7-2018

Context recovery in location-based social networks

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

CONTEXT RECOVERY IN LOCATION-BASED SOCIAL NETWORKS

CHONG WEN HAW

SINGAPORE MANAGEMENT UNIVERSITY

2018

Context Recovery in Location-based Social Networks

by

Chong Wen Haw

Submitted to School of Information Systems in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in Information Systems

Dissertation Committee:

Lim Ee Peng (Supervisor/Chair)
Professor of Information Systems
Singapore Management University

Jiang Jing
Associate Professor of Information Systems
Singapore Management University

Steven Hoi
Associate Professor of Information Systems
Singapore Management University

Teow Loo Nin
Distinguished Member of Technical Staff
DSO National Laboratories

Singapore Management University
2018

Context Recovery in Location-based Social Networks

by

Chong Wen Haw

Abstract

This dissertation addresses context recovery in Location-Based Social Networks (LBSN), which are platforms where users post content from various locations. With this general LBSN definition, many existing social media platforms that support user-generated location relevant content using mobile devices could also qualify as LBSNs. Context recovery for such user posts refers to recovering the venue and the semantic contexts of these user posts. Such information is useful for user profiling and to support various applications such as venue recommendation and location-based advertising.

For venue context recovery, we focus on Twitter where the venue information is often missing. We frame the problem as fine-grained geolocation whereby we geolocate tweets to their specific posting venues such as a restaurant, a shop etc. There are three tracks of work which cover different geolocation scenarios. In the first track, we geolocate tweets for users with location history in the form of geocoded tweets. Our model exploits a key empirical finding that users are more likely to visit venues near where they have visited in the past. In our second track, we geolocate tweets posted by users with no location history, who represent a larger category of users. For these users, we exploit the following observations: (1) users tend to revisit same or similar venues and (2) users with more similar content history are also more similar in their visitation behavior. In our last geolocation track, we geolocate tweets contained in sequences posted by the same user within a short time interval. We exploit the empirical observation that given a short time interval, users tend to post from the same or nearby venues. All these observations have been validated using real world data.

For semantic context recovery, we focus on the entity linking problem, which seeks to recover the entity mentioned or implied in a short content post. We explore two tracks of work. In the first track, we conduct explicit entity linking to link mentions of named entities in tweets to the referent entities in a knowledge base. We show that by exploring spatial and temporal information, we are able to improve the linking performance. In the second track, we conduct implicit entity linking on the specific task of identifying local cuisines in food-related posts. We link such posts directly to food entities in a knowledge base without the need of mention extraction. Empirically, we show that food venues are focused around a limited number of food entities each. By exploiting this entity-focused characteristic, our proposed model outperforms the state-of-the-art baselines.

Contents

List of Figures

List of Tables

Acknowledgements

This dissertation will not have been possible without the support of many people. Firstly I am grateful to DSO for awarding me the scholarship to embark on my studies. I will like to thank my immediate supervisors in DSO, Dr Teow Loo Nin and Dr Ng Gee Wah for their guidance and mentorship. In particular, I was able to acquire much expertise and knowledge in the area of data analytics under Loo Nin's mentorship over the years. I am thankful for this. I will also like to thank the senior management in DSO: Dr How Khee Yin and the ex-CEO Mr Quek Gim Pew for supporting my scholarship application. Secondly, I will like to thank the funding agency for their research funding support, namely the National Research Foundation, Prime Minister's Office, Singapore under its International Research Centres in Singapore Funding Initiative.

While the PhD journey is a tough one, the right thesis advisor can help to smoothen the ride. I regard myself fortunate in this aspect. I am especially thankful to my thesis advisor Prof Lim Ee-Peng for his patient guidance and mentorship. He has provided me much insights and research direction, as well as mental support and encouragement. I am also thankful to my thesis committee members Prof Jiang Jing and Prof Steven Hoi for their research input to improve this dissertation.

During my studies, I had the opportunity to conduct research in Carnegie Mellon University (CMU) for two semesters. I will like to thank Prof William Cohen for hosting and supervising me during my CMU stint, Kathryn Mazaitis for her technical support, Prof Jason Hong and Dan Tasse for providing me access to their user mobility data for my research. I am also grateful to Barbara Diecks for her adminis-

trative support and going the extra mile to make exchange students like myself feel extremely at home in Pittsburgh.

I will also like to thank fellow colleagues and friends in SMU for their help in administrative and technical matters, namely Ong Chew Hong, Seow Pei Huan, Fong Soon Keat, Phoebe Yeo, Desmond Yap, Janice Ng, Jamie Chia and Philips Prasetyo. I am also thankful to Dai Bing Tian for his research input when I was still a relatively fresh PhD student.

Lastly, I will like to thank my parents and wife for their support. I am grateful to my wife for accompanying me through the ups and downs of a PhD journey.

Dedicated to my Wife and Parents

Publications

Publications based on the dissertation:

1. Wen-Haw Chong and Ee-Peng Lim, *Implicit Linking of Food Entities in Social Media*, ECML-PKDD 2018.
2. Wen-Haw Chong and Ee-Peng Lim, *Exploiting User and Venue Characteristics for Fine-grained Tweet Geolocation*, ACM TOIS, 2018.
3. Wen-Haw Chong and Ee-Peng Lim, *Exploiting Contextual Information for Fine-Grained Tweet Geolocation*, ICWSM 2017.
4. Wen-Haw Chong and Ee-Peng Lim, *Tweet Geolocation: Leveraging Location, User and Peer Signals*, CIKM 2017.
5. Wen-Haw Chong, Ee-Peng Lim and William W. Cohen, *Collective Entity Linking in Tweets Over Space and Time*, ECIR 2017.

Manuscript based on the dissertation and under review:

1. Wen-Haw Chong and Ee-Peng Lim, *Geolocation of Tweets in Temporal Proximity*, submitted to ACM TOIS.

Other publications not included in dissertation.

1. Wen-Haw Chong, Bing Tian Dai and Ee-Peng Lim, *Not All Trips are Equal: Analyzing Foursquare Check-ins of Trips and City Visitors*, COSN 2015.
2. Wen-Haw Chong, Bing Tian Dai and Ee-Peng Lim, *Did You Expect Your Users to Say This?: Distilling Unexpected Micro-reviews for Venue Owners*, HT 2015.

3. Wen-Haw Chong, Bing Tian Dai and Ee-Peng Lim, *Prediction of Venues in Foursquare Using Flipped Topic Models*, ECIR 2015.

Chapter 1

Introduction

1.1 Motivation and Problem

The prevalence and growing popularity of social media in recent years have led to growing research interest in the exploitation of related data. Besides providing users with a means to share content, many platforms have included features that allow users to share their locations. This provides a linkage between content and mobility patterns. For example, Twitter users have the option of geocoding their tweets with location coordinates. In Foursquare, a popular location app, users can 'check-in' to venues while posting their comments (referred to in Foursquare as shouts). Such location related social networking platforms are referred to as Location-Based Social Networks [62, 53, 10, 88] or LBSN in short.

Basic questions arise when one considers LBSN usage. Where is the user posting from? What is the user posting about? Such questions lead to the problem of recovering the venue and semantic context. Next, we discuss each problem in detail.

Venue Context Recovery. Compared to Foursquare, Twitter is a much more flexible and coarse-grained LBSN platform where users can choose to geocode their tweets or not. For geocoded tweets not pushed from any location apps, they are associated with only location coordinates. These indicate the user's approximate location and not the specific venue that he is at. In densely populated cities where

multiple venues may share the same or similar coordinates, location coordinates do not uniquely identify the venues. Furthermore many users do not geocode many of their tweets in the first place. Earlier studies [1, 36] have indicated that only 1 to 2% of tweets are geocoded.

The discussed Twitter characteristics motivate the problem of *fine-grained tweet geolocation*. This problem [47, 44, 6] is relatively less well explored than coarse-grained tweet geolocation [1, 36, 23, 70, 84, 76, 63] which geolocates tweets to regions/cities or some location coordinates. Essentially we seek to identify the specific venue from which a tweet is posted, e.g. a restaurant, an office etc. In doing so, we recover the venue context of a tweet. This supports applications such as location based advertising and promotions, venue recommendation and user profiling. For example, there is a difference in venue context between a user who is dining at a restaurant and another user who is visiting an adjacent shop. This is even though the location coordinates of both users do not differ much. Business owners may want to target one or the other with different marketing strategies.

To understand the problem of fine-grained geolocation, it is also useful to view it as analogous to document retrieval. One can regard a tweet as a query and candidate venues as documents. One can then rank the candidate venues for the targeted tweet, which is 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 cast as document retrieval, there are differences between documents and venues. Importantly, venues are ordered in the spatial sense while documents are not. Such spatial ordering results in user and venue characteristics which can be exploited for better geolocation. For example, users may tend to visit venues near where they conduct their main activities, e.g. near workplace.

Semantic Context Recovery. Besides the venue context, it is also useful to understand what users talk about in their content such as tweets or Foursquare shouts.

Thus it is desired to recover the semantic context. Formalizing this, we have the *entity linking* problem whereby one seeks to associate the right semantic concepts to the content. Under the general entity linking problem, we explore two task variants, namely Explicit Entity Linking (EL) and Implicit Entity Linking (IEL).

For Explicit Entity Linking, we link the mentions of named entities in tweets to the correct referent entity in some knowledge base. This task requires mention extraction to be first conducted. While explicit entity linking has been explored over many years for longer documents [57, 20, 74, 78, 77], entity linking in tweets is a relatively new research area [56, 51, 79]. The latter is highly challenging due to the extremely short nature of such posts. To mitigate this challenge, we shall exploit certain properties of LBSN not found in traditional documents, e.g. considering content that are posted close in space and time.

For Implicit Entity Linking (IEL), we link venue-associated posts, e.g. Instagram captions, food reviews in a post-specific rather than mention-specific manner. Each post is linked in its entirety without any mention extraction. IEL is a relatively new concept proposed in [66] and has the advantage of being able to link posts which do not mention entities explicitly. In addition, IEL circumvents the challenge of mention extraction on grammatically noisy, colloquial content. However the challenge of content brevity remains. To link each post, we shall exploit information from other same-venue posts.

1.2 Research Objectives

Based on recovering the venue and semantic context in LBSN posts, this dissertation has respectively formulated two main research objectives: (1) fine-grained tweet geolocation and (2) entity linking. For fine-grained tweet geolocation, we also extensively analyse the mobility patterns of users in LBSN, i.e. user behavior, in order to motivate our geolocation models. Thus, user behavior analysis is a secondary objective that supports fine-grained tweet geolocation. Subsequent sections

detail each objective.

1.2.1 Fine-grained Tweet Geolocation

In fine-grained tweet geolocation, we associate non-geocoded tweets to the specific venues from which they are posted. We cast fine-grained geolocation as a ranking problem. Given a test tweet, we rank venues such that high ranking venues are more likely to be the posting venue. We explore fine-grained tweet geolocation for different user scenarios, whereby users may or may not have location history.

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. Firstly, such venues easily number in the hundreds of millions. Secondly, 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. These are venues created on location apps by users and associated with some minimum number of posts. With this approach, the number of candidate venues typically range in the thousands.

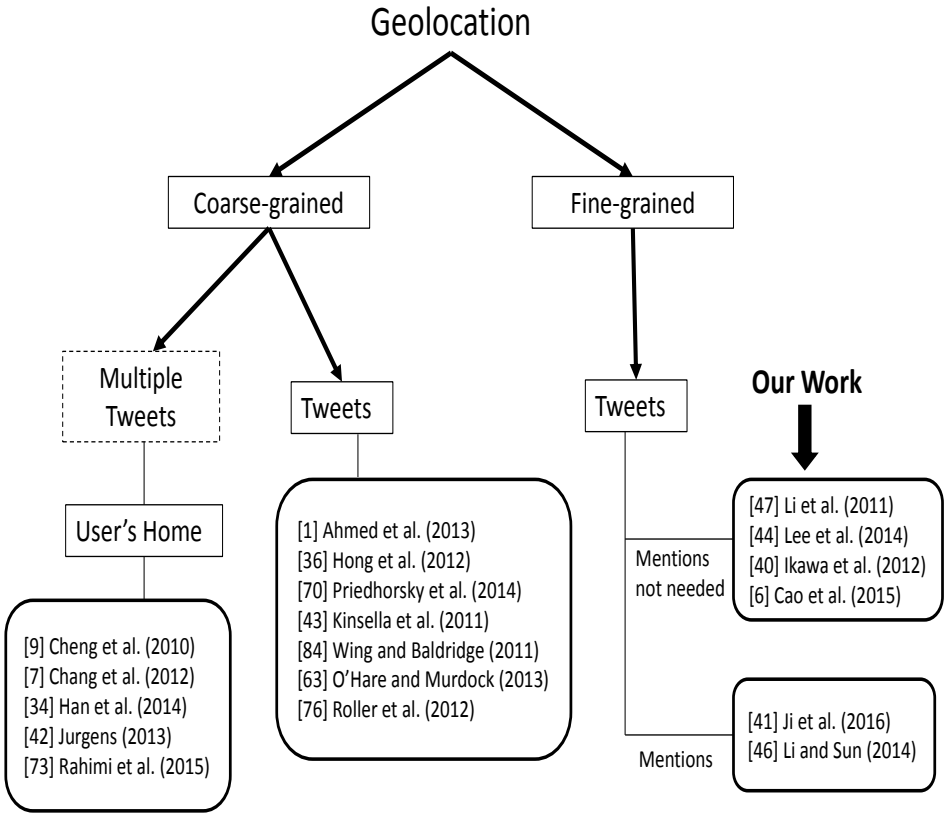
User Behavior Analysis. In this context, user behavior refers to the mobility patterns of users in LBSN. We conduct empirical analysis to provide further insights on user behavior, beyond what has been covered in prior work. We also aim to leverage on our empirical findings to design better geolocation models. Many works [69, 10, 22, 61, 62, 75, 88] have studied the mobility patterns of users, typically to support the inference of users' home locations, venue recommendation or next check-in prediction. Our task of fine-grained geolocation is different, but is intimately related to the mobility patterns of users as well. We design our analysis experiments with the geolocation task in mind, so as to surface characteristics to help in designing our geolocation models.

Overview. Figure 1.1 provides a high level overview of where we position the work in this dissertation relative to prior work. As shown in the figure, existing

geolocation work can be categorized into coarse-grained and fine-grained geolocation. Under coarse-grained geolocation (left branch in figure), the task can be on geolocating users to their home location by exploiting multiple tweets per user or geolocating individual tweets to their posting location. In the figure, the dashed box for the former task indicates that tweets are exploited, but their individual posting locations are not the main focus. Instead, the focus is on the users' home locations.

For both coarse-grained geolocation tasks, location granularities are at the region, neighborhood or coordinate level. For fine-grained geolocation (right branch), location granularities are specific venues. There is one task: geolocate individual tweets. Under this task, works can be divided into those that geolocate tweets without mention extraction and those that geolocate detected venue mentions in tweets (bottom right box). Our research focuses on the former task setting. Chapter 2 discusses the works in Figure 1.1 in greater detail.

Figure 1.1: A taxonomy of geolocation work

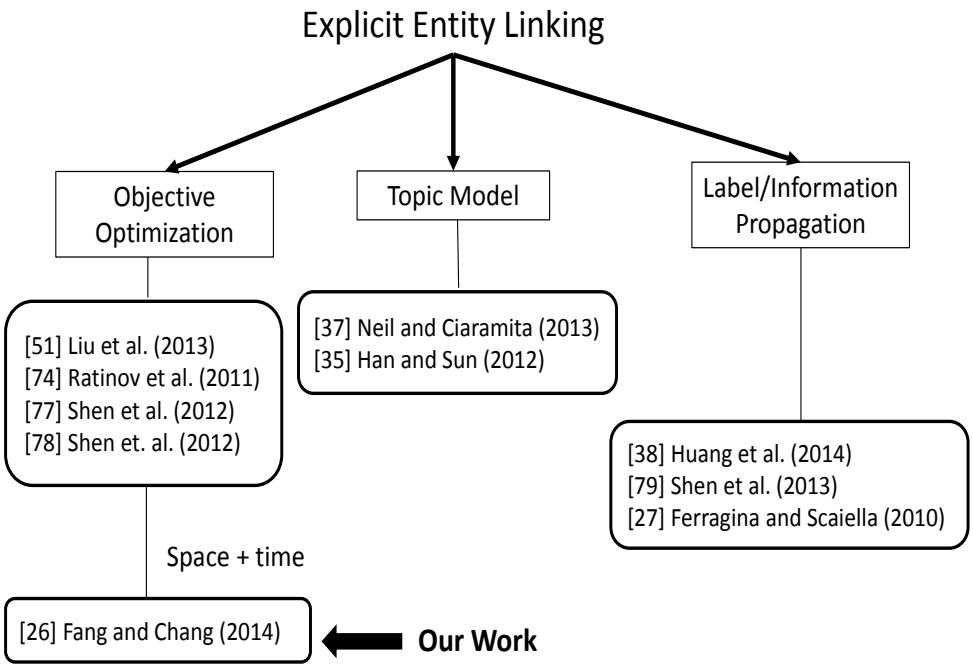


1.2.2 Entity Linking

For entity linking, we aim to make contributions in two different tasks: namely Explicit Entity Linking (EL) which links named entity mentions and Implicit Entity Linking (IEL) which links LBSN posts directly without mention extraction. Both tasks aim to recover the semantic context of what users are posting about. In our current work, we use Wikipedia as the knowledge base.

Explicit Entity Linking. Research on EL has been ongoing for many years, with much prior work in the literature. There is a wide variety of approaches using different frameworks and features. Figure 1.2 provides a highly simplified taxonomy of recent EL works. While these works can be categorized in many other ways, the presented taxonomy allows us to easily illustrate where our EL research fits in.

Figure 1.2: A simplified taxonomy of recent EL work



In Figure 1.2, the left branch lists recent approaches that construct and optimize objective functions. Various objective functions have been designed based on features such as word lexical forms, tweet content, entity popularity, strength of inter-entity relationships etc. However thus far, space and time has not been widely considered. One such work is [26] whereby Fang and Chang learn entity distribu-

tions for discrete location cells and time slots. Motivated by the characteristics of LBSN, we use space and time to construct and optimize a novel objective function for EL.

The middle branch lists recent EL approaches based on topic models. These are based on extensions of the Latent Dirichlet Allocation model [4]. Finally, the right branch lists approaches based on propagating entity labels or other information, e.g. inter-entity votes. Section 2.4.1 in Chapter 2 discusses the various EL work in more details.

Implicit Entity Linking. The IEL task is recently proposed by Perera et al. [66]. Compared to EL, there is very limited prior work [66, 56] and we omit a detailed categorization. As will be explained in Section 2.4.2, some existing EL works [26, 27] are also extensible to the IEL case. For IEL, our current research objective is the linkage of food-related posts to food entities. This is motivated by the popularity of dining activities and food-related posts in LBSN. However we envisage that our proposed IEL models can be generalized for linking other entity types.

1.3 Challenges

There are general challenges associated with mining information from LBSN posts. Firstly, the content is brief. For example, tweets are limited to 140 characters or 280 characters (from Nov 2017 onwards). Foursquare shouts and Instagram captions are also typically brief in content. Such characteristics may arise as users are posting from their mobile devices while on the move. The highly mobile nature and the less conducive input interface may mean there is less inclination to type long posts. In fact, Twitter indicates that the average tweet length has not increased much [1] even though they have doubled the character limit. Interestingly, most tweets are still shorter than 140 characters. In any case, such content brevity leads to a sparsity of information regardless of whether one is trying to do geolocation or entity linking.

[1] https://blog.twitter.com/official/en_us/topics/product/2017/tweetingmadeeasier.html

Compared to traditional documents (e.g. news articles) which are longer in length, it is more challenging for models to achieve good accuracies.

Secondly, the language in LBSN posts is highly colloquial, with improper grammar and sentence structure, spelling variations and social media lingos. Such characteristics lead to information sparsity as well since a word can appear in various surface forms, some of which have low usage frequency. This may impact the accuracies of language models. The improper grammar and sentence structure also impacts mention extraction of named entities, which is required for tasks such as explicit entity linking. Mentions may be extracted only partially or missed out, while some non-mentions may be mistaken for mentions.

Lastly, we highlight a challenge specific to fine-grained geolocation. Basically for any given tweet, the number of candidate posting venues is large. Even if one considers only venues with some social presence (associated with some minimum number of LBSN posts), there are easily thousands of candidate venues. Thus the problem is intrinsically a challenging one. There is also the challenge of obtaining ground-truth posting venues for an adequate number of tweets in order to conduct experiments. In Section 3.2, we shall discuss how we obtain such information.

1.4 Contributions

We summarize our major contributions.

1.4.1 Fine-grained Tweet Geolocation

For geolocation, we have two main user scenarios corresponding to users with and without location history. We cover these scenarios in three tracks of geolocation work. For each track, our contribution consists of empirical analysis of user behavior and proposed models.

In the first geolocation track (Chapter 3), we geolocate tweets from users with location history. Our contributions are:

- We show that users are *spatially focused* in being more likely to visit venues near each other and that venues near each other tend to be more similar in user-generated content, i.e. *spatial homophily*.

- Drawing on our empirical findings, we propose a spatially smoothed model (NB+S+T+U) that incorporates tweet content, posting time and user location history. Depending on the dataset and metric, our best performing model provides ranking accuracy improvement from 6% to 60% over the naive Bayes geolocation model.

In the second geolocation track (Chapter 4), we focus on users with no location history, but who have content history. For this track, we contribute the following:

- Empirically we show that users make repeat visits to venues. In addition, users with more similar tweet content history are more similar in their venue visitation history.

- We propose a model (LWQE-LW-CF) that exploits location, user and peer signals for better geolocation. The LWQE-LW-CF model incorporates location-indicative weighting to assign more weights to location-indicative words, query expansion of test tweets and collaborative filtering. In our experiments, LWQE-LW-CF performs 6% to 40% better than other baselines depending on the metric and dataset.

In the third geolocation track (Chapter 5), we again focus on users with content, but no location history. We explore the geolocation of tweets contained in sequences whereby tweets in each sequence are posted close in time by the same user.

- We verify empirically that users have the tendency to stay at the same or nearby venues given a short time period. We use this observation to design a temporal query expansion approach. This augments a test tweet with words from other tweets in the same sequence.

- We propose a novel model (HMM-Max) to geolocate tweets contained in sequences. Our model combines different query expansion approaches in a

novel fusion framework. Via stacking on a Hidden Markov Model, our model also captures sequential information. Performance improvements over baselines range from 4.5% to 45%.

1.4.2 Entity Linking

For entity linking, we have two tracks of work: explicit entity linking and implicit entity linking. For the first track of explicit entity linking (Chapter 6), we make the following contributions:

- We propose a new collective entity linking method to exploit event and geographical effects. We connect tweets close in space and time to form a tweet graph, and define a novel objective function over the graph. This mitigates the challenge of entity linking for overly brief content.
- We introduce a comparison-based evaluation approach to facilitate the comparison of unsupervised entity linking techniques when there is no labeled data. In addition, challenges such as noisy mention extraction and incomplete knowledge bases are mitigated.

For the second track of implicit entity linking (Chapter 7), we focus on linking posts from food venues to related food entities. Our contributions are:

- We analyzed food venues and highlight that such venues are focused around a limited set of food entities each. We termed this as the *entity-focused* characteristic.
- We design a novel implicit entity linking model EW-EWQE(v) which exploits the entity-focused characteristic in two ways. Firstly, the model augments each test post via query expansion to include words from other same-venue posts. Secondly, the model generates venue-based prior distribution over food entities in an initial entity linking stage. This prior is used to bias the entity scores for the next stage. We show EW-EWQE(v) to outperform state-of-the-art baselines.

1.5 Dissertation Structure

The rest of this dissertation is structured as follows. Chapter 2 first surveys work on the mobility behavior of LBSN users, followed by work related to tweet geolocation and entity linking. Chapters 3 to 5 cover our geolocation work, along with the associated empirical analysis. Chapter 3 describes our first geolocation track which geolocates tweets from users with location history. Chapter 4 describes our second geolocation track which focuses on users without location history. Our last geolocation track in Chapter 5 focuses on such users as well, but geolocates tweets contained in sequences. Chapter 6 discusses our work on explicit entity linking in tweets while Chapter 7 covers our implicit entity linking work. Finally we conclude with some suggestions for future work.

Chapter 2

Related Work

2.1 Mobility Behavior of LBSN Users

We discuss aspects of user behavior highlighted in prior work which motivate our own studies.

2.1.1 Mobility Patterns

We review related work that uses LBSNs to study mobility patterns. Typically the cited works carried out empirical analysis to support home location inference or next check-in prediction. While such tasks appear very different from tweet geolocation, there is an implicit linkage via mobility patterns. This is because tweets are posted by users as they commute or conduct their activities at some location.

Visitation Proximity to Home. In [69], Pontes et al. studied the relationship between home locations and mobility patterns on a coarse spatial scale. They analyzed user activities in Foursquare that are indicative of mobility patterns, e.g. tips (comments about visited venues) and venue mayorships (most frequent visitor) etc. They found that users tend to have such activities at their residing cities and that they frequently revisit venues. Cho et al. [10] utilizes check-ins and cell phone logs to show that users focus their visitations around individual activity centers such as home or workplace. This supports their formulation of a visitation model based on

Gaussian Mixtures. They also found that user revisit venues with substantial probability. Doan and Lim [22] conducted analysis at more fine-grained spatial resolution, within individual cities. They obtained the exact home coordinates of users by exploiting check-ins with indicative comments, e.g."Home sweet home!". On these users, they showed that the check-in probabilities decrease for venues with increasing distances from users' home locations. Furthermore, they assert a neighborhood competition effect whereby each user first chooses an area to check-in based on the area attractiveness and the distance from one's home. This is followed by choosing a winning venue based on it out-competing other nearby venues.

Other works [68, 83] implicitly exploit the idea that user visitation activities are spatially concentrated near their home locations. The work in [68] used majority voting and mean statistics on geocoded visitation data. Tasse et al. recursively partition space into grids of uniform cells, and then find the mode, i.e. cell with most number of check-ins. By repeating this process recursively, they are able to infer the home location.

Proximity between Consecutive Visitations. It was also found [61, 62, 75, 88] that consecutive venue visitations tend to be close in space. Thus, given a user's current venue, he is more likely to visit nearby venues than venues further away. Noulas et al. [61] showed that the probability distribution of spatial distance between consecutive check-ins exhibits a decreasing trend that resembles an inverse power law. Basically shorter distances are more likely to appear than longer distances, although the latter still has small, non-negligible probabilities. The study in [62] used the complementary cumulative distribution function on inter-check-in distances and arrived at very similar findings. There is also concurrence with the finding in [75] that human walk patterns exhibit statistically similar features as Levy walks. The study was of very high resolution, conducted using mobility track logs from participants carrying GPS receivers. It was found that people tend to visit nearby places and occasionally distant places. In another work [88], Yuan et al. studied Gowalla and Foursquare check-ins to surface a similar characteristic, which

they termed as spatial influence. They incorporate spatial influence in their model for venue recommendation, using a power law distribution to model the willingness of users to move between venues as a function of inter-venue distances.

Remarks. The discussed proximity characteristics can be generalized. If users tend to visit venues near their home [69, 10, 22], then by the transitivity property, users are also likely to visit venues near any of their previously visited venues. We termed such users as *spatially focused* users. Considering proximity between consecutive visitations [61, 62, 75, 88] and the observation that users revisit venues [10] or activity regions, we can arrive at a similar characteristic. We regard spatial focus as a much more general characteristic that is applicable even if one has no knowledge of a user's home location or current location. This has implications for tweet geolocation. Basically to geolocate a test tweet from a given user, one can leverage this characteristic in conjunction with the user's location history to refine the set of candidate posting venues. We shall investigate and exploit this characteristics in Chapter 3.

Venue Popularity with Time. Mobility patterns are affected by time of the day and day of the week. For example, dining venues are more popular at meal times while nightlife venues are more popular at late hours.

Venue popularities with time were studied extensively in prior work [53, 61, 88]. Long et al. [53] defined trending venues as venues that are popular at a certain time. They found that features such as events and venue-specific promotions can influence venue popularity at certain time periods. Noulas et al. [61] and Yuan et al. [88] incorporated temporal popularity into their models, together with other aspects, for predicting the next check-in. They considered temporal popularities of venues in two aspects: time of the day and day of the week.

