3-2019

Fine-grained geolocation of tweets in temporal proximity

Wen Haw CHONG
Singapore Management University, whchong.2013@phdis.smu.edu.sg

Ee Peng LIM
Singapore Management University, eplim@smu.edu.sg

DOI: https://doi.org/10.1145/3291059

Fine-grained Geolocation of Tweets in Temporal Proximity

WEN-HAW CHONG and EE-PENG LIM, Singapore Management University, Singapore

In fine-grained tweet geolocation, tweets are linked to the specific venues (e.g., restaurants, shops) from which they were posted. This explicitly recovers the venue context that is essential for applications such as location-based advertising or user profiling. For this geolocation task, we focus on geolocating tweets that are contained in tweet sequences. In a tweet sequence, tweets are posted from some latent venue(s) by the same user and within a short time interval. This scenario arises from two observations: (1) It is quite common that users post multiple tweets in a short time and (2) most tweets are not geocoded. To more accurately geolocate a tweet, we propose a model that performs query expansion on the tweet (query) using two novel approaches. The first approach *temporal query expansion* considers users' staying behavior around venues. The second approach *visitation query expansion* leverages on user revisiting the same or similar venues in the past. We combine both query expansion approaches via a novel fusion framework and overlay them on a Hidden Markov Model to account for sequential information. In our comprehensive experiments across multiple datasets and metrics, we show our proposed model to be more robust and accurate than other baselines.

CCS Concepts: • **Information systems** → **Data mining**; *Geographic information systems*;

Additional Key Words and Phrases: Tweet geolocation, temporal proximity, staying behavior

ACM Reference format:
Wen-Haw Chong and Ee-Peng Lim. 2019. Fine-grained Geolocation of Tweets in Temporal Proximity. *ACM Trans. Inf. Syst.* 37, 2, Article 17 (January 2019), 33 pages.
https://doi.org/10.1145/3291059

1 INTRODUCTION

Usage of social media platforms such as Twitter is becoming increasingly popular. This has led to opportunities to mine user posts to study various behavioral patterns or to support novel applications. In particular, there are potential benefits for location-sensitive applications such as venue recommendation, location-based advertising, urban planning, and so on. However, the sparsity of location information in user posts remains a technical challenge. Studies have shown that as much as 98% [1, 21] of the tweets posted are not geocoded. This motivates the problem of geolocating individual tweets whereby one infers the posting location of a tweet, based on some specification of location granularity. Location granularities can be at the city level [5, 6], grid cell [36, 46, 52], coordinates [1, 21], or at the finest level down to specific posting venues [4, 8, 9, 27, 30].

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its International Research Centres in Singapore Funding Initiative, and DSO National Laboratories.
Authors' addresses: W.-H. Chong and E.-P. Lim, Singapore Management University, 80 Stamford Road, Singapore, 178902, Singapore; emails: whchong.2013@phdis.smu.edu.sg, eplim@smu.edu.sg.

1.1 Fine-grained Geolocation

In this study, we infer the specific posting venues of tweets such as a shop, restaurant, and so on. For example, when a user posts a tweet "Just had pasta," we would like to geolocate it to a specific restaurant using its content and any auxiliary information. By doing so, we recover the venue context explicitly, an aspect that is absent when geolocating on a more coarse-grained level. For example, geolocation to coordinates or grid cells means that the posting location [1, 21] may be associated with many venues sharing the same coordinates or grid cell, e.g., venues in a high rise building. Following the problem formulation in References [8, 9, 30], we solve fine-grained geolocation as a ranking problem. Given a target tweet we rank venues such that high ranked venues are more likely to be the posting venue.

In the above problem definition, we assume that the tweet to be geolocated is posted from some venue within a known city, based on the profile of the posting user. For the problem to be challenging yet meaningful, we do not assume that we have all fine-grained venues within the city. First, such venues easily number in the hundreds of millions. Second, it is very costly to construct a knowledge base that covers all possible city venues. Instead we consider venues that have some minimal presence in social media, defined to be associated with some minimum number of tweets. Even with this consideration, the number of candidate venues typically range in the thousands.

1.1.1 Geolocation Scenarios. We geolocate *tweets contained in parent tweet sequences*,[1] whereby tweets in the same sequence are posted close in time by the same user. Sequences can be of any length larger than one. This scenario is motivated by our observation that it is common for users to post multiple tweets within a short time interval. For example, of 1,000 randomly sampled tweets from Singapore, 58.1% of them involves the user posting another tweet within 30 minutes of the first tweet. Repeating the analysis for Jakarta, such cases constitute 48.9%. Such user behavior can be due to various reasons such as to push out more content or to overcome the short message length constraint of individual tweets. In any case, tweet sequences are fairly common. Given a tweet targeted for geolocation, we can potentially improve geolocation accuracy by exploiting its parent tweet sequence. To our knowledge, such a problem scenario has not been previously studied for fine-grained geolocation.

In our geolocation scenario, we assume that no tweets in the parent sequence are associated with any location coordinates or posting venues. This is a prevalent and realistic scenario due to the scarcity of geocoded tweets. To make the task even more realistic, we also assume that the *target tweet's user has no observed location history*, i.e., has not posted any geocoded tweets. This allows our geolocation methods to be applicable to tweets from almost any users. Clearly, the geolocation task also becomes more challenging, since one is not able to exploit the home or activity regions [8] of the users to refine candidate posting venues.

1.1.2 Problem Definition. We now define our problem formally. Denote $\mathbf{w}$ as a tweet targeted for geolocation. Let $\mathbf{w}$ be posted by user u at time t, contained in a sequence $N_T(\mathbf{w}; u)$ whereby all other tweets in $N_T(\mathbf{w}; u)$ are posted by u at not more than T time away from t. Assuming that $\mathbf{w}$ is posted from a latent venue v from the set $\{v_i\}_{i=1}^{V}$ with V venues, our goal is to rank the V venues such that v is ranked as near the top as possible. To solve this problem, we shall exploit various information including those in $\mathbf{w}$, $N_T(\mathbf{w}; u)$, and so on.

1.1.3 Examples. To illustrate the usefulness of parent tweet sequences, Table 1 displays tweet pairs, each spanning a short time interval. These tweets are processed tweets originating from Foursquare (See Section 2.1). Tweets a1 and a2 are posted by one user while b1 and b2 are by

[1]We use the terms parent sequence, parent tweet sequence and tweet sequence interchangeably.

Table 1. Sample Pairs of Tweets

a1	(Nanyang Polytechnic, 08:36:20) "Morning rush to the airport and now I'm in school!"
a2	(Nanyang Polytechnic, 08:37:37) "Eyebag zzzzz"
b1	(Tampines MRT Station, 09:44:22) "Keep tripping."
b2	(Tampines Bus Interchange, 09:48:17) "Topped up my Ez-link"

Posting venue and time are in brackets. Tweets a1 and a2 are from one user while b1 and b2 are from another user.

another user. Consider a1 and a2 that are posted from Nanyang Polytechnic, a college venue. The user provides more information in a1, suggesting that he is in school. Since a1 precedes a2 by only 1 minute, we can use a1's content to augment a2 to better geolocate the latter. This helps when a tweet targeted for geolocation has little content or content unrelated to the posting venue. A similar argument applies for b1 and b2. b2 mentioned about topping up of Ez-link, the farecard used in Singapore's subway system (MRT[2]). This allows us to geolocate b1 to some subway station, thus improving geolocation accuracy. In the discussed examples, a1 and b2 are the more informative tweets that help to improve geolocation for their neighboring tweets. Certainly it is also possible for non-informative tweets to negatively affect geolocation accuracy for other tweets. The research question is then to design robust approaches such that on an overall basis, geolocation accuracy is improved.

1.2 Analogy to Document Retrieval

Geolocating tweets to specific posting venues is a problem that is analogous to document retrieval. One can regard a tweet targeted for geolocation as a query and candidate venues as akin to documents. Ranking the candidate venues for the targeted tweet is then akin to ranking the documents based on relevance to the query. However for tweet geolocation, there is only one posting venue per tweet, i.e., one relevant document per query.

While fine-grained geolocation can be casted as document retrieval, there are differences between documents and venues. Importantly, venues have a natural ordering in the spatial sense while documents do not. Tweets posted close in time and by the same user are also inadvertently ordered in space, since the user is likely to be posting from the same or nearby venues. Clearly, such spatial ordering are useful and can be exploited for better geolocation of tweets.

1.3 Approach

Given that fine-grained tweet geolocation is akin to document retrieval, certain techniques such as query expansions [3, 42, 53] can be adapted from the retrieval domain. We leverage on this to propose a probabilistic model that geolocates tweets contained in sequences. Our model does not rely on the need to explicitly identify informative and uninformative tweets in sequences. Instead, we treat each target tweet as a query and design query expansion approaches to augment it with additional words for better geolocation. The additional words are added both from tweets in the parent sequence, termed as *temporal query expansion* and from other tweets from the same user, termed as *visitation query expansion*. We also relate these query expansion approaches to intuitive

[2]Mass Rapid Transit.

user behavior. Basically temporal query expansion approach accounts for the user tendency to stay at the same or nearby venues given a short time period, i.e., staying behavior, while visitation query expansion accounts for revisits to the same or similar venues (even without explicitly observing the revisits). We combine both query expansion approaches in a novel fusion framework and overlay them on a Hidden Markov Model.

1.4 Challenges

Fine-grained geolocation is a challenging problem as there are thousands of candidate posting venues to consider. In addition, tweets are brief in content and highly colloquial. For example, the tweet "having dinner" may have been posted from any one of the numerous restaurants in the city or even at a user's home.

Although we are geolocating tweets contained in sequences, the geolocation scenario remains challenging as we assume there are no observed posting venues in the parent sequence. To understand this, consider the alternative scenario with observed venues. Then for a targeted tweet, the observed venues of adjacent tweets can be exploited for reducing the set of candidate venues. This is because within a short time interval, the user is likely to be posting either at the same venue or at nearby venues.

Outside of the tweet sequence, we also assume that a targeted tweet's user do not have any observed location history. Thus even if he only frequents a few venues, making it likely that the targeted tweet is posted from either one of these venues, these venues are unobserved and not easy to exploit in an explicit manner.

1.5 Contributions

Our contributions are as follows:

(1) We formulate the interesting problem of fine-grained geolocation of tweets contained in parent tweet sequences. To our knowledge, such a geolocation scenario is highly common but has not been previously investigated.
(2) We conduct empirical analysis to verify the tendency of users to stay at the same or nearby venues given a short time period, i.e., staying behavior. We also study the tendency of users to revisit venues. Such user behavior motivates the design of our models.
(3) We propose *temporal query expansion*, which accounts for the staying behavior of users. In this expansion approach, the target tweet is augmented with words from other tweets in its parent tweet sequence.
(4) We propose *visitation query expansion*, which augments the target tweet with semantically related words from the user's other tweets. This accounts for the user's repeat visits to the same or similar venues.
(5) We combine both query expansion approaches in a novel fusion framework, which is then overlaid on a Hidden Markov Model to capture sequential information. Through extensive experiments, we show that the resulting model is robust and outperforms pure query expansion approaches and other baselines. Depending on the dataset and metric, performance improvement ranges from 4+% to 40+% over the naive Bayes baseline.

1.6 Outline of Article

The next section discusses how we obtain venue-associated tweets for this work. Section 3 presents empirical analysis that motivates the query expansion components in our model. Section 4 describes our model while Section 5 presents experiment results, along with detailed analysis and case studies. We discuss related work in Section 6. Finally, we conclude in Section 7.

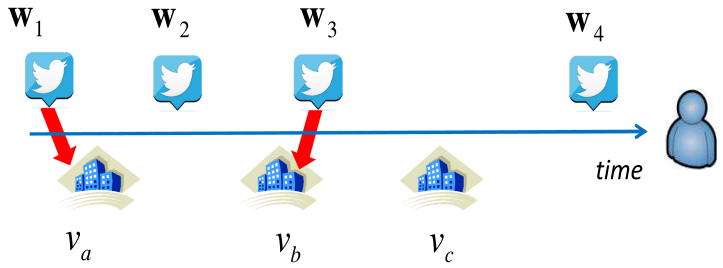


Fig. 1. Illustration of linking process.

2 TWEETS WITH POSTING VENUES

For model building and testing, we need ground truth datasets consisting of tweets with their posting venues. We use two types of tweets with such associations. The first type is content pushed to Twitter from location-apps, e.g., Foursquare. The second type is user-authored tweets in Twitter, which we link to venues using the approach from Reference [9]. Next we describe the processing steps for the two tweet types and the datasets constructed for this research.

2.1 Shouts

We use content pushed from Foursquare, a popular location app. In Foursquare, users can broadcast their comments to Twitter as they check-in to a venue. Following Foursquare terminology, we refer to such tweets as *shouts*. We process and treat shouts as normal tweets with known posting venues. This setup is a convenient source of ground truth and has been used in prior work [4, 30].

A shout contains the user-authored comment plus an app-generated portion indicating the check-in venue. We discard the latter portion and use only the comments for geolocation. For example, consider the sample shout "**Passport photo look retarded** (@ Immigration & Checkpoints Authority w/ 5 others)." Only the user-authored comments (bolded) are used for geolocation and empirical analysis.

2.2 Pure Tweets

We refer to tweets that are authored by users and non-retweets as *pure tweets* [8, 9]. Since most tweets are not geocoded, we assign some of them to venues by associating them with check-ins by their users in Foursquare around the same time. We first select users who post Foursquare shouts on Twitter. We then iterate through their pure tweets and link them to the posting venues of check-ins that occur around the same time. A pure tweet is assumed to be posted from a check-in venue when the tweet and check-in are performed close in time to each other by the same user. To minimize cases where a tweet is linked wrongly to a venue, we use a stringent threshold of 5 minutes for linking. Also note that this threshold of 5 minutes is used only for dataset construction. It is neither used to define our sequences nor has any relation to the time interval of sequences.

