"

Table 2: Distribution of the $3,977$ candidate POI mentions. The total number of POI's (including POI_p, POI_z and POI_f) is $2,056$.

$\#POI_p$	$\#POI_z$	$\#POI_f$	$\#NPOI$	$\#Unknown$	Total
307	$1,202$	547	$1,801$	120	$3,977$

respectively. Even the neighboring 4 tweets could provide little clue about the user's temporal awareness. After a discussion, the two annotators resolve the ambiguity and decided to use the label POI_z instead.

From the $4,000$ sampled tweets, 320 tweets are filtered away for containing mostly words in other languages than English.[5] In 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 (Kwak, Lee, Park, & Moon, 2010; Sakaki, Okazaki, & Matsuo, 2010; Weng & Lee, 2011). 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. A time-window is a predetermined duration 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 less 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.

[5] Singapore is a multi-cultural country. Some tweets are written in mixture of English, Chinese, Malay, Bahasa or other languages.

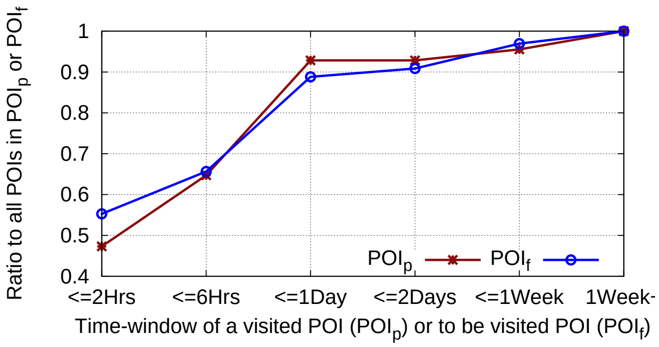

Figure 2: Time-window of the visits for POI_p and POI_f

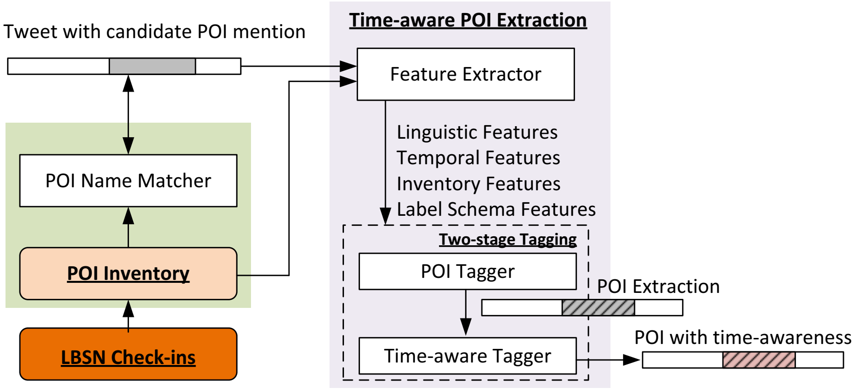

Figure 3: Overview of TS-Petar

3.1 Overview of TS-Petar

Given the problem definition, at first glance, it may seem that the problem can be easily addressed by using an off-the-shelf NER package to extract locations and then to label these locations using a 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 two-stage time-aware POI tagger, to extract POIs and to assign temporal awareness labels. The framework is named TS-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 extracted against the POI inventory and are then passed to a two-stage time-aware POI tagger for prediction. The two-stage POI tagger, based on the widely used linear-chain CRF model, takes in three types of contextual features: linguistic, temporal, and inventory features derived from the tweet and also the POI inventory. The CRF model also takes in features derived from the POI mentions, known as the schema features in this paper. Next, we detail the construction of POI inventory in Section 4 and the time-aware POI tagger in Section 5.

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)

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) 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 (see Table 3 for example check-in tweets). 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.

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 Foursquare users, who are also Twitter users, in Singapore in the duration of 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.[6] Note that, the check-in tweets are solely used for constructing the POI inventory and not used for evaluating TS-Petar.

4.2 POI Inventory Construction

Because check-in tweets are relatively well formatted, the POI names can be reliably extracted by applying hand-crafted 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 the POI names is 3.9 words. Plotted in Figure 4, 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 (Lieberman, Samet, & Sankaranarayanan, 2010). For example, *popular* is often used in tweets for *Popular*

[6] http://www.jurongpoint.com.sg/

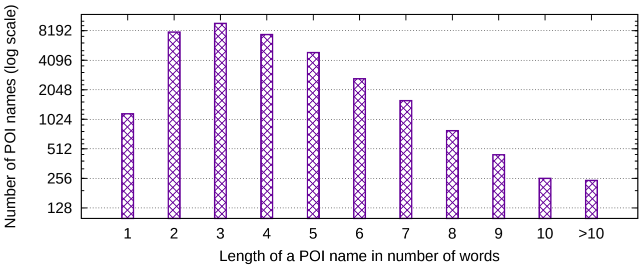

Figure 4: Length distribution of location names from check-ins

Bookstore. To ensure our POI inventory captures as many POI names mentioned in tweets as possible, 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. Note that this simple strategy of generating partial names could miss some acronyms of a POI. For example, we may not obtain "usa" from "United States of America". However, given that the POI inventory is constructed based on LBSN services, a kind of crowdsourcing location knowledge bases, we expect the common acronyms can be provided by the users' input. Developing a more sophisticated strategy is part of our future work. A practical 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, where *jurong east* is a region in Singapore. 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. Although such names appear in many check-in tweets, very unlikely these 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 two-stage time-aware POI tagger where the candidate POI mentions are first disambiguated at the first stage and these true POIs' temporal awareness are then resolved at the second stage.

5 Two Stage 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 DOESNT rain", the contextual and temporal cues like "*off to*" and "*now!*" are important information for the prediction of candidate POI mention *jp* (Jurong Point). 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 (Lafferty, McCallum, & Pereira, 2001). Also, it makes independence assumption for observation variables or features, enabling it for fast learning and inference.

The proposed two-stage 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 three classes of contextual features: *linguistic*, *temporal*, *inventory* features, and two labeling schemas: BILOU and OP schemas under a two-stage tagger strategy.

Next, we detail the features with the following notations. We use w_i to denote the i-th word in tweet t, x_i to denote the i-th word's lowercased form, and ℓ to denote a candidate POI name or mention.

5.1 Linguistic Feature

linguistic features are widely used in NER tasks and proven to be effective (Zhang & Johnson, 2003; Ratinov & Roth, 2009; Rüd, Ciaramita, Müller, & Schütze, 2011). In our implementation, we utilize 4 basic lexical features for a word w_i, 3 contextual features derived from the surrounding words of w_i, and part-of-speech (POS) tag and brown-clustering features.

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-numerics, 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 (Zhang & Johnson, 2003; Ratinov & Roth, 2009). 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}.

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.

Part-of-speech (POS) tag. We use TwitterNLP, a tweet-specific NLP tool to tag each tweet.[7] TwitterNLP incorporates tweet-specific features and outperforms conventional POS taggers in tweet domain (Ritter et al., 2011). Based

[7] http://github.com/aritter/twitter_nlp

on the Penn TreeBank tagset (Marcus, Santorini, & Marcinkiewicz, 1993), 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. We apply Brown clustering to capture the distributional similarity of words. Brown clustering is an algorithm that groups words that appear in similar contexts in a hierarchy (Brown, deSouza, Mercer, Pietra, & Lai, 1992). It helps to capture informal abbreviations and misspellings; for example, lexical variations like *shd, ishould, shudd, shuld, shoud, shud, shld, sould, shouldd* are clustered together with the modal verb *should*. As the result of Brown clustering, 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), following the work reported in (Ratinov & Roth, 2009).