Remarks. Temporal popularity directly links to the probability that a venue is the posting venue of a tweet, given a certain posting time. Since tweet posting time is readily observed, we include tweet posting time for modeling in Chapter 3.

The influence of Friends. Friends can affect the mobility patterns of users

to some limited extent. This motivates the development of collaborative filtering techniques for check-in venue prediction. Both [8] and [10] found that less than 10% of a user's check-in venues are also visited by his friends. Despite this small proportion, both works achieve convincing improvements by incorporating social information for prediction. For example, the model by Cho et al. [10] specifies that a user is more likely to check in at a venue at a certain time if one or more friends of his check in at the same or nearby venues at around the same time. In [8], Cheng et al. incorporate social regularization in their matrix factorization model, penalizing differences between the latent factors of each user and his friends.

Chong et al. [11] found that for a long-term visitor to a city, his check-in venues and friendships are weakly related such that the visitor is more likely to check-in to venues near those visited by his friends. This is even though compared to locals, visitors may not have deep social connections at the cities they are visiting. Gao et al. [31] also investigated if users' friendships affected their check-in behavior. They found that on average a pair of friends shares three times as much check-ins as a pair of strangers. They also computed cosine similarities between user pairs based on check-in venues and found similarities to be significantly higher for friends than strangers.

Remarks. In our work, we shall conduct empirical analysis that differs from the cited works. In Chapter 4, we show that users that are more similar in content history are also more similar in their visitation history. This analysis does not rely on the presence of any friendship links. We exploit this characteristic in a collaborative filtering framework to geolocate tweets from users with content history only.

2.1.2 Spatial Homophily of Locations

Users generate content differently depending on their locations. Interestingly locations that are near each other tend to have more similar user-authored content than locations further apart. We termed this as spatial homophily with respect to locations. This is discussed in further detail in Chapter 3.

Spatial homophily has been illustrated in prior work on geographical topic modeling [1, 36, 86, 24]. Generally such work surfaced region-specific topics. Ahmed et al. [1] proposed a hierarchical topic model that automatically infers both the hierarchical structure over content and over the size and position of geographical locations. In the topic hierarchy, topics at a higher level correspond to broad regions whereas topics at lower level correspond to more fine-grained locations, e.g. a neighborhood. Hong et al. [36] proposed an approach that models content in tweets based on topical influence, user's interest and geographical influence. The latter exerts its influence on the contents in tweets, causing the probability of certain words to deviate from a global background word distribution. Yin et al. [86] used tags from geocoded Flickr images to infer region-specific topics. Their model generates topics from regions whereby the geographical distribution of locations in each region follows a Gaussian distribution. Words close in space are more likely to belong to the same region and are more likely to be clustered into the same topic. We also note the work by Eisenstein et al. [24] who proposed a multi-level generative model based on cascading topic models. Their model recovers coherent topics and their regional variants, while identifying geographic areas of linguistic consistency. In short, the above cited works imply the presence of geographical topics or geographically influenced content. This supports the notion of spatial homophily on a coarse spatial level.

Some other works [25, 87, 18] used mobility patterns and venue features to infer neighborhoods of various functionalities or characteristics within a city, e.g. a shopping or residential neighborhood or neighborhoods with different demographics. Cranshaw et al. [18] clustered venues in a city based on both spatial proximity and social affinity. The latter is based on representing each venue as a bag of check-in users. They show that distinctive clusters arise, representing neighborhoods of different characteristics. Falher et al. [25] characterized neighborhoods using features derived from check-ins at neighborhood venues. They also explored finding neighborhoods of similar functions across different cities using the earth-mover's

distance as the metric. Yuan et al. [87] infer the functions of neighborhoods with the Dirichlet Multinomial Regression [59] topic model. They regard neighborhoods as documents, venue information as metadata and human mobility patterns from taxi rides as words.

In short, neighborhoods are clusters of venues having similar functions or characteristics. Thus users tend to post more similar content, resulting in the spatial homophily phenomenon.

2.2 Coarse-grained Geolocation

We review coarse-grained geolocation as it is a big research area that precedes fine-grained geolocation. Coarse-grained geolocation seeks to geolocate tweets or users at the city or region level. There are two different tasks as discussed next.

User Geolocation. The first task infers the home city or region of users by exploiting the content over multiple tweets posted by each user. For this, Cheng et al. [9] proposed a function that models the distribution of words over space, such that Location-Indicative (LI) words can be identified from model parameters. The idea is that such words should have high local focus and a fast dispersion, i.e. (1) it is very frequent at some central spatial point and (2) usage rapidly declines as one moves away from the central point. One can then use LI words found in the tweets of users to infer their home location. Chang et al. [7] also exploited location-indicative words. However to detect such words, they applied Gaussian Mixture Models (GMM) instead. Based on the notion that LI words should have probability mass concentrated on relatively few points, they used GMMs with relatively low number of components. Words with probability mass that are spatially focused on a small area are then picked out as LI words.

In [34], Han et al. compared various approaches such as statistical methods, e.g. hypothesis testing; information theory e.g. word entropy; and heuristics-based approaches e.g. TF-IDF to identify LI words. For each method, words are ranked

by their indicativeness and top ranking ones are used for geolocation. They found that geolocation performance of the various methods varies greatly with the number of top ranked words. They also found that user declared meta-data such as time zone, description etc. is useful for geolocation as they contain information that is complementary to that in tweet contents.

Jurgens [42] geolocated users based only on their social relationships, independent of any tweet content. The idea is to spatially propagate location assignments through the social network, using only a small number of initial locations. This assumes that users are likely to be near their friends. With the same intuition, Rahimi et al. [73] employed spatial propagation over friendship networks constructed from mentions in tweets. However they incorporated text-based geolocation priors into their network, showing that this joint exploitation of text and social network information performs better than text-only and network-only approaches.

Tweet Geolocation. For the second task, one geolocates individual tweets, instead of users. The approaches of [1, 36] are based on topic models. Ahmed [1] adapted the Nested Chinese Restaurant Process [3] to derive the nested Chinese Restaurant Franchise Process. With this adaption, they derive hierarchical topics whereby topics at a higher level correspond to broad regions whereas topics at lower level correspond to more fine-grained locations. In [36], Hong et al. employed the Sparse Additive Generative Model framework [23] based on modeling deviations caused by facets, e.g. a posting location will cause probabilities of certain words in a tweet to 'deviate' from some background distribution. In short, both topic modeling approaches assume some generative process for tweets, dependent on the posting location. Since topics are dependent on the posting location, the topic models can be used to geolocate tweets by inferring their topics.

In [70], Priedhorsky et al. modeled each word as a a Gaussian Mixture Model (GMM). To geolocate each tweet, the multiple GMMs corresponding to multiple words are linearly combined whereby words that are more location indicative are assigned higher weights. The works in [43] used naive Bayes to model the proba-

bility of words given locations. Given a tweet, one retrieves locations that have high probability of generating the tweet content.

Grid based approaches [84, 76, 63] have also been explored. Wing and Baldridge [84] discretize space into a uniform grid of square cells, followed by modeling the smoothed distribution of words for each cell. Test tweets are geolocated to the most similar cell based on the Kullback-Leibler (KL) divergence between word distributions or based on tweet content probability under a naive Bayes model. In [63], O'Hare and Murdock utilize uniform grids, the naive Bayes language model and some adaptation of spatial smoothing to geolocate Flickr photos using the photo tags. Instead of uniform grids, the work in [76] proposes an adaptive grid constructed using a k-d tree. This adapts to the training set size and geographic dispersion of the documents, i.e. more densely populated areas will be fitted with more numerous and smaller cells.

For each test tweet, the above works provide either a coordinate estimation [1, 36, 70] or a coarse discrete location, e.g. city/grid cell [43, 84, 76, 63]. This differs from fine-grained geolocation as will be explained next.

2.3 Fine-grained Geolocation

In contrast to coarse-grained geolocation, we work on fine-grained geolocation of tweets. This aims to link tweets to specific venues, e.g. geolocating a tweet "Flight delayed" to some airport venue, instead of a city, grid cell or a coordinate which may be associated with many venues.

Compared to coarse-grained geolocation, fine-grained geolocation is relatively less well explored. However certain approaches can be carried over. In [47], Li et al. modeled each venue as having some distribution over words. In an approach analogous to [84] for coarse-grained geolocation, tweets are geolocated using KL-divergence to the venue with the most similar word distribution. They also model venue probabilities based on posting time. This is linearly combined with the trans-

formed KL-divergences to form venue scores. However we note that their experiments are rather limited in that they only geolocate tweets posted from the ten most popular venues in each city. In [44], each venue generates words according to a fitted naive Bayes model, analogous to [43] for coarse-grained geolocation. However, not all test tweets will be geolocated. They regard tweets without any LI words as not tractable for geolocation. Such tweets are discarded. Hence there is a possibility in applications of discarding too many tweets. In [40], Ikawa et al. learned the keywords that are highly associated with locations from geocoded tweets generated by location apps. A test tweet that has at least one keyword is then geolocated to the location with highest cosine similarity. Again, there is the issue that test tweets without any key words are ignored.

In [6], Cao et al. conducted extensive feature engineering with content, location history and relationships. They specify and search for meta-paths in a network constructed from tweets, hashtags, friends, venues and Foursquare tips. Each meta-path is hand-crafted to capture certain intuitions, e.g. a user being more likely to post from a venue that his friends check-in to. The path counts of these meta-paths are used as features to a classifier which classifies whether a tweet is posted from a venue or not.

The works by [46, 41] require extracting venue mentions from tweets. In [41], Ji et al. proposed a framework to perform location recognition and location 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 [46], Li and Sun extract each location mention in a tweet and predict 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, which takes in lexical, grammatical, geographical and BILOU [1] schema features. For the discussed works [46, 41], we note that while colloquial mentions are handled, relying on mentions is a bottleneck. For example, a tweet 'safely landed' has no mentions, but is

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

indicative of the airport. Mention extraction is also a difficult problem on its own. In our work, we geolocate tweets even if no mentions exist.

2.4 Entity Linking

2.4.1 Explicit Entity Linking (EL)

In this section, we discuss Explicit Entity Linking (EL) work which precedes our work in Chapter 6. EL refers to linking mentions of named entities. Thus, mention extraction needs to be applied before EL can be conducted.

At a high level, many works formulate EL as maximizing some objective function that represents linking quality. Various objective functions have been proposed [77, 74, 51] often comprising some notion of coherence and features engineered from document and KB content. The work in [58] introduces a semantic relatedness measure to quantify coherence. The measure, derived from the Normalized Google Distance (NGD), uses only Wikipedia hyperlink structure and is inexpensive to compute. The main idea is that semantically related entities should share many common neighbors in Wikipedia. We use the same measure in our work.

There has been much EL work on long documents. Ratinov et al [74] quantified coherence based on [58] and Pointwise Mutual Information (PMI). In addition, they include features such as mention-associated text and content similarity between entities. The EL system LINDEN [78] ranks candidate entities using features derived from Wikipedia and the knowledge base's taxonomy. These features are linearly combined to form the scores of candidate entities for a given mention. The system then learns to rank candidate entities. In another system LIEGE [77], Shen at al. worked on linking entities in web lists. Besides NGD-based coherence, they also included coherence based on semantic category similarity which assumes that web lists tend to enumerate through entities of the same semantic category, e.g. a list of movies. This assumption does not apply for mentions in tweets. Unlike the above works, we focus on EL for very small text, i.e., tweets. We also assume a

non-supervised setting.

For linking individual mentions, Liu et al. [51] maximize an objective derived from coherence, mention-concept features and mention-mention features. The objective requires training of feature weights. In [79], the idea is to exploit user interest for linking. A user's initial interest score per entity is estimated from his tweets and quantified by combining coherence, content features and entity popularity. A user's interest scores over entities are initialized and propagated over a graph of entities linked by relatedness [58]. Given a new mention with multiple candidate entities, entities with higher interest score are preferred. Huang et al [38] use label propagation over a different form of graph. Graph nodes are mention-entity tuples, connected based on weighted combination of various relations, e.g. coreferencing mentions, semantic-relatedness[58] etc. After label propagation, high ranking tuples provide the linking results. Topic models [37, 35] have also been proposed for EL. Neil and Ciaramita [37] scaled up Latent Dirichlet Allocation [4] with parallel Gibbs sampling to make it appropriate for the EL task. They associate each Wikipedia entity with a topic. For the topic model of Han and Sun [35], they let topics generate the entities which in turn generate mentions and words.

Different from the above works, we consider orthogonal aspects such as spatial and temporal proximity between tweets. In terms of focused aspects, the work by Fang and Chang [26] is related. They learned entity distributions over time and large geographical areas (smallest area considered is $100km^2$) in a weakly supervised setting. In contrast, we work in the unsupervised setting and consider small geographical areas spanning hundreds of meters. For an unsupervised approach, TAGME [27] is applicable. Its key idea is: within the same document, candidate entities across mentions vote for each other. For a given mention, the entity with the highest prior is then selected from the top most voted entities. We shall also implement TAGME as a non-collective EL baseline.

2.4.2 Implicit Entity Linking (IEL)

Compared to explicit entity linking, Implicit Entity Linking (IEL) is less well explored. For IEL, Perera et al. [66] built information network to link entities and knowledge nodes, using factual knowledge from the knowledge base and contextual knowledge from labeled tweets. They then use graph features to rank entities. For implicit entity linking in tweets, Meij et al. [56] employed extensive feature engineering on content, page links and lexical word form. They then trained decision trees for ranking entities that are related to each tweet (rather than each mention). In contrast with both discussed IEL work, our IEL work in Chapter 7 assumes the posts in our training set are not entity-labeled, but are associated with venues. Thus our work explores a different task setting.

Some existing EL models can be easily extended to apply them for IEL. In our work (see Chapter 7), we have included such extensions as baselines. As discussed in the previous section, Fang and Chang [26] learned entity distributions over time and grid cells and integrate them into a base linking system. As a baseline, we have adapted their model by replacing the mention-to-entity linkage component with a post-to-entity linkage component. We also adapt the TAGME model [27]. In our extension, our voting entities are candidates for posts from the same venue, not mentions from the same document. Further details are provided in Section 7.4.3.

Part I

Venue Context Recovery

Chapter 3

Tweet Geolocation: Location History, Spatial Homophily and Temporal Popularity

3.1 Introduction

In this work [16, 14], we conduct *fine-grained geolocation* [44, 47, 46, 41], which links tweets to the specific venues from which they were posted e.g. restaurants, offices etc. We focus on tweets posted by users with location history, i.e. they have posted geocoded tweets in the past. We cast fine-grained geolocation as a learning to rank problem. Given a non-geocoded tweet from a city, we rank venues in the city such that highly ranked venues are more likely to be the posting venue.

Challenges. Tweets are short and colloquial, and may be posted from any one of the thousands of candidate venues in a given city or area of interest. Hence fine-grained tweet geolocation is highly challenging. For example, a tweet "having dinner" can arise from any of the numerous food venues or even at one's home. Some prior work [44] mitigated this challenge by performing fine-grained tweet geolocation for tweets with location-indicative words only, i.e. words used mostly at very few locations, e.g. "airport". Tweets with such words are thus easier to be

geolocated. Here, we geolocate both tweets with and without location-indicative words. To achieve better geolocation performance and to perform fine-grained geolocation on any tweets, we shall exploit the characteristics of users and venues, as surfaced by our empirical analysis.

For fine-grained geolocation, it is also challenging to acquire ground-truth data for meaningful experiments. Tweets have to be associated with the specific venues, instead of just the location coordinates. A popular strategy [6, 47] is to leverage on location-apps such as Foursquare where users associate their posts with specific venues. Besides adopting this, we also propose a novel strategy of linking tweets to venues based on Foursquare users posting tweets and check-ins within a short time period (see Section 3.2.2).

Empirical Analysis. For more effective geolocation, we first study some useful characteristics of venues and users, namely spatial homophily, spatial focus and the availability of location history. We first exploit the venues to investigate *spatial homophily* with respect to fine-grained spatial locations. Spatial homophily is a concept that has been studied at coarse geographical resolution [7, 1]. This concept means that social media content from the same city/region are more likely to share common words than content from different cities/regions, possibly due to geographical bias of language use in Twitter. For example, 'Tube' is commonly used to refer to the subway system in London, but hardly used in a similar fashion for Singapore. *However, at a much finer spatial scale such as between venues in a city, is spatial homophily still observable?* Our empirical studies indicated yes. Venues near each other tend to have more similar content than venues further apart in the same city. In other words, venues near each other have more similar text representations. Furthermore, spatial homophily is stronger for tweet content generated using a location-app (e.g. Foursquare) than that for tweet content that is posted not using a location-app. Next we focus on the user aspect. We show that while the proportion of geocoded tweets in Twitter is small [36, 1], they are posted by a substantial proportion of users. This justifies the design of personalized models that exploits user

location history in location-related applications. In addition, we show that users are *spatially focused* and are more likely to visit venues that are near each other. This characteristic can be readily incorporated into probabilistic models for geolocation.

Approach. Drawing from the various user and venue characteristics, we then propose several probabilistic geolocation models. We formulate our models such that parameters can be easily optimized in a learning to rank framework. We incorporate the loss function from [17] as a proxy for the ranking metric of mean reciprocal rank, along with novel adaptations to lower the computation complexity.

Via extensive experiments, we show that models incorporating user and venue characteristics such as venue temporal popularity and user location history consistently outperform pure content-based approaches. We also show our models to be useful even on tweets without words that are indicative of locations. This enables us to geolocate more tweets in applications.

Contributions. Our contributions are listed as follows:

1. We conduct empirical analysis to surface characteristics for exploitation in models. We show that spatial homophily exists at fine granularities such that venues near each other are more similar in content. We observe this effect to be stronger for tweet content generated in association with a location-app.

2. We show that 30% to 40% of users in Twitter have location history that are useful for model building. We also show that users are spatially focused in being more likely to visit venues near each other.

3. We propose several novel models for the fine-grained geolocation problem. For selected models, we optimized their parameter by minimizing an adapted loss function in a learning to rank framework.

4. Our experiments show that the various characteristics are useful for geolocation, with venue temporal popularity and user characteristics (location history and spatial focus) achieving large improvements. Depending on the dataset

and metric, our best performing model provides ranking accuracy improvement from 6% to 60% over the naive Bayes model.

This chapter is organized as follows. We first define two kinds of geocoded tweets in this study and the corresponding datasets in Section 3.2. We then cover the empirical study of both user and venue characteristics in Section 4.2. Section 3.4 presents our proposed fine-grained geolocation models. The experiment setup and results are given in Section 3.6. We conclude the chapter in Section 3.7.

3.2 Data for Geolocation

For our geolocation work, we require tweets with ground truth venues. To find them, we exploit users who are present in both Twitter and Foursquare. We use two types of geocoded tweets by these users. The first type consists of geocoded tweets from users who publish their Foursquare shouts at some venues using Twitter. The second type consists of pure tweets that we associate with venues using a very stringent criterion. For each type of tweets, we apply a different pre-processing step before using the data. We use the processed tweets for our empirical analysis on spatial homophily, spatially focused users as well as in our geolocation experiments.

3.2.1 Shouts (SHT)

These are tweets pushed from Foursquare, a highly popular location based social networking app. Such a setup is to construct a convenient source of tweets with ground truth venues and has been used in prior work [6, 47].

In Foursquare, users can write comments and broadcast them to Twitter while they check-in to a venue. Following Foursquare terminology, we refer to such tweets as *shouts*. As shown in Table 3.1, a shout contains the user-authored comment plus an app-generated portion indicating the check-in venue. We discard the latter portion which is trivial for geolocation and not meaningful for empirical analysis. Thereafter, we use only the comments for empirical analysis and geolocation.

Table 3.1: Sample shouts. Bolded portions are user-authored comments. Only this portion is used for empirical analysis and geolocation.

1	**Passport photo look retarded** (@ Immigration & Checkpoints Authority w/ 5 others)
2	**Dread dread dread work** (@ Orchard Central in Singapore)

3.2.2 Pure Tweets (TWT)

We refer to tweets that are authored by users and non-retweets as *pure tweets*. We iterate through users with Foursquare check-ins and extract their pure tweets whose posting venues can be accurately determined. Specifically, for each pure tweet from user u, we link it to u's check-in that is nearest in time. If the time difference is less than a specified threshold, then we assign the check-in venue as the tweet's posting venue. We use a stringent threshold of 5 minutes. This assumes the user is tweeting from where he check-ins, if both actions are within 5 minutes of each other.

Terminology. Subsequently we use 'tweets' to refer to both pure tweets and shouts. Where differentiation is required, we use each term explicitly, i.e. pure tweets or shouts.

3.2.3 Datasets

We collect data for users from Singapore (SG) and Jakarta (JKT). For Singapore, we collected 1,190,522 Foursquare check-ins from 2014, of which 30% involve shouts. The check-ins are posted by 29,301 users over 65,701 venues. We refer to this dataset as **SG-SHT**. Based on the previously discussed process, we also collected 90,250 pure tweets from 6424 users over 12,616 venues. We designate the dataset as **SG-TWT**. For Jakarta, the **JKT-SHT** dataset comprises 177,570 check-ins for the period 2015 to mid-2016, of which 49% are shouts. The check-ins are from 12,119 users over 45,213 venues. Linking the check-ins to pure tweets, we obtain only 1335 pure tweets (**JKT-TWT**) posted by 592 users from 886 venues. This small number is possibly due to platform API changes which affected crawling. We use JKT-TWT only for testing, not training.

For Singapore, tweets are mostly in English while for Jakarta, tweets are mostly in bahasa Indonesia. For both language types, words are represented in alphabets and easily processed for our models. It is not necessary to apply any machine translation for pre-processing.

3.3 Empirical Study

3.3.1 Spatial Homophily

Users in the same city/region generate more similar social media content when compared to another city/region [9, 7], due to geographical bias in language usage in Twitter. We refer this as spatial homophily with respect to locations. Does spatial homophily exist on a much smaller spatial scale such as between venues? To our knowledge, spatial homophily has not been studied at the venue level, thus motivating our analysis. Given venues in the same city, we compare the content of venues near each other versus venues which are far apart. If spatial homophily exists, then venues near each other should have more similar text representations.

Table 3.2: Average ratio statistic ($\bar{R}$) and average proportion of venues where nearest neighbors are more (or less) similar in content, compared to non-neighbors.

Dataset	Category	More similar	Less similar	Equally similar	$\overline{R}$
SG-SHT	Mixed	41.71%	19.14%	39.15%	0.516
	Food	50.61%	30.95%	18.44%	0.476
	Shop	35.72%	21.18%	43.10%	0.486
SG-TWT	Mixed	36.38%	26.26%	37.36%	0.461
	Food	30.67%	25.94%	43.39%	0.438
	Shop	38.63%	29.51%	31.86%	0.461
JKT-SHT	Mixed	29.50%	17.09%	53.41%	0.470
	Food	30.52%	23.70%	45.78%	0.445
	Shop	32.20%	18.92%	48.88%	0.476

We have conducted an experiment to investigate spatial homophily based on the simple bag of words model. Table 3.2 presents the results. Within each dataset, we conduct two sets of analysis. In the first set (labeled as 'Mixed'), we compare venues near each other regardless of their functionality. In the second set, we con-

trol for functionality by comparing venues within the same category, e.g. comparing adjacent restaurants. The venue category labels are provided by Foursquare. There are ten categories based on functionality. For better representativeness, we use the two categories 'Food' and 'Shop', which cover more venues. Such analysis allows us to evaluate spatial homophily under mixed and non-mixed functionality conditions. Our intuition is that spatial homophily should be less observable under the mixed condition.

For brevity, we describe the procedure for the 'Mixed' analysis. If we are controlling for venue functionality, we only need to repeat the steps on venues of the targeted category. We treat each venue as a document and use its tweets to create a TFIDF vector. Let $c(w, v)$ be the frequency of word w at venue v, V be the number of distinct venues and $df(w)$ be the number of venues where w occur at least once. Then the w-th dimension of v's TFIDF vector is computed as $c(w, v) \log(1 + V/df(w))$. We then conduct the following:

- Find k venues nearest to v that are also below distance threshold ψ. This forms v's nearest neighbor set, denoted as $nb(v)$. If there are $l < k$ venues below distance threshold, $nb(v)$ will only include l venues.
- Compute average cosine similarity between v and nearest neighbors denoted as $\overline{cos}_{nb}(v)$.
- Randomly sample k venues more than distance ψ away as non-neighbors, denote as $nnb(v)$.
- Compute $\overline{cos}_{nnb}(v)$, the average cosine similarity between v and non-neighbors.
- Compute $\overline{dist}_{nb}(v) = \frac{1}{|nb(v)|} \sum_{v' \in nb(v)} d(v, v')$, i.e. the average distance from v to $nb(v)$ whereby $d(v, v')$ is the distance between v and v'. Also compute $\overline{dist}_{nnb}(v)$, the average distance from v to $nnb(v)$

Since Singapore and Jakarta are dense cities, we use $k = 5$ and $\psi = 500m$. After iterating over all venues with content, we tabulate the proportion of venues whose nearest neighbors are more similar than the non-neighbors i.e. $\overline{cos}_{nb}(v) >$

$\overline{cos}_{nnb}(v)$; and the proportion of venues whose nearest neighbors are less similar than non-neighbors. Since the non-neighbors are sampled randomly, we conduct 10 runs per city and average the proportions.

For each venue in each run, we also compare the cosine similarities of neighbors and non-neighbors with the following **ratio statistic**:

$$R(v) = \exp(\frac{-\overline{cos}_{nnb}(v)}{\overline{cos}_{nb}(v)}) \tag{3.1}$$

where the exponential function avoids computation error caused by dividing by zero. $R(v)$ is larger when in terms of content, v has less similar non-neighbors than neighbors. For each run, we average $R(v)$ over venues to obtain the **average ratio statistic** $\overline{R}$.

Table 3.2 displays the average ratio statistics and the averaged proportions. Venues with identical $\overline{cos}_{nb}(.)$, $\overline{cos}_{nnb}(.)$ fall under the 'Equally similar' column in the table. These identical value cases involve venues with no common words i.e. $\overline{cos}_{nb}(v) = \overline{cos}_{nnb}(v) = 0$. Other venues fall under the 'more similar' or 'less similar' column. Table 3.2 shows that proportions in the 'more similar' column are consistently higher for all datasets than the 'less similar' column. This implies spatial homophily since venues are more similar to their neighbors than to random non-neighbors. For example in SG-SHT, on average, 50.61% of food venues are more similar to their food venue neighbors while 30.95% are less similar, when compared against non-neighbors of the food category. The difference between these two proportions is greater for SG-SHT than SG-TWT, suggesting that the spatial homophily effect is stronger for shouts than pure tweets.

Refer to the $\overline{R}$ values in Table 3.2. If there is no difference in cosine similarities between neighbors and non-neighbors, then Equation (3.1) indicates that $\overline{R}$ is expected to be $\exp(-1)$=0.368. As can be seen, all values are higher than this. On average, a venue's non-neighbors are less similar in content than neighbors. This again indicates spatial homophily. $\overline{R}$ is also higher for SG-SHT than SG-TWT across all categories. This reaffirms that spatial homophily is stronger for shouts.

Table 3.3: Venues (in brackets <>) near each other and sample shouts demonstrating spatial homophily.

M1	<Cha Cha Cha Mexican Restaurant & Bar> 'Hehe finally satisfied ma Mexican food craving w momsie'
M2	<El Patio Mexican Restaurant & Wine Bar> 'Mexican Hogmany food with @joanniewalker'
N1	<Executive Cafe> 'Hotpot at NTU. Yum <3 with Lem'
N2	<McDonald's> 'At NTU's North Spine.'

One possible explanation is that for pure tweets, users tend to share more diverse topics, which can be quite unrelated to their current venues. Different from pure tweets, shouts are authored by users as they check-in to some venues. They then broadcast their shouts to the Twitter, intentionally sharing their venues. Thus users may be more likely to mention aspects related to current venues or the local area. This also implies that pure tweets are harder to geolocate compared to shouts.