Figure 1 illustrates the linking process with an example. The user writes pure tweets ($\mathbf{w}_1$, $\mathbf{w}_2$, $\mathbf{w}_3$, $\mathbf{w}_4$) in Twitter as well as checkin to venues (v_a, v_b, v_c) in Foursquare. Pure tweets and checkins are displayed from left to right in chronological order on the timeline. In this example, $\mathbf{w}_1$ is nearest in time to the first checkin involving venue v_a. If the time difference is less than 5 minutes, then we assume $\mathbf{w}_1$ to have been posted from v_a. Similarly, $\mathbf{w}_3$ is assumed to be posted from v_b. In this example, $\mathbf{w}_2$ and $\mathbf{w}_4$ are not assigned any venues as they are each posted too far away in time from the nearest checkin. Their posting venues remain unknown.

Our linking approach requires the pure tweets' users to have visited the linked venues around the time they post their pure tweets. This is more stringent than just using geocoded pure tweets

Table 2. Dataset Statistics

	SG-SHT	SG-TWT	JKT-SHT
Tweets	361,899	90,250	86,343
Users	29,301	6,424	12,119
Venues	65,701	12,616	45,213
Tweets/user	12.35	14.05	7.12
Tweets/venue	5.51	7.15	1.91

with location coordinates and assigning them to the nearest venues. The latter is unsatisfactory in an urban setting, since many venues may share the same location coordinates, e.g., in a high rise building.

Terminology. For the rest of this article, "tweets" refer to both pure tweets and shouts. Where differentiation is required, we use each term explicitly, i.e., pure tweets or shouts.

2.3 Datasets

We collect data for users from Singapore (SG) and Jakarta (JKT). For Singapore, we collected 1,190,522 Foursquare check-ins from the year 2014, of which 361,899 (30.4%) involve shouts, which we regard as tweets. The check-ins are posted by 29,301 users over 65,701 venues. We use only the shouts for profiling venues in terms of content. We refer to this dataset as **SG-SHT**. Based on the previously discussed process, we also collected 90,250 pure tweets from 6,424 users over 12,616 venues, which we designate as **SG-TWT**. For Jakarta, the **JKT-SHT** dataset comprises 177,570 check-ins for the period 2015 to mid-2016, of which 86,343 (48.6%) are shouts. The check-ins are from 12,119 users over 45,213 venues.

Subsequently, we conduct our experiments over the 3 datasets: SG-SHT, SG-TWT, and JKT-SHT.[3] The dataset statistics are summarized in Table 2. As can be seen, JKT-SHT has the smallest number of posts per user and venue, on average. Hence information may be sparser, when compared to the other two datasets.

3 EMPIRICAL ANALYSIS

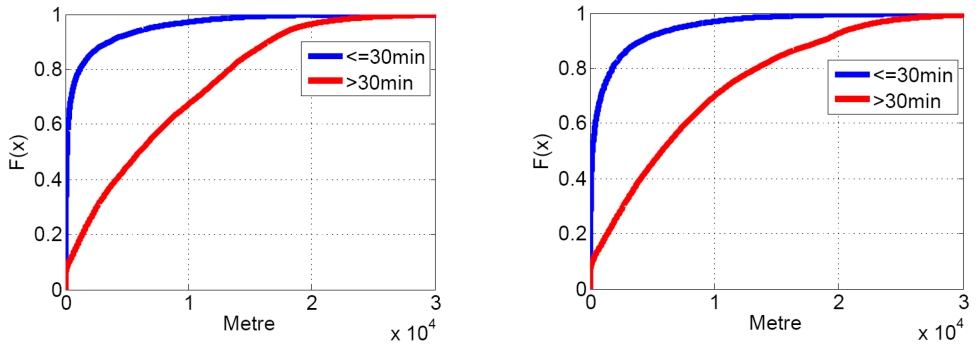
We conduct several empirical studies to verify our intuitions about user behavior and to motivate the design of our models.

3.1 Staying Behavior

Staying behavior refers to users' tendencies to remain at the same venue or traverse only between nearby venues given a short observed time interval. This is intuitive, since, first, some time is required for users to conduct activities at venues, e.g., work, school, dining. Second, time is also required for a user to move from one venue to another. If only a short time have lapsed, then a user is less likely to have travelled far.

In our first empirical study, we show that staying behavior is an established property. Basically for a given user, his consecutive shouts posted close in time are likely to have been posted from venues near each other. To analyze this, we compute the distances between sampled pairs of shouts, whereby each pair is posted by a common user within 30 minutes. We compare this against a null model whereby sampled pairs are posted by a common user more than 30 minutes apart.

[3]Due to platform API changes that affected crawling, the pure tweet dataset for Jakarta is small with only 1335 pure tweets. Hence, we omit it.



(a) CDF (SG-SHT)

(b) CDF (JKT-SHT)

Fig. 2. CDF for distances between sampled shout pairs. Each pair is posted by a common user. Shout pairs are differentiated by pairs posted within 30 minutes of each other (≤30 min); and pairs posted more than 30 minutes apart (>30 min). X-axis is distance in meters.

Figure 2 shows the Cumulative Distribution Functions (CDF) for Singapore (SG-SHT) and Jakarta (JKT-SHT). In each graph, the blue curve represents sample pairs within 30 minutes (≤30 min) while the red curve is for sample pairs more than 30 minutes (>30 min) apart. Evidently, both graphs display strong evidence of staying behavior. In both cases, the blue curve lies to the left of the red curve, thus shouts within 30 minutes of each other are more likely to be posted from nearer venues, compared to the null model. For example in Figure 2(a) for Singapore, more than 95% of sample pairs with posting time difference ≤30 min are posted at distances of 10,000m or below. In contrast, a similar distance covers only around 64% of sample pairs with posting time difference > 30 min. Figure 2(b) shows a similar trend for Jakarta.

3.2 Visitation Behaviour

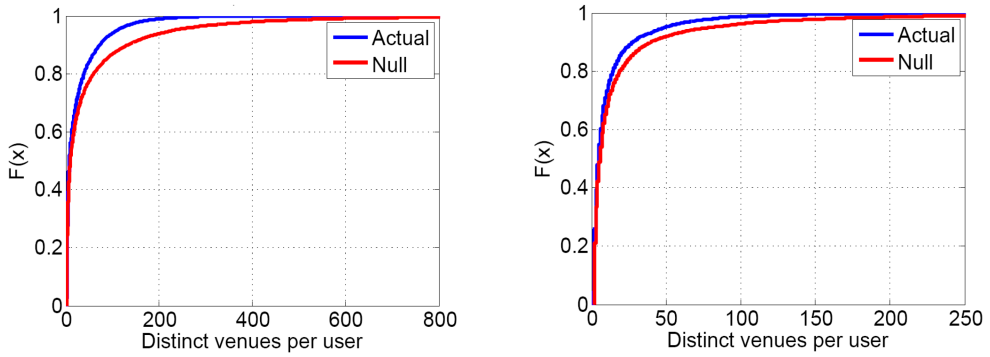
Besides staying behavior, we can potentially exploit other visitation behavior that users exhibit. In particular, users may visit the same venue multiple times for recurring activities, e.g., work, or visit venues around a common area or functionality, e.g., movie theatres. This means the possibility of augmenting an uninformative target tweet with more informative words from his other tweets. The following repeat visit scenarios arise:

- **Same venue**: The user may have tweeted from the target tweet's venue before and used more informative words.
- **Same spatial region**: The same user tweeting from venues near each other may mention local geographical features. For example, consider a user tweeting about dinner at a certain quayside restaurant. If he frequently tweets about dinner and the quay from other venues in the area, then these other tweets can be indicative of the target tweet's venue due to word co-occurrence relationships.
- **Same function**: The target tweet's venue may belong to a functional group of venues that the user frequents, e.g., nightclubs. Functionally related words, e.g., "clubbing" can be indicative of the target tweet's venue.

3.2.1 Repeat Visits to Same Venue. In our first empirical analysis, we measure repeat visits to the same venue, which is the most straightforward to quantify. We examine shouts and tabulate the frequencies of repeated visits to venues, on a per user basis. Given user u and venue v, denote

Table 3. Analysis of Repeat Visit to Same Venues

	Singapore	Jakarta
No. of tuples	603,198	108,428
tuples with freq=1	465,256 (77.13%)	88,219 (81.36%)
tuples with freq>1	137,942 (22.87%)	20,209 (18.64%)



(a) CDF (SG-SHT) (b) CDF (JKT-SHT)

Fig. 3. CDF for distinct venues per user.

the user-venue tuple as (u, v). We then iterate through all shouts and tabulate the frequencies of each tuple. Repeat visits are then user-venue tuples that occur more than once.

Table 3 shows that the proportion of repeat visits is substantial at 22.87% for Singapore (SG-SHT) and 18.64% for Jakarta (JKT-SHT). Thus, repeat visits to the same venue is an established user behavior. This and the other visitation behavior imply the presence of more informative words beyond the target tweet and its parent sequence. With proper query expansion techniques, one can exploit these words to more accurately geolocate the target tweet.

3.2.2 User Focus on Venues. Alternatively, we can quantify venue revisits by studying if users are focused on a smaller set of venues than expected. This means comparing the number of distinct venues visited against some null model where repeat visitation behavior is absent. If a user repeatedly visits one or more venues, then we expect him to be posting multiple tweets from a smaller set of distinct venues, when compared against the null model.

For each user u with multiple tweets, we first compute the number of distinct venues that his tweets are posted from. We then compute the expected number of distinct venues under the null model as:

- For each tweet from u, sample a venue v based on global venue probability i.e., venue popularity. Add to venue set $\mathbb{V}_{null}(u)$.
- Compute the size of $\mathbb{V}_{null}(u)$. This is the distinct venue count under the null model.

As the null model involves sampling, we conduct 10 runs and take the average expected count for each user. We conduct this empirical analysis on 22,488 users from SG-SHT and 8419 users from JKT-SHT who have posted at least twice. For users who have posted only once, the number of distinct venues is one and not meaningful to study.

Figure 3 plots the CDF of distinct venues visited per user for Singapore (SG-SHT) and Jakarta (JKT-SHT). In each graph, the blue curve represents the actual count while the red curve is for

counts from the null model (averaged over 10 runs). For both graphs, the blue curve lies to the left of the red curve. This indicates that users have repeat visitation behavior and visit fewer distinct venues than expected under the null model. For example in Figure 3(a) for SG-SHT, close to 100% of the users post from 200 distinct venues or less in the actual data. In comparison, the null model has a corresponding proportion of around 90%. For Figure 3(b) for JKT-SHT, the differences between the actual and null model count is smaller, since JKT-SHT is relatively more sparse in terms of tweets per user (see Table 2). Nonetheless, the differences are easily perceivable. Around 95% of users post from 50 distinct venues or less in the actual data. Under the null model, the corresponding proportion is around 90%. Hence there is evidence that users are revisiting some of the venues in their travel patterns. Therefore, we hope to achieve better geolocation accuracy by exploiting such behavior in our models.

4 MODELS

In this section, we first describe the base model, followed by the proposed query expansion and fusion approaches. We use each model to compute the probabilities of candidate venues given the tweet content and other information. We then rank venues based on the computed probabilities. Also note that in subsequent discussions, *Temporal neighbors* of the target tweet refer to other tweets in its parent sequence.

4.1 Base Model (Nb)

We use the naive Bayes model from References [26, 27] as the base model for query expansion. It models the word generative probabilities of a venue by accumulating and smoothing word frequencies over all tweets posted from the venue. The probability of word w given venue v is computed as

$$p(w|v) = \frac{f(w,v) + \alpha}{f(.,v) + W\alpha}, \tag{1}$$

where W is the vocabulary size, $f(w,v)$ is the frequency of word w at venue v, $f(.,v) = \sum_w f(w,v)$ and α is the smoothing parameter that can be tuned or set at 1 for Laplace smoothing.

Given a target tweet $\mathbf{w}$, we can rank venues by:

$$p(v|\mathbf{w}) \propto p(v) \prod_{w \in \mathbf{w}} p(w|v)^{c(w,\mathbf{w})}, \tag{2}$$

where $c(w,\mathbf{w})$ is the frequency of word w in $\mathbf{w}$ and $p(v)$ is the probability of venue v that can be estimated globally from posting frequencies.

4.2 Temporal Query Expansion (Temporal)

Staying behavior suggests that a user posting multiple tweets within a short time is likely to be posting from the same or nearby venues. Hence given a target tweet, words from other tweets in its parent sequence may be informative. Formally, given a tweet $\mathbf{w}$ posted by user u, we define its parent sequence $N_T(\mathbf{w};u)$ as tweets from the same user u that are posted not more than T time away from $\mathbf{w}$'s posting time. T is known as the *parent time window*. It can be tuned but is expected to be small, e.g., 0.5 hour.

We propose temporal query expansion to augment the target tweet with candidate words based on their occurrence frequencies in the parent sequence and weighted by temporal proximity to the target tweet. Words occurring closer in time to the target tweet are assigned greater weights than words occurring further away in time. To model this, we use the exponential kernel [16]. Let target tweet $\mathbf{w}$ be posted by user u at time t, with the set of temporal neighbors from the parent

sequence $N_T(\mathbf{w};u)$, whereby the jth tweet of $N_T(\mathbf{w};u)$ is denoted as $\mathbf{w}_j$ and posted at time t_j by the same user u. The set of temporal neighbors fulfils the condition $|t - t_j| \leq T, \forall \mathbf{w}_j \in N_T(\mathbf{w};u)$. We then weigh each word w as

$$\delta_S(w, \mathbf{w}; u) = c(w, \mathbf{w}) + \sum_{\mathbf{w}_j \in N_T(\mathbf{w};u)} c(w, \mathbf{w}_j) \exp(-S|t - t_j|), \tag{3}$$

where $c(w, \mathbf{w}_j)$ counts occurrences of w in $\mathbf{w}_j$ and the kernel parameter S is a tunable time decay factor. S controls the rate at which word influence diminishes with time difference within the interval T. A larger S corresponds to a larger decay rate. Note that word influence is 0 outside the interval T. Hence even if S=0, there is no decay only within the interval T.