5.2 Temporal Feature

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.[8] 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 (past participle) and VBD (past tense) are assigned score -1; VBZ (3rd person singular), VBP (non-thirrd person singular present), VBG (gerund/present participle) and VB (verb base form) assigned with score 0.

[8] The list is available at `https://sites.google.com/site/lichenliangpage/`

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 the right-hand side of the POI mention. If a tweet contains no verb, the aforementioned 12 features are set to "NULL".

The closest time-trend word. Besides the closest verb to a candidate POI mention, we also consider the closest word that matches an entry 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 the 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 Inventory Feature

The POI inventory is constructed by exploiting Foursquare check-in tweets, each of which is associated with a latitude/longitude coordinate. Also, each POI candidate can be from multiple check-ins. Here, we build several inventory related features by exploiting the spacial and check-in frequency factors as well as the characteristics of the POI candidate itself.

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

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

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 grid have $R(\ell) = 0$. Chain restaurants like McDonald's and Starbucks have much larger $R(\ell)$ values. We empirically investigate the impact of varying grid sizes in Section 6.2.

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 grids 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 μ_i and σ_i be the average and the standard deviation of all k_ℓ's of length i, the confidence of POI name ℓ of length i, denoted by $F(\ell)$, is defined in Equation 2, where 5 is a scaling constant.

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

Multiple candidate POI mentions. 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.

Characteristic features of a POI candidate. We observe that many multi-word POIs contain numeric values in between (*e.g.,* "waiting taxi at airport terminal 1"). Hence, we consider whether a multi-word POI candidate contains numeric values and whether the last word is a numeric value. Furthermore, longer POI candidates carry less ambiguity than the shorter ones (*e.g., popular bookstore* is more specific than a single word *popular*). In this sense, we consider the length of a POI candidate, and check whether any sub-sequence of a multi-word POI candidate matches another POI candidate in the inventory, and the length of the longest sub-sequence matched(*e.g., airport terminal 1* contains *airport*, *terminal*, and *terminal 1*).

5.4 Labeling Schema Feature

In this work, we shape the task of time-aware POI extraction as a sequence labeling problem by using linear-chain CRF model. To encode the knowledge from the POI inventory in this labeling task, we pre-label all candidate POI name mentions in tweets and use these labels as features to learn the CRF model (see Section 3.1 and Figure 3).

BILOU Schema. Existing works have demonstrated the importance of the labeling schema for sequence labeling techniques, and BILOU has been recognized as being an effective labeling schema (Ratinov & Roth, 2009; Liu et al., 2011). BILOU provides expressive representation for the boundary and constituent of a candidate POI name mention. More specifically, BILOU schema identifies **B**eginning, **I**nside and **L**ast word of a multi-word candidate POI mention, and **U**nit-length POI mention. The words that do not appear in any POI mention are identified by **O**utside words. For example, tweet t_3 in Table 1 is labeled as follows by using BILOU schema for *training* a POI disambiguation model.

We're\O all\O for\O Asian\O delights\O !\O Thai\B express\L today\O ,\O suki\B sushi\L tomorrow\O

OP Schema. As an alternative schema to BILOU, we concatenate the constituent words of a candidate POI mention in a tweet as a single token, and then label this token. Previous studies on product name recognition from forum discussions adopted this labeling strategy and demonstrated its effectiveness (Yao & Sun, 2015). Specifically, each candidate POI mention is treated as a single token by adding underscores to connect its constituent words. For example, the three words in *airport terminal 1* is rewritten as a single token *airport_terminal_1*. After rewriting the original tweet, OP schema is then used, where **P** and **O** identifies a candidate POI mention and an outside word respectively. Following the earlier example, tweet t_3 in Table 1 is labeled below by using OP schema.

We're\O all\O for\O Asian\O delights\O !\O Thai_express\P today\O ,\O suki_sushi\P tomorrow\O

Remark. Because of the POI inventory, the candidate POI mentions in a tweet can be pre-labeled with BILOU/OP schemas. The pre-labels are passed to the CRF classifier as BILOU/OP schema features in both training and testing phases, shown in Figure 3. For BILOU schema, pre-labels are expected to enhance the model by explicitly encoding the label dependencies. A similar strategy was used for NER in (Kazama & Torisawa, 2007). For OP schema, no explicit enhancement is required instead. In our implementation, the BILOU/OP 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. Under the OP schema, because a multi-word POI candidate is concatenated to be a single token, most linguistic features working for an ordinary token are set to "NA" for the words contained in a candidate POI mention. Table 4 summarizes all the features used in TS-Petar and indicates which features are applicable for words appearing in candidate POI mentions.

5.5 Two-Stage Tagger

After identifying the candidate POI mentions and extracting the corresponding contextual features, the CRF-based tagger is expected to identify the true POIs and their temporal awareness.

In our previous work (C. Li & Sun, 2014), named Petar, a single CRF tagger is utilized to simultaneously disambiguate the candidate POI mentions and resolve the temporal awareness by using the BILOU schema. However, representing POI_p, POI_z, POI_f and $NPOI$ under BILOU schema leads to a large label space and a large number of feature weights need to be learnt. Given tweets' shortness and free writing style, the resultant model may not generalize well to unseen samples. We therefore propose a two-stage framework which decomposes the time-aware POI extraction as two subtasks: *candidate POI mention disambiguation* and *POI temporal awareness resolution*.

More specifically, we train a POI disambiguation tagger and a POI time-aware tagger by using two linear-chain CRF models. Figure 5 demonstrates one configuration of the two-stage time-aware POI extraction procedure. Given an input tweet with candidates POI mentions and the corresponding features, we first identify the true POIs by using the disambiguation tagger. Then, the identified POIs are fed into the time-aware tagger to assign one of the three labels: POI_p, POI_z and POI_f.

This two-stage framework has several benefits: 1) The two sub-taggers have fewer number of feature weights to learn, leading to a better generalization ability; 2) Each sub-tagger accommodates different feature settings for optimal

Table 4: Summary of features used in TS-Petar. The features that are applicable to the words that appear in candidate POI mention ($w_i \in \ell$) and the words that do not ($w_i \notin \ell$), under the two labeling schemas (BILOU and OP) are indicated by ✓in this table.

Feature Description and Summary	BILOU		OP	
Linguistic Features	$w_i \notin \ell$	$w_i \in \ell$	$w_i \notin \ell$	$w_i \in \ell$
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	✓	✓	✓	-
2. Bag-of-words of the 5-word context window, the preceding two words, and the following two words, respectively	✓	✓	✓	-
3. POS tags of the preceding word w_{i-1}, the current word w_i, and the following word w_{i+1} based on TwitterNLP	✓	✓	✓	-
4. Word group features by Brown clustering based on the 4th, 8th, and 12th bits of the path	✓	✓	✓	-
Temporal Features				
1. The time-trend score $T(t)$ of the tweet in range of $[-1, 1]$, discretized into 20 binary features with interval of 0.1	-	✓	-	✓
2. The closest verb to a candidate location name ℓ based on TwitterNLP POS tagging; The tense label of the verb, the distance of the verb to ℓ, and whether the verb is to the left of ℓ. "NULL" is used if no verb is detected.	-	✓	-	✓
3. 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 ℓ, and whether w appears to the left of ℓ. "NULL" is used if no such word.	-	✓	-	✓
Inventory Features				
1. Spatial randomness of the location name ℓ, $R(\ell)$	-	✓	-	✓
2. Location name confidence $F(\ell)$	-	✓	-	✓
3. Indicator of multiple candidate POI mentions	-	✓	-	✓
4. Numeric values and whether the last token appears to be numeric value; the length of ℓ, whether ℓ contains any short POI candidate, and the maximum length of any matched shorter POI candidate.	-	✓	-	✓
Labeling Schema Feature				
1. Pre-labels of the current, the proceeding, and the following words	✓	✓	✓	-

performance. This would also help us better understand the important factors for these two subtasks respectively; 3) The reduction in feature space also leads to faster training and inference time. In the evaluation, we intensively investigate the impact of different feature settings for these two sub-taggers respectively. The experimental results demonstrate that the proposed two-stage framework is indeed more effective and efficient, reported in the next Section.