Interestingly, the 'Mixed' experiment which does not control for venue functionality exhibits spatial homophily effects that are rather comparable to 'Food' and 'Shop'. On inspection, we observed various contributing factors. While moving around adjacent venues of different functionalities, users may mention local spatial characteristics, events or be using unique words, e.g. mentions of friends.

Table 3.3 illustrates examples of spatial homophily. Shouts M1 and M2 are from Mexican restaurants near each other. User mentions of Mexican food contribute to content similarity between venues. For shouts N1 and N2, they are posted from venues in Nanyang Technological University (NTU), a university in Singapore. Thus NTU constitutes a local spatial feature and its mentions increase content similarity between venues on campus.

3.3.2 Location History

As the proportion of geocoded tweets is small, one may easily assume that they are contributed by an equally small proportion of users. For such users, the geocoded

tweets constitute a personal location history, which can be used to build more accurate models to geolocate their non-geocoded tweets. However are such models widely applicable to users? We therefore need to investigate the proportion of users with personal location history.

For the purpose of this empirical analysis, we randomly sample 50,000 Twitter users from Singapore for 2014 and from Jakarta for June to Dec 2016. The only sampling condition is that each sampled user has posted at least one tweet during the study period. Sampled users may or may not be active on Foursquare. Table 3.4 shows the statistics compiled.

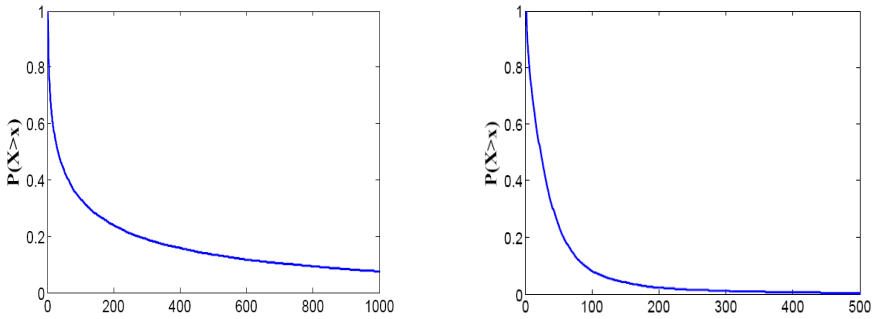
As expected from prior work [36, 1], the proportion of geocoded tweets is tiny at 3.22% for Singapore and 4.62% for Jakarta. However we find that the proportion of users posting geocoded tweets is substantial. For ease of discussion, denote the set of users who posted at least one geocoded tweet as $\{u\}_g$. Table 3.4 shows that in Singapore, $\{u\}_g$ comprises 30.34% of the sampled users. This is much larger than the value of 3.22% if one does a naive inference based on the fraction of geocoded tweets. Similarly in Jakarta, $\{u\}_g$ is substantial at 41.96% of the users. Such proportion characteristics arise because users in $\{u\}_g$ post both geocoded and non-geocoded tweets, with the latter at much larger counts. The last two rows of Table 3.4 illustrates this. On average, a Singapore user in $\{u\}_g$ post 289.69 geocoded tweets and 4532.98 non-geocoded tweets. A similar bias in tweeting behavior can be observed for Jakarta.

Intuitively, an average user is constrained by geographical, social or personal factors. This leads to venue revisits, or the conduct of many activities (e.g. work) in geographically localized regions. Now, consider a user in $\{u\}_g$. He has geocoded tweets with location coordinates. Such location history may provide useful information on his visit routines and activity regions. We can then build a personalized model of the user, that better geolocates his other non-geocoded tweets. Obviously, this also requires sufficient geocoded tweets per user, thus motivating our next analysis.

Table 3.4: Statistics for 50,000 sampled users from Singapore (2014) and from Jakarta (June to Dec, 2016).

	Singapore	Jakarta
Total Tweets	136,548,216	20,466,019
Geocoded Tweets	4,394,378 (3.22%)	946,432 (4.62%)
Users with geocoded tweets, $\{u\}_g$	15,169 (30.34%)	20,982 (41.96%)
Ave. geocoded tweets / user in $\{u\}_g$	289.69	45.11
Ave. non-geocoded tweets / user in $\{u\}_g$	4532.98	157.48

For users in $\{u\}_g$, we examine their distribution of geocoded tweets. This gives a sense of the proportion of users with sufficient location history for learning a model. Figure 3.1 displays the Complementary Cumulative Distribution (CCDF) plot. The plots show that many users in $\{u\}_g$ have adequate number of geocoded tweets. For example, Figure 3.1(a) indicates that for Singapore, around 40% of the users in the $\{u\}_g$ set has more than 50 geocoded tweets over a one year period. For Jakarta, over a half year period, the corresponding proportion is around 25%.



(a) Singapore (b) Jakarta

Figure 3.1: CCDF for users in $\{u\}_g$. X-axis = no. of geocoded tweets per user

3.3.3 Spatially Focused Users

We say a user is *spatially focused* if he or she is more likely to visit venues that are near his/her other visited venues. For each user, we compute a distance-based statistic to quantify the extent of spatial focus. We compare this against the expected distance statistic when a user visits the same number of venues in a random manner. We term the latter as the null model. We conduct our analysis on users with

geocoded tweets tied to Foursquare (datasets SG-SHT and JKT-SHT).

Denote $\mathbb{V}_u$ as the set of venues visited by user u. We iterate through each venue in $\mathbb{V}_u$ and compute the distance to the nearest neighboring venue. This is averaged over all venues in $\mathbb{V}_u$. If the distance statistic is small, relative to the null model (to be defined), then there is stronger evidence of spatial focus. Formally, the distance statistic is:

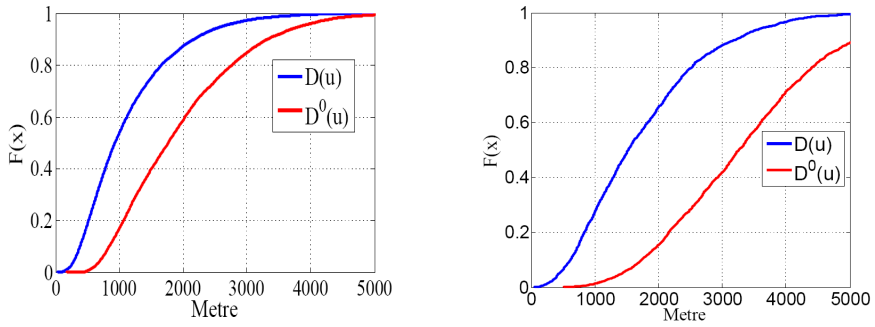
$$D(u) = \frac{1}{|\mathbb{V}_u|} \sum_{v \in \mathbb{V}_u} \min_{v' \in \mathbb{V}_u \setminus v} d(v, v') \tag{3.2}$$

where $d(,)$ measures spatial distance. $D(u)$ is easy to compute. It neither assumes any parametric form for the spatial distribution, nor knowledge of the number of spatial clusters.

The null model computes the expected distance statistic if the user is not spatially focused, but visiting venues at random. For the null model, we reassign each unique visit of user u to a random venue and obtain a random venue set $\mathbb{V}_u^0$ of the same size as $\mathbb{V}_u$. We then apply Equation (3.2) again to compute the distance statistic $D^0(u)$. Note that to ascertain the presence of spatial focus, it is important to compare $D(u)$ versus the null model rather than just examining its actual value. The reason is that $D(u)$ can be small even if a user is not spatially focused. For example, assume a huge geographical area containing many points which equally split the area. Let these points correspond to the coordinates visited by user u. When the number of points is sufficiently large, then $D(u)$ is small, although u is not spatially focused. However in this case, if we apply the null model, $D^0(u)$ will be small as well and close to $D(u)$. Thus by comparing both values, we can avoid drawing the wrong conclusion that u is spatially focused.

Figure 3.2 plots the Cumulative Distribution Function (CDF) of the distance statistics for Singapore and Jakarta. For each city, there is clear evidence that users are spatially focused. The red curve for the null model statistic consistently lies to the right of the blue curve for the user statistic. This implies that venues visited by users are spatially nearer each other than random. For example, Figure 3.2(a) shows that if users are visiting venues randomly (red curve), then we expect only

60% to have distance statistic of 2000 metres or less. However, the actual behavior (blue curve) indicates that the corresponding proportion is around 90%. For Jakarta in Figure 3.2(b), 65% of users (blue) have distance statistic of 2000 metres or less, much higher than the expected proportion of 15% based on the null model (red).



(a) Singapore (b) Jakarta

Figure 3.2: CDF of Distance statistic of users (blue) vs null model (red). (X-axis=distance in metres)

Remarks. In short, our empirical analysis highlights that users with geocoded tweets form a significant group in Twitter, much more than what one would expect from the proportion of geocoded tweets. We also observe strong evidence that users tend to visit venues that are spatially near each other. These motivate the design of personalized models based on users' location history.

3.3.4 Venue Temporal Popularity

Each tweet is associated with a posting time, which provides a modeling linkage to venue temporal popularities. Intuitively, different venues are more popular at different times of the day, e.g. dining venues are more popular at meal times while nightlife venues are more popular at late hours. This directly affects the probability that a venue is the posting venue of a given tweet at different times of the day.

Venue temporal popularities were studied extensively in prior work [53, 61, 88]. Also, tweet posting time is always observed, in contrast to user location history. Hence we omit empirical analysis and coverage studies. Instead we capture the discussed intuitions by including tweet posting time for modeling. This improves

geolocation performance significantly, as will be discussed in the experiment results.

3.4 Models

We first describe a baseline model for geolocation. We then propose several models that draw on the empirical analysis findings as well as incorporate additional contextual information. We elaborate the associated notations in an inline manner for ease of reading.

3.4.1 Naive Bayes (NB)

We denote the naive Bayes model from [44, 43] as NB. This models the tweet content associated with each venue as a bag of words **w**. Let W be the vocabulary size of tweet words. We use $c(w, v)$ as the frequency of word w at venue v and $c(., v)$ to denote $\sum_w c(w, v)$. Given a tweet, we then rank venues by the venue probability $p(v|\mathbf{w}) \propto p(v) \prod_{w \in \mathbf{w}} p(w|v)$. The probability of word given venue $p(w|v)$ is

$$p(w|v) = \frac{c(w, v) + \alpha}{c(., v) + W\alpha} \tag{3.3}$$

where α is the smoothing parameter which can be tuned or set at 1 for Laplace smoothing. Equation (3.3) can be interpreted in a Bayesian framework using the concept of conjugate priors. Equation (3.3) corresponds to the posterior predictive distribution of a multinomial distribution with a Dirichlet distribution as the prior. In this case, the Dirichlet distribution is symmetric and parameterized by α.

3.4.2 Spatial Smoothing (NB+S)

Our earlier empirical analysis had demonstrated the presence of spatial homophily where venues near each other are more similar in content. To consider this effect, we add spatial smoothing to the naive Bayes model NB. For each word w at the

ego venue v, we extend the definition of $p(w|v)$ with word frequencies of v's set of spatial neighbors, denoted by $nb(v)$. The spatially smoothed $p(w|v)$ is defined as:

$$p(w|v) = \frac{c(w,v) + \alpha + \frac{\gamma}{|nb(v)|} \sum_{v_i \in nb(v)} c(w,v_i)}{c(.,v) + W\alpha + \frac{\gamma}{|nb(v)|} \sum_{v_i \in nb(v)} c(.,v_i)} \tag{3.4}$$

where $0 \leq \gamma \leq 1$ is the weight factor. By setting γ, we adjust the spatial smoothing strength on word frequencies from v's neighbors. When $\gamma = 1$, a word w found in every v's neighbor will be equivalent to a single w occurrence in v. Otherwise, the words from neighbors are weighted less than the native words in v. Also recall that our earlier analysis shows that spatial homophily exist even without controlling for venue functionalities. Thus we do not need to restrict neighbors to be of the same category as the ego venue.

Similar to Equation (3.3), Equation (3.4) can be interpreted as the posterior predictive distribution resulting from a Dirichlet-multinomial conjugate pair. The difference is that the multinomial distribution specific to each venue is now adjusted with contributions from its spatial neighbors.

3.4.3 Tweet Posting Time (NB+S+T)

The previous models mainly exploit the tweet content. As tweet content is short, ranking accuracy may be low due to information sparsity. We thus explore user and/or venue characteristics to improve performance. As previously mentioned, the posting time of tweets is readily available. This ties to the characteristic that certain venues are more popular at different time of the day, making them more likely to be the posting venues of tweets. Hence given a tweet posted at time of day t, we incorporate time into the model as follows:

$$p(v|\mathbf{w}, t) \propto p(v|t) \prod\nolimits_{w \in \mathbf{w}} p(w|v) \tag{3.5}$$

where $p(v|t)$ accounts for venue popularity at time of day t.

A simple approach to compute $p(v|t)$ is to discretize t into time bins, e.g. hourly and estimate the venue distribution for every bin. However there are boundary effects which are counter-intuitive. For example, consider discretizing by hourly bins where each bin starts on the hour. Then $t = 2359$ hrs and $t = 0001$ hrs are in different bins, although they are only 2 minutes apart. In contrast, $t = 0001$ hrs and $t = 0059$ hrs are 58 minutes apart, but in the same bin.

Instead of binning, we model time of day t as a continuous variable which is more intuitive. We estimate $p(v|t)$ in an approach motivated by kernel density estimation (KDE) [49]. For time of day t, define a time interval of length $T(t)$ which covers t. Denote by $f(v, t)$ the number of user visits to venue v in the interval $T(t)$ and let $f(\cdot, t) = \sum_v f(v, t)$. Given a test tweet with time of day t, we compute

$$p(v|t) = \frac{f(v,t) + \beta}{f(\cdot,t) + V\beta} \tag{3.6}$$

where V is the number of distinct venues and β is the smoothing parameter. β can be tuned or learnt (see Section 3.5).

Defining a time interval and counting the venues within is similar to applying a uniform kernel in KDE. The time interval length $T(t)$ is analogous to the kernel bandwidth. Instead of adopting a fixed interval length, we use *adaptive bandwidth selection* [49]. Basically given a test tweet posted at time of day t, we search for the k training tweets closest in time of day to define the time interval, i.e. $f(\cdot, t) = k$. To do this efficiently, we use a k-d tree structure [30]. Given a set of training tweets $\mathbb{T}$, insertion and search using the k-d-tree has average complexity of $\mathcal{O}(\log |\mathbb{T}|)$. We index all training tweets after converting their posting times to 2-dimensional Cartesian coordinates. Formally, let time of day t be represented as the number of seconds past midnight. We compute the corresponding Cartesian coordinate (t_x, t_y) as:

$$\begin{aligned} t_x &= \sin(t/3600) \\ t_y &= \cos(t/3600) \end{aligned} \tag{3.7}$$

Following Equation (3.7), we can readily apply k-d trees and Euclidean distance to facilitate k nearest neighbor computation given any time of day query.

Using the parameter k, adaptive bandwidth selection is able to adjust the time interval length locally based on data density. Basically during timings with sparse training points (e.g. midnight), the interval length is longer to cover k nearest neighboring training tweets, while during timings with dense training points (e.g. dinner), the interval length is shorter. This is also intuitive from the Bayesian point of view. Consider Equation (3.6) where $f(v,t)$ and $f(\cdot,t)$ are actual observations while β and $V\beta$ are pseudo-observations. In fixing $f(\cdot,t) = k$, we effectively use k to control the relative importance of actual and pseudo-observations to be consistent across all test tweets.

3.4.4 User Location History (NB+S+T+U)

Earlier, we showed that a substantial proportion of users have location history in the form of geocoded tweets. On average, such users also post many non-geocoded tweets, which may be targeted for geolocation. Here we use location history to build models that are personalized to each user.

Consider the previous model NB+S+T. From Equation (3.5), this model can be interpreted as a Bayesian network where the time of day node generates the venue node which then generates the words. We now let the venue node generate the user node as well. Thus we now define:

$$p(v|\mathbf{w},t,u) \propto p(v|t)p(u|v)\prod_{w\in\mathbf{w}} p(w|v) \tag{3.8}$$

Since location history are specific to users, it is more intuitive to compute $p(v|u)$ instead of $p(u|v)$. $p(v|u)$ can also be represented by two dimensional distributions over geographical space, which is convenient for interpretation and visualization. By the property $p(u|v) = p(v|u)p(u)/p(v)$ and assuming constant $p(u)$, $p(v)$, we have $p(u|v) \propto p(v|u)$. Thus the probability term $p(u|v)$ in Equation (3.8) can be

replaced by $p(v|u)$.

To model $p(v|u)$, recap that the spatial focus property means users are more likely to visit venues spatially near previously visited venues. To capture this idea, we extend the distance statistic from Equation (3.2) and define $p(v|u)$ as:

$$p(v|u) \propto \exp(-S \cdot \min_{v' \in \mathbb{V}_u} d(v, v')) \tag{3.9}$$

where $\mathbb{V}_u$ is defined previously as the set of venues in u's location history and $S \geq 0$ is the decay parameter. A large S means that $p(v|u)$ decreases faster with increasing distance between v and the nearest venue in $\mathbb{V}_u$. Equivalently, we are making the model more sensitive to the spatial focus property. In contrast, if $S = 0$, we disregard the spatial focus property.

Equation (3.9) defines an affinity vector over venues, specific to user u, whereby $p(v|u)$ are the vector elements. This vector is fixed if user u's location history is not updated. Thus one can precompute the affinity vectors for users to geolocate their tweets more efficiently. Lastly, note that for notation simplicity, we have defined Equation (3.9) in terms of distances between venues. In fact, it is not required for the specific venues to be known in the location history. It suffices for only the location coordinates of geocoded tweets to be known. Thus the proposed model is applicable to more users, including those whose geocoded tweets are not associated with specific venues.

Query Likelihood Model. Equation (3.8) can be interpreted as a query likelihood model (see Section 12.2 of [54]) in the framework of traditional document retrieval. In the query likelihood model, the probability of document $\mathbf{d}$ given query q is computed as $p(\mathbf{d}|q) \propto p(q|\mathbf{d})p(\mathbf{d})$. In Equation (3.8), venues are analogous to documents while the test tweet's user and content comprises a query with accompanying meta-information. The posting time is used to assign a non-uniform prior to the venues (i.e. documents).

3.5 Learning to Rank

Given a tweet, one desires its posting venue to be ranked high. Thus there is only one relevant venue and the Mean Reciprocal Rank (MRR) is a suitable metric. Consider a set of tweets $\mathbb{T}$. Let the ranked position of the $i-$th tweet's posting venue v_i be r_i, where $0 \leq r_i \leq V-1$. MRR with respect to tweet set $\mathbb{T}$ is defined as:

$$\text{MRR}(\mathbb{T}) = \frac{1}{|\mathbb{T}|} \sum_{i=1}^{|\mathbb{T}|} \frac{1}{1+r_i} \tag{3.10}$$

which is the average of the reciprocal ranks for each tweet in $\mathbb{T}$.

We can optimize parameters of our models with respect to MRR via tuning or Learning to Rank (LTR). For models with few parameters e.g. NB and NB+S, tuning can be done with grid search over the parameter space in order to maximize MRR directly. However for more complicated models with more parameters, tuning cost increases at an exponential rate. In contrast, LTR requires a proxy function in place of MRR and may be susceptible to local optima. However LTR can utilize gradient information for more fine-grained optimization and scales better with increasing model parameters. Considering the computation cost of tuning versus LTR and the number of model parameters, we apply different approaches to different models. For NB and NB+S, we adopt tuning based on grid search. For NB+S+T and NB+S+T+U, we adopt LTR. To further motivate our choice of using LTR instead of tuning, assume that each parameter is tuned over a grid of τ values. Then for NB+S+T+U which has 4 parameters, tuning requires applying the model on the tuning set for a total of τ^4 times. This is much more expensive than for example, tuning for NB+S, which only requires applying the model for τ^2 times.

LTR requires one to define an appropriate objective function. Firstly, Equation (3.10) can be re-expressed as:

$$\text{MRR}(\mathbb{T}) = \frac{1}{|\mathbb{T}|} \sum_{i=1}^{|\mathbb{T}|} [1 + \sum_{v \neq v_i} \mathbf{I}(p_\Theta(v) > p_\Theta(v_i))]^{-1} \tag{3.11}$$

where $\mathbf{I}(.)$ is the indicator function and $p_\Theta(v_i)$ is the probability of the posting venue v_i for the i-th tweet computed from some geolocation model Θ (e.g. NB+S). Thus maximizing MRR is equivalent to maximizing some function constructed from multiple 0-1 loss functions. Note that the indicator function has gradient of 0 except at the point of discontinuity, where the gradient is ill-defined. Hence it is infeasible to maximize MRR directly [17] via LTR. Instead, one approximates MRR maximization by minimizing a proxy loss function, whereby a good proxy should approximate the 0-1 loss well, while retaining sufficient gradient for learning. Various loss functions are possible, e.g. logistic loss. However, in recent work, [17] has proposed the log-log loss function as a better alternative to logistic loss. This motivates us to introduce the log-log loss function into our models that have been selected for LTR. For the selected models, we construct the loss function over venue pairs for minimization.

3.5.1 Loss Function

For a posting venue v_i to be ranked high, $p_\Theta(v_i)$ should be large while $p_\Theta(v)$ should be small for $v \neq v_i$, i.e. non-posting venues. For computation convenience, we use log probabilities for ranking. Let $z_\Theta(v_i, v) = \ln p_\Theta(v_i) - \ln p_\Theta(v)$. The log-log loss function for a tweet with posting venue v_i is:

$$L_\Theta(v_i) = \sum_{v \neq v_i} \ln(1 + \ln(1 + \exp(-z_\Theta(v_i, v)))) = \sum_{v \neq v_i} \ln(1 + R_\Theta(v_i, v)) \quad (3.12)$$

where $R_\Theta(v_i, v) = \ln(1 + \exp(-z_\Theta(v_i, v)))$. To obtain the global loss function, one computes and sums Equation (3.12) over the set of tweets considered:

$$G_\Theta(\mathbb{T}) = \sum_{i=1}^{|\mathbb{T}|} L_\Theta(v_i) \quad (3.13)$$

3.5.2 Re-parameterization

With the loss function defined, we can perform gradient descent to minimize it. However there are constraints on the parameters. The smoothing parameters α, β and S are required to be non-negative. The spatial weight factor γ has to satisfy the constraint $0 \leq \gamma \leq 1$. Instead of constrained optimization, we incorporate the above constraints by re-parameterizing the model as follows:

$$\begin{aligned} \alpha &= x_\alpha^2 \\ \gamma &= (1 + \exp(-x_\gamma))^{-1} \\ \beta &= x_\beta^2 \\ S &= x_S^2 \end{aligned} \tag{3.14}$$

where x_α, x_β, x_S and x_γ are the new set of parameters. These can now be easily learnt from unconstrained optimization.

3.5.3 Gradients

We minimize the loss function via stochastic gradient descent. Here, we illustrate deriving the gradient for one parameter x_S for one model: NB+S+T+U model. For notation brevity, let Θ represent NB+S+T+U. By chain rule,

$$\frac{\partial L_\Theta(v_i)}{\partial x_S} = \sum_{v \neq v_i} \frac{\partial \ln(1 + R_\Theta(v_i, v))}{\partial R_\Theta(v_i, v)} \frac{\partial R_\Theta(v_i, v)}{\partial z_\Theta(v_i, v)} \frac{\partial z_\Theta(v_i, v)}{\partial x_S} \tag{3.15}$$

For NB+S+T+U, we have $p_\Theta(v) = p(v|\mathbf{w}, t, u)$, thus:

$$\frac{\partial z_\Theta(v_i, v)}{\partial x_S} = \frac{\partial \ln p(v_i|\mathbf{w}, t, u)}{\partial x_S} - \frac{\partial \ln p(v|\mathbf{w}, t, u)}{\partial x_S} \tag{3.16}$$

where

$$\frac{\partial \ln p(v|\mathbf{w}, t, u)}{\partial x_S} \propto \frac{\partial \ln p(u|v)}{\partial x_S} = -2x_S \cdot \min_{v' \in \mathbb{V}_u} (d(v, v')) \tag{3.17}$$

and $\partial \ln p(v_i|\mathbf{w}, t, u)/\partial x_S$ is computed similarly. The gradients for other model parameters are derived in a similar manner.

3.5.4 Complexity Reduction

Let $\mathbb{T}$ be the set of training tweets, and V be the number of distinct venues. For each training tweet, there is one posting venue and V-1 non-posting venues. Consequently for each training tweet, if we construct pairwise loss between the posting venue and all other non-posting venues, then there are V-1 pairs. The overall computational complexity for training is then $\mathcal{O}(|\mathbb{T}|V)$.

We can reduce the complexity by reducing the number of pairs considered per training tweet. The simplest approach is to randomly sample M proportion of pairs per training tweet (e.g. $M = 0.25$) such that $MV < V$-1. On top of this random sampling scheme, we propose further adaptations to reduce the complexity while enabling changes in the loss function to be more correlated to changes in MRR. We achieve this by assigning greater weights to training tweets which contribute more to MRR. Such tweets already have their posting venues ranked high and are intuitively more important. For example, assume two tweets at the start of training: tweet 1 with its venue ranked at position 0, i.e. $r_1 = 0$, and tweet 2 with its venue ranked at position 99, i.e. $r_2 = 99$. The overall MRR is $(\frac{1}{0+1} + \frac{1}{99+1})/2 = 0.505$, with a contribution of 0.5 from tweet 1 and 0.005 from tweet 2. Tweet 1 is thus much more important than tweet 2. As training proceeds, model parameters evolve and may lead to changes in the venue rankings of both tweets. However, any changes in the rank of tweet 1's venue will affect MRR much more than tweet 2.

The loss function as defined by Equations (3.12) and (3.13) do not reflect the varied importance of training tweets. Furthermore, given some reduction in the loss, not all reciprocal ranks associated with test tweets are simultaneously improved. Instead there is a mixture of improvement, decline or no change. Continuing from the earlier example, it is plausible for a given loss reduction to improve the ranking of tweet 2's venue to position 49, while tweet 1's associated ranking may drop to

position 1. This leads to a reduced MRR of $(\frac{1}{1+1} + \frac{1}{49+1})/2 = 0.26$, even though the loss has decreased. Hence to better correlate loss reduction with MRR improvement, it is important to improve or maintain the ranking accuracy for tweets already associated with high reciprocal rank. To achieve better correlation, we let more important training tweets contribute more pairs. Specifically for the i-th tweet at the start of the training phase, we construct the pairwise loss to $M_i(V-1)$ other venues where M_i is a proportion computed as:

$$M_i = \frac{M}{1 + \exp(-1/r_{i,0})} \tag{3.18}$$

where $r_{i,0}$ is the ranked position of the posting venue for the i-th tweet at the start of training, i.e. 0-th iteration. Basically tweets contribute more pairs (are assigned more importance), based on their associated reciprocal rank such that $M_i = M$ for $r_{i,0} = 0$ and is close to $0.5M$ for large values of $r_{i,0}$. For example, a tweet with its posting venue perfectly ranked at the start of training contributes (V-1)M pairs to the global loss function while a tweet with a very poorly ranked venue contributes close to only 0.5(V-1)M pairs. The computational complexity is now $\mathcal{O}(V \sum_i^{|\mathbb{T}|} M_i)$. Except for extreme and unlikely cases where all posting venues are perfectly ranked at the start of training, the new computational complexity is lower than $\mathcal{O}(|\mathbb{T}|V)$, enabling training to be conducted faster.

3.6 Experiments

3.6.1 Setup

We conduct fine-grained geolocation experiments to:

1. Compare our models with each other and other state of the art baselines.

2. Assess the importance of incorporating various user and venue characteristics such as user location history and temporal venue popularity.

We split the datasets SG-SHT, SG-TWT, JKT-SHT into training, tuning and test sets. Model parameters are learnt from the training set to minimize the loss on the tuning set. We include venues as ranking candidates only if they have at least 5 tweets in the training set. We also filter out stop words and rare words (frequency <4). The test set consists of test cases of tweets, each posted from some venue by a user with location history. On inspection, we noticed 'easy' test cases, where a user repeatedly uses a highly unique word every time he posts from a certain venue. This makes the unique word highly indicative of the posting venue, leading to high ranking accuracy for such cases. To make the problem more challenging, we filter them from the training set as follows: for each test case with user u and posting venue v, we exclude u's other tweets posted at v from the training set. In other words during training, applied models do not observe any postings of u from venue v.