Considering that $c(w, \mathbf{w})$=$c(w, \mathbf{w}) \exp(-S|t - t|)$, then Equation (3) can be viewed as a weighted sum of exponential kernels. It covers three possible cases of word occurrences as follows:

- Word w occurs only in the target tweet. Equation (3) reduces to $\delta_S(w, \mathbf{w}; u) = c(w, \mathbf{w})$.
- Word w occurs only in the temporal neighbors. $c(w, \mathbf{w})$=0 and only the right-most term of Equation (3) is retained.
- Word w is in both the target tweet and temporal neighbors. The weight for w is summed over its occurrences in both the target tweet and temporal neighbors.

We incorporate $\delta_S(w, \mathbf{w}; u)$ into our base model as follows:

$$p(v|\mathbf{w}, N_T(\mathbf{w};u)) \propto p(v) \prod_{\{w:\delta_S(w,\mathbf{w};u)>0\}} p(w|v)^{\delta_S(w,\mathbf{w};u)}, \tag{4}$$

whereby it suffices to consider the set of words with $\delta_S(w, \mathbf{w}; u) > 0$. Interestingly, Equation (4) corresponds to a weighted naive Bayes model, which was previously applied for classification [14, 55]. In the prior work with weighted naive Bayes, the goal was to improve classification accuracy via feature weighting based on distributional differences between classes. Here via temporal query expansion, we have derived a weighted naive Bayes model for tweet geolocation. Instead of classification accuracy, we shall tune the model with respect to ranking accuracy (see Section (5)).

4.3 Visitation Query Expansion (Visit)

In visitation query expansion, we expand the target tweet with words from the user's other tweets that may be indicative of the posting venue, due to different repeat visit scenarios. We note that visitation query expansion is applicable for geolocating both tweets with and without temporal neighbors. Also note that in our considered geolocation scenario, the target tweet's user have no location history (see Section 1.1.1). Hence tweets acting as a source of candidate words are neither geocoded nor associated with any posting venues.

Given target tweet $\mathbf{w}$ (i.e., query) from user u, we score candidate words w' that appears in u's other tweets and where $w' \notin \mathbf{w}$. The scoring aims to assess w''s suitability for augmenting $\mathbf{w}$ and are designed to reflect the relationship strength to the original query words $w \in \mathbf{w}$. Many scoring schemes exist and various suitable kernel functions can be used. For simplicity, we adopt a cosine similarity scheme [11]. This uses the normalized form of the dot product kernel, also referred to as the cosine kernel.

Let $\mathbf{I}_u(w)$ be a vector of indicator functions for the presence of word w in u's tweets. For a candidate word w' whereby $w' \notin \mathbf{w}$, we compute its *average relatedness* $\alpha(w', \mathbf{w}; u)$ to the original

query words, as the average of cosine kernels:

$$\begin{aligned}\alpha(w', \mathbf{w}; u) &= \frac{1}{|\mathbf{w}|} \sum\nolimits_{w \in \mathbf{w}} \frac{\langle \mathbf{I}_u(w'), \mathbf{I}_u(w) \rangle}{\|\mathbf{I}_u(w')\| \|\mathbf{I}_u(w)\|} \\ &= \frac{1}{|\mathbf{w}|} \sum\nolimits_{w \in \mathbf{w}} \frac{d_u(w', w)}{\sqrt{d_u(w) d_u(w')}},\end{aligned} \tag{5}$$

where $0 \leq \alpha(w', \mathbf{w}; u) \leq 1$, $d_u(w', w)$ is the count of u's tweets with both w' and w; and $d_u(w)$ is the count of u's tweets with w. Intuitively, words that co-occur more are more related. However average relatedness is dampened if one or both words are overly common.

Let $\{\mathbf{w}'\}_u$ denote the set of non-target tweets of user u. For a target tweet $\mathbf{w}$ from u, $\{\mathbf{w}'\}_u$ also includes the temporal neighbors of $\mathbf{w}$ if there are any. We incorporate the word weights $\alpha(w', \mathbf{w}; u)$ into our base model as follows:

$$p(v|\mathbf{w}, \{\mathbf{w}'\}_u) \propto p(v) \prod_{w \in \mathbf{w}} p(w|v)^{c(w, \mathbf{w})} \prod_{\substack{\{w': w' \notin \mathbf{w}, \\ \alpha(w', \mathbf{w}; u) > 0\}}} p(w'|v)^{\alpha(w', \mathbf{w}; u)}. \tag{6}$$

Equation (6) highlights that there are two groups of words: words already in the target tweet and words that are newly added. Each occurrence of a target tweet word has implicit weight of 1, while newly added words are weighted between 0 and 1 depending on their relatedness to the target tweet.

Finally, we note that query expansion can be conducted over the global set of tweets, instead of a user-specific set. This captures different notions rather than revisit behavior, while being more expensive and less personalized. For example, consider a target tweet with the word "dinner." Such a common word occurs in many tweets, leading to a huge set of candidate words for consideration. Geolocation may also be biased towards popular dinner venues, rather than being personalized to the target tweet's user. Nonetheless, for less common words or users with few tweets in their history, considering the global set of tweets may overcome information sparsity. We defer such exploration to future work.

4.4 Fusion Framework

In this section, we introduce a fusion framework to combine the above two query expansion approaches while mitigating the noise effects of any uninformative tweets from the target tweet's user.

Our query expansion approaches are based on kernels and fusing them is akin to *multiple kernel learning* [17]. In multiple kernel learning, one combines multiple kernels computed over different feature sets or capturing different data point similarities, such that the combined kernels perform better for the end task. Here, we fuse the kernels of temporal and visitation query expansions to compute a final weight for each word in the expanded target tweet. To capture both staying and repeat visitation behavior of users, we propose a novel "Max" combination approach. In addition, we consider simple kernel combination schemes such as linear and product combinations [10]. Our subsequent experiments show that the "Max" combination approach is more robust, performing either on par or better than the linear and product combination scheme across all datasets.

4.4.1 Max Combination (Max). Consider augmenting a targeted tweet $\mathbf{w}$ from u with candidate word w. Temporal query expansion prescribes augmentation using a weight of $\delta_S(w, \mathbf{w}; u)$ for w while visitation query expansion prescribes a weight of $\alpha(w, \mathbf{w}; u)$. At geolocation time, it is not known which candidate weight should be assigned or equivalently, whether staying or repeat visitation behavior is more important. Intuitively, one can adopt a catch-all approach to cover both behavior types. Considering the union of behaviors, then the candidate weight is either $\delta_S(w, \mathbf{w}; u)$

or $\alpha(w, \mathbf{w}; u)$, whichever weight is of larger value. This leads to the "Max" combination approach, where we adopt the maximum weight for each word over temporal and visitation query expansion. The intuition is that words are relevant for geolocating the target tweet *either* due to them being close in time (i.e., in the parent sequence), or being semantically related to the target tweet. Equivalently we cover both different behaviors: the user revisits the same or similar venue and/or stays around the posting venue of the target tweet. Formally, we compute:

$$p(v|\mathbf{w}, \{\mathbf{w}'\}_u) \propto p(v) \prod_{\substack{\{w:\delta_S(w,\,\mathbf{w};\,u)>0|\\ \alpha(w,\,\mathbf{w};\,u)>0\}}} p(w|v)^{\max(\delta_S(w,\mathbf{w};u),\,\alpha(w,\mathbf{w};u))}, \tag{7}$$

where the product of $p(w|v)$'s is computed over the union of words with non-zero weights from temporal query expansion and those from visitation query expansion. Equation (7) also means words from the target tweet are always assigned weights from temporal query expansion, i.e., $\delta_S(w, \mathbf{w}; u)$. For such words, $\delta_S(w, \mathbf{w}; u) \geq c(w, \mathbf{w}) \geq \alpha(w, \mathbf{w}; u)$. For words not in $\mathbf{w}$, their final weights depend on which query expansion scheme gives larger weights.

4.4.2 Linear Combination (Linear). The linear scheme defines the weight of a candidate word w as $\lambda\delta_S(w, \mathbf{w}; u) + (1 - \lambda)\alpha(w, \mathbf{w}; u)$, which leads to the following model:

$$p(v|\mathbf{w}, \{\mathbf{w}'\}_u; \lambda) \propto p(v) \prod_{\substack{\{w:\delta_S(w,\,\mathbf{w};\,u)>0|\\ \alpha(w,\,\mathbf{w};\,u)>0\}}} p(w|v)^{\lambda\delta_S(w,\mathbf{w};u)+(1-\lambda)\alpha(w,\mathbf{w};u)}, \tag{8}$$

where λ is the linear combination weights. In the linear scheme, each word is assigned a fixed proportion of importance based on its temporal proximity and relatedness to the target tweet. Thus, for every target tweet, one assumes a fixed relative importance from revisiting and staying behavior.

4.4.3 Product Combination (Product). Finally, the product scheme defines the weight of candidate word w as $\delta_S(w, \mathbf{w}; u) \times \alpha(w, \mathbf{w}; u)$. The resulting model is then

$$p(v|\mathbf{w}, \{\mathbf{w}'\}_u) \propto p(v) \prod_{\substack{\{w:\delta_S(w,\,\mathbf{w};\,u)>0|\\ \alpha(w,\,\mathbf{w};\,u)>0\}}} p(w|v)^{\delta_S(w,\mathbf{w};u)\times\alpha(w,\mathbf{w};u)}. \tag{9}$$

In the product scheme, a word has non-zero weight only if it is both semantically related *and* in temporal proximity to the target tweet. This assumes a stringent case where both revisiting *and* staying behavior must be present.

4.5 Sequential Information (HMM-Max)

Given that we are geolocating tweets contained in parent sequences, sequential information may help to improve geolocation, e.g., users may follow certain visit sequence in their daily travels. So far, neither temporal nor visitation query expansion explicitly models sequential information. To exploit such information, we adapt the sequence modeling approach from Reference [31] based on Hidden Markov Models (HMM). We model the hidden states in the Markov chain as venues and emissions as the tweet words. The probability that a tweet is posted from a venue is then computed from marginalizing over the hidden states in the sequence. This is done using the forward-backward algorithm [43].

Given a HMM model Θ and targeted tweet $\mathbf{w}$, we denote the marginalized venue probability as $p(v|\mathbf{w}, N_T(\mathbf{w}; u), \Theta)$. Figure 4 illustrates an example that computes $p(v = A|\mathbf{w}_2, N_T(\mathbf{w}_2; u), \Theta)$, the probability that $\mathbf{w}_2$ is posted from venue A. For simplicity, the tweet sequence contains only two tweets $\mathbf{w}_1$ and $\mathbf{w}_2$, whereby each tweet can be posted from one of two possible venues, A

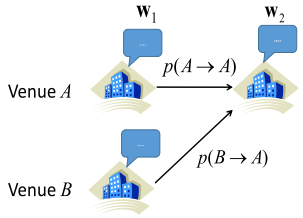


Fig. 4. HMM example: Computing the probability that $\mathbf{w}_2$ is posted from venue A.

or B. The required probability is computed as $p(v = A|\mathbf{w}_2, N_T(\mathbf{w}_2; u), \Theta) \propto p(\mathbf{w}_2|A)[p(\mathbf{w}_1|A)p(A \to A) + p(\mathbf{w}_1|B)p(B \to A)]$, whereby $p(B \to A)$ denotes the transition probability from B to A by the user. Thus one marginalizes over possible posting venues for $\mathbf{w}_2$'s temporal neighbors. In this example, the user may post $\mathbf{w}_1$ from A and then remains at A to post $\mathbf{w}_2$, or posts $\mathbf{w}_1$ at B and then moves to A to post $\mathbf{w}_2$. In our experiments, we estimate the transition probabilities between venues from observed transitions in the training set. Since it is impossible for users to transit between venues that are too far apart given a short time interval, the transition matrix is sparse. This facilitates the computation of marginal probabilities.

We can use $p(v|\mathbf{w}, N_T(\mathbf{w}; u), \Theta)$ directly as a baseline to rank venues. However we conjecture that query expansion contributes orthogonal information that should improve geolocation performance. Thus we stack our "Max"-based model over the HMM-based approach to exploit all information facets. Specifically, we compute:

$$p(v|\mathbf{w}, \{\mathbf{w}'\}_u) \propto p(v|\mathbf{w}, N_T(\mathbf{w}; u), \Theta) \prod_{\substack{\{w:\delta_S(w,\,\mathbf{w};\,u)>0|\\ \alpha(w,\,\mathbf{w};\,u)>0\}}} p(w|v)^{\max(\delta_S(w,\mathbf{w};u),\,\alpha(w,\mathbf{w};u))}. \tag{10}$$

Equation (10) is of similar form to Equation (7), except that given target tweet $\mathbf{w}$, we bias its venue probabilities with $p(v|\mathbf{w}, N_T(\mathbf{w}; u), \Theta)$ instead of the global distribution $p(v)$.

4.5.1 Limiting Cases. Given tweet $\mathbf{w}$ from user u targeted for geolocation, different scenarios can arise. For example, $\mathbf{w}$ may or may not have temporal neighbors or share common words with u's non-target tweets $\{\mathbf{w}'\}_u$ (see Section 4.3). Interestingly, HMM-Max is a highly general model that can be used for geolocation in various scenarios. It reduces to different models for the following scenarios:

- $\mathbf{w}$ has temporal neighbors and common words with tweets from $\{\mathbf{w}'\}_u$:
  The presence of temporal neighbors enables construction of the Markov chain and temporal query expansion. The presence of common words enables visitation query expansion. Hence all aspects of the HMM-Max model applies.
- $\mathbf{w}$ has temporal neighbors, but no common words with tweets from $\{\mathbf{w}'\}_u$:
  Markov chain construction and temporal query expansion apply, but visitation query expansion does not apply. HMM-Max reduces to a HMM model stacked with a naive Bayes model weighted with temporal query expansion, i.e., Equation (10) reduces to

$$p(v|\mathbf{w}, \{\mathbf{w}'\}_u) \propto p(v|\mathbf{w}, N_T(\mathbf{w}; u), \Theta) \prod_{\{w:\delta_S(w,\,\mathbf{w};\,u)>0\}} p(w|v)^{\delta_S(w,\mathbf{w};u)} \tag{11}$$

- $\mathbf{w}$ has no temporal neighbors, but has common words with tweets from $\{\mathbf{w}'\}_u$:
  Markov chain construction and temporal query expansion are no longer applicable. In this scenario, HMM-Max is equivalent to a naive Bayes model weighted only with visitation query expansion. Equation (10) reduces to Equation (6).