5.6 Time Complexity

There are three steps in TS-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 (C. Li, Sun, & Datta, 2013). 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 earlier are simple to derive. Specifically, the prior probabilities of x_i being in capitalization and in all-capitalization forms, Brown clustering, and inventory 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 two cascaded CRF-based taggers.

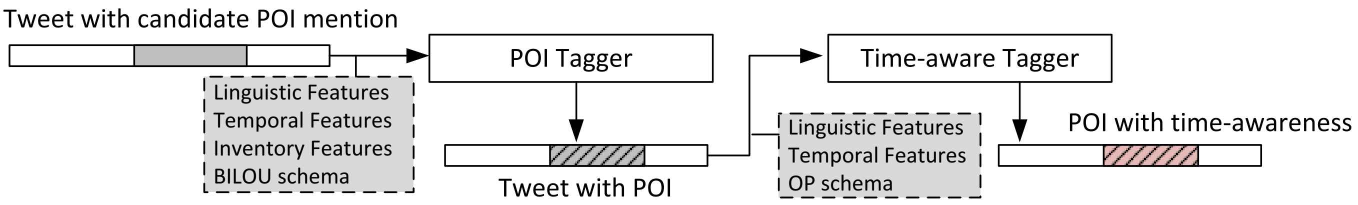

Figure 5: An configuration of two-stage tagger

On a workstation with a 1.86GHz Xeon quad-core CPU and 12GB of RAM, TS-Petar conducts POI mention disambiguation for about 613 raw tweets in a second (*i.e.,* 2.21M/Hr), and temporal awareness resolution for about 437 tweets with POIs in a second (*i.e.,* 1.57M/Hr) separately. In the cascaded manner, TS-Petar processes about 373 raw tweets in a second (*i.e.,* 1.34M/Hr) by using a single CPU core. That is, TS-Petar can be easily applied in large-scale real-time setting with parallel computing technique.

6 Experiments

We conduct two sets of experiments. The first set is to understand the usefulness of the different feature settings under the two-stage framework. In the second set of experiments, the proposed TS-Petar method is compared against several baselines including Petar (C. Li & Sun, 2014).

6.1 Experiment Setup

Recall that in Section 3, 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 and POI_f are 307 and 547 respectively. The remaining $1,202$ POIs are under the POI_z category (see Table 2). In our experiments, we use this manually annotated data as groundtruth and evaluate TS-Petar and other 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 metrics are calculated over 5 folds so that each subset is used as test set once. We randomly generate 10 sets of such 5-fold partitions, and report the averaged performance and conduct paired t-test based on the F_1 values.

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 TS-Petar method is implemented with the linear-chain CRF model by CRF++ toolkit with default settings for the system parameters.[9] 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.

[9] https://code.google.com/p/crfpp/

Table 5: Effectiveness of the features for POI disambiguation under BILOU and OP schemas. The better performance (BILOU vs OP schema) for each feature setting are highlighted in boldface ($*$ indicates the difference is statistically significant at 0.01 level). The best performance for each measure under the *POI* and *NPOI* categories is underlined († indicates that the difference regarding the best performance is statistically significant at 0.01 level).

Feature	BILOU Schema						OP Schema					
	POI			*NPOI*			*POI*			*NPOI*		
	Pr	*Re*	F_1	*Pr*	*Re*	F_1	*Pr*	*Re*	F_1	*Pr*	*Re*	F_1
*Lin*guistic	**0.8944**	**0.8782**	$\mathbf{0.8862}^{\dagger}_{*}$	**0.8643**	**0.8821**	$\mathbf{0.8731}^{\dagger}_{*}$	0.8810	0.8762	$0.8786^{\dagger}$	0.8601	0.8654	$0.8627^{\dagger}$
*Tem*poral	**0.6083**	**0.6664**	$\mathbf{0.6360}^{\dagger}_{*}$	**0.5744**	**0.5119**	$\mathbf{0.5413}^{\dagger}_{*}$	0.6002	0.6516	$0.6248^{\dagger}$	0.5632	0.5052	$0.5326^{\dagger}$
*Inv*entory	0.7415	**0.7323**	$\mathbf{0.7369}^{\dagger}_{*}$	**0.6998**	0.7097	$0.7047^{\dagger}$	**0.7454**	0.7068	$0.7256^{\dagger}$	0.6888	**0.7239**	$\mathbf{0.7059}^{\dagger}$
Tem+Inv	0.7309	0.7400	$0.7354^{\dagger}$	0.7003	0.6906	$0.6954^{\dagger}$	**0.7456**	**0.7533**	$\mathbf{0.7494}^{\dagger}_{*}$	**0.7151**	**0.7091**	$\mathbf{0.7121}^{\dagger}_{*}$
Lin+Tem	**0.8878**	**0.8808**	$\mathbf{0.8843}^{\dagger}_{*}$	**0.8657**	**0.8734**	$\mathbf{0.8695}^{\dagger}_{*}$	0.8768	0.8758	$0.8763^{\dagger}$	0.8589	0.8600	$0.8595^{\dagger}$
Lin+Inv	$\underline{\mathbf{0.9007}}$	$\underline{\mathbf{0.8831}}$	$\underline{\mathbf{0.8918}}_{*}$	$\underline{\mathbf{0.8699}}$	$\underline{\mathbf{0.8893}}$	$\underline{\mathbf{0.8795}}_{*}$	0.8959	0.8743	$0.8850^{\dagger}$	0.8609	0.8845	$0.8725^{\dagger}$
Lin+Tem+Inv	**0.8950**	**0.8816**	$\mathbf{0.8882}^{\dagger}_{*}$	**0.8676**	**0.8824**	$\mathbf{0.8749}^{\dagger}_{*}$	0.8884	0.8704	$0.8793^{\dagger}$	0.8559	0.8757	$0.8657^{\dagger}$

6.2 Evaluation of TS-Petar

In TS-Petar, we utilize a two-stage framework by cascading the two subtasks, *candidate POI mention disambiguation* and *POI temporal awareness resolution*. In this section, we investigate the performance of the two subtasks separately, under different feature settings. Specifically, we evaluate the effectiveness of the *Lin*guistic, *Tem*poral and *Inv*entory features as well as their combinations under the two labeling schemas BILOU and OP.