For each dataset, we conduct 20 runs where for each run, we sample 5000 tweets for testing/tuning and use the remainder for training. From the sampled set, we use 1000 tweets for tuning and the remainder for testing. Due to various filtering discussed above, the number of test cases per run is less than 4000. The average numbers of test cases are reported with the results for each experiment.

3.6.2 Models Applied

We compare the following models:

- KL: This model [47] assigns scores to venues based on posting time information, e.g. hour of day, and the Kullback-Leibler divergences between the smoothed language models of tweets and venues. The KL-divergences are transformed and linearly combined with the venue probabilities to form ranking scores.

- TFIDF: We represent venues and tweets as TFIDF vectors in terms of content. Given a test tweet, we use cosine similarity to retrieve and rank venues. This

is very similar to the method in [40].

- GMM: This models [7] each word as a Gaussian mixture over 2-d space, and a test tweet as the product of Gaussian mixtures. Venues are ranked by the probability that the product of Gaussian mixtures generate their coordinates. Since words that are indicative of spatial regions should have relatively few number of modes, we follow [7] and set the number of clusters to 3.

- VDOC: The topic model VDOC in [12] models the generation of check-ins and venue-related comments in the form of Foursquare tips. By treating tweets as tips and ignoring the check-in mode, we extend VDOC to model tweet generation. To generate each tweet, the venue first generates the topic. The topic then generates the posting user and the tweet words. In our experiments, we used 40 topics, after observing that this is sufficient for optimal ranking performance.

- KDE: This [39] integrates kernel density smoothing into multinomial naive Bayes to geolocate tweets to grid cells. Given cell c, geolocation is based on the probability $p(c)\prod_{w\in\mathbf{w}}p(w|c)$ whereby $p(c)$ and $p(w|c)$ are smoothed using Gaussian kernels. To apply the method for geolocating to venues, we extend it to compute $p(v|c)p(c)\prod_{w\in\mathbf{w}}p(w|c)$. Given venue v located in cell c, $p(v|c)$ is estimated by counting tweets posted from venue v, over all tweets posted within cell c. We experiment with grid sizes of 1 km and 500 m and report results from the latter due to its better performance.

- NB: This is the naive Bayes, content-only approach from [44, 43]. We observed better performance with uniform venue probabilities, i.e. $p(v|\mathbf{w}) \propto \prod_{w\in\mathbf{w}}p(w|v)$ and report the associated results. We tune the smoothing parameter α using grid search: α is varied from 0.1 to 1.5 in steps of 0.1. The value associated with the optimal tuning set MRR is then selected.

- NB+S: This extends the NB model with spatial smoothing. For spatial smoothing, we use $k = 5$ nearest neighbors of each venue to smooth the word prob-

abilities. We tune the smoothing parameter α and the spatial weight factor γ using grid search over (α, γ) values from (0.1,0.0) to (1.5,1.0), at intervals of 0.1.

- NB+S+T: This uses content with spatial smoothing plus tweet posting time which relates to venue temporal popularity. For parameter learning, we apply LTR with mini-batch stochastic gradient descent. We set M (see Equation (3.18)) at 0.25. We use 15 epochs and 50 mini-batches where each mini-batch consists of 20 tweets. To account for local optimal, we randomly initialize and train 5 instances per model. We then select the instance with the highest tuning set MRR to apply on the test set.

- NB+S+T+U: This uses content with spatial smoothing, posting time and user location history, thus exploiting all user and venue characteristics. We apply LTR with the same set-up as described above.

3.6.3 Results on Shouts

In the first experiment, we train and test models on the datasets SG-SHT and JKT-SHT. Tables 3.5 and 3.6 present results for Singapore and Jakarta shouts respectively. Note that MRR figures are not directly comparable across datasets since we are ranking with different venue sets. For JKT-SHT, there are also fewer venues to rank, making it easier to achieve high MRR.

In both Tables 3.5 and 3.6, KL, TFIDF, GMM, VDOC and KDE substantially underperform the NB model. KL includes posting time information, but fails to outperform NB anyway. Evidently, modeling each shout with a smoothed language model, as done by KL is inadequate. This in turn affects the computing of KL divergences between the word distributions of shouts and venues. TFIDF also consistently has low MRR, partly being because it is not optimized for ranking. Also if a test shout and a posting venue share no common word, cosine similarity is 0 and the venue will be ranked low. This may be overly stringent. The topic model

Table 3.5: Average MRR for SG-SHT. On average, there are 2626.2 test cases and 10814.5 venues to rank per run.

Model	MRR	Improvement over NB
KL	0.04021	-58.045%
TFIDF	0.03571	-62.740%
GMM	0.02495	-73.967%
VDOC	0.03683	-61.571%
KDE	0.06655	-30.561%
NB	0.09584	0%
NB+S	0.09620	0.376%
NB+S+T	0.09966	3.986%
NB+S+T+U	0.10271	7.168%

Table 3.6: Average MRR for JKT-SHT. On average, there are 975.9 test cases and 2713.75 venues to rank per run.

Model	MRR	Improvement over NB
KL	0.03019	-77.667%
TFIDF	0.04193	-68.982%
GMM	0.09767	-27.748%
VDOC	0.05849	-56.732%
KDE	0.10665	-21.105%
NB	0.13518	0%
NB+S	0.13623	0.777%
NB+S+T	0.14618	8.137%
NB+S+T+U	0.14824	9.661%

VDOC performs poorly despite its model complexity. This may be due to the fact that model parameters are optimized with respect to the formation of coherent topics rather than with respect to MRR. For GMM, performance is poor as we have to geolocate even shouts where words do not have peaky Gaussian distributions. Among the approaches inferior to NB, KDE is the best performing. Primarily, this approach models and smooths the word distributions of grid cells, instead of venues. Thus word distributions are learnt at a coarser level and sub-optimal for fine-grained geolocation.

Both Tables 3.5 and 3.6 exhibit similar trends from the NB model onwards. MRR improves as we add spatial smoothing and additional characteristics to the models. For adjacent models, e.g. NB vs NB+S, we have also conducted significance testing with the Wilcoxon signed rank test. The differences between all

models are statistically significant at p-value of 0.05.

Comparing NB and NB+S, spatial smoothing improves MRR slightly, which can be attributed to the presence of spatial homophily. The improvement is small but consistent across different runs. This may be due to the limited strength of spatial homophily at fine granularities. We also note that prior work on coarse-grained geolocation [9] had reported limited improvement from spatial smoothing, even when using location-indicative words only. For example, in [9], which geolocates users' cities with accuracy as the metric, the improvement from spatial smoothing is less than 1%. We also reason that even without smoothing, we are already capturing much of the spatial homophily effect. Recall that this means venues near each other have more similar content. In the NB model, we are modeling the venue content directly anyway, thus implicitly accounting for spatial homophily in a downstream manner.

For both cities, substantial improvement comes from exploiting temporal venue popularity and location history. For example, NB+S+T provides 3.986% improvement over NB in Table 3.5. For Jakarta in Table 3.6, the corresponding improvement is 8.137%. Thus venue popularity with time of the day plays a role. Adding user location history helps to increase MRR even more, with NB+S+T+U being consistently the best performing model in both tables. This shows that location history is highly useful. Also recap that our modeling approach captures the idea that users are spatially focused in being more likely to visit venues that are near each other. The experiment results further validate this.

3.6.4 Results on Pure Tweets

In this experiment, we train and test our models on pure tweets from Singapore (SG-TWT). Results are displayed in Table 3.7. We only rank venues appearing in pure tweets. This results in an average of 2783.55 venues to rank per run. Also, JKT-TWT has too few pure tweets for training and we do not use it in this experiment.

The trend in Table 3.7 is mostly similar to that of the previous experiment on

shouts. KL, TFIDF, GMM and VDOC are poor performers. KDE performs better than these techniques, but loses out slightly to NB. Spatial smoothing again provides only slight improvement over the NB model, although it is statistically significant over 20 paired runs. The exploitation of venue temporal popularity and user location history provides very sharp improvement. NB+S+T+U again has the highest MRR with over 60% improvement from NB.

Typically, MRR is not compared across experiments that rank different number of items. However here, we can make certain statements by comparing Tables 3.7 and 3.5. In Table 3.7 for pure tweets, we rank fewer venues, but obtain mostly lower MRR than Table 3.5 for shouts. Since we have fewer venues to rank, the task should have been easier, resulting in a higher MRR. The lower MRR thus implies that it is more challenging to rank venues for pure tweets than shouts. This observation is also consistent with our empirical analysis (Table 3.2), whereby we have observed spatial homophily to be stronger for shouts than pure tweets. Also, pure tweets may be about more diverse topics not related to the posting venue. Obviously this will impact ranking accuracy.

If the contents of pure tweets are not highly indicative of venues, then characteristics such as temporal venue popularity and user location history become relatively more important. This is illustrated by the huge gains in MRR as we move from model NB to NB+S+T / NB+S+T+U. The percentage improvement is much larger in Table 3.7 than the case for shouts in Table 3.5.

3.6.5 Applying Shout Models to Pure Tweets

In this experiment, we explore if models that are trained to rank using shouts (i.e. model NB and extensions) will perform well on pure tweets. The motivation is that in applications, it is easier to form training sets using shouts which are already associated with venues, than tweets which require labeling or some linking process. We apply the models trained on SG-SHT to test tweets from SG-TWT. We also train models with JKT-SHT and test on JKT-TWT. For test cases, we use pure tweets

Table 3.7: Average MRR for SG-TWT. On average, there are 2061.9 test cases and 2783.55 venues to rank per run.

Model	MRR	Improvement over NB
KL	0.03790	-33.310%
TFIDF	0.02059	-63.769%
GMM	0.01385	-75.629%
VDOC	0.01986	-65.054%
KDE	0.05349	-5.877%
NB	0.05683	0%
NB+S	0.05718	0.612%
NB+S+T	0.07600	33.526%
NB+S+T+U	0.09229	62.015%

Table 3.8: Average MRR from applying SG-SHT models to test on SG-TWT. On average, there are 31946.2 test cases and 10814.5 venues to rank per run.

Model	MRR	Improvement over NB
NB	0.04021	0%
NB+S	0.04028	0.1741%
NB+S+T	0.04993	24.173%
NB+S+T+U	0.05821	44.765%

which contain one or more words from the shout content vocabulary. We use the set of shout venues for ranking. This makes it possible to compare with the results for shouts.

Tables 3.8 and 3.9 depict the respective results for Singapore and Jakarta. The trend is similar to training/testing with pure tweets or shouts. Spatial smoothing contributes a small improvement while substantial improvements occur as we model additional characteristics. Clearly, temporal venue popularity and location history remain highly important.

For each city, we cross-compare the results for pure tweets and shouts, i.e. Ta-

Table 3.9: Average MRR from applying JKT-SHT models to test on JKT-TWT. On average, there are 363.15 test cases and 2713.75 venues to rank per run.

Model	MRR	Improvement over NB
NB	0.10571	0%
NB+S	0.10596	0.237%
NB+S+T	0.14043	32.845%
NB+S+T+U	0.14241	34.718%

bles 3.8 vs 3.5, and Tables 3.9 vs 3.6. Clearly, MRR is consistently lower for pure tweets across all models. This affirms again that pure tweets are more challenging to geolocate than shouts. This is so even though we are using pure tweets from users who also posted shouts. This should limit the differences in topics and vocabulary.

3.6.6 Stratified Experiment

Finally, we compare geolocation for tweets with and without Location-Indicative (LI) words. We also examine if we can obtain meaningful geolocation accuracy for the latter. LI words suggest a venue or spatial region with high probability, e.g. 'airport'. Typically ignoring tweets without LI words can improve performance [7, 9] for the task of inferring a user's home location. This is because users typically post multiple tweets, some of which are more informative of their home location. However we have a different task of geolocating individual tweets. Tweets without LI words were considered not appropriate for fine-grained geolocation and excluded in an earlier work [44]. Equivalently, they were regarded as noise. Depending on the strictness of the criteria for detecting LI words, a substantial fraction of data may be discarded. This is rather undesired in applications.

We adopt the approach in [44] to detect LI words. Basically LI words have high occurrence probability in at least one venue and occur at relatively few venues. Words are scored based on the TFIDF measure as follows:

$$LI(w) = \max_{v} \{p(w|v) \log(\frac{V}{df(w)})\} \tag{3.19}$$

Equation (3.19) encapsulates some word popularity effects due to the term $p(w|v)$. Thus more popular words tend to have higher scores, although this is offset to some extent by the lower inverse document frequency inherent in such words. Empirically, we observe a larger fraction of tweets indicated as containing LI words, compared to other word scoring measures [7]. In [44], Lee et al. applied the NB model after using Equation (3.19) to filter out tweets with no LI words. Here we conduct

more extensive experiments by stratifying tweets based on the absence or presence of LI words, followed by applying our proposed models on both types of tweets.

If tweets without LI words are not meaningful for geolocation, then when geolocating such tweets, the expected ranking performance is equal to geolocating random noise. This means that the ranking of candidate venues is random, with uniform probabilities over all reciprocal rank outcomes. The expected Reciprocal Rank (RR) from random ranking can then be computed as:

$$\mathrm{E}_{\text{Random}}[RR] = \sum_{i=1}^{V} p(\frac{1}{i})\frac{1}{i} = \frac{1}{V}\sum_{i=1}^{V}\frac{1}{i} \tag{3.20}$$

where 'Random' is a model that does random ranking. The expected MRR then follows by averaging over the number of geolocated tweets. Subsequently for tweets without LI words, we shall compare each model's MRR against the expected MRR from random ranking.

Equation (3.19) results in location-indicative scores that are dataset dependent, e.g. V varies across different datasets. Instead of specifying dataset dependent thresholds, we designate the top 5% scoring words as LI words for each dataset. Our experiment setup is similar as in Section 3.6.1, except that test tweets are now stratified into tweets with LI words (denote as set $\mathbb{L}$) and tweets without LI words ($\neg\mathbb{L}$). We compute MRR for each set of test tweets, i.e. MRR($\mathbb{L}$) and MRR($\neg\mathbb{L}$).

Table 3.10 displays the results of the stratified experiment for all datasets. Also included in the table is the expected mean reciprocal rank for the model 'Random', which randomly ranks candidate venues. This regards the tweets as noise, independently of whether they contain LI words or not. Hence MRR values are equal across both tweet sets $\mathbb{L}$ and $\neg\mathbb{L}$. As shown in the table, it is easier to geolocate tweets with LI words than tweets without. Consistently across all models for all datasets, MRR($\mathbb{L}$) is substantially higher than MRR($\neg\mathbb{L}$). For both MRR values, there is also an improving trend as we incorporate more characteristics into the models. From the trend corresponding to MRR($\mathbb{L}$), it is clear that even if we adopt the filtering process of [44] and focus only on geolocating tweets from $\mathbb{L}$, our proposed

Table 3.10: Results for stratified experiment. $\mathbb{L}$ and $\neg\mathbb{L}$ are respectively the set of test tweets with and without LI words, with associated mean reciprocal rank of MRR($\mathbb{L}$) and MRR($\neg\mathbb{L}$). The model 'Random' denotes a random ranking model. Statistics and results shown are averaged over 20 runs.

Dataset	Statistics	Models	MRR($\mathbb{L}$)	MRR($\neg\mathbb{L}$)
SG-SHT	$\|\mathbb{L}\|$=1726.5 $\|\neg\mathbb{L}\|$=899.7 V=10814.5	NB	0.11748	0.05441
		NB+S	0.11755	0.05529
		NB+S+T	0.11841	0.06376
		NB+S+T+U	0.12184	0.06608
		Random	9.123E-4	
SG-TWT	$\|\mathbb{L}\|$=484.65 $\|\neg\mathbb{L}\|$=1577.25 V=2783.55	NB	0.11270	0.03983
		NB+S	0.11352	0.04007
		NB+S+T	0.12154	0.06204
		NB+S+T+U	0.13441	0.07939
		Random	3.057E-3	
JKT-SHT	$\|\mathbb{L}\|$=464.05 $\|\neg\mathbb{L}\|$=511.85 V=2713.75	NB	0.22806	0.05125
		NB+S	0.22954	0.05195
		NB+S+T	0.23153	0.06912
		NB+S+T+U	0.23279	0.07191
		Random	3.126-3	
JKT-TWT	$\|\mathbb{L}\|$=137.25 $\|\neg\mathbb{L}\|$=225.9 V=2713.75 (Based on JKT-SHT venues)	NB	0.19240	0.05279
		NB+S	0.19285	0.05288
		NB+S+T	0.20687	0.09996
		NB+S+T+U	0.20609	0.10354
		Random	3.126-3	

approaches provide consistent improvements.

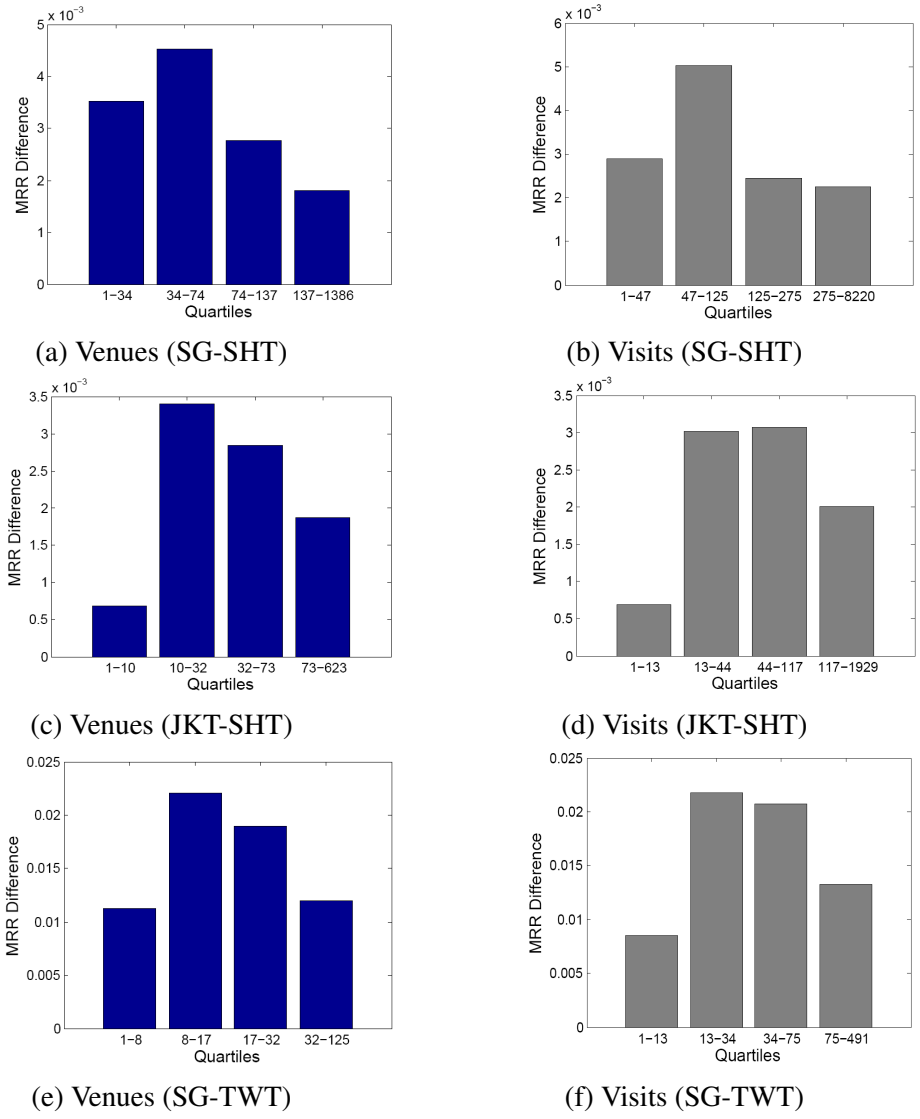
Importantly, Table 3.10 shows that MRR($\neg\mathbb{L}$) for various models is orders of magnitude higher than the random baseline (model Random). For example in SG-SHT, the model NB+S+T+U gives MRR of 0.06608, which is 72.43 times that of 9.123E-4 from random ranking. This implies that we are achieving meaningful geolocation accuracy even for tweets without LI words. Secondly for tweet set $\neg\mathbb{L}$, there is consistent improvement in geolocation accuracy attained from our models. Hence, there are useful information that can be progressively incorporated to geolocate such 'noisy' tweets. Thus, it may not be necessary to discard such tweets, as advocated in [44].

3.6.7 Performance Analysis

The goal of this analysis is to examine how the performance gains attained by NB+S+T+U over NB+S+T vary with the amount of users' location history. To this end, we quantify location history with two criteria: the number of distinct venues that a user had visited (i.e. posted tweets from) and the number of visits that he had accumulated over all venues. For each dataset with multiple runs (SG-SHT, JKT-SHT and SG-TWT), we accumulate test tweets over 10 runs and group them into four bins of equal sizes based on their users' location history, i.e. the first bin corresponds to users with the least history while the last bin corresponds to users with the most history. Since we used four bins, the bins are also referred to as quartiles and we use both terms interchangeably.

For each test tweet, we subtract the reciprocal rank attained by NB+S+T from that obtained from NB+S+T+U. This difference is then averaged over all test tweets within each quartile. Figure 3.3 plots the MRR differences for each dataset, based on the two binning criteria of distinct venues and visit counts. In each sub-figure of Figure 3.3, numbers below each bin indicate the range of location history covered. Also, ties have to be distributed between bins such that the bins are equal-sized. For example, the left most bin of Figure 3.3(a) covers test tweets whose users have distinct venues ranging from 1 to 34 in their location history. Users of test tweets in the second bin have distinct venues ranging from 34 to 74. Thus some users in these two bins share the same distinct venue count of 34.

Across all quartiles for both binning criteria, NB+S+T+U provides gains in MRR over NB+S+T. This is consistent across the three datasets. However the extent of improvement varies across different quartiles. A pattern emerges whereby the largest MRR gains are usually attained over the second and/or third bin from the left. Equivalently, improvement is largest for users with a moderate amount of location history, compared to users with less or more location history. For example, in sub-figure 3.3(f) which corresponds to SG-TWT, MRR gains are largest for the middle two bins, i.e. tweets from users with visit counts ranging from 13 to 75.



(a) Venues (SG-SHT)

(b) Visits (SG-SHT)

(c) Venues (JKT-SHT)

(d) Visits (JKT-SHT)

(e) Venues (SG-TWT)

(f) Visits (SG-TWT)

Figure 3.3: Average differences in MRR between models NB+S+T+U and NB+S+T. Test tweets are divided into bins/quartiles based on the number of distinct venues ('Venues') and the number of visits ('Visits') in their users' location history. The number of binned tweets are 25,898 for SG-SHT, 9429 for JKT-SHT and 19,978 for SG-TWT. For sub-figures (a), (c) and (e), labels on the X-axis represent the range of distinct venues covered by each bin. For sub-figures (b), (d) and (f), X-axis labels are the range of visit counts covered by each bin.

Tweets from users with less ($\leq$ 13) or more ($\geq$ 75) visits in their location history experience less improvement.

Clearly, sparse location history limits the extent of improvement that NB+S+T+U can make. However it can be unintuitive that gains are not monotonic with respect to the amount of location history. One reason is that user behavior is confounded with the amount of location history such that users with more location history are also visiting more venues all over the city and exhibiting a long tailed effect. This may cancel out some of the benefits derived from more location history. For example, it is more difficult to geolocate tweets for a user who spreads his visits over hundreds of venues, compare to another user who is mainly focused on a few dozen venues. In separate studies, we have measured the entropy of the users' distributions over venues. This is found to be higher for users with higher number of visits in their location history. Consistent with this, we also found the number of visits to be highly correlated with the number of distinct venues, with the Pearson's correlation exceeding 0.85 across all three datasets. Thus, users with higher visit counts are also spreading his visits more widely over different venues, possibly making their tweets harder to geolocate.

3.6.8 Case Studies

In this section, we first illustrate examples where sample tweets are geolocated more accurately from the inclusion of temporal venue popularity and user location history for modeling. We then examine cases where the inclusion of location history does not provide improvements. This motivates the case for future work.

3.6.8.1 Temporal Venue Popularity

In Table 3.11, we compare sample tweets geolocated using the models NB+S and NB+S+T. Tweet S1's posting venue is a popular shopping mall in Singapore, <Ion Orchard>. Based on the venue probabilities from model NB+S, the posting venue is placed at position 2 (i.e. $r_{S1} = 2$), behind two other venues, both of which

Table 3.11: Sample test tweets from SG-SHT to illustrate improvement of NB+S+T over NB+S. For each tweet, bolded words are words used for geolocation, i.e. after filtering off stop-words and rare words. ΔRR is the difference in reciprocal rank of the posting venue when one applies NB+S+T versus NB+S. The last two columns r show the ranked position of posting venues obtained under each model (in brackets). Note that the best possible ranked position is 0, corresponding to reciprocal rank of 1. See Equation (3.10).

ID	Time of day	<Posting venue>:Tweet content	ΔRR	r (NB+S)	r (NB+S+T)
S1	16:10:53	<Ion Orchard>: '**Remind** me to never **step** into **ion** on a **Sunday**..'	0.667	2	0
S2	18:15:59	<Golden Village (Yishun)>: '**White House** Down!'	0.4	9	1

are Catholic churches. This can be explained by the fact that tweets posted from churches often contain the term 'Sunday' due to Sunday services. However with the posting time of 16:10:53, i.e. a Sunday afternoon, it is more probable for the tweet to be posted from the mall rather than from churches. This is because malls tend to be more popular than churches on Sunday afternoons. NB+S+T is able to exploit this additional information and assigns higher probability to <Ion Orchard>, making the posting venue the top ranked. The change in reciprocal rank is thus $\Delta RR = \frac{1}{(0+1)} - \frac{1}{(2+1)} = 0.667$.

For S2, the tweet was posted from <Golden Village (Yishun)>, a movie theatre. In this case, the tweet mentioned a movie title and is indicative of movie theatres. Hence for both geolocation models, the top ranking candidate venues for the tweet are all movie theatres. However even in this case, posting time information is still useful since the movie theatres differ in popularities based on time of the day. This may be due to differences in the screening schedule across different theatres. With the exploitation of temporal venue popularity, NB+S+T ranks the actual posting venue at position 1, an improvement of 8 places over that achieved by NB+S.

3.6.8.2 Location History

Table 3.12 lists three sample tweets that have been geolocated using the models NB+S+T and NB+S+T+U. Recall that the latter model assumes that each user is

more likely to post from candidate venues near his other visited venues. Thus for each tweet, we also list the distance from the posting venue to the nearest venue in the posting user's training venues (second column of Table 3.12). Also recall in our experiment setup that each user's set of training venues specifically excludes posting venues of his test tweets.

Tweet S3 is posted from a bus station <Woodlands Regional Bus Interchange>. For S3's user, his nearest venue in the training set is 42.2m away. This turns out to be a subway station <Woodlands MRT Station>. While S3's content is indicative of a bus-related venue, there are many such venues (e.g., bus stops, bus interchanges, etc.) in Singapore. With the tweet content and spatial smoothing, NB+S only manages to rank the posting venue at position 8. By further exploiting a user's location history, NB+S+T+U geolocates S3 with higher accuracy, ranking the posting venue at position 5. This example is intuitive as well for Singapore since many commuters have to transfer between subways and buses when commuting. Thus both subway and bus stations are frequently co-visited. S4 is posted from a Korean restaurant <Manna Story>. The user's nearest training venue is just 49.5m away, which we observed to be a Starbucks cafe. Basically the user is conducting his activities such as dining and drinking at venues around the same area. For S5, the user's nearest training venue is 956.8m away, which is a library venue. In this case, there is ranking improvement even though the nearest venue is relatively far from the posting venue, as compared to the previous two examples. Hence the spatial focus property may still be applicable even if posting venues are sparsely distributed over space.