- $\mathbf{w}$ has no temporal neighbors and no common words with tweets from $\{\mathbf{w}'\}_u$:
  Both Markov chain construction and query expansion are not applicable. HMM-Max reduces to a naive Bayes model as characterized by Equation (2). Hence in the worst case scenario of highly sparse information, performance will be comparable to applying the models from References [26, 27].

4.6 Computational Complexity

We first examine the computational complexity of query expansion. Given a targeted tweet $\mathbf{w}$ from user u, the complexity of temporal query expansion depends on the length of $\mathbf{w}$'s parent sequence and can be written as $O(|N_T(\mathbf{w};u)|)$. For visitation query expansion, the complexity depends on the number of other tweets from u that contains words from $\mathbf{w}$, denoted as $D(\mathbf{w};u)$. These other tweets can be retrieved efficiently in $O(|D(\mathbf{w};u)|)$ time using an inverted index [57], which indexes tweets based on their constituent words. We then only need to compute weights for candidate words in the retrieved tweets. This means the number of words for consideration is usually much smaller than the entire word vocabulary. Depending on the words in $\mathbf{w}$, visitation query expansion can involve a few or a substantial number of tweets from u's non-target tweets. In the worst case, all non-target tweets are involved. In contrast, temporal query expansion usually involves fewer tweets due to the time interval constraint. Hence typically $|D(\mathbf{w};u)| > |N_T(\mathbf{w};u)|$. In this case, the complexity in the "Max" fusion framework is dominated by visitation query expansion and can be written as $O(|D(\mathbf{w};u)|)$.

For incorporating sequential information, the main computation complexity lies in the forward-backward algorithm. To geolocate $\mathbf{w}$ from user u, the basic algorithm has a complexity of $O(|N_T(\mathbf{w};u)| \times V^2)$, whereby V is the number of venues. However in practice, one needs not compute transitions over all possible venue pairs. The transition matrix is highly sparse due to user mobility patterns and the physical constraint that within a short time interval, it is not possible to traverse between venue pairs that are too far apart. Thus when computing possible transitions from a given venue, one only needs to consider observed transitions in the training set with optional probability smoothing for venue pairs that are not too far apart. This reduces the complexity to $O(|N_T(\mathbf{w};u)| \times \gamma \times V^2)$ where $0 < \gamma < 1$ is the average fraction of venues that each venue can transit to. Thus complexity is dependent on the transitional characteristics of the dataset.

5 EXPERIMENTS

We explore fine-grained geolocation models that incorporate different query expansion approaches and fusion schemes. We also implement other baselines for comparison. For each dataset, we conduct 20 runs that differ by randomly partitioning tweets into three sets: training, tuning and testing. In each run for each dataset, we first obtain the pool of tweets with temporal neighbors. From such tweets, we randomly sample 5000 tweets, from which 40% is used as the tuning set and 60% is used as the test set. All other tweets, including those without temporal neighbors, are used as the training set.

We select posting venues with at least three training tweets as candidate venues. To simulate the scenario where the temporal neighbors of test tweets have no observed posting venues, we process the training tweets as follows: If a user has one or more tweets sampled for testing/tuning, then we hide the posting venues of all his tweets in the training set. Thus, the training set mixes tweets with unknown posting venues and other tweets whose posting venues are retained. In training, we estimate the word distributions $p(w|v)$ using the tweets with observed venues. The training set is also used as a source of candidate words for query expansion and to estimate the transition probabilities for HMM-based models.

We use the tuning set to tune model parameters to optimize Mean Reciprocal Rank (see Section (5.1)). For models utilizing temporal query expansion (e.g., Temporal, Max etc.), tuning is done for the scaling parameter S for the exponential kernel. We use a grid with logarithmic intervals: $\{0, 0.01, 0.1, 1.0\}$. For the linear combination scheme, the linear combination weight λ is jointly tuned as well using a uniform grid from 0.1 to 0.9 at intervals of 0.1.

For the test set, we discard test tweets that are not posted from venues within the training set. Tweets with only stop words and rare words (with frequency < 3) are also discarded. The number of test tweets and candidate venues after filtering are reported in the tables in Section 5.2.

We compare the following models:

- **KL**: This approach [30] derives scores for venues by transforming and combining Kullback-Leibler divergences between the language models of tweets and venues, with the probabilities that venues generate tweets at different times of the day.
- **PTE**: PTE [49] or Predictive Text Embedding is a state-of-the-art graph embedding method for heterogeneous graphs. By treating venues as labels, we can use PTE to learn continuous vector representation for words, tweets and venues. The graph consists of word nodes, tweet nodes and venue nodes, connected via the following edge types: word-word, tweet-word and venue-word. An edge is created between a pair of words that co-occur in tweet(s). An edge is formed between a tweet and each word in the tweet. Finally, an edge is defined between a venue and a word if the word appears in some tweet(s) associated with the venue. For each new test tweet, we compute its representation by averaging over the representations of its constituent words. We then compute the cosine similarities of the test tweet to venue representations for ranking venues. We use an embedding dimension of 200 and 400 million negative samples. This performs better than the default parameters in Reference [49].
- **KDE**: This method [22] integrates kernel density smoothing with unigram language models to geolocate tweets to grid cells. Given a cell c, one computes $p(c) \prod_{w \in \mathbf{w}} p(w|c)$ whereby $p(c)$ and $p(w|c)$ are smoothed using Gaussian kernels. To geolocate tweets to venues, we extend the method by computing $p(v|c)p(c) \prod_{w \in \mathbf{w}} p(w|c)$, where probability of venue v given cell c, $p(v|c)$ is estimated by counting tweets posted from venue v, over all tweets posted within cell c. We use a grid size of 500 m. We tune the kernel parameter on a grid with logarithmic intervals $\{0.01, 0.1, 1.0, 10.0\}$.
- **Nb**: The base model from Equation (2) with Laplace smoothed word probabilities.
- **Temporal**: Temporal query expansion as shown in Equation (4).
- **Visit**: Visitation query expansion as shown in Equation (6).
- **Max**: The max combination scheme that combines the temporal and visitation query expansion approaches. See Equation (7)
- **Linear**: Temporal and visitation query expansion combined via linear combination. See Equation (8)
- **Product**: Both query expansion approaches combined via product combination. See Equation (9)
- **HMM**: This is the approach from Reference [31] based on Hidden Markov Models. We adapt it for our work by modeling venues as the hidden states.
- **Max-HMM**: The test tweet is first query expanded using "Max," denote as $\tilde{\mathbf{w}}$. We treat $\tilde{\mathbf{w}}$ as an observed tweet within a sequence and compute its marginal venue probabilities $p(v|\tilde{\mathbf{w}}, N_T(\mathbf{w}; u), \Theta)$ where Θ is the fitted HMM model. We use the marginal venue probabilities to rank venues.

- **HMM-Max**: The HMM model is first applied to compute the marginal venue probabilities, followed by stacking of the "Max" model, as shown in Equation (10)

There are other fine-grained geolocation methods in the literature that are not considered here, largely due to additional assumptions about users and social media platforms [4, 8].

We use three parent time window settings: T=1 hour, 0.5 hour, and 0.25 hour to define temporal neighbors. To recap the purpose of T, if T=1 hour, then any training tweet posted by the user within 1 hour (e.g., 10 min) of his test tweet is defined as a temporal neighbor. While T can be set to any interval, using a short interval such as 5 min may generalize to too few test cases while using a long interval (e.g., days) leads to long Markov chains and increased computation cost. Also, a long duration is unnecessary for temporary query expansion due to the kernel parameter S acting as a time decay factor (see Equation (3)).

5.1 Metrics

As we are solving fine-grained geolocation as a ranking problem and geolocating to venues within a small geographical area (within a city), ranking metrics are more appropriate than geographical distance metrics. The latter is not able to differentiate between venues that are stacked on top of each other, e.g., in a high rise building, or adjacent venues with essentially the same location coordinates.

We use the standard ranking metric *Mean Reciprocal Rank* (MRR) for evaluation. MRR was previously also used in Reference [8] to evaluate fine-grained geolocation. This metric is appropriate, since each tweet is posted from only one venue, which is desired to be ranked high. Given a tweet $\mathbf{w}_i$, let the rank of its posting venue be $r(\mathbf{w}_i)$, where $r(\mathbf{w}_i) = 0$ for the top rank. Over M test tweets, MRR is defined as

$$\text{MRR} = \frac{1}{M} \sum_{i=1}^{M} \frac{1}{r(\mathbf{w}_i) + 1}, \tag{12}$$

which is simply averaging over the reciprocal ranks.

MRR adopts micro-averages that favor popular venues contributing a larger proportion of tweets. In practical applications e.g., geolocating a stream of tweets, this is a realistic and reasonable evaluation metric. However for further analysis, we introduce the *macro-averaged* version of MRR, known as VMRR, to remove the effects of popular venues. In fact, the relationship between MRR and VMRR is akin to that between the micro-F and macro-F measures.

We compute VMRR by grouping test tweets by posting venues, followed by averaging the MRRs of the groups of test tweets from different posting venues. Formally, over V distinct venues:

$$\text{VMRR} = \frac{1}{V} \sum_{i=1}^{V} \text{MRR}(v_i), \tag{13}$$

where $\text{MRR}(v_i)$ is MRR values for test tweets from venue v_i. Thus each venue contributes one value for summation in Equation (13), regardless of the number of test tweets it contributes. VMRR helps to ascertain if improvements in MRR is biased towards more popular venues or spread over venues of differing popularities. A robust model should perform well in both MRR and VMRR.

5.2 Results

Tables 4, 5, and 6 display the results for datasets SG-SHT, SG-TWT, and JKT-SHT, respectively. For each dataset and metric, we use the Wilcoxon signed rank test to assess statistical significance between models. The best results or group of results are boldfaced. Models are described as on par or comparable if the signed ranked test does not indicate statistically significant differences at

Table 4. SG-SHT Results Averaged over 20 Runs

Models	T = 1 hour		T = 0.5 hour		T = 0.25 hour	
	MRR	VMRR	MRR	VMRR	MRR	VMRR
KL	0.03057 (−56.19%)	0.02170 (2.94%)	0.02861 (−59.62%)	0.02027 (−3.93%)	0.02710 (−62.85%)	0.02078 (−12.47%)
PTE	0.03593 (−48.51%)	0.02754 (30.65%)	0.03402 (−51.99%)	0.02593 (22.89%)	0.03278 (−55.06%)	0.02660 (12.05%)
KDE	0.05684 (−18.54%)	0.02037 (−3.37%)	0.05567 (−21.44%)	0.01937 (−8.20%)	0.05568 (−23.66%)	0.02088 (−12.05%)
Nb	0.06978	0.02108	0.07086	0.02110	0.07294	0.02374
Temporal	0.07036 (0.83%)	0.02145 (1.76%)	0.07220 (1.89%)	0.02259 (7.06%)	0.07516 (3.04%)	0.02550 (7.41%)
Visit	0.07145 (2.39%)	0.02113 (0.24%)	0.07257 (2.41%)	0.02135 (1.18%)	0.07469 (2.40%)	0.02377 (0.13%)
Max	0.07114 (1.95%)	0.02152 (2.09%)	0.07314 (3.22%)	0.02230 (5.69%)	0.07555 (3.58%)	0.02524 (6.32%)
Linear	0.07108 (1.86%)	0.02123 (0.71%)	0.07326 (3.39%)	0.02202 (4.36%)	0.07552 (3.54%)	0.02474 (4.21%)
Product	0.07100 (1.75%)	0.02260 (7.21%)	0.07243 (2.22%)	0.02332 (10.52%)	0.07547 (3.47%)	0.02682 (12.97%)
HMM	0.07401 (6.06%)	0.02380 (12.90%)	0.07539 (6.39%)	0.02496 (18.29%)	0.07848 (7.60%)	0.02777 (16.98%)
Max-HMM	0.07420 (6.33%)	0.02362 (12.05%)	0.07564 (6.75%)	0.02484 (17.73%)	0.07858 (7.73%)	0.02776 (16.93%)
HMM-Max	**0.08122 (16.39%)**	**0.03053 (44.83%)**	**0.08110 (14.45%)**	**0.03074 (45.69%)**	**0.08494 (16.45%)**	**0.03453 (45.45%)**

Bracketed numbers are percentage improvement over Nb. On average for T = 0.25 hour, M = 1047.9, V = 11344.2 on average.

p-value of 0.05. Across Tables 4 to 6, HMM-Max is consistently the best or among the best models. For all models, VMRR is also consistently lower than MRR, as expected from the correction of venue popularity effects. In the following, we further elaborate the results.

Baselines. Tables 4 to 6 show that KL, KDE, and PTE perform substantially worse than Nb and other models across all datasets and metrics. KL's poor performance indicates that modeling each tweet with a smoothed language model is inadequate, probably due to the brevity in content. This affects the computation of divergence values between the language models of tweets and venues. KDE out-performs KL but is still inferior to Nb. Although KDE works well for coarse-grained geolocation [22], word distributions are learnt at a grid cell level and are sub-optimal for fine-grained geolocation. PTE performs poorly for MRR, but does well for VMRR, although it is still inferior to HMM-Max. Since more popular venues contribute more to MRR, the results indicate that PTE's learnt representations of such venues may need further improvement. It will be interesting in future work to explore how to vary the distributions of edge samples during PTE training to achieve this.