Candidate POI Mention Disambiguation. Given a tweet with identified candidate POI mentions, we measure the performance of by counting the correct categorizations of *POI* and *NPOI* labels. Table 5 reports the performance of this subtask with different feature settings under BILOU and OP schemas, respectively. We make the following observations:

1. If each of the three types of features is used alone, then *Lin*guistic features achieve the best performance for both *POI* and *NPOI* under either BILOU or OP schema. This observation is consistent with our previous finding (C. Li & Sun, 2014), where *lexical* features (excluding POS tagging and Brown clustering features in *Lin*guistic) were the winner in classifying *POI* and *NPOI*. *Tem*poral features perform the worst for the POI candidate disambiguation. This is under expectation since the extracted temporal features are less relevant to the task of POI mention disambiguation. While in (C. Li & Sun, 2014) we did not consider the characteristic features of a POI candidate in *geographical* feature set, we find that these features largely boost up the disambiguation performance of *POI*. In detail, under BILOU/OP schemas, F_1 score is 0.7369/0.7256 by using *Inv*entory features, while using *geographical* features in (C. Li & Sun, 2014) only achieved an averaged F_1 of 0.6604 over the 10 runs. As for *NPOI*, there is a slight performance degradation for the two feature sets (*i.e.,* 0.7047/0.7059 vs 0.7077).

2. With two or three types of features, the feature combination *Lin+Inv* significantly outperforms other feature combinations for both *POI* and *NPOI* in terms of F_1. In terms of precision and recall, *Lin+Inv* also achieves the best results than other feature combinations. The inclusion of *Tem*poral features, *e.g., Lin+Tem+Inv*, leads to the suboptimal performance in almost all metrics under both BILOU and OP schemas. This is consistent with the effectiveness of each single feature set. In summary, *Lin+Inv* is the optimal feature setting for the POI candidate disambiguation.

Table 6: Performance comparison for POI disambiguation under BILOU, BIO and OP schemas by using *Linguistic* and *Inventory* features. The best performance are highlighted in boldface. $*$ indicates the difference regarding BIO schema is statistically significant at the 0.01 level.

Schema	*POI*			*NPOI*		
	Pr	*Re*	F_1	*Pr*	*Re*	F_1
BILOU	**0.9007**	**0.8831**	**0.8918**	**0.8699**	**0.8893**	**0.8795**
BIO	0.8984	0.8819	0.8901	0.8685	0.8867	0.8775
OP	0.8959	0.8743	0.8850_*	0.8609	0.8845	0.8725_*

3. Different label schemas produce varying performance gains. Generally, BILOU schema is significantly superior to OP schema for the task of candidate POI mention disambiguation in most cases. Specifically, in all feature settings involving *Lin*guistic features, the performance of *POI* and *NPOI* by using BILOU is significantly better than that using OP schema.

Note that most linguistic features for the words contained in a candidate POI mention are simply ignored (*i.e.,* set to "NA") under the OP schema. This calls for a question: *whether this knowledge loss is responsible for the performance degradation when using the OP schema.* To answer this question, we conduct another set of experiments by using *Lin*+*Inv* feature combination with word extensions, under the OP schema. More specifically, in addition to *Lin*+*Inv*, we explicitly add bag-of-word (BOW) feature, *i.e.,* the constituent words in a candidate POI mention[10]. This BOW feature delivers a F_1 score of 0.8874 and 0.8746, for *POI* and *NPOI* respectively. However, this set of results remains significantly worse than that using BILOU schema at 0.01 level, which partially suggests that the dependencies between the constituent words play an important role for the disambiguation task. Because OP schema does not incorporate such information into the model, the disambiguation performance deteriorates to some extent.

We argued in Section 5.5 that designing a single classifier under BILOU schema like Petar leads to a large number of feature weights to be learnt, and may not generalize well to unseen samples. Here, we conduct another set of experiments for the disambiguation task by using *Lin*+*Inv* under BIO labeling schema. BIO schema identifies **B**eginning, **I**nside and **O**utside word of a multi-word candidate POI mention. It is more compact than BILOU schema, but can model more dependencies between constituent words than OP schema. Table 6 reports the performance comparison among these three labeling schemas. We observe that using BIO schema significantly performs better than using OP schema. This is expected since the dependencies between consecutive words is very userful for POI mention disambiguation, just like what we discussed above for OP and BILOU comparison. However, we also note that using BILOU schema is marginally more beneficial than using BIO schema. This is consistent with the existing NER works (Ratinov & Roth, 2009), since BILOU schema is a more expressive schema. This also validates the hypothesis that each sub-tagger accommodates different feature settings for optimal performance.

Compared with Petar reported in (C. Li & Sun, 2014), we add several features in TS-Petar and re-group all features. To be more specific, we add several characteristic features of a candidate POI into geographical features of Petar, and name the new set of features as inventory features in TS-Petar. Our experimental results show that excluding these characteristic features from *Inv*+BILOU setting (denoted as *Inv*+BILOU-*C*) leads to significant performance degradation from 0.7369/0.7047 to 0.7239/0.6829, for *POI* and *NPOI* respectively. Also, we notice that *Inv*+BILOU

[10] All numeric tokens are represented by "NUM".

Table 7: Performance comparison for POI disambiguation under *Lin*+*Inv*+BILOU with varying grid size. The best performance are highlighted in boldface.

Schema	*POI*			*NPOI*		
	Pr	*Re*	F_1	*Pr*	*Re*	F_1
0.5KM×0.5KM	0.8987	**0.8848**	0.8917	**0.8712**	0.8866	0.8788
1KM×1KM	**0.9007**	0.8831	**0.8918**	0.8699	**0.8893**	**0.8795**
2KM×2KM	0.8995	0.8842	**0.8918**	0.8708	0.8876	0.8791
5KM×5KM	0.8987	**0.8848**	0.8917	**0.8712**	0.8866	0.8788
-SR	0.8982	0.8845	0.8913	0.8709	0.8860	0.8784

Table 8: Effectiveness of features for temporal awareness resolution under BILOU/OP schema. The better performance (BILOU vs OP) for each feature setting is highlighted in boldface ($*$ indicates the difference is statistically significant at 0.01 level), and the overall best performance is underlined († indicates that the difference regarding the best performance is statistically significant at 0.01 level; ◇ indicates that the difference regarding BIO schema by using *Linguistic* and *Inventory* features is statistically significant at 0.01 level).

Feature	POI_f			POI_z			POI_p		
	Pr	*Re*	F_1	*Pr*	*Re*	F_1	*Pr*	*Re*	F_1
*Lin*guistic + BILOU	0.5911	0.4331	0.4998†	0.6642	0.8643	0.7512†	0.3150	0.0954	0.1463†
*Lin*guistic + OP	**0.6854**	**0.4835**	**0.5670**$^{\dagger}_{*}$	**0.6757**	**0.9030**	**0.7729**$^{\dagger}_{*}$	**0.5510**	**0.1092**	**0.1798**$^{\dagger}_{*}$
*Tem*poral + BILOU	0.6921	0.5616	0.6200†	0.7326	0.8362	0.7810†	0.6048	0.4748	0.5318†
*Tem*poral + OP	**0.7191**	**0.5899**	**0.6481**$^{\dagger}_{*}$	**0.7477**	**0.8509**	**0.7960**$^{\dagger}_{*}$	$\underline{\mathbf{0.6213}}$	$\underline{\mathbf{0.4859}}$	$\underline{\mathbf{0.5452}}_{*}$
*Inv*entory + BILOU	**0.3442**	**0.0362**	**0.0649**$^{\dagger}_{*}$	**0.5847**	0.9729	0.7304†	0.0	0.0	0.0†
*Inv*entory + OP	0.2183	0.0149	0.0278†	0.5846	$\underline{\mathbf{0.9867}}$	**0.7342**$^{\dagger}_{*}$	0.0	0.0	0.0†
Lin+Tem+BILOU	0.7179	0.6088	0.6589$^{\dagger\diamond}$	0.7428	0.8381	0.7875$^{\dagger\diamond}$	0.5735	0.4456	0.5013$^{\dagger\diamond}$
Lin+Tem+OP	$\underline{\mathbf{0.7432}}$	$\underline{\mathbf{0.6386}}$	$\underline{\mathbf{0.6869}}_{*}$	$\underline{\mathbf{0.7610}}$	**0.8505**	$\underline{\mathbf{0.8032}}^{\diamond}_{*}$	**0.6132**	**0.4856**	**0.5419**$^{\diamond}_{*}$
Lin+Tem+BIO	0.7315	0.6316	0.6779	0.7523	0.8396	0.7936	0.5986	0.4630	0.5220