3.6.8.3 Negative Cases

To motivate further research, we examine negative cases where NB+S+T+U performs worse than NB+S+T. Table 3.13 lists three such test tweets. Tweet S6 is mainly written in Malay and posted from <Universal Studios Singapore>, a theme park. The user is not spatially focused around S6's posting venue, with the nearest venue in his location history being the airport at around 21 km away. On investi-

Table 3.12: Sample test tweets from SG-SHT to illustrate improvement of NB+S+T+U over NB+S+T. Here, ΔRR is the difference in reciprocal rank of the posting venue when one applies NB+S+T+U versus NB+S+T. The second column shows the distance of the posting venue to the next nearest venue visited by the same user. Other notations as in Table 3.11

ID	Dist. to nearest user venue (m)	<Posting venue>:Tweet content	ΔRR	r (NB+S+T)	r (NB+S+T+U)
S3	42.2	<Woodlands Regional Bus Interchange>: '**Hahaha** 168 **bus ride** with **mah homie** - with Eezah'	0.056	8	5
S4	49.5	<Manna Story>: '**Korean food @OldLadyFang**'	0.3	4	1
S5	956.8	<Republic Polytechnic>: '8am **class**'	0.076	14	6

gation, we also found that the user has extremely sparse location history, with the airport constituting the only training venue. This makes it difficult for NB+S+T+U to exploit location history. Compared to the model NB+S+T, performance drops. In particular, one candidate venue near the airport is scored higher than the posting venue, pushing the latter down to ranked position 3. However performance drops is limited since NB+S+T+U also exploits other information such as tweet content and time. In particular, the words 'transformer' and 'mummy' refer to rides at <Universal Studios Singapore> and are indicative of the theme park. Hence although there are numerous other candidate venues nearer the airport, they are not scored higher than the posting venue.

S7 is posted from a border crossing in the west of Singapore. The user is not spatially focused around this venue with his nearest training venue at around 22 km away. In contrast to S6's user, S7's user has substantial location history. However most of his visits are focused on venues in the central and northern part of Singapore, far from where S7 is posted. Thus the user deviates from his usual activity area, which NB+S+T+U is not able to account for. The posting venue is ranked lower at position 24, with some venues from central and north Singapore being scored higher by NB+S+T+U.

Finally, S8 is posted from <Ikea> with the nearest user venue at about 2.7 km away, which is a less drastic case than S6 and S7. This user has some number of

Table 3.13: Sample test tweets where NB+S+T+U results in poorer performance over NB+S+T. Notations as in Table 3.12

ID	Dist. to nearest user venue (m)	<Posting venue>:Tweet content	ΔRR	r (NB+S+T)	r (NB+S+T+U)
S6	21,663.6	<Universal Studios Singapore>: '**Transformer ama mummy nya keren** parah. **Mau lagi**.'	-0.083	2	3
S7	21,875.0	<Tuas Checkpoint (Second Link)>: 'Off to **jb yay**'	-0.293	2	24
S8	2727.7	<Ikea>: '**Meatballs** for **tea**'	-0.0571	4	6

visits in his location history, however he is more active in the central business and shopping area of Singapore, rather than the suburb area where <Ikea> is located. Hence there is insufficient spatial focus around <Ikea> for NB+S+T+U to better geolocate S8.

In short, the cases discussed here highlight scenarios where NB+S+T+U may be inadequate and are grounds for future work. S6 pertains to users with sparse location history, which may be common for tourists or new users and is akin to the cold start problem. A possible mitigation for this is to include geometric weights into the NB+S+T+U model (Equation 3.8) such that the relative importance between tweet content, posting time and location history can be tailored to each user. For new users with little location history, the latter can be assigned smaller importance. S7 and S8 pertain to users who deviate significantly from their usual visitation behavior. This can be due to users seeking novelty [90] and visiting new venues, or users changing their visitation behavior over time. The latter can be for various reasons, e.g. change of workplace, shifting of houses etc. For better geolocation, it will be interesting in future work to incorporate the aspects of novelty seeking and behavior evolution into our models.

3.7 Concluding Remarks

We show that many users have location history in the form of geocoded tweets and that users are spatially focused, with a tendency to visit venues near each other. We

also show the presence of spatial homophily at fine granularities such that venues near each other are more similar in content. Following our empirical studies, we proposed several models for fine-grained geolocation. We achieve large improvements in ranking accuracy with the inclusion of contextual information such as posting time and location history. In our next geolocation track, we shall explore the geolocation of tweets from users without location history.

Chapter 4

Tweet Geolocation: Location, User and Peer Signals

4.1 Introduction

In this chapter [15], we focus on fine-grained geolocation of tweets from users who do not share their location history. Without location history, we are motivated to exploit content history for better geolocation. As per Chapter 3, we solve geolocation as a venue ranking problem. Given a non-geocoded tweet from a city, we rank venues in the city such that highly ranked venues are more likely to be the posting venue.

We propose a model that exploits location, user and peer signals for better geolocation. We list each model aspect below, together with the intuitions (italicized):

- We use **location-indicative weighting** to assign more weights to location-indicative words. *Such words are more important for inferring venues than other words.*
- We expand test tweets via **query expansion** and geolocate the expanded tweet. *Users have habits or constraints, often making repeated visits to the same or related venues.*
- We propose **collaborative filtering** to propagate location information across

users connected via content similarities. *Users with more similar tweet content history may be more similar in their location history.*

The intuitions will be elaborated along with each model aspect in Section 4.3. For user and peer signals, we also justify the associated intuitions with empirical analysis in Sections 4.2.2 and 4.2.3. While each signal leads to some improvement in geolocation, we achieve the best overall results fusing all three signals. We utilize the same datasets discussed in Section 3.2.3 of Chapter 3. Depending on the dataset and metric, we achieve 6% to 40% improvement over the baseline.

4.2 Empirical Analysis

4.2.1 Scenario Study

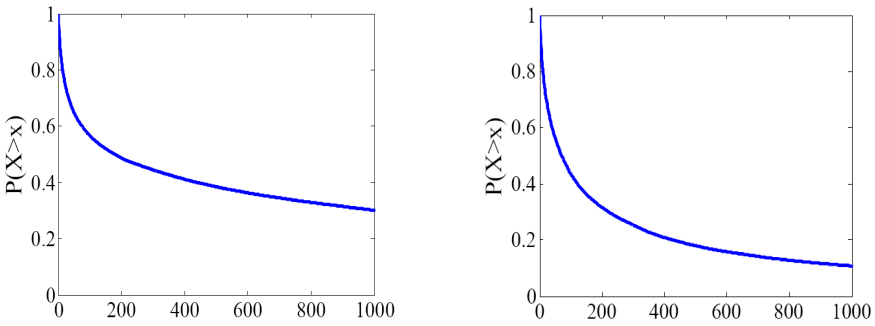
While not explicitly mentioned in the previous chapter, there is clearly a large proportion of users with no location history but substantial content history. In this section, we quantified this proportion. Such users motivate our research focus of improving geolocation by exploiting a user's content history.

We reuse our earlier empirical analysis results from Section 3.3.2 and present statistics that are relevant to the current chapter. Recap that we have randomly sampled 50,000 Twitter users from Singapore for 2014 and from Jakarta for June to Dec 2016, with the condition that each user has posted at least one tweet during the study period. Table 4.1 shows statistics with respect to users with only content history. The count values are higher for Singapore due to the longer study period considered, however the conclusion is the same for both cities.

Table 4.1 shows that there is a substantial proportion of users with no geocoded tweets. For brevity of discussion, denote this set of users as $\mathbb{U}_c$. Users in $\mathbb{U}_c$ have no location history and only content history from their non-geocoded tweets. $\mathbb{U}_c$ constitutes 69.66% of the sampled users for Singapore and 58.04% for Jakarta. These users still generate substantial numbers of non-geocoded tweets, e.g. each user in $\mathbb{U}_c$ have an average of 1820.02 tweets over a one year period for Singapore.

Table 4.1: Statistics for 50,000 sampled users from Singapore (2014) and from Jakarta (June to Dec, 2016).

	Singapore	Jakarta
Total Tweets	136,548,216	20,466,019
Geocoded Tweets	4,394,378 (3.22%)	946,432 (4.62%)
Users with only content history $\mathbb{U}_c$	34,831 (69.66%)	29,018 58.04%
Average tweets for users in $\mathbb{U}_c$	1820.02	558.80



(a) Singapore (b) Jakarta

Figure 4.1: CCDF of average tweet count for $\mathbb{U}_c$ users.

The tweet distributions further illustrate that $\mathbb{U}_c$ users have rich content history. Figure 4.1 plots the Complementary Cumulative Distribution Function (CCDF) of average tweet count for $\mathbb{U}_c$ users. For Singapore, around 55% of users have more than 100 tweets over a one year period, while for Jakarta, the proportion is around 45% over a half year period. Thus even though users in $\mathbb{U}_c$ have no location history, there is substantial content history. How can one exploit this for better geolocation?

We also point out that some approaches are made more complicated by the lack of location history. For example, collaborative filtering is more straightforward given location history. In such cases, to geolocate a tweet from user u, one will exploit other users similar to u in location history. However if u has no location history at all, then visitation similarities can no longer be computed.

4.2.2 User Signals

To tap on user signals, we expand a test tweet with additional words from the same user and then geolocate the expanded tweet. This assumes the user's other tweets

Table 4.2: Repeat Visit Analysis

	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%)

have words which are indicative of the test tweet's venue, i.e. test venue. The presence of such words can be explained by several user behaviour aspects:

- **Repeat visits**: The user may have tweeted from the test venue before and used more informative words.
- **Nearby visits**: The same user tweeting from venues near each other may mention local geographical features. For example, assume we are geolocating a user's first tweet from a quayside restaurant. If he had previously visited neighboring restaurants and mention about the quay, then this will be indicative of the test venue to some extent.
- **Functionally related visits**: The test venue may belong to a functional group of venues that the user frequently tweets from, e.g. nightclubs. Functionally related words, e.g. 'clubbing' will indicate a clubbing test venue with some probability even if the test venue is being visited for the first time.

We empirically study only the aspect of repeat visits in this section. This suffices to motivate the use of user signals. We examine shouts and tabulate the frequencies of repeated visits to venues, on a per user basis. Given user u and venue v, we denote the user-venue tuple as (u, v). We iterate through all shouts and tabulate the frequencies of each tuple. Repeat visits are then simply user-venue tuples that occur more than once. We use the datasets SG-SHT and JKT-SHT for our analysis.

Table 4.2 shows that the proportion of repeat visits is substantial at 22.87% for Singapore and 18.64% for Jakarta. Thus, repeat visits is an established user behavior. This and the earlier discussed behavior aspects imply the possible presence of more informative words beyond the test tweet and justify query expansion. For example, consider a regular visitor to a restaurant. His different tweets from the same

Table 4.3: Query Expansion example.

Query Tweet	"2nd day of orientation"
Query words	{day, orientation}
Sample tweets linked by common word 'day'	"Graduation day" "Last day of exam then holiday!!" ...
Query and Added words with weights	{ (day, 1.0), (orientation, 1.0), (exam,0.113), (graduation, 0.063), (school, 0.048), (holiday,0.045),...}

restaurant may share a common word of 'dinner'. However other words may differ especially if he orders and tweets about different dishes for each visit.

Table 4.3 illustrates an example. The first row displays the test/query tweet which was sent from a school. The user had in fact tweeted from the same venue multiple times. This is shown in the third row containing sample tweets linked by the common word 'day'. If informative words, e.g. 'exam' from these other tweets are added to the test tweet, then one will be able to better geolocate the test tweet.

Thus treating each test tweet as a query, we consider query expansion techniques based on word co-occurrence, to augment each test tweet with additional words. We discuss further details and revisit Table 4.3 in Section 4.3.2.

4.2.3 Peer Signals

For each city, there is a smaller fraction of users not in the set $\mathbb{U}_c$ as shown in Table 4.1, e.g. 30.34% for Singapore. These users have both content history, based on their tweet content and location history, based on their geocoded tweets. We also observed that many of such users have geocoded tweets in the form of Foursquare checkins/shouts, from which we are able to obtain their visitation probabilities over venues. Hence such users provide linkages between content and visitation behavior, which may be useful for geolocating the tweets of users from $\mathbb{U}_c$. The following question then arises: *Are users that are more similar in content history also more similar in their location history?* If this is true, then one can devise collaborative filtering models for geolocation, based on content similarities. This motivates our

empirical analysis.

In the terminology of multi-view learning [81], users with both content and location history have two views: a content view and a venue view. In our empirical analysis, we compute a user representation in each view. First, we treat each user u as a document and aggregate his tweet words in a content TFIDF vector $\mathbf{t}_u$. This represents the user in the content view. Considering the venue view, we reckon that users are more similar if they share common, less popular venues. Venues that are highly popular are somewhat analogous to stop-words and should contribute less to similarity. Clearly, this reasoning supports the TFIDF representation as well. Thus we also compute a venue TFIDF vector $\mathbf{l}_u$ to represent user u in the venue view.

For each user u, we then compute the following:

- In content view, find k nearest neighbors of u based on cosine similarity between $\mathbf{t}_u$ and the vectors of other users. Denote as the set $nb(u)$. Also sample k dissimilar users (i.e. cosine similarity of 0) as non-nearest neighbors. Denote as the set $nnb(u)$.
- Switching to venue view, compute average cosine similarity between u and his content view neighbors: $p_{nb}(u) = \frac{1}{|nb(u)|} \sum_{u_i \in nb(u)} sim(\mathbf{l}_u, \mathbf{l}_{u_i})$. Repeat a similar computation with sampled non-nearest neighbors $nnb(u)$ to obtain $p_{nnb}(u)$.

Over multiple users, we compute the mean venue view similarities for the nearest neighbor and non-nearest neighbor sets: $\overline{p_{nb}}$ and $\overline{p_{nnb}}$. We also count the fraction of cases where $p_{nb}(u) > p_{nnb}(u)$. For such cases, a user's nearest neighbors are more similar on average than non-nearest neighbors. Recap that neighbors are defined based on content view and similarity comparisons are based on the venue view.

We studied 4271 users from Singapore (SG-SHT) and 911 users from Jakarta (JKT-SHT), who have at least 20 shouts. We experiment with $k = 10, 500$, respectively representing small and large nearest neighbor sets. Table 4.4 displays the proportion and the mean similarities.

Table 4.4: Profile analysis for Singapore and Jakarta users.

	$\overline{p_{nb}}$	$\overline{p_{nnb}}$	$p_{nb}(u) > p_{nnb}(u)$
SG-SHT, k=10	1.67E-2	4.76E-3	63.52 %
SG-SHT, k=500	1.19E-2	4.77E-3	77.08%
JKT-SHT, k=10	1.60E-2	2.76E-2	71.79%
JKT-SHT, k=500	9.81E-3	3.06E-3	64.65%

Table 4.4 provides evidence that the content and venue views are correlated. Compare the mean similarity values in the venue view, $\overline{p_{nb}}$ and $\overline{p_{nnb}}$. Consistently, across cities and different k values, content-based nearest neighbors give higher mean similarities than non-nearest neighbors. At a micro-level, the last column also indicates that most users see their content-based neighbors having more similar location history than sampled non-neighbors.

The empirical results indicate that a collaborative filtering approach may be feasible for our geolocation scenario. Basically to better geolocate the tweet of users with only content history, we shall propagate information from users with both content and venue history.

4.3 Models

This section discusses three main model aspects: location-indicative weighting, query expansion and collaborative filtering. Respectively, these aspects incorporate location, user and peer signals. We also describe the fusion framework. For ease of reading, we define notations in an in-line manner.

Similar to our previous geolocation track (see Section 3.4.1), we start with the naive Bayes model [44, 43] and subsequently build on it. For ease of reading, we recap the model here. For notation simplicity, assume that every word in a test tweet $\mathbf{w}$ is unique. We also follow [44, 43] and assume a constant venue probability $p(v)$. Given test tweet $\mathbf{w}$, the probability of venue v is $p(v|\mathbf{w}) \propto \prod_{w \in \mathbf{w}} p(w|v)$. Venues are then ranked by $p(v|\mathbf{w})$.

The naive Bayes model is extremely fast and had been shown to work well. It is

also easily extended to capture our intuitions, as we will discuss next.

4.3.1 Location-Indicative Weighting

This aspect incorporates location signal with the intuition that *Location-indicative words are more important for inferring venues than other words.* Such words indicate one or more venues with high probabilities, i.e. high $p(v|w)$. This concept differs from the venue probabilities over words $p(w|v)$ which is prescribed by the naive Bayes model. For example, a dining venue v may have high probability for the word 'dinner', i.e. high $p(\text{'dinner'}|v)$. However if there are many dining venues, 'dinner' may not necessarily indicate the venue with high probability i.e. low $p(v|\text{'dinner'})$. If a tweet mentions dinner and venue-specific dishes or characteristics, then the latter words are more location-indicative and should be given more importance in contributing to $p(v|\mathbf{w})$.

To capture the discussed intuition, we propose a location-indicative weighting scheme which assigns weights on a continuous scale. Thus there is no necessity to threshold words as location-indicative or not, as some prior work [44] had done. Our weighting scheme can be readily introduced into the naive Bayes model. Interestingly, combining naive Bayes with weighting schemes had been previously explored [89, 28] for improving accuracy in classification tasks. Here we show that for the very different problem of tweet geolocation, the framework is also applicable provided that one uses appropriate weighting schemes. The framework results in a weighted naive Bayes model as:

$$p(v|\mathbf{w}) \propto \prod\nolimits_{w \in \mathbf{w}} p(w|v)^{\beta(w)} \tag{4.1}$$

where $\beta(w)$ is the weight for word w, and is to be determined. In practice, to avoid underflow errors, we use the logarithmic form:

$$\ln p(v|\mathbf{w}) \propto \sum\nolimits_{w \in \mathbf{w}} \beta(w) \ln p(w|v) \tag{4.2}$$

Equation (4.2) also shows that the 'β's are equivalent to weights in a linear mixture of log probabilities. Ranking with both equations (4.2) and (4.1) are equivalent.

Weights. In our context, locations are discrete venues and akin to documents. Hence we can quantify the location-indicative characteristic of words by applying the vector space model. Words that are location-indicative will have large inverse-document frequencies, i.e. they occur in fewer venues. Formally given word w, we set its weight as:

$$\beta(w) = \log(1 + V/df(w)) \tag{4.3}$$

where V is the number of distinct venues and $df(w)$ is the number of venues where w occurs at least once. $\beta(w)$ is computed for all words that meet a minimum support frequency. Rare noisy words are excluded.

We next describe the query expansion portion of the model before covering how location-indicative weighting and query expansion can be combined.

4.3.2 Query Expansion of Test Tweets

This aspect incorporates user signal with the intuition that *users have habits or constraints, often making repeated visits to the same or related venues.* This is intuitive e.g. work or school are usually carried out repeatedly at the same venue, or a user may have favourite hangouts. In Section 4.2.2, we have shown repeated visits to be an established user behavior. We also discussed that users visiting venues that are near or similar in function to the test tweet's venue justifies query expansion as well.

To the best of our knowledge, query expansion has been largely used for document retrieval [71, 21, 85]. Adapting it for the purpose of tweet geolocation is a novel idea. In our context, the query refers to the test tweet. Geolocating a test tweet on its own is difficult due to its short length and missing contextual information. With query expansion, we seek to retrieve words from related tweets to fill in the missing information. Given a test tweet, we iterate through its words and add

co-occurring words from the user's other tweets. The added words are also scored appropriately.

Given query/test tweet **w** from user u, we score candidate words w' which appears in u's other tweets and where $w' \notin \mathbf{w}$. The scoring aims to assess w''s suitability for adding to the query and are designed to reflect the relationship strength to the original query words $w \in \mathbf{w}$. Many scoring schemes exist and we adopt a cosine similarity scheme [21]. For a candidate word w', we compute its average relatedness $\Omega(w', \mathbf{w}; u)$ to the original query words:

$$\Omega(w', \mathbf{w}; u) = \frac{1}{|\mathbf{w}|} \sum\nolimits_{w \in \mathbf{w}} \frac{d_u(w', w)}{\sqrt{d_u(w) d_u(w')}} \tag{4.4}$$

where $d_u(w', w)$ is the count of u's tweets containing both w' and w; and $d_u(w)$ is the count of u's tweets containing w. Equivalently, each summand in Equation (4.4) is the cosine similarity between boolean indicator vectors of w' and w where vector dimensions correspond to u's tweets and vector values indicates presence/absence of words. Intuitively, words that co-occur more are more related. However relatedness is dampened if one or both words are overly common.

We add all words w' with $\Omega(w', \mathbf{w}; u) > 0$ to the query. By definition, the relatedness scores are bounded between 0 and 1. Thus original query words have an implicit weight of 1 while added words are weighted less or at most equal. After expanding the query, we use the relatedness scores as weights in the naive Bayes model. Let **w'** comprise the set of added words for the tweet **w** from user u. We again derive a weighted naive Bayes model:

$$\ln p(v|\{\mathbf{w}, \mathbf{w'}\}, u) \quad \propto \quad \sum_{w \in \mathbf{w}} \ln p(w|v) \; + \; \sum_{w' \in \mathbf{w'}} \Omega(w', \mathbf{w}; u) \ln p(w'|v) \tag{4.5}$$

Given that $0 \leq \Omega(w', \mathbf{w}; u) \leq 1$, Equation (4.5) illustrates that the original query words $w \in \mathbf{w}$ have greatest importance in the naive Bayes model while newly added words $w' \in \mathbf{w'}$ have varying degrees of importance based on how related they are to the query. Table 4.3 illustrates query expansion for a sample tweet. The original

query words are 'day' and 'orientation' (after stop word and rare word exclusion). The last row of the table shows the query words and added words along with their weights after query expansion.

Lastly, we note that since the added words **w'** are from the user's other tweets, we have in fact introduced some user personalization in the model.

4.3.3 Concept Fusion

We envisage both location-indicative weighting and query expansion to be useful for geolocation. This suggests a weighted naive Bayes model that combines both concepts. Intuitively, a word is important only when it is both location-indicative and highly related to the test tweet. Consider the cases where either requirement is not satisfied. If a word is not location-indicative, then it is less useful for geolocation even if it is in the original query or is a highly related word. Conversely, a location-indicative, but unrelated word to the query will introduce noise and hurt geolocation accuracy.

We capture our discussed intuitions by multiplying weights from location-indicative weighting and query expansion. We formulate the weighted naive Bayes model as follows:

$$\ln p(v|\{\mathbf{w}, \mathbf{w'}\}, u) \propto \sum_{w \in \mathbf{w}} \beta(w) \ln p(w|v) + \sum_{w' \in \mathbf{w'}} \beta(w') \Omega(w', \mathbf{w}; u) \ln p(w'|v) \tag{4.6}$$

4.3.4 Collaborative Filtering

Lastly we incorporate peer signals with collaborative filtering, based on the intuition:*Users with more similar tweet content history may be more similar in their location history.* This is also supported by our empirical analysis in Section 4.2.3.

While collaborative filtering has been much used for venue recommendation [8, 48, 45], we adapt it for the different problem of tweet geolocation. We also focus on users with no location history. Hence our work is very much different.

We use collaborative filtering to estimate the user visitation distribution to venue $p(v|u)$. This distribution is then used to personalize the naive Bayes model or the weighted variants. For example, the naive Bayes model can be extended as $p(v|\mathbf{w}, u) \propto p(v|u) \prod_{w \in \mathbf{w}} p(w|v)$. However $p(v|u)$ is not directly computable for users without location history, i.e. set $\mathbb{U}_c$ (See Table 4.1). To overcome this, we use collaborative filtering to propagate visitation information from users not in $\mathbb{U}_c$ to those within. Propagation is via content similarities since content history exist for all users.

Let $\mathbf{t}_u$ be the representation of u in the content view. Many forms of representations are possible. For simplicity, we use the vector space model with users as documents. We represent each user as a TFIDF vector where vector dimensions correspond to words. To compute $p(v|u)$ for user $u \in \mathbb{U}_c$, we first estimate u's visit frequencies to venues from similar users in the content view. Let $nb(u)$ contain k users with location history and who are most similar to u in terms of content cosine similarity. Also denote $\hat{c}(u, v)$ as the estimated frequency from user u to venue v. We compute:

$$\hat{c}(u, v) = \frac{1}{S} \sum\nolimits_{u' \in nb(u)} sim(\mathbf{t}_u, \mathbf{t}_{u'}).c(u', v) \tag{4.7}$$

where $c(u', v)$ is the observed frequency from user u' to venue v, $sim(,)$ is cosine similarity and S sums the similarities for normalization. We then use $\hat{c}(u, v)$ to compute:

$$p(v|u) = \frac{\hat{c}(u, v) + 1}{\sum_{v'} \hat{c}(u, v') + V} \tag{4.8}$$

Lastly, we extend Equation (4.6) with the probability $p(v|u)$:

$$\ln p(v|\{\mathbf{w}, \mathbf{w'}\}, u) \propto \ln p(v|u) + \sum_{w \in \mathbf{w}} \beta(w) \ln p(w|v) + \sum_{w' \in \mathbf{w'}} \beta(w') \Omega(w', \mathbf{w}; u) \ln p(w'|v) \tag{4.9}$$

It can be seen that Equation (4.9) encapsulates all our main model aspects: location-indicative weighting, query expansion and collaborative filtering.

4.3.4.1 Weighted Similarities

In our collaborative filtering approach, we are propagating visitation information over content similarities. We can again apply location-indicative weighting when computing content similarities between users. It is possible to modify the content similarity measure such that visitation and content similarities are more correlated.

The intuition is that *users are more similar in their location history if they are more similar in their usage of location-indicative words*. For example, two users who share common mentions of a restaurant-specific dish will be more likely to have visited the same restaurant, compared to users who only share mentions of 'dinner'. This implies that location-indicative words should be given greater importance in the similarity function between content history. This is easily done by incorporating weights in the cosine similarity between users. Given two users u and u', we compute the weighted cosine similarity as:

$$wsim(\mathbf{t}_u, \mathbf{t}_{u'}) = \frac{\sum_w \beta(w)^2 \mathbf{t}_u(w).\mathbf{t}_{u'}(w)}{\| \mathbf{t}_u \|_2 \| \mathbf{t}_{u'} \|_2} \tag{4.10}$$

where $\beta(w)$ was discussed in Equation (7.3) and $\mathbf{t}_u(w)$ is the w-th dimension of vector $\mathbf{t}_u$. In one variant of collaborative filtering, we replace $sim(,)$ with $wsim(,)$ in Equation (4.7).

4.4 Experiments

We apply the models to a series of fine-grained geolocation experiments. For each dataset (see Section 3.2 of Chapter 3), we conduct 10 runs where each run differs from the others by test and training set partitioning. For SG-SHT, we sample 5000 tweets for testing. For the smaller datasets, SG-TWT and JKT-SHT, we sample 2000 tweets for testing. Tweets not sampled for testing are used for model training. The JKT-TWT dataset (1335 tweets) is too small for training. It is used only in a single run as a test set for the model trained on JKT-SHT.

Recap that we are focusing on the scenario where users of test tweets have no lo-

cation history. To represent this scenario, we process the training tweets as follows: If a user has one or more tweets sampled for testing, we iterate through his training tweets and hide their venues (if any). This is repeated for all users of test tweets. Thus, the training set consists of a mixture of tweets with hidden venue associations due to the owners having some tweets selected for testing, and other tweets whose venue associations are retained. We consider a venue as candidate for ranking only if it is associated with at least 3 training tweets. We also exclude stop words and rare words with frequency < 3. Due to such filtering, the number of test cases per run is less than the number of sampled test tweets. The average number of test cases and venues to rank are reported in Section 4.4.3 which discusses the results of each dataset.

We compare the unweighted naive Bayes model (NB) [44, 43] with variants incorporating different signal combinations (in brackets):

- LW (Location): Location-indicative weighting as indicated in Equation (4.2)
- QE (User): Query expansion as indicated in Equation (4.5)
- LWQE (Location+User): Fusion of query expansion and location-indicative weighting. Refer Equation (4.6).
- LWQE-P (Location+User): LWQE with a Laplace-smoothed global venue popularity model. This is Equation (4.9) with $p(v|u)$ replaced by $p(v)$.
- LWQE-CF (Location+User+Peer): LWQE with a personalized venue distribution $p(v|u)$ from collaborative filtering. Refer Equation (4.9).
- LWQE-LW-CF (Location+User+Peer): Collaborative filtering utilizes location-indicative weighting when computing cosine similarities between users. All other aspects are similar to LWQE-CF.