HMM [31] out-performs the Nb model for both MRR and VMRR for datasets SG-SHT (Table 4) and SG-TWT (Table 5). For JKT-SHT (Table 6), it is on par for MRR and performs better for VMRR. Thus sequential information exploited by HMM provides useful information, even when one omits

Table 5. SG-TWT Results Averaged over 20 Runs

Models	T = 1 hour		T = 0.5 hour		T = 0.25 hour	
	MRR	VMRR	MRR	VMRR	MRR	VMRR
KL	0.02837 (−63.14%)	0.01411 (−14.28%)	0.02947 (−62.30%)	0.01543 (−7.22%)	0.02881 (−62.78%)	0.01496 (−9.77%)
PTE	0.03149 (−59.08%)	0.01782 (8.26%)	0.03336 (−57.32%)	0.01921 (15.51%)	0.03189 (−58.80%)	0.01878 (13.27%)
KDE	0.05141 (−33.20%)	0.01607 (−2.37%)	0.05278 (−32.47%)	0.01703 (2.41%)	0.05181 (−33.07%)	0.01605 (−3.20%)
Nb	0.07696	0.01646	0.07816	0.01663	0.07741	0.01658
Temporal	**0.08399 (9.13%)**	0.01873 (13.79%)	0.08496 (8.70%)	0.01881 (13.11%)	**0.08390 (8.38%)**	0.01868 (12.67%)
Visit	0.07851 (2.01%)	0.01654 (0.49%)	0.07951 (1.73%)	0.01669 (0.36%)	0.07859 (1.52%)	0.01664 (0.36%)
Max	**0.08408 (9.52%)**	0.01845 (12.09%)	**0.08563 (9.56%)**	0.01880 (13.05%)	**0.08429 (8.89%)**	0.01859 (12.12%)
Linear	0.08383 (8.93%)	0.01819 (10.51%)	0.08487 (8.59%)	0.01827 (9.87%)	**0.08390 (8.38%)**	0.01819 (9.71%)
Product	0.07805 (1.42%)	0.01723 (4.68%)	0.07947 (1.68%)	0.01775 (6.74%)	0.07856 (1.49%)	0.01744 (5.19%)
HMM	**0.08429 (9.25%)**	0.01874 (13.85%)	0.08529 (9.12%)	0.01890 (13.65%)	**0.08411 (8.66%)**	0.01876 (13.15%)
Max-HMM	**0.08483 (10.23%)**	0.01926 (17.01%)	**0.08541 (9.28%)**	0.01963 (18.04%)	**0.08436 (8.98%)**	0.01886 (13.75%)
HMM-Max	**0.08486 (10.27%)**	**0.02020 (22.72%)**	**0.08604 (10.08%)**	**0.02102 (26.40%)**	**0.08446 (9.11%)**	**0.02030 (22.44%)**

On average per run, M = 1302.7, V = 1911.15 for T= 0.25 hour.

any query expansion. However as will be subsequently discussed, query expansion will provide further performance gains.

Query Expansion. The two query expansion approaches "Temporal" and "Visit" outperform or are on par with the base model "Nb" across all three datasets. For SG-SHT (Table 4), "Visit" achieves small, but statistically significant improvement over "Nb" for MRR for all T settings, while being on par for VMRR. "Temporal" improves slightly over "Nb," except for T=1 hour where the MRR gains are not significant. For SG-TWT (Table 5), query expansion also works well, with "Temporal" achieving much larger gains than "Visit" over the base model. This is consistent across metrics and T settings. This matches our earlier empirical results in Section 3.1, showing that consecutive tweets in SG-TWT are likely to be from the same posting venue (since each linked check-in links to 1.5 pure tweets on average). Finally, for JKT-SHT (Table 6), both query expansion approaches provide consistently improved or on-par performance over different T settings and metrics.

Fusion Approaches. Comparing the fusion approaches: "Max," "Linear," and "Product" over different datasets, one sees that "Max" performs consistently well over the different datasets and is the more robust fusion approach. For MRR on SG-SHT (Table 4), the performance of "Max" is statistically equivalent with "Linear" and "Product" for all T settings. For VMRR on SG-SHT, "Product" performs better than other fusion approaches, including the proposed "Max" approach.

Table 6. JKT-SHT Results Averaged over 20 Runs

Models	T = 1 hour		T = 0.5 hour		T = 0.25 hour	
	MRR	VMRR	MRR	VMRR	MRR	VMRR
KL	0.05735 (−53.23%)	0.02714 (−21.31%)	0.05028 (−52.98%)	0.02474 (−29.66%)	0.04760 (−51.09%)	0.02553 (−37.66%)
PTE	0.05778 (−52.88%)	0.03459 (0.29%)	0.05265 (−50.76%)	0.03614 (2.76%)	0.04270 (−56.13%)	0.03473 (−15.19%)
KDE	0.07906 (−35.53%)	0.02370 (−31.28%)	0.07133 (−33.29%)	0.02375 (−32.47%)	0.06394 (−34.31%)	0.02973 (−27.40%)
Nb	0.12263	0.03449	0.10693	0.03517	0.09733	0.04095
Temporal	0.12482 (1.79%)	0.03579 (3.77%)	0.10878 (1.73%)	0.03671 (4.38%)	0.09841 (1.11%)	0.04289 (4.74%)
Visit	0.12336 (0.60%)	0.03475 (0.75%)	0.10850 (1.47%)	0.03538 (0.60%)	0.09894 (1.65%)	0.04091 (-0.10%)
Max	0.12543 (2.28%)	0.03598 (4.32%)	0.10928 (2.20%)	0.03623 (3.01%)	**0.09984 (2.58%)**	0.04290 (4.76%)
Linear	0.12445 (1.48%)	0.03551 (2.96%)	0.10821 (1.20%)	0.03582 (1.85%)	**0.09839 (1.09%)**	0.04183 (2.15%)
Product	0.12373 (0.90%)	0.03524 (2.17%)	0.10710 (0.16%)	0.03540 (0.65%)	0.09711 (−0.23%)	0.04109 (0.34%)
HMM	0.12276 (0.11%)	0.03705 (7.42%)	0.10662 (−0.29%)	0.03747 (6.54%)	0.09734 (0.01%)	0.04459 (8.89%)
Max-HMM	0.12628 (2.98%)	0.03961 (14.85%)	**0.11100 (3.81%)**	0.04018 (14.25%)	**0.10087 (3.64%)**	**0.04733 (15.58%)**
HMM-Max	**0.12825 (4.58%)**	**0.04144 (20.15%)**	**0.11182 (4.57%)**	**0.04160 (18.28%)**	**0.10098 (3.75%)**	**0.04840 (18.19%)**

On average per run, M = 204.5, V = 3063.45 for T = 0.25 hour.

By definition, "Product" uses words in the intersection set from both visitation and query expansions, whereas "Max" uses the union set. We observe that in some cases, the union set can be large, with non-informative words that affects performance. For VMRR, cases with less popular venues acquire greater importance and there occurs enough such cases in SG-SHT for "Max" to underperform "Product." However for MRR in SG-SHT, "Max" is on par with "Product." The latter is also inconsistent and performs poorly on other datasets. For SG-TWT (Table 5), "Max" is the best fusion approach for most combination of metrics and T settings, while "Product" does poorly. "Linear" is slightly inferior to "Max," except for the case of MRR with T = 0.25 hour. However "Linear" achieves this at the expense of more tuning costs. For JKT-SHT (Table 6), "Max" again outperforms the other two fusion approaches in most cases.

Note that for each dataset, "Max" also achieves performance that is on par or slightly better than what is achieved alone by query expansion. It appears to be fairly unaffected by the weaker method. This is obvious from comparing "Max" vs "Temporal" and Visit." For example on SG-SHT (Table 4), "Visit" performs better than "Temporal" for MRR while for VMRR, "Temporal" performs better. With "Max" fusion, we obtain a more robust model, achieving MRR on par with "Visit" and VMRR on par with "Temporal." For another dataset SG-TWT, "Temporal" clearly outperforms "Visit" across all metrics and T settings. In this case, "Max" consistently achieves performance comparable with "Temporal". In fact, for MRR with T = 0.5 hour, "Max" also outperforms

"Temporal" with statistical significance. In short, although both query expansion approaches were useful, we achieve more consistent and robust gains after applying "Max" combination.

Stacking with HMM. While "Max" performs well, further performance gains can be achieved by stacking with HMM in an appropriate manner. Across all datasets and metrics, HMM-Max is consistently the best or among the best performing models. Intuitively, each target tweet's parent sequence has useful sequential information and HMM-Max is able to exploit this. Over the base model, performance gains for VMRR are especially impressive, ranging from around 18% for SG-TWT and JKT-SHT (Tables 5 and 6) to more than 40% for SG-SHT (Table 4). For MRR, gains range from 3+% for JKT-SHT to 10+% for SG-SHT and SG-TWT.

HMM-Max mostly outperforms Max-HMM. Although both models incorporate sequential information, the former turns out to be a better combination approach. Also, exploiting sequential information without query expansion (i.e., HMM) is not optimal. Although "HMM" mostly outperforms "Nb" (except for MRR in JKT-SHT), it is inferior to HMM-Max in most cases. For example, in SG-SHT (Table 4), HMM loses out by a large margin to HMM-Max over both metrics for both T settings. Such results show that query expansion exploits information that is orthogonal to sequential information, resulting in more effective geolocation.

5.3 Analysis by Venue Popularity

Given that HMM-Max is the best performing and most robust model, we examine how its accuracy varies with venue popularity. Our analysis also serves to improve our understanding of how geolocation accuracy may be affected by data characteristics. We quantify venue popularity by the venue probability $p(v)$, which we compute based on the global proportion of tweets posted from each venue. For each run, we divide test tweets into 3 equal-sized bins of low, medium and high popularity based on the probability of their posting venues. MRR is computed for each bin. We repeat this for 10 runs with the setting of $T = 1$ hour and compute the average bin-specific MRR. Figure 5 displays the results for SG-SHT, SG-TWT, and JKT-SHT. The graphs in each row arise from the same dataset and are arranged from left to right in increasing order of venue popularity. For comparison, we also illustrate the performance for HMM.

Figure 5 shows that it is easier to geolocate tweets posted from more popular venues than less popular ones. This trend is consistent across all datasets as well as across both HMM and HMM-Max models. For example, in Figure 5(c) for SG-SHT tweets from high popularity venues, HMM-Max achieves an average MRR of 0.22, much higher than 0.0039 in Figure 5(a) for low popularity venues. For JKT-SHT, the corresponding figures for HMM-Max are 0.35 in Figure 5(i) versus 0.0053 in Figure 5(g) for high and low popularity venues respectively. HMM follows the same trend. Intuitively, popular venues are associated with more tweets, which helps to build more complete venue profiles. They may also have distinct or dominant characteristics that attract users and are mentioned more in tweets, e.g., unique dishes in a popular restaurant. These factors will increase the geolocation accuracy for tweets posted from such venues.

Relative to HMM, the percentage improvement attained by HMM-Max is larger for less popular venues. In Figure 5(a) for low popularity venues, HMM-Max's average MRR of 3.88e-3 is a 92% improvement over HMM's value of 2.02e-3. For high popularity venues in Figure 5(c), the corresponding relative improvement is around 5.6%. For other datasets, the same trend persists although the magnitude of relative improvement differs. For example, for JKT-SHT, HMM-Max's relative improvement over HMM is less drastic than SG-SHT for low popularity venues, i.e., 15.9% in Figure 5(g). However, relative improvement is even smaller for high popularity venues at 3.17% in Figure 5(i). We also note that for SG-TWT, HMM-Max outperforms HMM for low and medium popularity venues (ee Figures 5(d) and (e)), but is on par for high popularity venues in Figure 5(f).

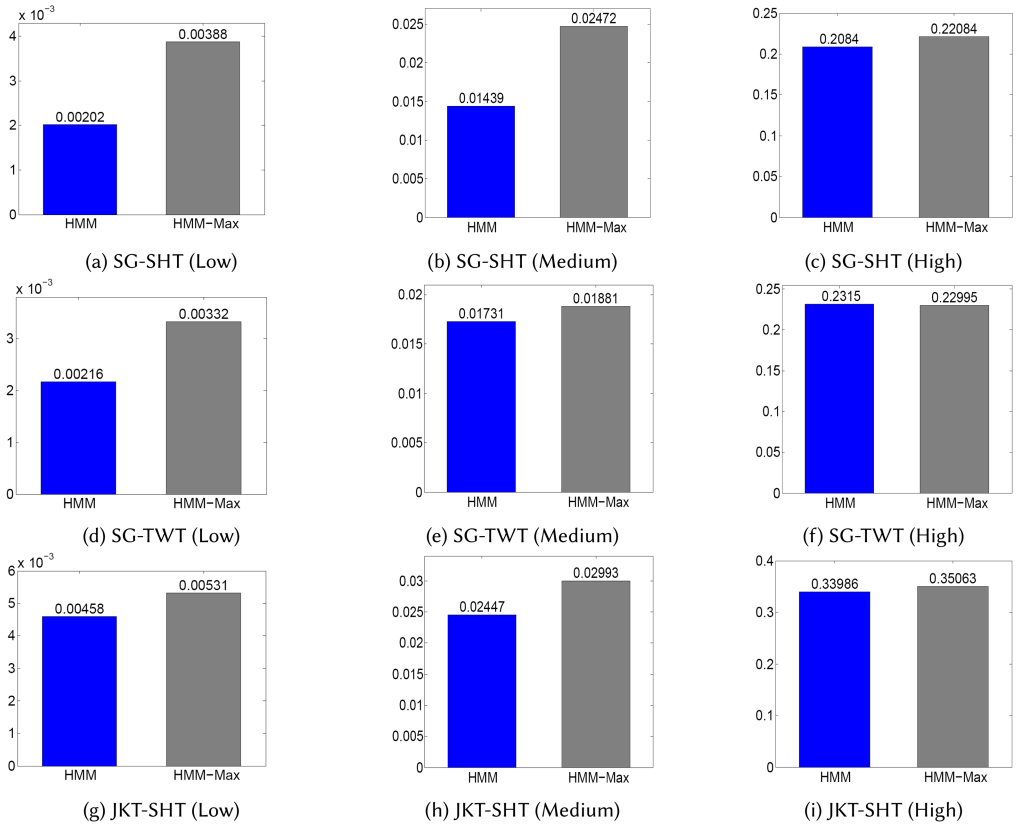


Fig. 5. Average MRR of HMM (blue) and HMM-Max (gray) for test tweets from venues of different popularities. Each row corresponds to a dataset.

We can conclude that the relative improvement provided by HMM-Max declines with increasing venue popularity. Such a trend may be because tweets from more popular venues are already geolocated fairly well and it is harder to achieve larger relative improvements. However there is still significant absolute improvement in MRR, i.e., a difference of 0.0124 in Figure 5(c). Since MRR is a top-heavy metric, small changes in the ranking positions near the top have large effects. Thus, HMM-Max still provides meaningful improvements in MRR when one considers absolute rank improvements of the posting venues. In short, it is reassuring that HMM-Max outperforms or is on par with HMM's performance across venues of different popularity.