achieves a much better F_1 than PETAR using geographical features for *POI* (0.7369 vs 0.6624), but remains a very close F_1 for *NPOI* (0.7059 vs 0.7085). That is, *Inv*+BILOU obtains a large *Re* gain for *POI*, which also leads to a fierce *Re* degradation for *NPOI*. Similar performance pattern is also observed for *Inv*+BILOU-*C*, which is equivalent to using geographical features of PETAR. Because POI disambiguation tagger has a much smaller label space than PETAR (*i.e.,* 50%), the experimental results demonstrate that a single POI disambiguation tagger designed in TS-PETAR is more robust than a single overall time-aware POI extraction classifier provided by PETAR.

Spatial randomness feature described in Section 5.3 is used to measure the pattern of spatial distribution of each POI candidate. We calculate normalized entropy value by putting each check-in tweet into the grids with a size of 1KM×1KM. Here we investigate the effect of grid size for POI extraction. Table 7 reports the performance of varying grid size (*i.e.,* 0.5KM, 1KM, 2KM and 5KM) as well as excluding this feature (denoted as -SR) under *Lin*+*Inv*+BILOU setting. We observe that the grid size does not impact the performance of POI extraction much, and excluding this feature just results in negligible performance loss[11]. This feature provides contextual semantic that determines the meaning of a POI candidate mention rather than the recognition of a POI mention.

We further evaluate the performance of two labeling schemas for efficiency. Training a disambiguation classifier by using *Lin*+*Inv*+*OP* takes on average of 25.32s, while the counterpart with BILOU schema takes 76.78s. OP schema also leads to faster inference. However, given the limited size of the test set, the difference between the inference time of the two schemas is marginal.

[11] The difference between any two settings in Table 7 is not statistically significant at 0.2 level.

Temporal Awareness Resolution. Time-aware classifier is responsible for classifying a POI to one of the three categories (*i.e.,* POI_f, POI_z and POI_p) based on the temporal context of the tweet. To evaluate the effectiveness of the features, we only consider the true POIs and their temporal awareness manually assigned in our dataset as training and testing instances in this set of experiments. The cascaded performance of temporal awareness resolution will be discussed in the next section.

Table 8 reports the experimental results. Three observations are made:

1. If each of the three types of features is used alone, *Tem*poral features achieve the best performance for POI_f, POI_z, and POI_p by using either BILOU or OP schema. *Inv*entory performs the worst for temporal awareness resolution. It hardly hits anything for POI_f and POI_p. This result is expected as temporal features are designed to capture the temporal awareness while inventory features do not provides much information about a POI's temporal context. As a result, almost all instances are classified into POI_z, the majority among the three temporal classes, leading to a very high recall for POI_z (*i.e.,* 0.9729/0.9867).

2. The inclusion of *Lin*, *i.e., Lin+Tem*, achieves a much better performance for POI_f and POI_z with either BILOU or OP. However, linguistic features contribute negatively to the classification of POI_p with both labeling schemas. We manually examine the classification results and find that some POI_z instances are wrongly classified as POI_p. One possible reason is that linguistic features over the limited training instances of POI_p introduce some bias. On the other hand, the contextual knowledge provided by the linguistic features (*e.g., off to*, *head down*) is beneficial to identify the *future*-trend.

3. OP schema is more suitable for temporal awareness resolution. With the same feature setting, OP schema brings significant improvement over BILOU, except for using *Inv*entory features. For instance, when using *Lin+Tem*, F_1 score increases from 0.6589 to 0.6869, 0.7875 to 0.8032, and 0.5013 to 0.5419, respectively for POI_f, POI_z and POI_p categories. Because using OP schema results in fewer number of feature functions, the corresponding parameter estimation could be more reliable, leading to the superior performance. The same observation was also made recently in the work of named entity recognition in travel-related queries (Cowan et al., 2015).

Because *Inv*entory is proven to be less useful for temporal awareness resolution in our experiments, we do not list the results of feature combinations involving inventory features in Table 8. We also evaluate the performance of using BIO schema under *Lin+Tem* setting (the last row in Table 8). We note that OP schema significantly outperforms BIO schema which in turn performs significantly better than using BILOU schema. This is reasonable since identifying a POI's temporal awareness require less dependency knowledge about the constituent words of the POI mention. The reduction in label space by using OP schema significantly boost the generalization ability of the time-aware classifier. Similar to the case of training a disambiguation classifier, the training time by using *Lin+Tem+OP* takes 64.73s on average, while the same feature set using BILOU schema takes 148.72s.

In summary, our results suggest that the proposed two-stage framework is a flexible solution for time-aware POI extraction. The optimal performance can be obtained by using *Lin+Inv*+BILOU setting for candidate POI mention disambiguation, and *Lin+Tem*+OP setting for temporal awareness resolution.

Table 9: The performance comparison of different methods. The best results are highlighted in boldface († indicates that the difference regarding the best performance is statistically significant at 0.01 level).

Method	POI			POI_f			POI_z			POI_p			$NPOI$		
	Pr	Re	F_1	Pr	Re	F_1	Pr	Re	F_1	Pr	Re	F_1	Pr	Re	F_1
RA	0.5332	0.7527	0.6242†	0.1270	0.2059	0.1571†	0.3150	0.2700	0.2908†	0.0840	0.2689	0.1280†	0.4711	0.2506	0.3271†
POI/Tem_{ts}	0.5321	**1.0**	0.6946†	0.3827	0.4706	0.4221†	0.3955	0.6989	0.5052†	0.2151	**0.7475**	0.3341†	0.0	0.0	0.0†
POI/Tem_{ttw}	0.5321	**1.0**	0.6946†	0.3492	0.3364	0.3427†	0.3351	**0.9193**	0.4911†	0.3409	0.0492	0.0860†	0.0	0.0	0.0†
KNN	0.7349	0.6323	0.6797†	0.5207	0.4303	0.4711†	0.5141	0.5453	0.5292†	0.4054	0.0561	0.0983†	0.6391	0.7404	0.6860†
StanfordNER	**0.9325**	0.6925	0.7948†	0.5474	0.4513	0.4947†	0.5858	0.5240	0.5531†	0.2564	0.1223	0.1655†	0.7295	**0.9430**	0.8226†
Petar	0.9072	0.8413	0.8730†	0.6862	0.5502	0.6107†	0.6739	0.7096	0.6912†	0.5161	0.3459	0.4141†	0.8333	0.9021	0.8663†
Petar$_{new}$	0.9085	0.8445	0.8753†	0.6731	0.5476	0.6039†	0.6723	0.7071	0.6892†	0.5272	0.3521	0.4221†	0.8363	0.9032	0.8685†
Petar$_{BIO}$	0.9057	0.8383	0.8707†	**0.7004**	0.5713	**0.6292**	**0.6819**	0.7144	0.6977	**0.5551**	0.3715	0.4451†	0.8305	0.9007	0.8642†
TS-Petar	0.9007	0.8831	**0.8918**	0.6791	**0.5794**	0.6253	0.6704	0.7305	**0.6992**	0.5465	0.4266	**0.4790**	**0.8699**	0.8893	**0.8795**

6.3 Methods Comparison

In this section, we compare TS-Petar with four baseline methods. For the performance comparison of POI candidate disambiguation, we use the results of TS-Petar with *Lin*+*Inv*+BILOU setting (*i.e.,* first stage). Then, TS-Petar with *Lin*+*Tem*+OP setting (*i.e.,* second stage) is applied on the results from the first stage. Results of the second stage are reported for the comparison of temporal awareness resolution. The following five methods are evaluated in our experiments.