We also compare with the following baseline models:

- KL: This model [47] assigns scores to venues based on time information and the Kullback-Leibler divergences between the language models of tweets and venues. The scores are then used to rank venues.

- GMM: This model [7] represents each word as a Gaussian mixture over 2-d space, and a test tweet as the product of Gaussian mixtures. Venues are ranked by the probability of the product of Gaussian mixtures generating their coordinates. As in [7], we set the number of clusters to 3.
- TM: [12] proposes topic models to generate Foursquare check-ins and tips. Among them, we use the Udoc model to learn topics for both training tweets associated with and not associated with venues.[1] For each tweet, Udoc generates a user-dependent topic which generates the tweet words. If the tweet is associated with a venue, the venue is generated conditional on the topic. In our experiments, we used 40 topics, which exhibits optimal ranking performance.

4.4.1 Metrics

We again use Mean Reciprocal Rank (MRR) as the primary evaluation metric. This has been defined earlier in Equation (3.10) of Chapter 3.

MRR considers micro-averages. For randomly sampled test cases, popular venues will contribute a larger proportion of tweets, and be more important in determining MRR. In practical applications e.g. geolocating a stream of tweets, this is realistic and there is no reason to avoid this. However for further analysis, we consider the case where all venues are treated as equally important, regardless of their popularities. Thus we introduce a second evaluation metric, denote as Macro-MRR. This is simply the macro-averaged version of MRR. For all test cases from the same posting venue, we average their MRR such that each test venue contributes only one value. We then do a second averaging over distinct test venues. Formally, let $\mathbb{T} = \bigcup_{v=1}^{V} \mathbb{T}_v$ where $\mathbb{T}_v$ is the set of test cases from venue v. We compute:

$$\text{Macro-MRR}(\mathbb{T}) = \frac{1}{V} \sum_{v=1}^{V} \text{MRR}(\mathbb{T}_v) \tag{4.11}$$

[1] We found the Vdoc model to perform worst as it can only model tweets associated with venues. Results omitted.

where MRR($\mathbb{T}_v$) is MRR computed over the set of test cases $\mathbb{T}_v$ and V is the number of distinct test venues.

4.4.2 Result Summary

We summarize the MRR and Macro-MRR results for every dataset in Tables 4.5 and 4.6. We conduct significance testing except for JKT-TWT which has just a single run. For other datasets, the "model 1 $<$ model 2" notation means that model 2 significantly outperforms model 1 by the Wilcoxon signed rank test. In each row, models are arranged from left to right in ascending order of performance. Models that are not significantly different at p-value of 0.05 are grouped in brackets. For example, Table 4.5 shows that for SG-SHT, QE performs better than NB for MRR and the results are statistically significant. For the same metric and data set, LWQE-CF and LWQE-LW-CF perform the best, but they are not statistically different from each other. In rare cases, we list a model twice if it is statistically insignificant against two closest models (in terms of performance), but the two models are significant against each other.

While there are permutations in model ordering, some general trend holds. Comparing against NB, GMM and KL performs poorer while QE and LW performs better. TM's performance is mixed and tends to be poorer for Macro-MRR. For the MRR metric in Table 4.5, QE is better than NB in all datasets while LW outperforms NB in SG-SHT, JKT-SHT and is on-par in SG-TWT. For the Macro-MRR metric in Table 4.6, QE performs better than NB, except for SG-SHT. In the same table, LW outperforms NB in all datasets.

While QE and LW perform relatively well against NB, we achieve more consistent improvement by fusing both approaches. This is illustrated by the model LWQE. In both Tables 4.5 and 4.6, LWQE always outperform NB. It is also typically better than QE or LW alone, lying to the right of both models for most cases. Finally, we achieve the best results with LWQE-CF and LWQE-CF-LW-CF, which combines location-indicative weighting, query expansion and collaborative filter-

Table 4.5: Result Summary for MRR

SG-SHT: {GMM} < {KL} < {TM} < {NB} < {QE} < {LW} < {LWQE} < {LWQE-P} < {LWQE-CF, LWQE-LW-CF}
SG-TWT: {GMM} < {KL} < {NB, LW} < {QE, LWQE} < {LWQE-P,TM} < {TM, LWQE-CF, LWQE-LW-CF}
JKT-SHT: {KL} < {GMM} < {TM, NB} < {LW} < {QE} < {LWQE} < {LWQE-P} < {LWQE-LW-CF, LWQE-CF}
JKT-TWT: KL < GMM < LW < NB < LWQE < LWQE-P < QE < LWQE-CF < LWQE-LW-CF < TM

Table 4.6: Result Summary for Macro-MRR

SG-SHT: {GMM, TM} < {KL} < {QE} < {NB} < {LWQE-P} < {LWQE < LWQE-CF} < {LWQE-LW-CF, LW}
SG-TWT: {GMM} < {KL} < {TM} < {NB} < {QE} < {LW} < {LWQE-P, LWQE} < {LWQE-CF} < {LWQE-LW-CF}
JKT-SHT: {GMM, TM, KL} < {NB} < {QE} < {LW} < {LWQE, LWQE-P}< {LWQE-CF, LWQE-LW-CF}
JKT-TWT: GMM < TM < KL < NB < QE < LW < LWQE-P < LWQE < LWQE-CF < LWQE-LW-CF

ing. Except for one case (Macro-MRR on SG-SHT), these two models are consistently the best performers.

4.4.3 Detailed Results

Tables 4.7, 4.8 and 4.9 display the average MRR and Macro-MRR values over 10 runs for SG-SHT, SG-TWT and JKT-SHT respectively. Table 4.10 displays the results where models are trained on JKT-SHT and tested on JKT-TWT in a single run.

As shown in Tables 4.7 to 4.10, both KL and GMM perform poorly, underperforming even the NB model. As tweets are very short, modeling each with a smoothed language model, as done by KL is inadequate. This in turn affects the computing of KL divergences between the word distributions of tweets and venues. Even with the inclusion of time information, performance is not promising. For GMM, performance is poor as we have to geolocate even tweets where words do not have peaky Gaussian distributions. The topic model TM has mixed performance. It achieves good MRR values for SG-TWT in Table 4.8 and JKT-TWT in Table 4.10,

Table 4.7: SG-SHT results. Bracketed numbers are percentage improvement over NB. Best results are bolded. On average, there are 3248.5 test cases and 9209.1 venues to rank per run.

Models	MRR	Macro-MRR
KL	0.0447 (-54.98%)	0.0254 (-27.22%)
GMM	0.0317 (-68.08%)	0.0119 (-65.90%)
TM	0.0665 (-33.03%)	0.0125 (-64.18%)
NB	0.0993	0.0349
LW	0.1049 (5.69%)	**0.0406 (16.55%)**
QE	0.1008 (1.53%)	0.0339 (-2.72%)
LWQE	0.1066 (7.38%)	0.0402 (15.29%)
LWQE-P	0.1074 (8.20%)	0.0399 (14.46%)
LWQE-CF	0.1088 (9.61%)	0.0402 (15.46%)
LWQE-LW-CF	**0.1090 (9.84%)**	0.0405 (16.12%)

Table 4.8: SG-TWT results. On average, there are 1049.9 test cases and 2672.5 venues to rank per run.

Models	MRR	Macro-MRR
KL	0.0275 (-53.15%)	0.0136 (-28.42%)
GMM	0.0170 (-71.04%)	0.0119 (-37.37%)
TM	0.0666 (13.46%)	0.0151 (-20.53%)
NB	0.0587	0.0190
LW	0.0596 (1.5%)	0.0221 (16.09%)
QE	0.0638 (8.57%)	0.0196 (2.91%)
LWQE	0.0646 (9.96%)	0.0230 (20.71%)
LWQE-P	0.0650 (10.70)	0.0230 (20.60%)
LWQE-CF	0.0674 (14.79%)	0.0236 (23.81%)
LWQE-LW-CF	**0.0675 (14.89%)**	**0.0238 (25.06%)**

but performs poorly for MRR for other datasets, as well as for the Macro-MRR metric. The poor Macro-MRR performance implies that TM is biased to a larger extent towards more popular venues and works poorly when all venues are treated as equally important.

For shouts and pure tweets of both Singapore and Jakarta, LW improves over NB much more substantially for Macro-MRR than MRR. This can be seen by comparing the rows 'LW' and 'NB' in Tables 4.7 to 4.10. For example in Table 4.7, LW improves over NB by 5.69% for MRR. For Macro-MRR, the corresponding improvement is much larger at 16.55%. This trend means that test tweets posted from less popular venues experience relatively larger improvement from location-

Table 4.9: JKT-SHT results. On average, there are 626 test cases and 2492.8 venues to rank per run.

Models	MRR	Macro-MRR
KL	0.0759 (-54.52%)	0.0259 (-27.04%)
GMM	0.1296 (-22.35%)	0.0232 (-34.65%)
TM	0.1657 (-0.72%)	0.0250 (-29.58%)
NB	0.1669	0.0355
LW	0.1691 (1.32%)	0.0403 (13.69%)
QE	0.1716 (2.82%)	0.0372 (4.82%)
LWQE	0.1737 (4.08%)	0.0424 (19.42%)
LWQE-P	0.1760 (5.42%)	0.0425 (19.77%)
LWQE-CF	**0.1778 (6.51%)**	0.0435 (22.58%)
LWQE-LW-CF	0.1777 (6.48%)	**0.0437 (23.06%)**

Table 4.10: JKT-TWT results. There is 1 run with 475 test cases and 4299 venues to rank.

Models	MRR	Macro-MRR
KL	0.0521 (-43.68%)	0.0205 (-8.89%)
GMM	0.0678 (-26.70%)	0.0148 (-34.22%)
TM	0.1130 (22.16%)	0.0157 (-30.22%)
NB	0.0925	0.0225
LW	0.0924 (-0.11%)	0.0284 (26.07%)
QE	0.0959 (3.66%)	0.0266 (17.84%)
LWQE	0.0933 (0.90%)	0.0305 (35.11%)
LWQE-P	0.0956 (3.43%)	0.0301 (33.58%)
LWQE-CF	0.0975 (5.46%)	0.0313 (38.73%)
LWQE-LW-CF	**0.0982 (6.19%)**	**0.0322 (42.86%)**

indicative weighting. As less popular venues are associated with fewer tweets, information is sparse for modeling and it is harder to geolocate their tweets. For such venues, location-indicative words becomes relatively more important for a geolocation model.

We now compare QE to NB. The result for QE is mixed for SG-SHT (Table 4.7). it achieves a small improvement of 1.53% in MRR, but results in a slight dip of 2.72% for Macro-MRR. For SG-TWT (Table 4.8), JKT-SHT (Table 4.9) and JKT-TWT (Table 4.10), QE is more consistent in improving over NB for both metrics. Generally the results indicate room for improvement. We note that query expansion may be noisy and expand a test tweet with words that are less relevant to the test venue. This also depends on the word relatedness function. We have currently

used a relatively simple cosine similarity based function. While more complicated selection mechanisms [29] can be explored, the current query expansion technique is already shown to be useful over the combination of datasets and metrics.

LWQE combines the intuitions of LW and QE, i.e. words are more important if they are *both* location-indicative and highly related to the query. As can be seen, LWQE mostly outperforms LW or QE. In 6 out of 8 dataset-metric combinations, LWQE outperforms both LW and QE. For example in Table 4.8 for SG-TWT, LWQE's Macro-MRR is 0.023, better than QE (0.0196) or LW (0.0221) alone.

For non-collaborative filtering models, LWQE and LWQE-P are best performers. Comparing both models in Tables 4.7 to 4.10, LWQE-P is always better than LWQE for the MRR metric, but not for Macro-MRR. This is expected since LWQE-P utilizes a globally estimated venue distribution $p(v)$ which is related to venue popularity. Venue popularity is however controlled for in Macro-MRR.

We now compare the collaborative filtering model LWQE-CF, against LWQE and LWQE-P, which are best performing models without collaborative filtering. Across Tables 4.7 to 4.10, LWQE-CF always improve on MRR and Macro-MRR against both LWQE and LWQE-P. Hence information propagated from users with visitation history is useful for geolocating the tweets of users with only content history. Since propagation is across content similarities, it also affirms our stated intuition that users that are more similar in content history are more similar in their visitation behavior.

Lastly we note that LWQE-LW-CF is either comparable (Table 4.9) or provides very small improvement (Tables 4.7, 4.8 and 4.10) over LWQE. This is probably due to the fact that other aspects of the model, e.g. collaborative filtering, location-indicative weighting already captures much existing information that are useful for geolocation.

Table 4.11: Sample test tweets from SG-SHT to illustrate location-indicative weighting. Modeled words are italicized and sized proportionately to their assigned weights. r_X denotes the ranked position of posting venue under the model X. ΔRR_X=change in reciprocal rank incurred by model X over the Nb model.

		ΔRR_{LW}	r_{Nb}	r_{LW}
S1	"*Singapore*'s ***Tallest Balloon*** Sculpture."	0.163	26	4
S2	"***Chingay*** *work* last *day*"	0.321	83	2
S3	"*Morning* ***Karaoke***?"	0.389	8	1

4.4.4 Case Studies

We now present some example cases to show the effects of using LW and QE. Table 4.11 displays sample test tweets from SG-SHT, where geolocation is improved by location-indicative weighting, i.e. the model LW. Also displayed is the change in reciprocal rank ΔRR_{LW}, which is computed as $\Delta RR_{LW} = \frac{1}{(r_{LW}+1)} - \frac{1}{(r_{Nb}+1)}$, where r_X denotes the ranked position of posting venue under the model X. Note that the best possible ranked position is 0.

Within each test tweet, modeled words are italicized and sized proportionately to their assigned location-indicative weights. For example in tweet S1, LW assigns largest weights to the words 'Tallest' and 'Balloon'. These are words that appear in relatively fewer venues and are more location-indicative. Compared to not weighting the words, reciprocal rank improves by 0.163, due to the ranked position of the posting venue being elevated from 26 to 4.

Similarly, tweet S2 is better geolocated due to the emphasis on location-indicative words. S2 is posted from a parade preparation venue. 'Chingay' refers to an annual parade event held in the city area of Singapore, thus the word is highly indicative of venues associated with the parade. In S3, emphasizing 'Karaoke' increases the probabilities for venues providing such entertainment activity. Since karaoke venues are relatively few in number, the actual karaoke venue of the test tweet is elevated in rank.

Table 4.12 displays sample test tweets from SG-SHT, where geolocation is improved by query expansion, i.e. the model QE. The user posting tweet S4 had also visited the posting venue multiple times. On its own, S4 is not informative since

Table 4.12: Sample test tweets from SG-SHT. Below each tweet, we list up to 5 added words that are most related to the query, along with their relatedness score. Notations as in Table 4.11.

		ΔRR_{QE}	r_{Nb}	r_{QE}
S4	"Breakfast!" (teddy,0.120), (buying,0.104), (lemak,0.085) (nasi,0.085), (prata,0.070), ...	0.046	75	16
S5	"2nd time spiderman2" (captain,0.25),(america,0.25)	0.008	24	20

there are many dining venues where one can have breakfast. However on another visit to the same venue, the user mentioned having "Nasi Lemak" [2] for breakfast. This is a dish which the test venue is popular for, resulting in the ranking improvement. The last tweet S5 is associated with functionally related visits (Section 4.2.2), instead of repeated visits. The user visited the posting venue (a movie theatre) once to catch the movie "Spiderman", and another theatre to catch "Captain America". Due to query expansion, the latter's title words are added to S5. In this case, the test venue screens "Captain America" as well. Thus the added words are relevant although they arise from a different venue. This improves geolocation since the expanded tweet now describes venue characteristics more effectively.

4.4.5 Parameter Sensitivity Studies

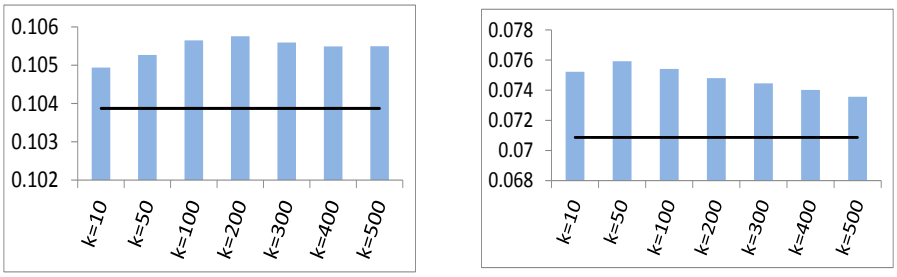
In the collaborative filtering portion, we propagate visitation information from the test user's k nearest neighbors, where similarity is based on content history. In our experiments, we have omitted tuning for k and simply use $k = 500$. In this section, we show that ranking accuracy is not particularly sensitive to the value of k.

Figures 4.2 and 4.3 display the MRR results for Singapore and Jakarta respectively. As Macro-MRR exhibits similar robustness to MRR, we have plotted only the results for the latter. For each figure, the vertical bars are MRR values for the model LWQE-LW-CF with different k values. For comparison, we also plot the MRR value of LWQE as a horizontal line. Recall that LWQE is a model that per-

[2] Malay name for a rice dish cooked with coconut milk

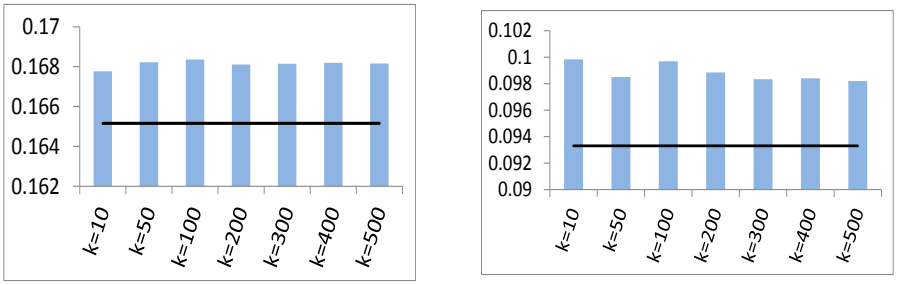
forms well, but does not include any collaborative filtering. Both figures show that MRR is not sensitive to the value of k, remaining in a narrow band as we vary k from 10 to 500. For all k values, LWQE-LW-CF also consistently gives higher MRR than LWQE which is reassuring.

This study shows that collaborative filtering easily improves ranking accuracies, even when one omits the potentially expensive tuning or learning process.



(a) SG-SHT (b) SG-TWT

Figure 4.2: MRR variation with different k values for LWQE-LW-CF. On Singapore datasets.



(a) JKT-SHT (b) JKT-TWT

Figure 4.3: MRR variation with different k values for LWQE-LW-CF. On Jakarta datasets.

4.5 Concluding Remarks

In this chapter, we have focused on fine-grained geolocation for users without location history. We achieve better geolocation of each user's tweets by exploiting three

types of signals from locations, users and peers. Our model is widely applicable in Twitter where many users post frequently but neglect to geocode any of their tweets. Such users have rich content history, but no location history.

Our model aspects capture the signals based on intuitive ideas. Firstly through location-indicative weighting, we place more importance on words that are indicative of venues. Secondly through query expansion, we add potentially informative words to test tweets before geolocation. Lastly, we use collaborative filtering to propagate visitation information from users with location history to those without. Our best model incorporates all three aspects.

Chapter 5

Tweet Geolocation: Same-User Tweets in Temporal Proximity

5.1 Introduction

In this last geolocation track, 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, out of 1000 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 scenario has not been previously studied for fine-grained geolocation.

In our geolocation scenario, we assume that no tweets in the parent sequence are

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

associated with any location coordinates or posting venues. This is a prevalent and realistic scenario due to the scarcity of geocoded tweets. Similar to our previous track in Chapter 4, we 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 [14] of the users to refine candidate posting venues.

Table 5.1: Sample pairs of tweets. Posting venue and time are in brackets. Tweets a1 and a2 are from one user while b1 and b2 are from another user.

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"

Examples. To illustrate the usefulness of parent tweet sequences, Table 5.1 displays tweet pairs, each spanning a short time interval. These tweets are Foursquare shouts pushed to Twitter (See Section 3.2). Tweets a1 and a2 are posted by one user while b1 and b2 are by another user. Consider a1 and a2 which 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 one 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 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 which 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.

[2]Mass Rapid Transit

The research question is then to design robust approaches such that on an overall basis, geolocation accuracy is improved.

5.1.1 Approach.

Given that fine-grained tweet geolocation is akin to document retrieval, certain techniques such as query expansions [85, 5, 71] 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 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.

5.1.2 Challenges.

We have discussed the challenges of fine-grained tweet geolocation earlier in Section 3.1. Although in the current track, we are geolocating tweets contained in sequences, the geolocation scenario remains challenging. This is because 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.

5.1.3 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.

Section 5.2 presents empirical analysis that motivates the query expansion components in our model. Section 5.3 describes our model while Section 5.4 presents

experiment results, along with detailed analysis and case studies. We conclude the chapter in Section 5.5.

5.2 Empirical Analysis

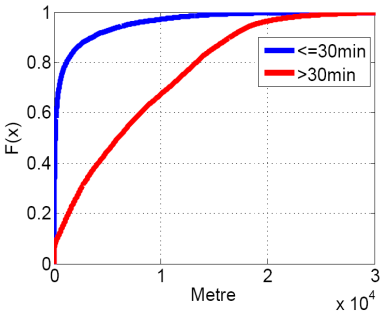
We conduct several empirical studies to verify our intuitions about user behavior and to motivate the design of our models.

5.2.1 Staying Behavior

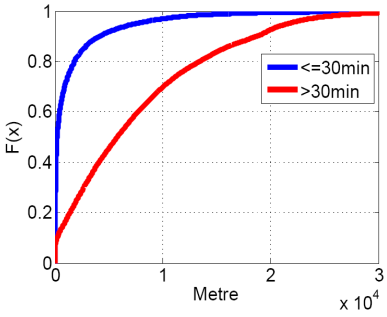
Staying behavior refers to users' tendencies to remain at the same venue or traverse only between nearby venues given a short observed time interval. This is intuitive since firstly, some time interval is required for users to conduct activities at venues, e.g. work, school, dining. Secondly, time is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

Figure 5.1 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 ($\leq$ 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 5.1(a) for Singapore, more than 95% of sample pairs with posting time difference $\leq$ 30 min are posted



(a) CDF (SG-SHT)



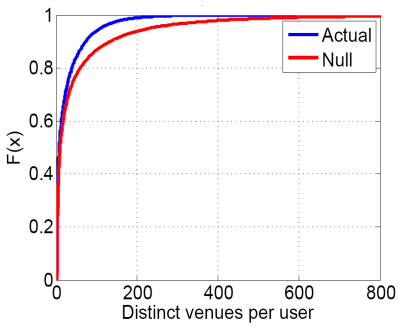
(b) CDF (JKT-SHT)

Figure 5.1: 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 ($\leq$ 30 min); and pairs posted more than 30 minutes apart ($>$ 30 min). X-axis is distance in meters.

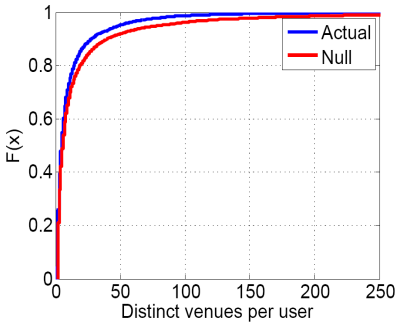
at distances of 10,000 meters or below. In contrast, a similar distance covers only around 64% of sample pairs with posting time difference $>$ 30 min. Figure 5.1(b) shows a similar trend for Jakarta.

5.2.2 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 re-curring activities, e.g. work, or visit venues around a common area or functionality, e.g. movie theatres. This has been discussed earlier in Section 4.2.2, along with an empirical analysis on repeat visits.



(a) CDF (SG-SHT)



(b) CDF (JKT-SHT)

Figure 5.2: CDF for distinct venues per user.

In this section, we provide an additional empirical analysis which quantifies venue revisits by studying if users are focused on a smaller set of venues than ex-

pected. 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, 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 venue 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 venue is one and not meaningful to study.

Figure 5.2 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 5.2(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 5.2(b) for JKT-SHT, the differences between the actual and null model count is smaller, but still 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.

5.3 Models

In this section, we first describe the base model, followed by the proposed query expansion and fusion approaches. In subsequent discussions, *Temporal neighbors* of the target tweet refer to other tweets in its parent sequence.

5.3.1 Base Model (NB)

We use the naive Bayes model from [44, 43] as the base model for query expansion. This model has been used in our earlier geolocation tracks (See Sections 3.4.1 and 4.3). We redefine the model here in a slightly different notation to facilitate the subsequent explanation of query expansion.

The naive Bayes 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:

$$p(w|v) = \frac{f(w,v) + \alpha}{f(.,v) + W\alpha} \tag{5.1}$$

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 which can be tuned or set at 1 for Laplace smoothing.

Given a target tweet **w**, we can rank venues by:

$$p(v|\mathbf{w}) \propto p(v) \prod_{w \in \mathbf{w}} p(w|v)^{c(w,\mathbf{w})} \tag{5.2}$$

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

5.3.2 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 **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 **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 hr.

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 [32]. Let target tweet **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 j-th 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 fulfills the condition $|t - t_j| \leq T, \forall \mathbf{w}_j \in N_T(\mathbf{w}; u)$. We then weigh each word w as:

$$\delta_S(w, \mathbf{w}; u) = c(w, \mathbf{w}) + \sum_{\mathbf{w}_j \in N_T(\mathbf{w};u)} c(w, \mathbf{w}_j) \exp(-S|t - t_j|) \tag{5.3}$$

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 (5.3) 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 (5.3) 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 rightmost term of Equation (5.3) 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:

$$p(v|\mathbf{w}, N_T(\mathbf{w}; u)) \propto p(v) \prod_{\{w:\delta_S(w,\mathbf{w};u)>0\}} p(w|v)^{\delta_S(w,\mathbf{w};u)} \tag{5.4}$$

whereby it suffices to consider the set of words with $\delta_S(w, \mathbf{w}; u) > 0$. Interestingly, Equation (5.4) corresponds to a weighted naive Bayes model, which was previously applied only for classification [89, 28]. 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 the very different problem of tweet geolocation.

5.3.3 Visitation Query Expansion (Visit)

In this part of the model, we reuse the query expansion component presented earlier in Section 4.3.2. Recap that we expand the target tweet with words from the user's other tweets which may be indicative of the posting venue, due to repeat visits to same or similar venues. In this chapter, we termed this component as visitation query expansion to differentiate it from temporal query expansion. We note that visitation query expansion is applicable for geolocating both tweets with and without temporal neighbors. Also recap that in our considered geolocation scenario, the target tweet's user have no location history (see Section 5.1). Hence tweets acting as a source of candidate words are neither geocoded nor associated with any posting venues.

For ease of reading, we reproduce Equation 4.4 here as well as make clear its interpretation as a kernel function. Recap that given a target tweet **w** (i.e. query) from user u, we score candidate words w' which appears in u's other tweets and where $w' \notin \mathbf{w}$ as:

$$\begin{aligned}\Omega(w', \mathbf{w}; u) &= \frac{1}{|\mathbf{w}|} \sum\nolimits_{w \in \mathbf{w}} \frac{d_u(w', w)}{\sqrt{d_u(w) d_u(w')}} \\ &= \frac{1}{|\mathbf{w}|} \sum\nolimits_{w \in \mathbf{w}} \frac{\langle \mathbf{I}_u(w'), \mathbf{I}_u(w) \rangle}{\|\mathbf{I}_u(w')\| \|\mathbf{I}_u(w)\|}\end{aligned} \tag{5.5}$$

where $\mathbf{I}_u(w)$ is a vector of indicator functions for the presence of word w in u's tweets. Equation (5.5) makes it clear that $\Omega(w', \mathbf{w}; u)$ is a normalized form of the dot product kernel, also referred to as the cosine kernel. Subsequently we shall integrate it in our model in the context of multiple kernel learning.

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

$$p(v|\mathbf{w}, \{\mathbf{w}'\}_u) \propto p(v) \prod_{w \in \mathbf{w}} p(w|v)^{c(w,\mathbf{w})} \prod_{\substack{\{w': w' \notin \mathbf{w}, \\ \Omega(w', \mathbf{w}; u) > 0\}}} p(w'|v)^{\Omega(w', \mathbf{w}; u)} \tag{5.6}$$

Equation (5.6) 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.