5.4 Analysis by Distinct Venues per User

In this section, we study the relation between geolocation performance and the number of distinct venues that each user visit. The latter characteristic varies across users and will directly impact models that aim to exploit visitation behavior for geolocation. At one end, there are users who are focused on a small set of venues. At the other extreme, there are highly active users who post from a large number of venues, possibly due to novelty seeking behaviour [56] or to project an interesting image of themselves on social media [37].

Note that in our experiments, if a user has one or more tweets selected for testing, we mask the venues of all his tweets in the training set. This is in line with our discussed scenario in

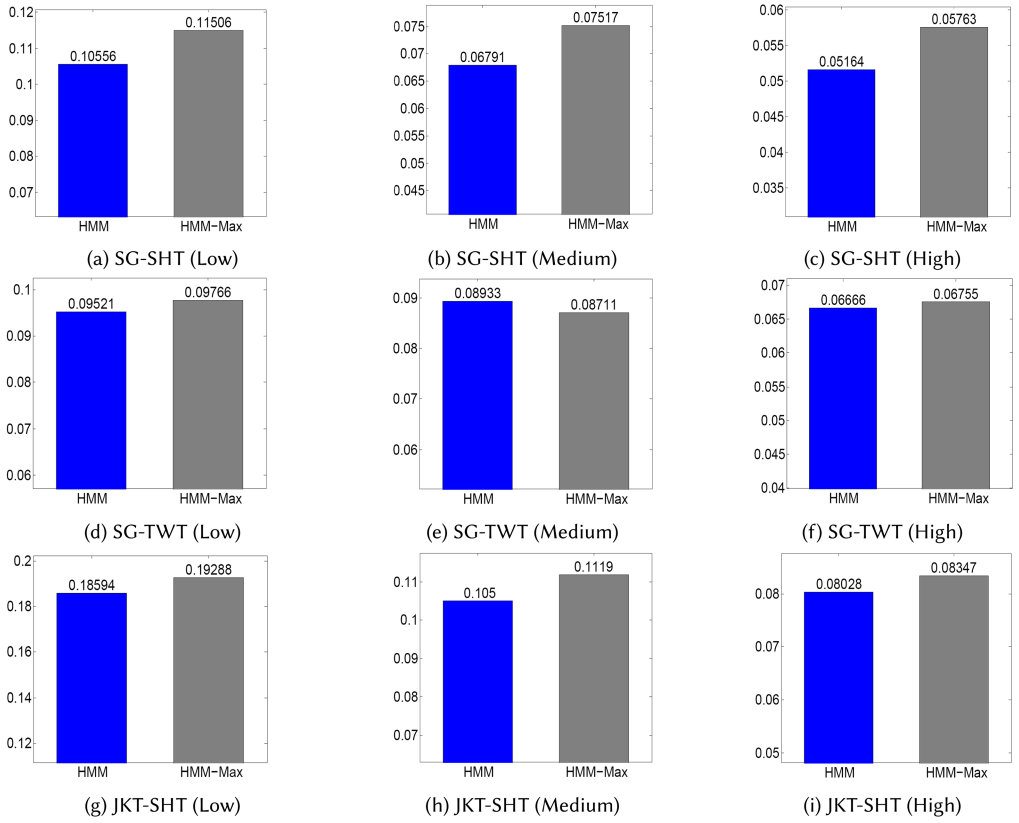


Fig. 6. Average MRR of HMM (blue) and HMM-Max (gray) for test tweets from users with different number of distinct venues in training tweets.

Section 1.1.1. Here, for the purpose of analysis, we unmask the venues of the training tweets for such users. For each test tweet, we compute the number of distinct venues that its user had visited over his tweets in the training set. Based on this statistic, we divide test tweets into three equal-sized bins, corresponding to the cases where the user has low, medium and high number of distinct venues. We then compute the MRR for each bin. We repeat this procedure for 10 runs with the setting of $T = 1$ hour and average the bin-specific MRR across the runs.

Figure 6 plots the average bin-specific MRR for all datasets. Across all 3 datasets, there is a common trend that geolocation performance drops as the number of distinct venues per user increases. In 6(a) corresponding to the "Low" bin for SG-SHT, HMM-Max achieves average MRR of 0.115. With increasing distinct venues per user, HMM-Max's MRR drops to 0.0752 in Figure 6(b) and finally to 0.0576 for the "High" bin in Figure 6(c). HMM follows the same trend. For both SG-SHT and JKT-SHT, HMM also performs consistently poorer than HMM-Max in each bin. For SG-TWT, HMM-Max outperforms HMM for the "Low" and "High" bin, while under-performing the latter on the "Medium" bin. Overall, both models can be regarded as on-par for SG-TWT (See Table 5, $T = 1$ hour, MRR metric).

Intuitively, if users are focused on a narrower set of venues, it may be easier to geolocate their tweets. Each user posts a finite number of tweets and spreading this over fewer venues will generally mean that information is less sparse. In contrast, if users are visiting a large number or highly diverse venues, then geolocation becomes more challenging. Our result shows that even in this

scenario, HMM-Max can better mitigate the effects and is more robust across datasets, compared to HMM. Interestingly, the better performance arises from simply overlaying a query expansion process on HMM to exploit both repeat visitation and staying behavior.

5.5 Case Study

As our proposed query expansion and Max combination approaches are more novel than the well studied HMM, we focus our case studies on the former portion. For ease of analysis, we compare non-sequential models, i.e., "Temporal," "Visit," and "Max." We first discuss positive cases that illustrate the usefulness of query expansion and Max combination. We then examine negative cases, which are grounds for future work.

5.5.1 Positive Cases. There are numerous examples where test tweets are geolocated more accurately from query expansions, as well as with Max combination. For ease of discussion, we use example cases where the test tweet is augmented with relatively few words, and contained in sequences of length two. Table 7 displays geolocation cases from SG-SHT with $T = 1$ hour. These are extracted from sample runs of the main experiment in Section 5. Each case consists of a pair of tweets: a test tweet (bolded) and its temporal neighbor. For each case, the words used for geolocation and associated weights are illustrated for different query expansion methods. Finally, the last row of each case displays the ranked position that each method attained for the test tweet's posting venue.

In Table 7, case A illustrates the usefulness of temporal neighbors and temporal query expansion. The test tweet A2 and its temporal neighbor A1 are posted from "Marina Bay Sands Hotel," known to have impressive city views. Hence the word "view" is indicative of the venue. With the "Temporal" method, a greater weight is placed on the word "view" due to its occurrence in both A1 and A2. This improves the ranking of the posting venue to position 2, i.e., $r(\mathbf{w} = \text{A2})=2$. For the "Visit" method, the word set is similar to that of "Temporal." This is because A2's words only co-occur with the words in A1. Consequently "Max" is also restricted to the same word set as "Temporal" and "Visit." However, the kernel parameters are now tuned over a tuning set that considers combined word sets for each tuning tweet. In this case, the tuned kernel learns a time decay of 0 within interval T (i.e., $S = 0$ in Equation (3)). This increases the weight of word "view" such that "Max" matches the performance of "Temporal."

Case B is another example that highlights the usefulness of temporal information. Tweets B1 and B2 are near duplicates of each other. Both mentioned a movie being screened at a theatre venue "Golden Village." By considering temporal neighbors, the informative word "conjuring" is given larger weights. Since this is indicative of the movie theatre, geolocation is improved. In contrast, visitation query expansion based on the method "Visit" is unable to augment the test tweet due to the lack of co-occurring words. By using "Max" fusion, one retains the geolocation improvement provided by temporal query expansion.

For case C, both the temporal neighbor C1 and other tweets from the user are useful for geolocating C2. C2 is posted from an airport departure hall. The base model "Nb" ranks its posting venue at position 121. By exploiting C2's predecessor C1, "Temporal" improves the ranking to 64. This is due to the word "flying," which is indicative of the airport. For the "Visit" method, some improvement is achieved as well by adding the word "boarding" to the test tweet. Finally the "Max" method uses the union of word sets considered by both "Visit" and "Temporal." This ranks the posting venue at position of 21, better than "Visit" and "Temporal."

Finally case D corresponds to the case where the temporal neighbor is not useful as a result of the tuned parameters not being optimal to this example. Fortunately other tweets from the user's history are useful. In D1, the user tweets about going to "ICA," which is an acronym for D2's posting venue "Immigration & Checkpoints Authority." However, tuning on a separate set of

Table 7. Sample Geolocation Cases/Tweet Sequences from SG-SHT

Case A	A1	(Marina Bay Sands Hotel, 14:22:29) "Zimmer bezogen... *City-View*"
	A2	**(Marina Bay Sands Hotel, 14:23:23)** **"*Und Garden*/Rennstrecken *View*..."**
	Temporal	(view, 1.58), (garden, 1.0), (und, 1.0), (city, 0.58)
	Visit	(view, 1.0), (garden, 1.0), (und, 1.0), (city, 0.33)
	Max	(view, 2.0), (garden, 1.0), (und, 1.0), (city, 1.0)
	$r(\mathbf{w} = \text{A2})$	Nb: 4, Temporal: 2, Visit: 6, Max: 2
Case B	B1	(Golden Village, 22:44:14) "Few minuted to The *Conjuring*."
	B2	**(Golden Village, 22:45:14)** **"Few *minutes* to The *Conjuring*."**
	Temporal	(conjuring, 1.55), (minutes, 1.0)
	Visit	(conjuring, 1.0), (minutes, 1.0)
	Max	(conjuring, 1.55), (minutes, 1.0)
	$r(\mathbf{w} = \text{B2})$	Nb: 14, Temporal: 10, Visit: 14, Max: 10
Case C	C1	(Changi International Airport, 15:55:26) "*Flying* out"
	C2	**(Terminal 1 Departure Hall, 15:56:39)** **"*Upgraded* again. Thank you *KLM*!"**
	Temporal	(upgraded, 1.0), (klm, 1.0), (flying, 0.48)
	Visit	(upgraded, 1.0), (klm, 1.0), (au, 0.5), (revoir, 0.5), (boarding, 0.35), (faith, 0.17), (alexandra, 0.17)
	Max	(upgraded, 1.0), (klm, 1.0), (au, 0.5), (revoir, 0.5), (flying, 0.48), (boarding, 0.35), (faith, 0.17), (alexandra, 0.17)
	$r(\mathbf{w} = \text{C2})$	Nb: 121, Temporal: 64, Visit: 78, Max: 21
Case D	D1	(Jurong East MRT Interchange, 14:34:36) "To *ICA*"
	D2	**(Immigration & Checkpoints Authority, 15:12:54)** **"*Change passport*!"**
	Temporal	(passport, 1.0), (change, 1.0), (ica, 1.05e-10)
	Visit	(passport, 1.0), (change, 1.0), (collecting, 0.5)
	Max	(passport, 1.0), (change, 1.0), (collecting, 0.5), (ica, 1.05e-10)
	$r(\mathbf{w} = \text{D2})$	Nb: 1 , Temporal: 1, Visit: 0, Max: 0

For ease of discussion, each case consists of a pair of tweets. The test tweet is bolded while its temporal neighbor is unbolded. In each tweet, modeled words are italicized (after omitting rare and stop-words). For each case, words and associated weights are sorted and illustrated for different query expansion methods. The last row of each case displays the ranked position that each method attained for the test tweet's posting venue.

tweets had resulted in a strong time decay for word weights. Given the substantial time difference of 38 minutes between D1 and D2, the weight of "ica" is overly small and has negligible effect on D2's geolocation. However by visitation query expansion, one is able to augment D2 by the word "collecting." This is a word strongly indicative of the posting venue that is a government building where users frequently tweet about collecting their immigration-related documents. Thus visitation query expansion improves geolocation by including an additional informative word. This improvement is also retained by Max combination.

Table 8. Sample Geolocation Cases from SG-SHT Where Current Query Expansion Approaches Do Not Improve Performances

Case E	E1	(ION Orchard, 18:38:14) "*tiredddddd*"
	E2	**(Cineleisure Orchard, 18:39:08)** **"*Running errand*"**
	Temporal	(running, 1.0), (errand, 1.0), (tiredddddd, 0.58)
	Visit	(running, 1.0), (errand, 1.0)
	Max	(running, 1.0), (errand, 1.0), (tiredddddd, 0.58)
	$r(\mathbf{w} = \text{E2})$	Nb: 6, Temporal: 7, Visit: 6, Max: 7
Case F	**F1**	**(Rooftop Infinity Edge Pool, 20:45:06)** **"*Finally* here to see the *Infinity Pool* and get to see the *awesome night view* of the *Singapore Skyline*"**
	F2	(Sky on 57, 20:47:38) "*Enjoying* the *nightview* of *Singapore Skyline* while *enjoying light snacks*"
	Temporal	(singapore, 1.22), (skyline, 1.22), (infinity, 1.0), (awesome, 1.0), (finally, 1.0), (night, 1.0), (pool, 1.0), (view, 1.0), (enjoying, 0.44), (light, 0.22), (snacks, 0.22)
	Visit	(singapore, 1.0), (skyline, 1.0), (infinity, 1.0), (awesome, 1.0), (finally, 1.0), (night, 1.0), (pool, 1.0), (view, 1.0), (enjoying, 0.44), (light, 0.22), (snacks, 0.22), (sweet, 0.13),(reached, 0.13), (flight, 0.13), (hours, 0.13)...
	Max	(singapore, 1.22), (skyline, 1.22), (infinity, 1.0), (awesome, 1.0), (finally, 1.0), (night, 1.0), (pool, 1.0), (view, 1.0), (enjoying, 0.44), (light, 0.22), (snacks, 0.22), (sweet, 0.13),(reached, 0.13), (flight, 0.13), (hours, 0.13)...
	$r(\mathbf{w} = \text{F1})$	Nb: 11, Temporal: 12, Visit: 14, Max: 16

The usefulness of temporal neighbors and other tweets from the user's history vary over cases A to D, resulting in temporal and visitation query expansions providing different extents of improvement over the base model "Nb." In all cases, Max fusion is able to handle the different scenarios and match the better performing method. This indicates that using Max fusion is more robust than either temporal or visitation query expansion alone.