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.

POI and Temporal Expression Matching (POI/Tem). We have constructed the POI inventory and time-trend word dictionary for candidate POI extraction and temporal feature extraction. A simple approach is to take all matched candidate POI mentions against the POI inventory as true POIs. Then, we resolve the temporal awareness for a POI by checking the matching words in the tweet against the time-trend word dictionary. We can consider the time-trend score $T(t)$ of the tweet against the time-trend word dictionary (denoted POI/Tem_{ts}): 1) POI_p, if $T(t) < 0$; 2) POI_f, if $T(t) > 0$; 3) otherwise, POI_z. Similarly, we take the time-trend score of the closed time-trend word for a POI in its tweet to determine the temporal awareness (denoted POI/Tem_{ttw}).

K-Nearest Neighbor. KNN is non-parametric method that has achieved good accuracy in many classification tasks. Here, each candidate POI mention ℓ is represented by its surrounding 4 words (*i.e.,* the context words), denoted by $\mathcal{W}$. The similarity between two POI names ℓ_a and ℓ_b is calculated by Jaccard coefficient. Note that, we do not weigh the words using TFIDF because many high-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.,* $k = 10$).

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

StanfordNER. Also known as CRFClassifier, it is a state-of-the-art sequence labeling system which achieves robust performance across different domains. 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.[12]

[12] http://nlp.stanford.edu/software/crf-faq.shtml#gazette

Petar. As being proposed in our previous work (C. Li & Sun, 2014), Petar is based on linear-CRF model for time-aware POI extraction. It utilizes linguistic, grammatical, geographical and BILOU schema features to *directly* infer a candidate POI as being one of POI_f, POI_z, POI_p and $NPOI$ categories, through a single classifier. In contrast, TS-Petar adds additional characteristic features of a candidate POI, and regroup all these features into linguistic, temporal and inventory categories. Here, we report the performance of Petar by using the combination of lexical and grammatical features, because this feature setting achieves the best performance (C. Li & Sun, 2014). Moreover, we evaluate the performance of Petar by using BIO schema (denoted as **Petar**$_{BIO}$). We also report the performance of Petar by using the combination of *Linguistic*, *Inventory* and *Temporal* features described in this work, since these features are useful for each subtask or both (denoted as **Petar**$_{new}$).

Table 9 reports the experimental results of the seven methods. We make the following observations:

1. TS-Petar achieves the best performance for POI extraction (*i.e.,* POI candidate disambiguation), followed by Petar$_{new}$ for both POI and $NPOI$. Specifically, TS-Petar obtains a much better recall than Petar$_{new}$ for POI category (0.8831 vs 0.8445). While a better precision is obtained by StanfordNER (0.9325), the low recall (0.6925) hinders its application in real systems. Note that TS-Petar and Petar based methods obtain a much better recall than StanfordNER for POI. We believe the higher recall attributes to the pre-labels of candidate POI mentions by using BILOU/BIO schema. However, the pre-labels may bring in noise, resulting in the slight degradation of precision. Regarding $NPOI$, StanfordNER obtains a better recall than TS-Petar (0.9430 vs 0.8893) but much poorer precision instead (0.7295 vs 0.8699). Since StanfordNER uses a CRF classifier without pre-labels, most POI candidates feeded into it are biased towards $NPOI$. This difference explains the much better recall StanfordNER achieved for $NPOI$. Note that POI/Tem only deliver an precision of 0.5321 for POI extraction. This indicates that the POI mention ambiguity is not a trivial problem.

2. Petar$_{new}$ performs marginally better than Petar and Petar$_{BIO}$ for POI extraction. Note that the inclusion of geographical features in Petar hurts the performance of POI extraction (C. Li & Sun, 2014). Thus, the results suggest that several characteristic features added in inventory feature set indeed provide discriminative power for POI mention disambiguation. Also, in contrast to Petar$_{new}$, much better recall and precision achieved by TS-Petar for POI and $NPOI$ respectively suggests that a single POI disambiguation tagger is more appropriate for the task of POI disambiguation.

3. TS-Petar achieves the best performance for temporal awareness categories POI_z and POI_p in terms of F_1, and is slightly worse than Petar$_{BIO}$ for POI_f (0.6253 vs 0.6292). While TS-Petar and Petar$_{BIO}$ have tied for POI_f and POI_z, TS-Petar plays significantly better for POI_p. This suggests that OP schema is more beneficial to temporal awareness resolution. In terms of F_1, TS-Petar consistently outperforms StanfordNER by 26.4%, 26.4% and 189.4% for POI_f, POI_z, and POI_p respectively. This result suggests that the conventional lexical features alone are not discriminative enough for temporal awareness resolution.

4. All the nine 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. Note that TS-Petar outperforms all other methods in terms of precision, recall and F_1 for POI_p. Given smaller number of training instances, the OP schema results in less number of parameters to be estimated, thus leads to a better classification model. Overall, we argue that detection of POI_f and POI_z is more meaningful for the downstream applications, *e.g.,* context-aware personalization or recommendation applications.

5. Compared with the CRF-based models, KNN, POI/Tem and RA deliver much poorer performance. The much better performance achieved by KNN for POI_f and POI_z validates the importance of contextual information for temporal awareness resolution in tweets. As discussed above, POI_p is a much difficult task compared to POI_f and POI_z. POI/Tem_{ts} performs much better than KNN, POI/Tem_{ttw} and RA in POI_p (0.331 vs 0.0983/0.0860/0.1280). This suggests that the contextual linguistical information is unable to capture the *past*-trend. The feature study in Table 8 also validates this finding, *i.e., Lin+Tem* performs worse than *Tem* under both BILOU and OP schemas (0.5013/0.5419 vs 0.5318/0.5452). POI/Tem_{ts} performs significantly better than POI/Tem_{ttw} for the task of temporal awareness, especially for POI_p. Because of the free writing styles, the closed time-trend words are not reliable to infer the temporal awareness. It is observed that POI/Tem_{ts} obtains a much higher recall of 0.7475. This indicates that the whole tweet often carries the *past*-trend when a POI_p temporal awareness is expressed by the user.

Note that in our previous study (C. Li & Sun, 2014), we have conducted extensive experiments to validate the effectiveness of each individual features for Petar, such as *ContextWindow, TimeTrend, ClosestVerb, BILOU schema*, etc. In this work, we observe similar results. Hence, we do not repeat these results.