5.3.4 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* [33]. 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. In order 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 [19]. 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.

5.3.4.1 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 $\Omega(w, \mathbf{w}; u)$. 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 $\Omega(w, \mathbf{w}; u)$, 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:

$$p(v|\mathbf{w}, \{\mathbf{w}'\}_u) \propto p(v) \prod_{\substack{\{w:\delta_S(w,\mathbf{w};u)>0| \\ \Omega(w,\mathbf{w};u)>0\}}} p(w|v)^{\max(\delta_S(w,\mathbf{w};u),\Omega(w,\mathbf{w};u))} \tag{5.7}$$

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 (5.7) 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, $\delta_S(w, \mathbf{w}; u) \geq c(w, \mathbf{w}) \geq \Omega(w, \mathbf{w}; u)$. For words not in $\mathbf{w}$, their final weights depend on which query expansion scheme gives larger weights.

5.3.4.2 Linear Combination (Linear)

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

$$p(v|\mathbf{w}, \{\mathbf{w}'\}_u; \lambda) \propto p(v) \prod_{\substack{\{w:\delta_S(w,\mathbf{w};u)>0| \\ \Omega(w,\mathbf{w};u)>0\}}} p(w|v)^{\lambda\delta_S(w,\mathbf{w};u)+(1-\lambda)\Omega(w,\mathbf{w};u)} \tag{5.8}$$

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.

5.3.4.3 Product Combination (Product)

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

$$p(v|\mathbf{w}, \{\mathbf{w}'\}_u) \propto p(v) \prod_{\substack{\{w:\delta_S(w,\mathbf{w};u)>0| \\ \Omega(w,\mathbf{w};u)>0\}}} p(w|v)^{\delta_S(w,\mathbf{w};u)\times\Omega(w,\mathbf{w};u)} \tag{5.9}$$

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.

5.3.5 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 [52] 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 [72].

Given a HMM model Θ, denote $p(v|\mathbf{w}, N_T(\mathbf{w}; u), \Theta)$ as the marginalized venue probability. The transition probabilities between venues are estimated 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 which should improve geolocation performance. Thus we stack our 'Max'-based model

over the HMM-based approach to exploit all information facets. Specifically, we compute:

$$p(v|\mathbf{w}, \{\mathbf{w}'\}_u) \propto p(v|\mathbf{w}, N_T(\mathbf{w}; u), \Theta) \prod_{\substack{\{w:\delta_S(w,\mathbf{w};u)>0| \\ \Omega(w,\mathbf{w};u)>0\}}} p(w|v)^{\max(\delta_S(w,\mathbf{w};u),\Omega(w,\mathbf{w};u))} \tag{5.10}$$

Equation (5.10) is of similar form to Equation (5.7), 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 $p(v)$.

5.3.5.1 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 5.3.3). 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 apply.
- $\mathbf{w}$ 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 (5.10) reduces to:

$$p(v|\mathbf{w}, \{\mathbf{w}'\}_u) \propto p(v|\mathbf{w}, N_T(\mathbf{w}; u), \Theta) \prod_{\{w:\delta_S(w,\mathbf{w};u)>0\}} p(w|v)^{\delta_S(w,\mathbf{w};u)} \tag{5.11}$$

- $\mathbf{w}$ has no temporal neighbors, but has common words with tweets from $\{\mathbf{w}'\}_u$:
  Markov chain construction and temporal query expansion are no longer ap-

plicable. In this scenario, HMM-Max is equivalent to a naive Bayes model weighted only with visitation query expansion. Equation (5.10) reduces to Equation (5.6).

- **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 (5.2). Hence in the worst case scenario of highly sparse information, performance will be comparable to applying the models from [44, 43].

5.3.6 Computational Complexity

We first examine the computational complexity of query expansion. Given a targeted tweet **w** from user u, the complexity of temporal query expansion depends on the length of **w**'s parent sequence and can be written as $\mathcal{O}(|N_T(\mathbf{w}; u)|)$. For visitation query expansion, the complexity depends on the number of other tweets from u which contains words from **w**, denoted as $D(\mathbf{w}; u)$. These other tweets can be retrieved efficiently in $\mathcal{O}(|D(\mathbf{w}; u)|)$ time using an inverted index [91], 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 **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 $\mathcal{O}(|D(\mathbf{w}; u)|)$.

For incorporating sequential information, the main computation complexity lies in the forward-backward algorithm. To geolocate **w** from user u, the basic algorithm has a complexity of $\mathcal{O}(|N_T(\mathbf{w}; u)| \times V^2)$, whereby V is the number of venues. However in practice, one need 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 $\mathcal{O}(|N_T(\mathbf{w}; u)| \times \gamma \times V^2)$ 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.

5.4 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 (see Section 3.2 of Chapter 3), we conduct 20 runs which differ by randomly partitioning tweets into 3 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 3 training tweets as candidate venues. Test tweets with posting venues not among the candidate venues are discarded. 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 5.4.1.

We compare the following models:

- **KL**: This approach [47] 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.
- **KDE**: This method [39] 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 (5.2) with Laplace smoothed word probabilities.
- **Temporal**: Temporal query expansion as shown in Equation (5.4).
- **Visit**: Visitation query expansion as shown in Equation (5.6).
- **Max**: The max combination scheme which combines the temporal and visitation query expansion approaches. See Equation (5.7)
- **Linear**: Temporal and visitation query expansion combined via linear combination. See Equation (5.8)
- **Product**: Both query expansion approaches combined via product combination. See Equation (5.9)
- **HMM**: This is the approach from [52] 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 (5.10)

There are other fine-grained geolocation methods in the literature which are not

considered here, largely due to additional assumptions about users and social media platforms [14, 6].

We use two parent time window settings: T=1 hr and T=0.5 hr to define temporal neighbors. To recap the purpose of T, if T=1 hr, then any training tweet posted by the user within 1 hr (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 (5.3)).

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, 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. Such tweets are also used to estimate the transition probabilities for HMM-based models. We use Laplace smoothing for $p(w|v)$ and tune other parameters to optimize MRR on the tuning set. For models utilizing temporal query expansion, 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.

As per our earlier experiments, we use MRR and Macro-MRR as the evaluation metrics. Refer to Section 4.4.1.

Table 5.2: SG-SHT results averaged over 20 runs. Bracketed numbers are percentage improvement over NB. On average for T=1 hr, there are M=1239.5 test tweets and V=10539.4 venues to rank per run. For T=0.5 hr, M=1136.8, V=10959.3 on average.

Models	MRR (T=1 hr)	Macro-MRR (T=1 hr)	MRR (T=0.5 hr)	Macro-MRR (T=0.5 hr)
KL	0.03057 (-56.19%)	0.02170 (2.94%)	0.02861 (-59.62%)	0.02027 (-3.93%)
KDE	0.05684 (-18.54%)	0.02037 (-3.37%)	0.05567 (-21.44%)	0.01937 (-8.20%)
NB	0.06978	0.02108	0.07086	0.02110
Temporal	0.07036 (0.83%)	0.02145 (1.76%)	0.07220 (1.89%)	0.02259 (7.06%)
Visit	0.07145 (2.39%)	0.02113 (0.24%)	0.07257 (2.41%)	0.02135 (1.18%)
Max	0.07114 (1.95%)	0.02152 (2.09%)	0.07314 (3.22%)	0.02230 (5.69%)
Linear	0.07108 (1.86%)	0.02123 (0.71%)	0.07326 (3.39%)	0.02202 (4.36%)
Product	0.07100 (1.75%)	0.02260 (7.21%)	0.07243 (2.22%)	0.02332 (10.52%)
HMM	0.07401 (6.06%)	0.02380 (12.90%)	0.07539 (6.39%)	0.02496 (18.29%)
Max-HMM	0.07420 (6.33%)	0.02362 (12.05%)	0.07564 (6.75%)	0.02484 (17.73%)
HMM-Max	**0.08122 (16.39%)**	**0.03053 (44.83%)**	**0.08110 (14.45%)**	**0.03074 (45.69%)**

5.4.1 Results

Tables 5.2, 5.3 and 5.4 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 do not indicate statistically significant differences at p-value of 0.05. Across Tables 5.2 to 5.4, HMM-Max is consistently the best or among the best models. For all models, Macro-MRR 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 5.2 to 5.4 show that KL and KDE performs substantially

Table 5.3: SG-TWT results averaged over 20 runs. On average per run, M=1290.7, V=1914.2 for T=1 hr, and M=1296.6, V=1912.1 for T=0.5 hr

Models	MRR (T=1 hr)	Macro-MRR (T=1 hr)	MRR (T=0.5 hr)	Macro-MRR (T=0.5 hr)
KL	0.02837 (-63.14%)	0.01411 (-14.28%)	0.02947 (-62.30%)	0.01543 (-7.22%)
KDE	0.05141 (-33.20%)	0.01607 (-2.37%)	0.05278 (-32.47%)	0.01703 (2.41%)
NB	0.07696	0.01646	0.07816	0.01663
Temporal	**0.08399 (9.13%)**	0.01873 (13.79%)	0.08496 (8.70%)	0.01881 (13.11%)
Visit	0.07851 (2.01%)	0.01654 (0.49%)	0.07951 (1.73%)	0.01669 (0.36%)
Max	**0.08408 (9.52%)**	0.01845 (12.09%)	**0.08563 (9.56%)**	0.01880 (13.05%)
Linear	0.08383 (8.93%)	0.01819 (10.51%)	0.08487 (8.59%)	0.01827 (9.87%)
Product	0.07805 (1.42%)	0.01723 (4.68%)	0.07947 (1.68%)	0.01775 (6.74%)
HMM	**0.08429 (9.25%)**	0.01874 (13.85%)	0.08529 (9.12%)	0.01890 (13.65%)
Max-HMM	**0.08483 (10.23%)**	0.01926 (17.01%)	**0.08541 (9.28%)**	0.01963 (18.04%)
HMM-Max	**0.08486 (10.27%)**	**0.02020 (22.72%)**	**0.08604 (10.08%)**	**0.02102 (26.40%)**

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 [49], word distributions are learnt at a grid cell level and are sub-optimal for fine-grained geolocation.

HMM [52] out-performs the NB model for both MRR and Macro-MRR for datasets SG-SHT (Table 5.2) and SG-TWT (Table 5.3). For JKT-SHT (Table 5.4), it is on par for MRR and performs better for Macro-MRR. Thus sequential information exploited by HMM provides useful information, even when one omits any query expansion. However as will be subsequently discussed, query expansion will

Table 5.4: JKT-SHT results averaged over 20 runs. On average per run, M=297.6, V=2520.8 for T=1 hr, and M=277.3, V=2795.6 for T=0.5 hr

Models	MRR (T=1 hr)	Macro-MRR (T=1 hr)	MRR (T=0.5 hr)	Macro-MRR (T=0.5 hr)
KL	0.05735 (-53.23%)	0.02714 (-21.31%)	0.05028 (-52.98%)	0.02474 (-29.66%)
KDE	0.07906 (-35.53%)	0.02370 (-31.28%)	0.07133 (-33.29%)	0.02375 (-32.47%)
NB	0.12263	0.03449	0.10693	0.03517
Temporal	0.12482 (1.79%)	0.03579 (3.77%)	0.10878 (1.73%)	0.03671 (4.38%)
Visit	0.12336 (0.60%)	0.03475 (0.75%)	0.10850 (1.47%)	0.03538 (0.60%)
Max	0.12543 (2.28%)	0.03598 (4.32%)	0.10928 (2.20%)	0.03623 (3.01%)
Linear	0.12445 (1.48%)	0.03551 (2.96%)	0.10821 (1.20%)	0.03582 (1.85%)
Product	0.12373 (0.90%)	0.03524 (2.17%)	0.10710 (0.16%)	0.03540 (0.65%)
HMM	0.12276 (0.11%)	0.03705 (7.42%)	0.10662 (-0.29%)	0.03747 (6.54%)
Max-HMM	0.12628 (2.98%)	0.03961 (14.85%)	**0.11100 (3.81%)**	0.04018 (14.25%)
HMM-Max	**0.12825 (4.58%)**	**0.04144 (20.15%)**	**0.11182 (4.57%)**	**0.04160 (18.28%)**

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 5.2), 'Visit' achieves small, but statistically significant improvement over 'NB' for MRR for both T settings, while being on par for Macro-MRR. 'Temporal' improves slightly over 'NB', except for T=1 hr where the MRR gains are not significant. For SG-TWT (Table 5.3), 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 5.2.1, 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 5.4), both query expansion approaches provide

consistently small improvements over different T settings and metrics, except for one case: Macro-MRR for 'Visit' at T=0.5 hr. Nonetheless, 'Visit' still outperforms 'NB' under MRR.

Fusion Approaches. On the whole, 'Max' performs consistently well over the different datasets and is the more robust fusion approach. We compare the fusion approaches: 'Max', 'Linear' and 'Product' over different datasets. For MRR on SG-SHT (Table 5.2), the performance of 'Max' is statistically equivalent with 'Linear' and 'Product' for both T=1 hr and 0.5 hr. For Macro-MRR on SG-SHT, 'Product' performs better than other fusion approaches. However, it performs poorly on other datasets. For SG-TWT (Table 5.3), 'Max' is the best fusion approach, while 'Product' does poorly. 'Linear' is slightly inferior to 'Max' even though the former incurs more tuning costs. For JKT-SHT (Table 5.4), 'Max' again outperforms the other two fusion approaches.

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 5.2), 'Visit' performs better than 'Temporal' for MRR while for Macro-MRR, 'Temporal' performs better. With 'Max' fusion, we obtain a more robust model, achieving MRR on par with 'Visit' and Macro-MRR 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 hr, '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 in-

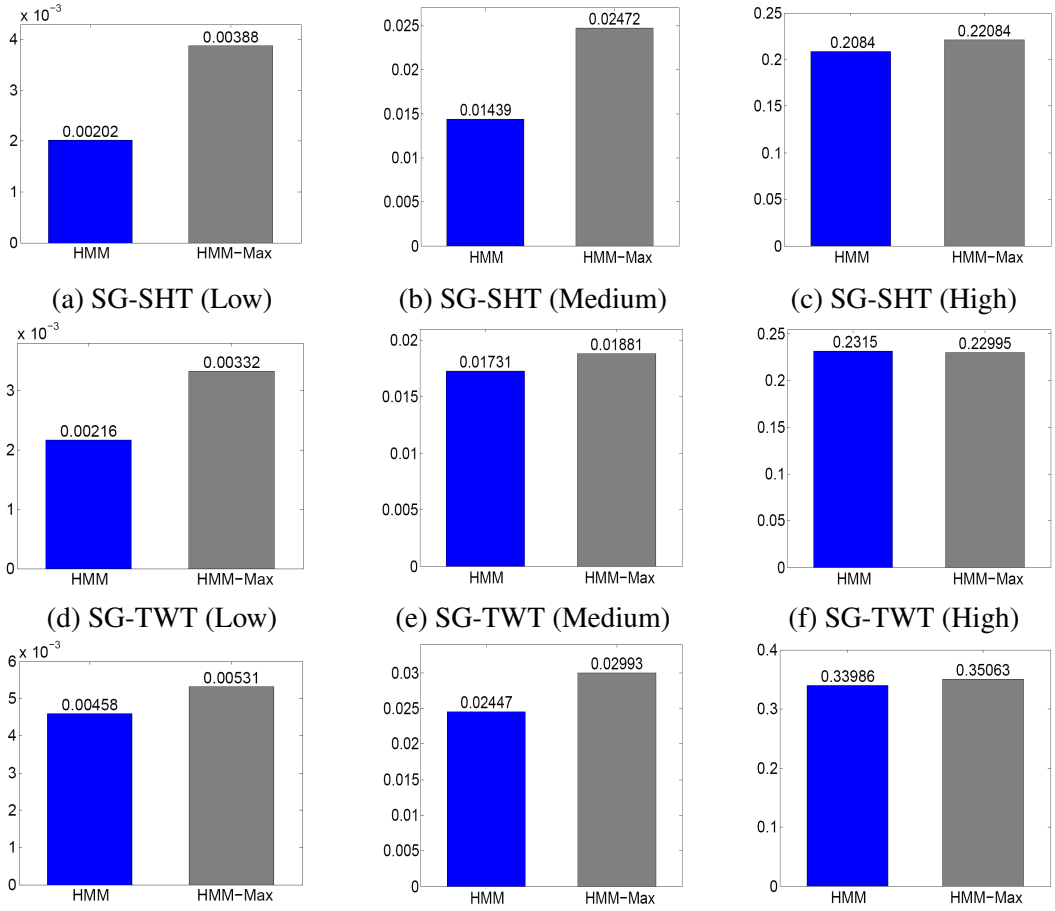
formation and HMM-Max is able to exploit this. Over the base model, performance gains for Macro-MRR are especially impressive, ranging from around 20% for SG-TWT and JKT-SHT (Tables 5.3 and 5.4) to more than 40% for SG-SHT (Table 5.2). For MRR, gains range from 4+% for JKT-SHT to 10+% for SG-SHT and 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 5.2), 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.

5.4.2 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 hr and compute the average bin-specific MRR. Figure 5.3 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 5.3 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 5.3(c) for SG-



(a) SG-SHT (Low) (b) SG-SHT (Medium) (c) SG-SHT (High)

(d) SG-TWT (Low) (e) SG-TWT (Medium) (f) SG-TWT (High)

(g) JKT-SHT (Low) (h) JKT-SHT (Medium) (i) JKT-SHT (High)

Figure 5.3: Average MRR of HMM (blue) and HMM-Max (gray) for test tweets from venues of different popularities. Each row corresponds to a dataset.

SHT tweets from high popularity venues, HMM-Max achieves an average MRR of 0.22, much higher than 0.0039 in Figure 5.3(a) for low popularity venues. For JKT-SHT, the corresponding figures for HMM-Max are 0.35 in Figure 5.3(i) versus 0.0053 in Figure 5.3(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 5.3(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

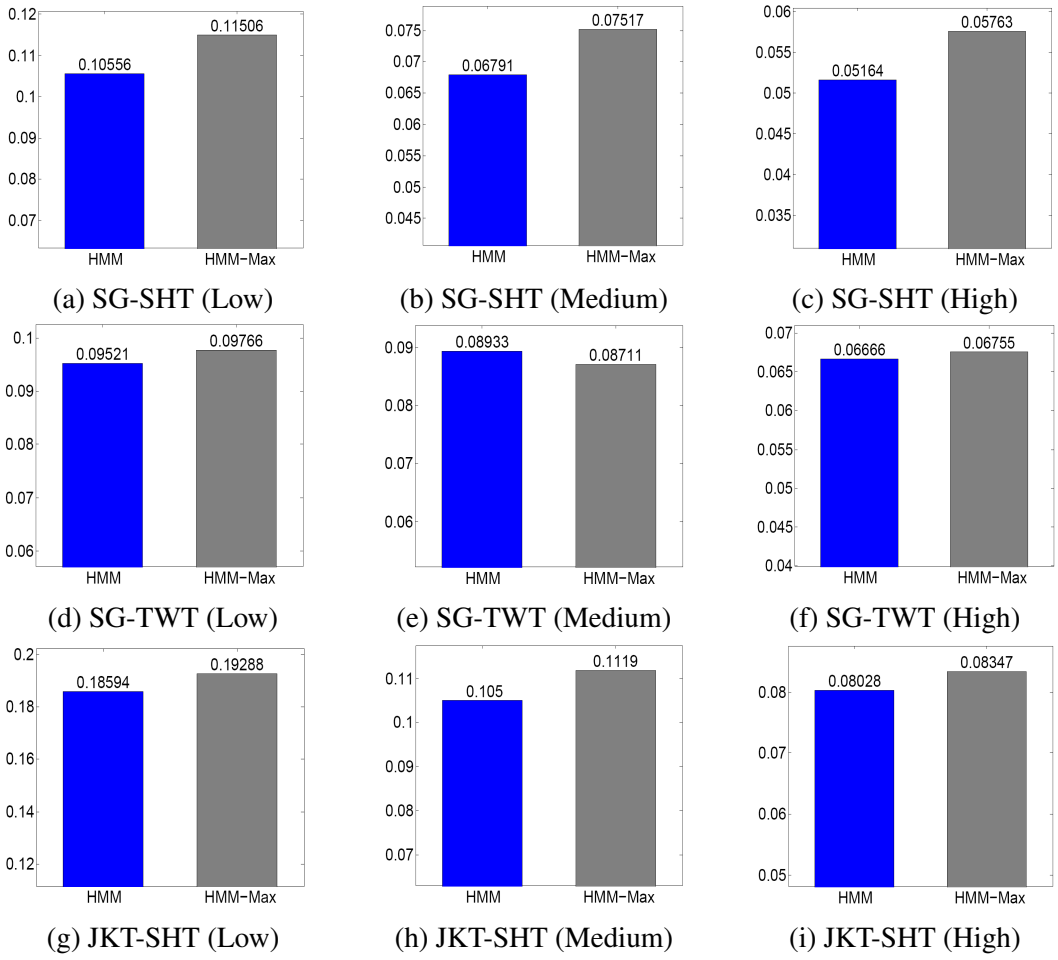
high popularity venues in Figure 5.3(c), the corresponding relative improvement is around 5.6%. For other datasets, the same trend persists although the magnitude of relative improvement differs. For example, for JKT-SHT, HMM-Max's relative improvement over HMM is less drastic than SG-SHT for low popularity venues, i.e. 15.9% in Figure 5.3(g). However, relative improvement is even smaller for high popularity venues at 3.17% in Figure 5.3(i). We also note that for SG-TWT, HMM-Max outperforms HMM for low and medium popularity venues (See Figures 5.3(d),(e)), but is on par for high popularity venues in Figure 5.3(f).

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 5.3(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.

5.4.3 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 visits. 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 [90] or to project an interesting image of themselves on social media [65].

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



(a) SG-SHT (Low) (b) SG-SHT (Medium) (c) SG-SHT (High)

(d) SG-TWT (Low) (e) SG-TWT (Medium) (f) SG-TWT (High)

(g) JKT-SHT (Low) (h) JKT-SHT (Medium) (i) JKT-SHT (High)

Figure 5.4: Average MRR of HMM (blue) and HMM-Max (gray) for test tweets from users with different number of distinct venues in training tweets.

scenario at the start of this chapter. 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 3 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 hr and average the bin-specific MRR across the runs.

Figure 5.4 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 5.4(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 5.4(b) and finally to

0.0576 for the 'High' bin in Figure 5.4(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 5.3, T=1 hr, 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.

5.4.4 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 which illustrate the usefulness of query expansion and Max combination. We then examine negative cases, which are grounds for future work.

5.4.4.1 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 5.5 displays geolocation cases from SG-SHT with T=1 hr. These are extracted from sample runs of the main experiment in Section 5.4. Each case consists of a pair of tweets: a test tweet (bolded) and its

Table 5.5: Sample geolocation cases/tweet sequences from SG-SHT. 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.

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), (flying, 0.48), (boarding, 0.35), (au, 0.5), (revoir, 0.5), (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

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 5.5, 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 which 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 (5.3)). 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 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 which 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.

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.

5.4.4.2 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 5.6 illustrates some examples.

Our experiment results and previous case studies have shown temporal neighbors to be generally useful. However there exist cases where they have no effect or worsen geolocation accuracy. For case E in Table 5.6, both tweets are from adjacent shopping malls. Incidentally, the temporal neighbor E1 provides no additional use-

Table 5.6: 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

ful 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 paper.

Case F in Table 5.6 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[3] 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 5.6. As can be seen, the added words included 'reached', 'flight' etc., 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 [90]. 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.

5.5 Concluding Remarks

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.

[3] A park venue: Gardens by the Bay

By ‘Max’ fusion of both query expansion approaches and stacking with HMMs, we achieve an effective and robust model for geolocation. In future work, it will be interesting to explore how other behavioral aspects such as cyclical visits and novelty seeking can be modeled to improve geolocation.

Part II

Semantic Context Recovery

Chapter 6

Explicit Entity Linking

6.1 Introduction

We frame the recovery of semantic context as the entity linking problem. In this chapter, we explore Explicit Entity Linking (EL) whereby we link mentions of named entities in tweets to the correct knowledge base entity. This is challenging as tweets are short. Thus mentions arise in short documents, which lack substantial content or context for deriving features. The sparsity of information motivates the use of collective linking, i.e. exploiting information from multiple tweets to link mentions in a single tweet. Prior work [38, 79] had considered collective linking over multiple tweets from the same user, and tweets linked by common terms or hashtags. In this work, we focus on the orthogonal aspects of space and time for collective linking. This is motivated by observations of tweeting behaviour with respect to events and geographical effects.

Our main contribution is a new collective entity linking method [13] to exploit event and geographical effects. We connect tweets close in space and time to form a tweet graph, and define a novel objective function over the graph. This mitigates the challenge of entity linking for overly brief content. In addition, we introduce a comparison-based evaluation approach in Section 6.4 that mitigates challenges in evaluation.

6.2 Motivating Characteristics

6.2.1 Event Effects

Tweets may be event related [2]. When tweet-worthy events occur, users may tweet about related entities, leading to an excess of related mentions in a space-time cube, i.e. a certain time period defined over a geographical area. Within a space-time cube, we can conduct collective linking and share linkage information across tweets. For example, the following are two actual tweets close in space and time: "*Stones*" and "Waiting for @*RollingStones* to come on stage so we can rock out *Singapore*". Consider the mentions in italics. The first tweet has insufficient context for linking *Stones*. The second tweet's mentions can be linked with much less ambiguity, since *RollingStones* refers to the band entity 'The Rolling Stones' with high probability [80]. Given the space-time proximity of both tweets, one can now use the second tweet's results to link the first tweet's *Stones* to the band with much more certainty.

6.2.2 Geographical Effects

Besides events, locations also affect tweeting behaviour. Certain entities may be more prevalent and mentioned more frequently at certain locations. Thus we can exploit geographical effects by collectively linking tweets that are close in space. For example, compare the following two tweets with mentions in italics: "*MBS* #throwback", "Standby for SHOWTIME! @ *Marina Bay Sands*". *MBS* in the first tweet is the surface form for many possible entities. The probability that it refers to 'Marina Bay Sands', a Singapore tourist attraction, is extremely low [80] at 0.000155. However if the second tweet with unambiguous mentions to 'Marina Bay Sands' occurs spatially near the first tweet, then it is much more plausible for the latter to be mentioning the same entity. Both event and geographical effects are often coupled due to events at Points of Interest (POI), e.g. concerts at a tourist attraction.

6.3 Approach

6.3.1 System Architecture

Our system architecture comprises of **Pre-processing**, **Local linking** and **Collective linking**. Given a set of tweets for entity linking, the first pre-processing step is mention extraction with an NER tool. The process is often noisy with mentions being omitted or extracted partially. To mitigate this, we apply TweetNLP [64], which was specially developed for tweets. Next, for each extracted mention, we use the Google lexicon [80] to identify candidate Wikipedia entities. The lexicon lists possible mentions $\{m\}$ for each entity e along with the occurrence probability $p(e|m)$ derived from web hyperlinks.

In local linking, mentions to entities are linked individually for each tweet, without considering information from other tweets. We implemented two local linking methods: TAGME [27] and Loclink, introduced in Section 6.3.2. Local linking can be used to initialize the entity assignments for collective linking.

In collective linking, each mention in a tweet is linked using information within that tweet and from other tweets. Collective linking comprises three steps:

- **Tweet Graph Construction**: We first construct a graph that connects tweets by spatio-temporal proximity. The tweet graph is used to propagate information. Section 6.3.3 describes the construction process.
- **Initialization**: This means assigning an initial entity to each mention for subsequent refinement. This can be done using the results from local linking or with some other heuristics. We have opted for the former.
- **Optimization**: We define an objective function over the tweet graph and search for entity assignments to optimize it. Refer to section 6.3.3.