5.5.2 Negative Cases. It is useful to also study cases where both temporal and visitation query expansions do not improve geolocation. For such cases, it is also difficult for Max combination to provide any improvements. Table 8 illustrates some examples.

Our experiment results and previous case studies have shown temporal neighbors to be generally useful. However there exists cases where they have no effect or worsen geolocation accuracy. For case E in Table 8, both tweets are from adjacent shopping malls. Incidentally, the temporal neighbor E1 provides no additional useful information to help geolocate test tweet E2. E1's content is not indicative of E2's posting venue. Using the former to augment the latter may then be akin to adding noise. Specifically with temporal query expansion, E2's posting venue is ranked at position 7, worse than the position of 6 obtained with the "Nb" base model. In this example, visitation query expansion does not provide additional informative words as well. Consequently, "Max" only manages to perform on par with "Temporal." On further analysis of case E, we observed the user to exhibit a cyclical visitation pattern, in the sense that he repeatedly visits E2's posting venue on evenings. If we augment E2 with words from the user's other tweets posted at around evenings, then more informative words such as "shopping" will be added to E2. This equates to

query expansion based on time of the day to model cyclical patterns. While the idea is intuitive, one caveat is that users may adhere to or deviate from their usual patterns, such that improving geolocation accuracy for this case may lead to worse accuracies in other cases. Hence further work can explore the robust fusion of cyclical models/approaches with the approaches in this article.

Case F in Table 8 covers a non-cyclical scenario. The user visits a rooftop swimming pool for the first time and posts tweet F1. He also posts F2 from an adjacent dining venue. Unfortunately, F2's content did not improve F1's geolocation. Due to the word "light" in F2, another candidate venue[4] popular for its night lightings were elevated in rank over F1's posting venue. Visitation query expansion was not useful as well, resulting in F1 being augmented with dozens of words. For brevity, we only list the top weighted words in Table 8. As can be seen, the added words included "reached," "flight" and so on, which are more indicative of the airport than F1's posting venue. Hence "Visit" performs worse than "Nb." Consequently, "Max," which combines the approach of "Temporal" and "Visit," also under-performs "Nb." When temporal neighbors are not useful, considering a user's visitation history may have some mitigating effect and still improve geolocation. However Case F pertains to users with significant deviations from their visitation history, e.g., tourists or users exploring new venues for the novelty factor [56]. Users may also evolve in their visitation behavior for more mundane reasons, e.g., change of workplace. For such cases, the current visitation query expansion approach is likely to be inadequate. In future work, it will be interesting to explore how novelty seeking and behavior evolution can be modeled and combined with the current approaches.

6 RELATED WORK

6.1 Coarse-grained Geolocation

We first review coarse-grained geolocation work that addresses a problem very different from fine-grained geolocation. The former is well studied and involves the tasks of (1) geolocating individual tweets or (2) geolocating users to home city/region/coordinates via their tweets.

6.1.1 Tweet Geolocation. The first task geolocates individual tweets to locations of different granularities such as cities, grid cells/regions or coordinates. Bo Han et al. [20] described geolocation to discrete locations (eg. cities) as akin to multiclass classification, while geolocation to coordinates is akin to multi-target regression.

For geolocation to cities (or coarse locations), [26] used naive Bayes to model the probability of words conditional on the cities. Each tweet is regarded as a bag of words and geolocated to cities with high probability of generating the tweet content. Highly related to this are grid based approaches [36, 46, 52] that also handles locations in a discrete manner. Wing and Baldridge [52] discretized space into a uniform grid of square cells, such that each cell can be modeled by a smoothed distribution of words. They then geolocate test tweets to the cells based either on the Kullback-Leibler (KL) divergence between word distributions of tweets and cells, or on tweet content probability under a naive Bayes model. In Reference [36], O'Hare and Murdock adopt uniform grids as well. They use the naive Bayes language model, along with spatial smoothing to geolocate Flickr photos, based on the accompanying photo tags. In a recent grid-based approach, [46] proposes to use adaptive grids constructed using a k-d tree, as a better alternative to uniform grids. Adaptive grids can vary the cell size to adapt to the training set size and geographic dispersion of the documents. Hence more densely populated areas will be fitted with more numerous and smaller cells.

[4] A park venue: Gardens by the Bay

For geolocating tweets to coordinates, topic models [1, 21] and spatial models [41] have been proposed. Ahmed [1] proposed the nested Chinese Restaurant Franchise Process. The process derives hierarchical topics such that higher level topics correspond to broad regions whereas lower level topics correspond to more fine-grained locations. In Reference [21], Hong et al. proposed a topic model under the Sparse Additive Generative Model framework [13]. The main idea is to model deviations caused by facets. For example, the probabilities of tweet words are made dependent on location coordinate such that they "deviate" from some background distribution. In both topic modeling works, topics are dependent on the posting coordinates, hence tweets are geolocated by inferring their topics. Besides topic models, spatial models are applicable as well. In Reference [41], Priedhorsky et al. modeled each word as a a Gaussian Mixture Model (GMM). To geolocate each tweet, the multiple GMMs corresponding to multiple words in a tweet are linearly combined, with higher weights place on more location-indicative words.

For each test tweet, the above works provide either a coordinate estimation [1, 21, 41] or a coarse discrete location, e.g., city/grid cell [26, 36, 46, 52]. Clearly, there is much difference from fine-grained geolocation that aims to infer the posting venue.

6.1.2 User Geolocation. This task infers the home city or region of users by exploiting the content over multiple tweets posted by each user. The approach in Reference [6] is to model the distribution of words over space and to use Location-Indicative (LI) words in the tweets to infer home locations. Such words are frequent at some central spatial point and their usage rapidly declines as one moves away from the central point. Chang et al. [5] defined LI words differently using Gaussian Mixture Models (GMM) instead. They identified LI words as those with GMM probability mass spatially focused on a small area. More approaches to identify LI words are compared in [19], where Han et al. grouped approaches into statistical methods such as hypothesis testing, information theory e.g., word entropy; and heuristics-based approaches such as TF-IDF . They found that geolocation performance of the various methods is sensitive to the number of top ranked words. Besides LI words, other information can be exploited for user geolocation. In Reference [25], Jurgens geolocated users based only on their social relationships, independent of any tweet content. Starting with a small number of initial locations from seed users, the approach spatially propagates location assignments through the social network. The assumption is that users are likely to be near their friends. With the same assumption, Rahimi et al. [44] propagates spatial labels over friendship networks constructed from user mentions in tweets. They also integrated priors from content-based geolocation into their network, showing that this joint exploitation of content and social network information out-performs content-only and network-only approaches.

Different from these mentioned works, we geolocate individual tweets, not users. Also, our geolocation granularity is to specific posting venues rather than cities or regions.

6.2 Fine-grained Tweet Geolocation

In contrast to coarse-grained geolocation, we conduct fine-grained tweet geolocation that links tweets to specific venues. Each tweet will be associated with one posting venue instead of a city, grid cell or a coordinate. The latter resolutions will associate each tweet with multiple venues. Furthermore, we consider fine-grained geolocation in the context of tweet sequences.

While fine-grained geolocation is relatively less explored than coarse-grained geolocation, many approaches from the latter task are applicable. The approach by Li et al. [30] is analogous to that of Reference [52] for coarse-grained geolocation. They geolocate tweets to the venue with the most similar word distribution, based on KL-divergence. Furthermore, venue probabilities based on posting time are linearly combined with the transformed KL-divergences to form venue scores. We implement this approach as a baseline. In Reference [27], a naive Bayes model for words is fitted

to each venue, analogously to Reference [26] for coarse-grained geolocation. However, only tweets with location indicative words are geolocated, whereas tweets without such words are discarded. In our work, we geolocate tweets even if they have no location indicative words. In Reference [23], Ikawa et al. learned the keywords that are highly associated with locations from geocoded content pushed by location apps to Twitter. They geolocate tweets with at least one keyword, based on cosine similarity matching to venues. Cao et al. [4] engineered features with content, location history and relationships, whereby features are specific to Foursquare, e.g., venue categories, user mayorships, and so on. Based on such features, they classify if a tweet is posted from a venue or not. In contrast, we seek to develop a more general approach that do not require platform specific features.

The works by References [24, 28, 29] geolocate/link venue mentions in tweets. In Reference [24], Ji et al. performed location recognition and linking simultaneously in a joint search space. They formulated fine-grained geolocation as a structured prediction problem and proposed a beam search based algorithm. In Reference [28], Li and Sun predict for each location mention in tweets, whether the user has visited, is currently at, or will soon visit the mentioned location. They designed a Conditional Random Field (CRF) based location tagger. The tagger takes in inputs such as lexical, grammatical, geographical and BILOU[5] schema features. Li et al. [29] exploit location mentions over multiple tweets per user to infer the top-k locations for each user. Their system is based on name matching against locations organized in hierarchical trees, whereby coarse grained locations, e.g., neighborhoods are parents of more fine-grained locations, e.g., streets. Once the top-k locations are identified for a specific user, his location mentions in specific tweets can be linked with greater accuracy.

While the discussed works [24, 28, 29] were shown to handle colloquial mentions, relying on mentions remains a bottleneck. This requires mention extraction, which remains a challenging task for tweets. Also tweets can be location indicative without mentions, e.g., a tweet "safely landed" is indicative of the airport even though it has no mentions. In fact, from a quick inspection of a sample of our data, we estimate that around 90% of the tweets do not contain any venue mentions. In our work, we geolocate tweets regardless of whether they contain any mentions or not.

Last, to our knowledge, there has been no prior work on fine-grained geolocation of tweets contained in sequences. The closest work is by Liu and Huang [31] who conducted coarse-grained geolocation. They geolocated tweets in sequences to cities, using Hidden Markov Models. This is easily adapted to fine-grained geolocation. In this article, we implement the method in Reference [31] as a baseline.

6.3 User Behavior

Repeat visitation and staying behavior can be related to some prior work.

6.3.1 Proximity between Consecutive Visits. Previous studies [34, 35, 54] have shown that consecutive venue visits over time tend to be close in space. In [34], Noulas et al. showed that between consecutive venue visits, the probability distribution of spatial distances has a declining trend and resembles an inverse power law. Shorter distances have higher probabilities than longer distances, although the latter still has small, non-negligible probabilities. A separate study in Reference [35] applied the complementary cumulative distribution function on inter-check-in distances and made similar observations. In Reference [54], Yuan et al. studied venue visits from location-based social networks to surface a similar characteristic, which they termed as spatial influence. Finally in Reference [45], Rhee et al. studied the mobility track logs of participants carrying GPS receivers. They

[5]BILOU schema identifies Beginning, Inside and Last word of a multi-word location name, and Unit-length location name.

found that human walk patterns exhibit statistically similar features as Levy walks [51], whereby people tend to visit nearby places and occasionally distant places.

6.3.2 Proximity of Visits to Home Location. In References [7, 12, 39, 40, 50], it was found that users are more likely to visit venues near their home locations. Some of these studies [7, 40] also highlighted that repeat visits are common. In Reference [40], Pontes et al. studied user activities in Foursquare that are indicative of mobility patterns, e.g., posts about visited venues. They found strong relationships between home locations and mobility patterns, whereby users made more visits at their residing cities and that they frequently revisit venues. Cho et al. [7] studied check-ins and cell phone logs. They showed that users are spatially focused and tend to visit venues around individual activity centers, e.g., home or workplace. They also found venues revisits to be a significant aspect of user behavior. Doan and Lim [12] conducted fine-grained spatial analysis of users. They obtained user home coordinates via extracting check-ins with indicative comments, e.g.,"Home sweet home!" On these users, they found that visitation probabilities to venues decrease with increasing distances from users' home locations. Other works [39, 50] simply assume that user visits are spatially concentrated near their home locations, when inferring the home location of users. Pontes et al. [39] used majority voting and mean statistics on geocoded visit data. Tasse et al. [50] recursively partition space into uniform grids and compute the mode to infer the home location.

6.3.3 Remarks. Based on proximity between consecutive visits, tweets posted within a short time of each other are likely to be posted from the same or nearby venues. The latter can also be linked to the proximity of visits to home locations. Via the transitivity property, if a user visits multiple venues near his home, then these venues will generally be near each other. If the visits are made within a short period of time, then the visited venues also constitute a sequence bounded within a short time interval.

6.4 Query Expansion

Query expansion is traditionally used for document retrieval. The initial query is expanded by adding potentially relevant words using term weighting schemes. To enhance query expansion, Reference [11] used genetic programming to learn weighting schemes. Reference [53] compared global and local query expansion techniques. Basically, global techniques expand queries based on corpora-wide word co-occurrence information/relationships while local techniques exploit the top rank documents given an unexpanded query. An example of a global technique is Reference [42], which uses a similarity thesaurus to add words that are most similar to the query concept. For local techniques, an early work in Reference [3] expands a query with the most frequent terms and phrases from the initial top ranked documents. More recently, Lv and Zhai [32] exploit term positions and proximity to assign more weights to words that are closer to query words. Intuitively such words are more likely to be related to the query topic. Besides retrieval tasks, query expansion has also been applied for entity linking. In Reference [18], Gottipati and Jiang expanded queries using both the local contexts within the query documents and global world knowledge obtained from the Web. Such expanded queries can be linked more accurately to the correct knowledge base entity. Interestingly, our tweet geolocation task can also be interpreted as implicit entity linking [38], whereby a tweet is linked (without mention extraction), to a knowledge base of venues.

With the increasing quantity of user generated content in social media, query expansion has also been applied for tasks related to social media content. Bandyopadhyay et al. [2] applied query expansion to tweets to retrieve relevant tweets given a user query. Given initial key words, they retrieved web pages and used their titles as a source of expansion words to retrieve more tweets. Fresno et al. [15] worked on a different task of retrieving more relevant keywords. They designed a

query expansion approach for tweets to discover keywords related to natural hazard events. They exploited user, tweet location and time information, e.g., considering candidate words from tweets close in space and time to an event-related tweet.

Instead of the above tasks, we have adapted query expansion for the different task of fine-grained tweet geolocation. Our query expansion approaches can be considered as a local technique in the sense that it is personalized to each user, i.e., words are added from other tweets from the same user.