It is reported that a retrained StanfordNER for tweet-based location extraction achieved a much better F_1 (0.902) and a balanced *Pr*/*Re* performance (0.935/0.873) in (Lingad et al., 2013). According to our understanding, the feature setting used in our work for StanfordNER is as same as in (Lingad et al., 2013), which was designed for the CoNLL 2003 shared task. The ground truth annotations used in (Lingad et al., 2013) consist of words, hashtags and URLs which contained location information. However, the tweets used in (Lingad et al., 2013) were related to a variety of disasters, especially major disastrous events happened from late 2010 until late 2012. In our task, the locations to be extracted from tweets are fine-grained POIs within a city. We believe it is the intrinsic characteristics of the datasets that produce such *Pr*/*Re* unbalance and performance difference. As a reference, a retrained StanfordNER with the same feature setting in (C. Li et al., 2012) produced a very unbalanced *Pr*/*Re* (0.762/0.293) performance in the dataset SIN_g whose size is in the same scale as ours (*i.e.,* about 4K tweets).

We further measure the computational cost of Petar under its optimal setting; training Petar takes on average 332.18s. As reported in Section 6.2, TS-Petar consists of training two classifiers which results in a total of 141.51s by using the corresponding optimal settings. The POI disambiguation tagger takes 1.010 seconds to process all manually annotated tweets, while the POI time-aware tagger takes 3.374 seconds instead. With a single CRF classifier, Petar takes 4.123 seconds to finish the inference process. Given there is some tiny efficiency sacrifice, the flexibility provided by TS-Petar is more valuable. As a reference, training StanfordNER classifier takes 1460.68s on average and the inference takes 37.735 seconds[13].

[13] Note the training and inference of StanfordNER involve the feature extraction procedure, which makes the direct comparison unfair. However, based on the time complexity discussed in Section 5.6, TS-Petar and Petar are still superior in terms of efficiency.

In summary, our experimental results show that the proposed TS-Petar method achieves better results in disambiguating candidate POI mentions and resolving the temporal awareness from tweets. The significant improvement over Petar and its variants suggests that decomposing the time-aware POI extraction as two subtasks provides a more flexible approach. We can develop optimal feature settings and labeling schemas for each of the two subtasks alone. Moreover, the reduction in the number of parameters to be learnt significantly reduces the computational cost. Furthermore, our previous work devised a joint CRF tagger to simultaneously conduct the POI mention disambiguation and resolve the temporal awareness under an assumption that the two subtask are correlated with each other. That is, a joint model could help us capture the semantic information to its maximum for time-aware POI extraction. However, in this study, the comprehensive experimental results show that the two subtasks are not coupled and call for a different setting for their optimal performance respectively. That is, these two subtasks are shown to be independent to each other for a single tweet. On the other hand, if we consider multiple tweets from a user collectively, then the tweets together might give some clue on the dependency of the two subtasks. The proposed joint model does not take care of this case. Therefore, we cannot validate that the two subtasks are dependent. We will leave this as a part of our future work.

7 Related Work

POI Extraction. The most relevant work to ours are the approaches presented in (Rae et al., 2012; Lingad et al., 2013). Rae *et al.* proposed an approach to identify POI mentions in formal text (Rae et al., 2012). 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 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 significantly 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 (Lingad et al., 2013). 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 (Cheng et al., 2010). 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 (H.-W. Chang, Lee, Eltaher, & Lee, 2012), or estimate location by using Kullback-Leiber divergence with Dirichlet smoothing (Kinsella et al., 2011). Li *et al.* further considered the time dimension for location estimation in (W. Li et al., 2011). Mahmud *et al.* applied statistical learning approaches with an ensemble model to infer Twitter user's home location (Mahmud et al., 2012). A gazetteer containing references to US cities and states was used to build the training set. Han *et al.* showed that identifying location-indicative words could benefit positively for the task of twitter user location prediction (Han, Cook, & Baldwin, 2012). Then, they further conducted a comprehensive study of text-based twitter user geolocation (Han, Cook, & Baldwin, 2014). They studied the impact of location-indicative word selection, non-geotagged tweets, user language, and user-metadata as well as the temporal variance on model generalization. Besides using text information alone, several works incorporated the social network structure as a complementary information for the location prediction task. The recent representative works include (Rahimi, Cohn, & Baldwin, 2015; Jurgens, Finethy, McCorriston, Xu, & Ruths, 2015). 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 (Schulz, Hadjakos, Paulheim, Nachtwey, & Mühlhäuser, 2013). Many external resources and tools were used to derive the indicators, including timezone mapping, Geonames, DBPedia Spotlight as well as the links embedded in the tweets to Foursquare, etc..[14] Ikawa *et al.* exploited check-in tweets in Foursquare, and estimated location based on keyword match. These studies do not consider location name ambiguity nor 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 (Eisenstein et al., 2010). Hong *et al.* considered the user's interest to model users' geographic behavior (Hong et al., 2012). Recently, Yuan *et al.* further investigated spatial, temporal, and topical aspects to model users' geographic activities (Yuan, Cong, Ma, Sun, & Magnenat-Thalmann, 2013). Flatow *et al.* addressed the task of fine-grained location estimation by identifying hyper-local *n*-grams based on a collection of geo-tagged tweets of a specific geographic area (Flatow, Naaman, Xie, Volkovich, & Kanza, 2015). 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. Chen *et al.* proposed a clustering based solution for associating the geo-tagged tweets that contain a POI name to its real venue (*i.e.,* a particular bookstore, a starbucks) by exploiting the check-in information of Foursquare (Chen, Joshi, Miura, & Ohkuma, 2014). Our work can be considered as the first step before this procedure since they did not consider the informal abbreviations and ambiguity of POI mentions.

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 (Liu et al., 2011; Ritter et al., 2011). Liu *et al.* proposed a two-stage NER system for tweets (Liu et al., 2011). 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 linguistical features are incorporated into a CRF model for further refinement. Ritter *et al.* developed a pipelined NLP tool for tweets called TwitterNLP (Ritter et al., 2011). It consists of POS tagger, shallow parsing, capitalization classifier and named entity recognition. They

[14] `http://www.geonames.org/`

reported superior performance compared to the retrained existing state-of-the-art systems. In our implementation, we applied TwitterNLP for POS tagging the tweets. Li *et al.* proposed an unsupervised NER solution by splitting tweet into non-overlapping segments (C. Li et al., 2012; C. Li, Sun, Weng, & He, 2013, 2015). 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 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 two-stage tagger is developed for the subtasks of candidate POI mention disambiguation and temporal awareness resolution respectively. We investigate three sets of features and evaluate their effectiveness in the two subtasks. Our experimental result show that TS-Petar achieves promising performance against all baseline methods and is efficient for real-time applications. While many corporations just shout out their message in Twitter, TS-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 TS-Petar, we will investigate the context derived from historical tweets from a user and the effectiveness of considering social relationships of users. Regarding temporal awareness, currently we only classify the temporal awareness into three coarse categories. To further extend the algorithm to distinguish more fine-grained temporal windows for POI_f and POI_p is also part of the future work.

Acknowledgment

This research was supported by Singapore Ministry of Education Academic Research Fund Tier 2 (MOE2014-T2-2-066), National Natural Science Foundation of China (61502344), Natural Science Foundation of Hubei Province (2015CFB337), and Natural Scientific Research Program of Wuhan University (2042015kf0014,2042016kf0190). Chenliang Li is the corresponding author.

References

Brown, P. F., deSouza, P. V., Mercer, R. L., Pietra, V. J. D., & Lai, J. C. (1992). Class-based N-Gram Models of Natural Language. *Comput. Linguist.*, *18*(4), 467–479.

Chang, A. X., & Manning, C. D. (2012). SUTime: A Library for Recognizing and Normalizing Time Expressions. In *Proc. of LREC* (p. 3735-3740).