7 CONCLUSION

We have explored geolocation of tweets that are close in time to other tweets posted by the same user. Such a scenario is fairly common, but to our knowledge, has not been studied in prior work. In particular we treat test tweets as akin to queries and propose temporal and visitation query expansions. These are conceptually simple, but novel expansion approaches motivated by observed mobility patterns of users. By "Max" fusion of both query expansion approaches and stacking with HMMs, we achieve an effective and robust model for geolocation. In future work, we will explore more sophisticated query expansion and word weighting functions. For example, a test tweet's temporal neighbors may contain noise words and words that are more indicative of locations. In this case, it may be more optimal to identify and assign larger weights to the location-indicative words. As mentioned in the negative case studies, it will also be interesting to explore how other behavioral aspects such as cyclical visits and novelty seeking can be modeled to improve geolocation.

7.1 Future Work

There is room for future work. First, the current model is based on probability distributions over discrete word representations. There is no notion of semantic similarities between different words. As tweets are very short text, venue-word and word-word co-occurrences are sparse. These impact the visitation query expansion component used in HMM-Max as well as the estimation of probability distributions. In future work, one can explore replacing the current word representations with continuous representations such as Word2Vec [33], and possibly to query-expand targeted tweets based on cosine similarities between tweets and word representations.

Second, we have thus far geolocated tweets to venues in the knowledge base. Clearly it is also possible for users to post from new venues or venues that are not in the knowledge base. One possible approach to handle this challenge is to modify the current models to incorporate a confidence measure. When the confidence level of a model in linking a targeted tweet is lower than some specified threshold, then the tweet can be flagged as unlinkable. This is also the current approach adopted by some explicit entity linking approaches whereby unlinkable mentions are flagged as out-of-value [47, 48].

Another more interesting direction is to combine coarse-grained and fine-grained geolocation. Basically even if a tweet is posted from some venue not in the knowledge base, it may be possible to geolocate it to some neighborhood or a coarser parent venue. For example, consider a newly opened restaurant in an existing shopping mall, whereby the latter is represented in the knowledge base. If the restaurant is not in the knowledge base, then we cannot geolocate tweets to it, but we can geolocate tweets to its parent mall. Hence coarse-grained geolocation serves to complement fine-grained geolocation where the latter is not possible, or is not confident about its geolocation outcome. One can also explore how to achieve a consensus in geolocation results from both fine-grained and coarse-grained geolocation in a fused or ensemble model. The idea is that the inferred posting venue or ranked venue list from fine-grained geolocation should be consistent with the inferred neighborhood/parent venue from coarse-grained geolocation. For example, if fine-grained

geolocation indicates a tweet to be posted from some venue that is not in the posting neighborhood inferred by coarse-grained geolocation, then at least one of the geolocation approach is providing inaccurate results. Thus any inconsistencies can be used to refine the model to achieve better geolocation.

REFERENCES

[1] Amr Ahmed, Liangjie Hong, and Alexander J. Smola. 2013. Hierarchical geographical modeling of user locations from social media posts. In *Proceedings of the 22nd International Conference on World Wide Web (WWW'13)*. 25–36.

[2] Ayan Bandyopadhyay, Mandar Mitra, and Prasenjit Majumder. 2011. Query expansion for microblog retrieval. In *Proceedings of the 20th Text Retrieval Conference (TREC'11)*.

[3] Chris Buckley, Amit Singhal, Mandar Mitra, and Gerard Salton. 1996. New retrieval approaches using SMART: TREC 4. In *Proceedings of the 4th Text Retrieval Conference (TREC'96)*. 25–48.

[4] Bokai Cao, Francine Chen, Dhiraj Joshi, and Philip S. Yu. 2015. Inferring crowd-sourced venues for tweets. In *2015 IEEE International Conference on Big Data (Big Data'15)*. 639–648.

[5] H.-W. Chang, D. Lee, M. Eltaher, and J. Lee. 2012. @Phillies tweeting from Philly? Predicting Twitter user locations with spatial word usage. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM'12)*. 111–118.

[6] Zhiyuan Cheng, James Caverlee, and Kyumin Lee. 2010. You are where you tweet: A content-based approach to geo-locating Twitter users. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM'10)*. ACM, 759–768.

[7] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. 2011. Friendship and mobility: User movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'11)*. 1082–1090.

[8] Wen-Haw Chong and Ee-Peng Lim. 2017. Exploiting contextual information for fine-grained tweet geolocation. In *Proceeding of the 11th International AAAI Conference on Web and Social Media (ICWSM'17)*.

[9] Wen-Haw Chong and Ee-Peng Lim. 2017. Tweet geolocation: Leveraging location, user and peer signals. In *Proceedings of the 26th ACM Conference on Information and Knowledge Management (CIKM'17)*. ACM, 1279–1288.

[10] Nello Cristianini and John Shawe-Taylor. 2000. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press.

[11] Ronan Cummins. 2008. *The Evolution and Analysis of Term-Weighting Schemes in Information Retrieval (Doctoral Dissertation)*. Ph.D. Dissertation. National University of Ireland, Galway.

[12] Thanh-Nam Doan and Ee-Peng Lim. 2016. Attractiveness versus competition: Towards an unified model for user visitation. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM'16)*. 2149–2154.

[13] Jacob Eisenstein, Amr Ahmed, and Eric Xing. 2011. Sparse additive generative models of text. In *Proceedings of the 28th International Conference on Machine Learning (ICML'11)*. 1041–1048.

[14] J. T. A. S. Ferreira, D. G. T. Denison, and D. J. Hand. 2001. *Weighted Naive Bayes Modelling for Data Mining*. Technical Report. Department of Mathematics, Imperial College.

[15] Víctor Fresno, Arkaitz Zubiaga, Heng Ji, and Raquel Martínez-Unanue. 2015. Exploiting geolocation, user and temporal information for natural hazards monitoring in Twitter. *Proces. Leng. Natur.* 54 (2015).

[16] Marc G. Genton. 2001. Classes of kernels for machine learning: A statistics perspective. *J. Mach. Learn. Res.* 2 (2001), 299–312.

[17] Mehmet Gönen and Ethem Alpaydin. 2011. Multiple kernel learning algorithms. *J. Mach. Learn. Res.* 12 (2011), 2211–2268.

[18] Swapna Gottipati and Jing Jiang. 2011. Linking entities to a knowledge base with query expansion. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'11)*. 804–813.

[19] Bo Han, Paul Cook, and Timothy Baldwin. 2014. Text-based Twitter user geolocation prediction. *J. Artif. Intell. Res.* 49, 1 (Jan. 2014), 451–500.

[20] Bo Han, Afshin Rahimi, Leon Derczynski, and Timothy Baldwin. 2016. Twitter geolocation prediction shared task of the 2016 workshop on noisy user-generated text. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT'16)*. 213–217.

[21] Liangjie Hong, Amr Ahmed, Siva Gurumurthy, Alexander J. Smola, and Kostas Tsioutsiouliklis. 2012. Discovering geographical topics in the Twitter stream. In *Proceedings of the 21st International Conference on World Wide Web (WWW'12)*. 769–778.

[22] Mans Hulden, Miikka Silfverberg, and Jerid Francom. 2015. Kernel density estimation for text-based geolocation. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15)*. 145–150.

[23] Yohei Ikawa, Miki Enoki, and Michiaki Tatsubori. 2012. Location inference using microblog messages. In *Proceedings of the 21st International Conference on World Wide Web (WWW'12 Companion)*. 687–690.

[24] Zongcheng Ji, Aixing Sun, Gao Cong, and Jialong Han. 2016. Joint recognition and linking of fine-grained locations from tweets. In *Proceedings of the 25th International Conference on World Wide Web (WWW'16)*. 1271–1281.

[25] David Jurgens. 2013. That's what friends are for. Inferring location in online social media platforms based on social relationships. In *Proceedings of the 7th International AAAI Conference on Weblogs and Social Media (ICWSM'13)*.

[26] Sheila Kinsella, Vanessa Murdock, and Neil O'Hare. 2011. "I'm eating a sandwich in Glasgow": Modeling locations with tweets. In *Proceedings of the 3rd International Workshop on Search and Mining User-generated Contents (SMUC'11)*. 61–68.

[27] Kisung Lee, Raghu K. Ganti, Mudhakar Srivatsa, and Ling Liu. 2014. When Twitter meets foursquare: Tweet location prediction using foursquare. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous'14)*. 198–207.

[28] Chenliang Li and Aixing Sun. 2014. Fine-grained location extraction from tweets with temporal awareness. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'14)*. 43–52.

[29] Guoliang Li, Jun Hu, Kian lee Tan, and Jianhua Feng. 2014. Effective Location Identification from Microblogs. In *30th IEEE International Conference on Data Engineering (ICDE'14)*. 880–891.

[30] Wen Li, Pavel Serdyukov, Arjen P. de Vries, Carsten Eickhoff, and M. Larson. 2011. The where in the tweet. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM'11)*. 2473–2476.

[31] Zhi Liu and Yan Huang. 2016. Where are you tweeting?: A context and user movement based approach. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 1949–1952.

[32] Yuanhua Lv and ChengXiang Zhai. 2010. Positional relevance model for pseudo-relevance feedback. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'10)*. 579–586.

[33] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint* 1301.3781 (2013).

[34] Anastasios Noulas, Salvatore Scellato, Neal Lathia, and Cecilia Mascolo. 2012. Mining user mobility features for next place prediction in location-based services. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining (ICDM'12)*. 1038–1043.

[35] Anastasios Noulas, Salvatore Scellato, Cecilia Mascolo, and Massimiliano Pontil. 2011. An empirical study of geographic user activity patterns in foursquare. In *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media (ICWSM'11)*.

[36] Neil O'Hare and Vanessa Murdock. 2013. *Inf. Retriev.* 16, 1 (Feb. 2013), 30–62.

[37] Sameer Patil, Gregory Norcie, Apu Kapadia, and Adam Lee. 2012. Check out where I am!: Location-sharing motivations, preferences, and practices. In *Extended Abstracts on Human Factors in Computing Systems (CHI'12)*. 1997–2002.

[38] Sujan Perera, Pablo N. Mendes, Adarsh Alex, Amit P. Sheth, and Krishnaprasad Thirunarayan. 2016. Implicit entity linking in tweets. In *Proceedings of the 13th Extended Semantic Web Conference (ESWC'16)*. 118–132.

[39] Tatiana Pontes, Gabriel Magno, Marisa Vasconcelos, Aditi Gupta, Jussara Almeida, Ponnurangam Kumaraguru, and Virgilio Almeida. 2012. Beware of what you share: Inferring home location in social networks. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining Workshops (ICDMW'12)*. 571–578.

[40] Tatiana Pontes, Marisa A. Vasconcelos, Jussara M. Almeida, Ponnurangam Kumaraguru, and Virgílio A. F. Almeida. 2012. We know where you live: Privacy characterization of foursquare behavior. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing (UbiComp'12)*. 898–905.

[41] Reid Priedhorsky, Aron Culotta, and Sara Y. Del Valle. 2014. Inferring the origin locations of tweets with quantitative confidence. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW'14)*. 1523–1536.

[42] Yonggang Qiu and Hans-Peter Frei. 1993. Concept based query expansion. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'93)*. 160–169.

[43] Lawrence R. Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77, 2 (Feb. 1989), 257–286.

[44] Afshin Rahimi, Trevor Cohn, and Timothy Baldwin. 2015. Twitter user geolocation using a unified text and network prediction model. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL'15)*. 630–636.

[45] Injong Rhee, Minsu Shin, Seongik Hong, Kyunghan Lee, Seong Joon Kim, and Song Chong. 2011. On the levy-walk nature of human mobility. *IEEE/ACM Trans. Network.* 19, 3 (2011), 630–643.

[46] Stephen Roller, Michael Speriosu, Sarat Rallapalli, Benjamin Wing, and Jason Baldridge. 2012. Supervised text-based geolocation using language models on an adaptive grid. In *Proceedings of the 2012 Joint Conference on Empirical*

Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL'12). 1500–1510.

[47] Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. 2012. LINDEN: Linking named entities with knowledge base via semantic knowledge. In *Proceedings of the 21st International Conference on World Wide Web (WWW'12)*. 449–458.

[48] Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. 2013. Linking named entities in tweets with knowledge base via user interest modeling. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'13)*. 68–76.

[49] Jian Tang, Meng Qu, and Qiaozhu Mei. 2015. PTE: Predictive text embedding through large-scale heterogeneous text networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'15)*. 1165–1174.

[50] Dan Tasse, Alex Sciuto, and Jason I. Hong. 2016. Our house, in the middle of our tweets. In *Proceedings of the 10th International AAAI Conference on Web and Social Media (ICWSM'16)*.

[51] Gandhi Viswanathan, Frederic Bartumeus, Sergey V. Buldyrev, Jordi Catalan, U. L. Fulco, Shlomo Havlin, M. G. E. da Luz, Marcelo Leite Lyra, E. P. Raposo, and H. Eugene Stanley. 2002. Levy flight random searches in biological phenomena. *Physica A* 314 (2002), 208–213.

[52] Benjamin Wing and Jason Baldridge. 2011. Simple supervised document geolocation with geodesic grids. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL'11)*. 955–964.

[53] Jinxi Xu and W. Bruce Croft. 1996. Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96)*. 4–11.

[54] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat-Thalmann. 2013. Time-aware point-of-interest recommendation. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'13)*. 363–372.

[55] Nayyar A. Zaidi, Jesús Cerquides, Mark James Carman, and Geoffrey I. Webb. 2013. Alleviating naive Bayes attribute independence assumption by attribute weighting. *J. Mach. Learn. Res.* 14, 1 (Jan. 2013), 1947–1988.

[56] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, and Xing Xie. 2014. Mining novelty-seeking trait across heterogeneous domains. In *Proceedings of the 23rd International Conference on World wide web (WWW'14)*. 373–384.

[57] Justin Zobel, Alistair Moffat, and Kotagiri Ramamohanarao. 1998. Inverted files versus signature files for text indexing. *ACM Trans. Database Syst.* 23, 4 (1998), 453–490.