Chang, H.-W., Lee, D., Eltaher, M., & Lee, J. (2012). @Phillies Tweeting from Philly? Predicting Twitter User Locations with Spatial Word Usage. In *Proc. of ASONAM* (p. 111-118).

Chen, F., Joshi, D., Miura, Y., & Ohkuma, T. (2014). Social Media-based Profiling of Business Locations. In *Proc. of GeoMM* (pp. 1–6).

Cheng, Z., Caverlee, J., & Lee, K. (2010). You Are Where You Tweet: A Content-based Approach to Geo-locating Twitter Users. In *Proc. of CIKM* (pp. 759–768).

Cowan, B., Zethelius, S., Luk, B., Baras, T., Ukarde, P., & Zhang, D. (2015). Named Entity Recognition in Travel-Related Search Queries. In *Proc. of AAAI* (pp. 3935–3941).

Eisenstein, J., O'Connor, B., Smith, N. A., & Xing, E. P. (2010). A Latent Variable Model for Geographic Lexical Variation. In *Proc. of EMNLP* (p. 1277-1287).

Flatow, D., Naaman, M., Xie, K. E., Volkovich, Y., & Kanza, Y. (2015). On the Accuracy of Hyper-local Geotagging of Social Media Content. In *Proc. of WSDM* (pp. 127–136).

Han, B., Cook, P., & Baldwin, T. (2012). Geolocation Prediction in Social Media Data by Finding Location Indicative Words. In *Proc. of COLING* (pp. 1045–1062).

Han, B., Cook, P., & Baldwin, T. (2014). Text-Based Twitter User Geolocation Prediction. *J. Artif. Intell. Res. (JAIR)*, *49*, 451–500.

Hong, L., Ahmed, A., Gurumurthy, S., Smola, A. J., & Tsioutsiouliklis, K. (2012). Discovering Geographical Topics in the Twitter Stream. In *Proc. of WWW* (pp. 769–778).

Ikawa, Y., Enoki, M., & Tatsubori, M. (2012). Location Inference Using Microblog Messages. In *Proc. of WWW (Companion)* (pp. 687–690).

Jurgens, D., Finethy, T., McCorriston, J., Xu, Y. T., & Ruths, D. (2015). Geolocation Prediction in Twitter Using Social Networks: A Critical Analysis and Review of Current Practice. In *Proc. of ICWSM* (pp. 188–197).

Kazama, J., & Torisawa, K. (2007). Exploiting Wikipedia as External Knowledge for Named Entity Recognition. In *Proc. of EMNLP-CoNLL* (p. 698-707).

Kinsella, S., Murdock, V., & O'Hare, N. (2011). "I'M Eating a Sandwich in Glasgow": Modeling Locations with Tweets. In *Proc. of SMUC* (pp. 61–68).

Kwak, H., Lee, C., Park, H., & Moon, S. (2010). What is Twitter, a Social Network or a News Media? In *Proc. of WWW* (pp. 591–600).

Lafferty, J. D., McCallum, A., & Pereira, F. C. N. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proc. of ICML* (pp. 282–289).

Li, C., & Sun, A. (2014). Fine-grained location extraction from tweets with temporal awareness. In *Proc. of SIGIR* (pp. 43–52).

Li, C., Sun, A., & Datta, A. (2013). TSDW: Two-stage Word Sense Disambiguation using Wikipedia. *JASIST*, *64*(6), 1203-1223.

Li, C., Sun, A., Weng, J., & He, Q. (2013). Exploiting Hybrid Contexts for Tweet Segmentation. In *Proc. of SIGIR* (pp. 523–532).

Li, C., Sun, A., Weng, J., & He, Q. (2015). Tweet Segmentation and Its Application to Named Entity Recognition. *IEEE Trans. Knowl. Data Eng.*, *27*(2), 558–570.

Li, C., Weng, J., He, Q., Yao, Y., Datta, A., Sun, A., & Lee, B.-S. (2012). TwiNER: Named Entity Recognition in Targeted Twitter Stream. In *Proc. of SIGIR* (pp. 721–730).

Li, W., Serdyukov, P., de Vries, A. P., Eickhoff, C., & Larson, M. (2011). The Where in the Tweet. In *Proc. of CIKM* (pp. 2473–2476).

Lieberman, M. D., Samet, H., & Sankaranarayanan, J. (2010). Geotagging with Local Lexicons to Build Indexes for Textually-specified Spatial Data. In *Proc. of ICDE* (p. 201-212).

Lingad, J., Karimi, S., & Yin, J. (2013). Location extraction from disaster-related microblogs. In *Proc. of WWW (Companion)* (p. 1017-1020).

Liu, X., Zhang, S., Wei, F., & Zhou, M. (2011). Recognizing Named Entities in Tweets. In *Proc. of ACL* (p. 359-367).

Mahmud, J., Nichols, J., & Drews, C. (2012). Where Is This Tweet From? Inferring Home Locations of Twitter Users. In *Proc. of ICWSM*.

Marcus, M. P., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a Large Annotated Corpus of English: The Penn Treebank. *Comput. Linguist.*, *19*(2), 313–330.

Rae, A., Murdock, V., Popescu, A., & Bouchard, H. (2012). Mining the web for points of interest. In *Proc. of SIGIR* (p. 711-720).

Rahimi, A., Cohn, T., & Baldwin, T. (2015). Twitter User Geolocation Using a Unified Text and Network Prediction Model. In *Proc. of ACL* (pp. 630–636).

Ratinov, L., & Roth, D. (2009). Design challenges and misconceptions in named entity recognition. In *Proc. of CoNLL* (pp. 147–155).

Ritter, A., Clark, S., Mausam, & Etzioni, O. (2011). Named Entity Recognition in Tweets: An Experimental Study. In *Proc. of EMNLP* (p. 1524-1534).

Rüd, S., Ciaramita, M., Müller, J., & Schütze, H. (2011). Piggyback: using search engines for robust cross-domain named entity recognition. In *Proc. of ACL-HLT* (pp. 965–975).

Sakaki, T., Okazaki, M., & Matsuo, Y. (2010). Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors. In *Proc. of WWW* (pp. 851–860).

Schulz, A., Hadjakos, A., Paulheim, H., Nachtwey, J., & Mühlhäuser, M. (2013). A Multi-Indicator Approach for Geolocalization of Tweets. In *Proc. of ICWSM.*

Verhagen, M., Gaizauskas, R., Schilder, F., Hepple, M., Moszkowicz, J., & Pustejovsky, J. (2009). The TempEval Challenge: Identifying Temporal Relations in Text. *Language Resources and Evaluation*, *43*(2), 161-179.

Verhagen, M., Mani, I., Sauri, R., Knippen, R., Jang, S. B., Littman, J., … Pustejovsky, J. (2005). Automating Temporal Annotation with TARSQI. In *Proc. of ACL* (pp. 81–84).

Weng, J., & Lee, B.-S. (2011). Event Detection in Twitter. In *Proc. of ICWSM* (pp. 401–408).

Yao, Y., & Sun, A. (2015). Mobile phone name extraction from internet forums: a semi-supervised approach. *World Wide Web*, 1-23.

Yuan, Q., Cong, G., Ma, Z., Sun, A., & Magnenat-Thalmann, N. (2013). Who, Where, When and What: Discover Spatio-temporal Topics for Twitter Users. In *Proc. of KDD* (p. 605-613).

Zhang, T., & Johnson, D. (2003). A Robust Risk Minimization Based Named Entity Recognition System. In *Proc. of CoNLL* (pp. 204–207).