6.3.2 LocLink: A Local Linking Method

Local linking processes each tweet individually, assigning entities that are semantically related to each other to make each tweet coherent. To quantify coherence, we adopt the semantic relatedness measure proposed in [58]. Consider entity e_a. Denote other entities with outgoing links to e_a as the set $I(e_a)$. Equivalently, regard e_a as having $|I(e_a)|$ incoming neighbors. For a pair of entities e_a, e_b with overlapping incoming neighbors, semantic-relatedness is then computed as:

$$SR(e_a, e_b) = 1 - \frac{log(max\{|I(e_a)|, |I(e_b)|\}) - log|I(e_a) \cap I(e_b)|}{log(|W|) - log(min\{|I(e_a)|, |I(e_b)|\})} \tag{6.1}$$

where $I(e_a) \cap I(e_b)$ are entities which link to both e_a, e_b in Wikipedia and W is the total number of Wikipedia entities. If $I(e_a) \cap I(e_b) = \emptyset$, we set $SR(e_a, e_b) = 0$.

Intra-tweet Coherence Let d_i represent the i-th tweet containing $|m_i|$ mentions with set of linked entities $\mathbf{e}_i$. Also let m_{ia} be the a-th mention of d_i, with corresponding linked entity e_{ia}. We define the intra-tweet coherence as average semantic relatedness between its assigned entities:

$$C(d_i, \mathbf{e}_i) = \frac{1}{0.5|m_i|(|m_i| - 1)} \sum\nolimits_{a=1}^{|m_i|} \sum\nolimits_{b>a}^{|m_i|} SR(e_{ia}, e_{ib}) \tag{6.2}$$

Maximizing intra-tweet coherence makes each tweet as coherent as possible. However assigned entities can be rather obscure or rare. Hence a prior $p(e|m)$ is usually included [77, 51, 79] to favor more popular entities. In fact using only the prior for entity linking is a surprisingly strong baseline [50, 74], while including the notion of coherence improves performance further. We use the prior from [80] and define the objective function for tweet d_i as:

$$Q_i(d_i, \mathbf{e}_i) = \xi.C(d_i, \mathbf{e}_i) + \frac{\tau}{|m_i|} \sum\nolimits_{a=1}^{|m_i|} p(e_{ia}|m_{ia}) \tag{6.3}$$

where ξ and τ are combination weights. In the unsupervised setting, we simply let $\xi = \tau$ and assign entities to maximize Q_i. For single-mention tweets, coherence is

undefined and we simply assign the entity with the highest prior to the mention. We call the above local linking method as **LocLink**.

6.3.3 Collective Linking in Space and Time

Inter-tweet coherence. For collective linking, we exploit the fact that different tweets close in space and time may be related to the same event or have a common geographical effect, e.g. mentioning a common location. Therefore we expect some of the tweets to be *inter-coherent*. For computational efficiency, we shall only consider tweet pairs. Given tweets d_i and d_j with respective linked entity sets $\mathbf{e}_i$ and $\mathbf{e}_j$, we define the inter-tweet coherence as:

$$C(d_i, d_j, \mathbf{e}_i, \mathbf{e}_j) = \frac{1}{|m_i|.|m_j|} \sum\nolimits_{a=1}^{|m_i|} \sum\nolimits_{b=1}^{|m_j|} SR(e_{ia}, e_{jb}) \tag{6.4}$$

Tweet Graph Construction. Denote tweet d_i's timestamp as t_i and its location as l_i. In the simplest graph building scenario, we first retrieve geocoded tweets from a desired time interval and geographical area. For convenience, we call this a space-time cube although the geographical area need not be rectangular. For every pair of tweets d_i and d_j, we connect them if $|t_i - t_j| \leq \delta_t$ and $dist(l_i, l_j) \leq \delta_d$, where δ_t and δ_d are the respective thresholds for temporal and spatial proximities, and $dist()$ measures geographical distance.

We can relax the spatial requirement to include non-geocoded tweets. This assumes that non-geocoded tweets related to an event/POI may mention similar entities as the geocoded tweets. Thus from geocoded tweets in the initial space-time cube, we first extract mentions. We then query for more tweets with similar mentions and from same-city users (based on their profiles). We now have a mixture of tweets with and without location information. To consistently form the graph, we connect tweets based only on temporal proximity, i.e. $|t_i - t_j| \leq \delta_t$. Note that although individual edges are based on temporal proximity, the overall graph incorporates spatial-proximity since tweets are constrained to be from the initial

space-time cube or users in the same city.

Objective function. Let D and E be the set of nodes and edges respectively in the tweet graph. We define our objective function for collective linking:

$$Q(D, E, \mathbf{e}) = \frac{\alpha}{|D|} \sum_{i=1}^{|D|} C(d_i, \mathbf{e}_i) + \frac{\beta}{|E|} \sum_{(d_i,d_j)\in E}^{|E|} C(d_i, d_j, \mathbf{e}_i, \mathbf{e}_j) + \frac{\gamma}{|M|} \sum_{i=1}^{|T|} \sum_{a=1}^{|m_i|} p(e_{ia}|m_{ia}) \quad (6.5)$$

where $|M|$ is the total number of mentions, with set of linked entities $\mathbf{e}$; and α, β and γ are global combination weights. Essentially Q is a linear combination of intra-tweet coherence, inter-tweet coherence and the entity prior term. Thus Q encapsulates our earlier discussed intuitions about coherence and entity popularity. For a fixed set of weights, the optimization problem is to assign entities to mentions to maximize Q. For optimization, we use the decoding algorithm [51].

Parameter Settings. We consider unsupervised collective linking where labeled data is unavailable. Given that tuning/training is not possible, we consider two intuitive cases of averaging. In the first case, we use uniform weights in Q, i.e. $\alpha = \beta = \gamma$. We referred to this setting as *Uniform*. Alternatively, one can regard coherence and entity prior as very different notions and assign them equal importance. Hence in the second case, one averages over coherence and the entity prior, i.e. $\alpha = \beta, \gamma = \alpha + \beta$. We denote this setting *Avg(Coh,prior)*.

6.4 Comparison-Based Evaluation

Consider a typical entity linking approach where one provides some initial entity assignments for initialization. For example, we can initialize collective linking with local linking, rather than using random initialization or some heuristics. In comparison-based evaluation, we shall compare the initial and final linkings to determine if a change is an improvement (positive change), a degradation (negative change) or neither. This has several advantages which we discuss next.

Annotation Effort. Firstly, we only need to compare linkings which are different between two linking results. This reduces the data annotation effort, compared to traditional evaluation using accuracy [78], i.e. proportion of correctly linked mentions. For example, to compute accuracy for a dataset of 100 mentions, each mention first has to be linked to the correct KB entity, typically via manual annotation [56]. In our evaluation framework, the annotation effort depends on the linkage differences between techniques and is usually less. For example, if all 100 mentions are linked by local linking and collective linking suggested 5 changes, then we only need to examine 5 changes. Clearly, more positive than negative changes is desired and implies improved performance.

Incomplete KB & Imperfect Linking. No KB can cover all mentioned entities. One can ignore unlinkable mentions or link them to the catch-all NIL entity [78, 55, 79]. However this discards data that may be useful for evaluation. Related to this, there is also the notion of how fine-grained a linkage needs to be, in order to be considered correct. Mentions can be linked to entities at different type or instance granularities. If one considers all coarse-grained linkages as wrong, many linkages useful for comparing techniques will be discarded.

For example, consider Table 6.1. The tweet was sent from the game venue during a college football match between Duke and Indiana University. Linking the mention *Duke* to Wikipedia, the most fine grained entity is e_1, i.e. Duke University's football team. However a linking technique may miss this perfect linking and choose other entities. Table 6.1 also lists Wikipedia entities in decreasing order of relatedness to the actual football team. Consider two techniques, one linking *Duke* to e_2, the other to e_4. Clearly the former provides useful information, even though both techniques miss out on e_1. In such cases, we still want to differentiate both techniques instead of regarding both linkings as equally wrong. If e_1 is not in the KB, but parent organizations such as e_2 and e_3 are present, it is still possible and reasonable to compare linking performance on *Duke*, instead of just discarding the mention as unlinkable. This calls for a comparison-based kind of evaluation.

Table 6.1: A sample tweet with mentions (in Italics). Row 2 lists candidate Wikipedia entities for the mention *Duke*, in decreasing relatedness.

"Go *Duke*! *#PinstripeBowl @Yankee Stadium*"
• e_1: Duke_Blue_Devils_football: Duke University's football team
• e_2: Duke_Blue_Devils: Duke University's varsity sports team
• e_3: Duke_University: Duke University
• e_4: Duke: Monarch ruling over a duchy

Noisy Mention Extraction. Automated mention extraction is noisy. Often, incomplete sub-mentions are extracted. Even in cases where a mention should link to a unique entity, the notion of correct/wrong linking is less clear when sub-mentions are involved. Fortunately in comparison-based evaluation, we can compare entity assignments and pick the better one. For example, consider the tweet "Watching *Jeff Dunham @star* performing arts centre with the family", where mentions (in italics) were extracted with TweetNLP [64]. The complete venue mention is *star performing arts centre*. However the sub-mention *star* was extracted, constraining entity linking to link *star*. Instead of discarding such cases, one can still compare linking results, e.g. linking to 'Movie_star' is intuitively preferred over 'Star': a luminous sphere of plasma in space. On a related note, if an extracted mention is in fact not of a named-entity, such comparisons can also be used for evaluation.

6.4.1 Evaluating Changes

To evaluate changes, we define what constitutes each outcome. Firstly, we observe changes to often reduce or increase the specificity/granularity of linked entities. This leads to the consideration of parent-child relationships between entities in a type hierarchy. For brevity of discussion, we overload the term of entity types such that types can refer to semantic categories, organizations or locations. A super-type is decomposable into sub-types of finer granularities and this is applicable to semantic categories, instances, organizations and locations. For example entity e_1:'Duke_Blue_Devils_Football' is a sports team instance under the semantic category of 'American_football', and also a child organization of 'Duke_University'. For a location example 'New_York_City' (NYC) contains (and is the parent of) 'Madi-

son_ Square_Garden', a multi-purpose indoor arena.

Clearly, we are considering more parent-child relationships beyond the semantic categories in ontologies. Hence any automated evaluation using only ontologies, e.g. the Dbpedia ontology[1] will be highly incomplete. Instead we compare type information using Wikipedia content when assessing linkage changes, e.g. e_1's Wikipedia page starts with "*The Duke Blue Devils Football team represents Duke University in the sport of American football*".

We now discuss *positive changes* using Table 6.1:

- **Incorrect linking to parent entity / correct linking:** In this case, initial linking is unrelated and wrong, e.g. linking *Duke* to 'Duke', ruler of a Duchy. Changing the linking to either 'Duke_University' (a parent entity) or 'Duke_Blue_Devils_football' (the correct linking) is a positive change.
- **Parent entity to correct linking:** An example of this is changing the linking for *Duke* from 'Duke_University' to 'Duke_Blue_Devils_football'. Intuitively, this provides more specific information to the system user.
- **Ancestor entity to parent entity:** In this case, the final linking is still not perfect, however the information specificity is increased, e.g. changing the linking for *Duke* from 'Duke_University' to 'Duke_Blue_Devils'.
- **Incorrect sibling entity to parent entity:** We regard coarse-grained, related information as more useful than specific, but wrong information, e.g. if *Duke* is initially linked wrongly to 'Duke_Blue_Devils_men's_basketball' and changed to 'Duke_Blue_Devils', it counts as a positive change.

For the above, reversing the change direction counts as *negative changes*. In addition, changes can be neither positive nor negative, e.g. replacing an incorrect entity with another. Such "neither" changes also include changing an initial unrelated entity assignment to a sibling or child entity, although this arguably improves our understanding of the tweets involved. For example, if *Duke* in Table 6.1 is initially

[1] http://mappings.dbpedia.org/server/ontology/classes/

linked to 'Duke' and changed to 'Duke_Blue_Devils_men's_basketball', we count it as a neither. Section 6.5.4 provides examples from experiments.

6.4.2 Limitations

While we have discussed the advantages of comparison-based evaluation, it is also important to point out the limitations. Firstly, since comparison-based evaluation compares the results of model pairs, the number of inter-model comparisons and the number of inter-model changes to report will scale quadratically with the number of models compared. If one is comparing many models, then comparison based evaluation can lead to significant comparison effort and a large result table which may be harder to interpret. Secondly, while comparison based evaluation can be applied for both unsupervised and supervised models, the latter will imply that researchers have access to labeled data anyway, which cancels out the annotation effort advantage of comparison-based evaluation. For this reason, we see comparison based evaluation as being more likely to be used for comparing unsupervised models.

In short, it is crucial to consider one's experiment setup and resources for annotation when deciding whether to use comparison-based evaluation or the traditional accuracy-based evaluation.

6.5 Experiments

6.5.1 Data

We conduct experiments on New York City (NYC) and Singapore (SG) tweets. To obtain meaningful tweets for linking (instead of trivial blabber [56]), we collect tweets near POIs or in space-time cubes covering performance events. For NYC, we obtained geocoded tweets from the CHIMPS Lab[2] that are within 100 meters of five popular event venues. For each venue, we consider two evenings (18:00-22:00) in Dec 2015 with the most tweets, obtaining 10 space-time cubes with an average

[2]http://cmuchimps.org/

of 24.8 tweets. For each cube, we form a spatio-temporal tweet graph for collective linking where tweets within 1 hr and 100 m of each other are connected. For Singapore, we relax the spatial proximity requirement as discussed in Section 6.3.3 and obtain an average of 46.47 tweets over space-time cubes covering 17 performance events. The tweets are a mixture of geocoded and non-geocoded tweets. We connect tweets within 1 hr of each other. Note that although individual edges in the tweet graph are based on temporal proximity, there is still a coarse notion of spatial proximity as most tweets are from Singapore, a small geographical area.

Following tweet graph construction, we apply both manual and automated mention extraction. For the latter, we use TweetNLP. For manual mention extraction, we process all 10 space-time cubes for NYC and 8 space-time cubes (out of 17) for SG, selected based on largest number of tweets. We link all mentions regardless of whether the parent tweets are related to the POI or event.

6.5.2 Local Linking Baselines

We use collective linking to modify the results of local linking. Thus the latter are equivalent to baselines. We implement LocLink (Section 6.3.2) with uniform weights for the objective in Equation (6.3). We also implement TAGME [27], which is based on weighted voting among candidate entities.

6.5.3 Results

Results are summarized in Table 6.2 for New York City (NYC) tweets and Table 6.3 for Singapore (SG) tweets. Comparing collective linking to local linking, we see linkage improvements across all experiment settings. Consistently, collective linking makes more positive changes than negative changes, when applied on the results of local linking. In most cases, the ratio of positive to negative changes is larger than 2. The highest ratio is 12, for the experiment using NYC tweets with manually extracted mentions, TAGME for local linking and averaging over coherence and entity for Q, i.e, $Avg(coh, prior)$. The lowest ratio is 1.44, again on

NYC tweets and with TweetNLP, LocLink and $Avg(coh, prior)$.

Table 6.2: Results on NYC tweets. Bracketed numbers are counts of unique mentions over which changes occur. (Δ: total changes, +ve: total positive, -ve: total negative, Ratio: +ve/-ve. **: significant at p-value=0.01, *: sig. at p-value=0.05)

Local linking method		**LocLink**				**TAGME**			
Mentions	Setting	Δ	+ve	-ve	Ratio	Δ	+ve	-ve	Ratio
Manual	*Uniform*	43	22 (14)	9 (6)	2.44^{**}	73	37 (18)	6 (5)	6.17^{**}
Manual	*Avg(coh,prior)*	20	13 (9)	3 (3)	4.33^{*}	62	36 (18)	3 (3)	12.00^{**}
TweetNLP	*Uniform*	61	23 (14)	11 (10)	2.09^{*}	103	38 (19)	13 (12)	2.92^{**}
TweetNLP	*Avg(coh,prior)*	50	13 (8)	9 (7)	1.44	95	35 (18)	9(7)	3.89^{**}

Table 6.3: Results on SG tweets. Notations as in Table 6.2.

Local linking method		**LocLink**				**TAGME**			
Mentions	Setting	Δ	+ve	-ve	Ratio	Δ	+ve	-ve	Ratio
Manual	*Uniform*	59	22 (10)	7 (4)	3.14^{**}	93	38 (14)	8 (6)	4.75^{**}
Manual	*Avg(coh,prior)*	28	16 (7)	2 (2)	8.00^{**}	78	37 (16)	8 (6)	4.63^{**}
TweetNLP	*Uniform*	83	29 (10)	9 (7)	3.22^{**}	168	61 (21)	30 (8)	2.03^{**}
TweetNLP	*Avg(coh,prior)*	44	23 (8)	2 (2)	11.5^{**}	128	54 (23)	23 (6)	2.35^{**}

Our results are statistically significant. Considering positive and negative changes, we conducted significance testing with the binomial test. The null hypothesis is that the proportion of positive and negative changes is equal. Except for one setting (TweetNLP, LocLink and $Avg(coh, prior)$), we are able to reject the null hypothesis at p-value of 0.05.

In both Tables 6.2 and 6.3, we also tabulate the number of unique mentions (in brackets) over which changes are made. This provides another view of the results accounting for mention diversity. In the trivial case, if all mentions are identical and initially wrongly linked, then it is easy to achieve many positive changes just from correcting one unique mention. However this overstates the performance advantage of collective linking due to a lack of mention diversity. From both tables, we see that the number of unique mentions for positive changes is consistently larger than that for negative changes, which is reassuring.

Collective linking exerts much of its influence through inter-tweet coherence. Recall that for $Uniform$, we use uniform weights for Q, while for $Avg(coh, prior)$, weight for the entity prior is set equal to total weights from intra and inter-tweet

coherence. Thus in $Avg(coh, prior)$, inter-tweet coherence has smaller relative weight and plays a smaller role in affecting the linking results. This means that collective linking should suggest fewer changes. Indeed, we see that for a fixed mention extraction and local linking method, there are always fewer changes in $Avg(coh, prior)$ than $Uniform$.

6.5.4 Qualitative Analysis

Many, but not all changes are shared across experiments. We illustrate changes for one experiment on NYC using the following settings: TweetNLP for mention extraction, TAGME for local linking and uniform weighting for Q. Sample tweets are displayed in Tables 6.4 to 6.6, along with changes in the format: Initial entity $\rightarrow$ final entity. Readers can inspect Wikipedia entities by appending the entity name to the URL 'https://en.wikipedia.org/wiki/'.[3]

Positive Changes. Table 6.4 shows positive changes. Tweets N1 and N2 are from a college football match between Duke and Indiana University. The mention *Duke* in N1 is initially linked by TAGME to 'Duke': ruler of a Duchy. Collective linking then changed it to 'Duke_University'. Although this is not perfect, it is an improvement since Duke University is the parent organization of the football team involved. For N2, the final entity for *Hoosier* is correct in the strictest sense. Tweet N3 illustrates geographical effects, where surrounding tweets linked to NYC-related entities drive changes in the initial linking. For example, N3 is about a basketball game involving Syracuse University. Its final linking is a positive change, since an unrelated entity (a location in Italy) has been changed to a parent entity (university's location in NYC).

Negative Changes. Table 6.5 illustrates negative changes. N5's mention *World* is not from a named entity, but has been extracted by TweetNLP. It is impossible to automatically filter out all such mentions, hence linking is still conducted. The final linking in N5 is overly specific and wrong. N5 originates from NYC and

[3]e.g. entity 'Duke_University' for tweet N1 (Table 6.4) is described in 'https://en.wikipedia.org/wiki/Duke_University'.

Table 6.4: Examples of positive changes (in bold), with affected mentions in italics.

N1	"LETS GO *DUKE*!! #PinstripeBowl @Yankee Stadium" **Duke → Duke_University**
N2	"May be the post-season but finally getting to see the #*Hoosiers* play" **Hoosiers → Indiana_Hoosiers_football**
N3	"*Syracuse* game with my dad at The Garden-we're both alumni #cuse #cusenation #nyc" **Syracuse,_Sicily → Syracuse,_New_York**

Table 6.5: Examples of negative changes (in bold), with affected mentions in italics.

N5	"*World*'s Most Famous Arena for my sixth sporting event in two weeks..." **World → World_Wrestling_Entertainment**
N6	"Incredible spread by the @*yankees*. Choice of pork, chicken, hot dogs and burgers. Salad bar" **Yankee → New_York_Yankees**

surrounding tweets mentioned entities that drive the negative change. For example, mentions of NYC will drive the linking towards 'World_Wrestling_Entertainment' (WWE) since WWE's event had been held in NYC before. For N6, initial linking is to 'Yankee', which discusses usage of the word, including its usage in referring to Americans. The final linking is wrong and refers to an American baseball team.

Table 6.6: Sample changes (bold) for affected mentions (italics) that arguably improve tweet understanding, but are not counted as positive changes.

N9	"*Bowl* Games with Famiky #CandyStripes NotPinstripes #PinstripeBowl" **Bowl → Super_Bowl**
N10	"I Met Former UFC Fighter & *WWF* Wrestler Dan The Beast Severn At The MMA World Expo. Dan Is A..." **World_Wide_Fund_for_Nature → Hulk_Hogan**

Neither. Table 6.6 shows two examples where the final linking arguably improves our understanding of the tweet content. N9 is generated during a college football game. After collective linking, its mention *Bowl* is linked to a different series of football game, much better than the initial linking to 'Bowl', a container. N10's mention *WWF* is finally linked to a WWF wrestler, a more related entity than the initial linking to a nature conservation organization. Nonetheless such cases do not fall into our discussed scenarios in Section 6.4.1 and can be subjective to assess. Hence we do not count them as positive change.

6.6 Concluding Remarks

Motivated by event and geographical effects, we have proposed a collective entity linking approach for tweets over space and time. In addition, we proposed a comparison-based evaluation strategy that focuses on the linkage differences between competing entity linking techniques. This reduces manual annotation effort and mitigates challenges such as noisy mention extraction and incomplete KB. Our results show that collective linking over space and time performs much better than local linking techniques that process individual tweets. In extensive experiments, collective linking improves the linking quality of local linking.

Chapter 7

Implicit Entity Linking

7.1 Introduction

In the final track of this dissertation, we recover the semantic contexts of food-related posts. As dining comprises an important and interesting activity for many users, they post food-related microblogs or reviews on various platforms such as Instagram, Foursquare, Yelp, etc. Such user generated content can be mined for profiling food lovers or for food and dining venue recommendations. In fact, identifying the local cuisines in posts has been justified [60] as useful for applications such as helping tourists in their dining choices. In this chapter, we propose to link food-related posts to a knowledge base of food entities. Given a test post that mention or *merely imply* some food entity, the task is to rank food entities in order of relevance.

We refer to this problem of linking posts as *Implicit Entity Linking* (IEL) [66, 56]. In IEL, one links each test post to one or more related entities, without the need for mention extraction. This contrasts with the Explicit Entity Linking (EL) problem [51, 79, 38, 13] which links mentions of named entities and which we have explored in Chapter 6. Notably IEL circumvents the challenge of mention extraction in social media where posts are often grammatically noisy and colloquial. IEL also generalizes easily to various content scenarios. For example, consider the text snip-

pets “XX Chicken Rice”, “rice with chicken” and “having lunch”. These are cases where food entities are respectively mentioned via proper nouns, improper nouns and merely implied. All snippets can be processed via IEL while EL is mention-dependent and will process only the first snippet comprising proper nouns. Lastly, IEL is also easier to conduct if one is only focused on a certain entity type, e.g. food entities. There is no need to ensure that only mentions of the right type are extracted.

Problem Setting. We formulate IEL as a ranking problem. For each post, we rank candidate food entities such that high ranking entities are more likely to be related. We assume that posts are not labeled with food entities for training, but are associated with posting venues. Both assumptions are realistic. Firstly labeled data are typically expensive to obtain. Secondly venue information is often available for platforms such as Foursquare, Instagram, review websites etc. We use Wikipedia as the knowledge base to link against. However our proposed models are general and not specific to Wikipedia.

Contributions. Our contributions are (1) an empirical analysis whereby we highlight that venues are focused around a limited set of food entities each, i.e. *entity-focused characteristic* and (2) a series of models for IEL. Our best performing model comprises the following aspects:

- **Entity-Indicative Weighting**: We propose a weighting scheme in our model to assign more weights to entity-indicative words. The intuition is that such words are more important for inferring entities than other words.
- **Query Expansion:** The entity-focused characteristic implies that a test post is likely to share common food entities as other same-venue posts. Hence we augment each test post via query expansion to include words from other same-venue posts.
- **Venue-based Prior:** Leveraging the same entity-focused characteristic, we generate venue-based prior distribution over food entities in an initial entity linking stage. This prior is used to bias the entity scores for the next stage.

By combining all above aspects, our best model EW-EWQE(v) outperforms state-of-the-art baselines that have been adapted for implicit entity linking.

7.2 Empirical Analysis

7.2.1 Datasets

In our empirical analysis and subsequent experiments, we use data from Instagram and Burpple [1]. The latter is a popular food review website in Singapore. Both datasets are generated by users from Singapore, a city well known for its wide range of food choices. Since both datasets are from Singapore users, we link their posts against a list of 76 food entities derived from the Wikipedia page on Singapore's cuisines[2]. Further details are discussed in Section 7.4.2.

Table 7.1: Sample posts comprising Instagram captions and Burpple reviews.

Instagram	"super heavy lunch. and spicy! but its a must-try cafe! #food #foodporn #foodie #foodgasm #badoquecafe #instagood"
	"yesterday's lunch! #fishballnoodle #food #foodporn the soup was damn good"
Burpple	"*Signature Lamb Rack ($46++)* Very neat rectangular bricks of lamb, which we requested to be done medium-well.Nothing too impressive.. hurhur. Service is top -notch though"
	"*Good morning!* One of my favourite old school breakfast but he not his fav"

For Instagram, we collect highly popular food-related captions from 2015 using hashtags of food e.g. '#foodporn' [3], or food entities e.g. '#chillicrab'. Following data cleaning and merging of duplicate venues, we obtained 278,647 Instagram posts arising from 79,496 distinct venues. For Burpple, all its posts are food reviews and filtering by hashtags is not required. From Burpple, we obtained 297,179 posts over 13,966 venues. Table 7.1 illustrates four sample posts, two each from Instagram and Burpple. It can be seen that some posts are more informative about specific food entities than others. For example, the first instagram example does not

[1] https://www.burpple.com/sg

[2] https://en.wikipedia.org/wiki/Singaporean_cuisine

[3] the most popular food related hashtag on our Instagram dataset

reveal the food entity explicitly while the second example mentions fish ball noodle.

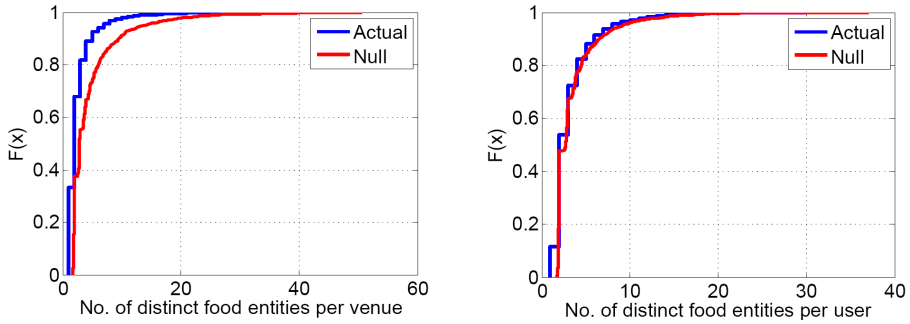
7.2.2 Analysis

A food venue typically focuses on some cuisines or food themes and is unlikely to serve an overly wide variety of dishes. For example, it is more probable for a restaurant to serve either Western or Asian cuisines, rather than both. Consequently, each food venue is likely to be associated with a limited number of food entities. We termed this as the *entity-focused characteristic*. To quantify this characteristic, we compare the number of distinct food entities per venue against a null model where the characteristic is absent. On average, we expect food venues to be associated with fewer food entities when compared against the null model.

For each venue v with multiple posts, we first compute the number of distinct entities over its posts. We then compute the expected number of distinct entities under the null model following the steps below:

- For each post from v, sample an entity e based on global entity probability i.e. entity popularity. Add to entity set $\mathbb{E}_{null}(v)$.
- Compute $|\mathbb{E}_{null}(v)|$, the distinct food entity count under the null model.

We conduct our analysis on 2308 venues from Instagram and 362 venues from Burpple which have at least two user-labeled posts each. Such posts contain entity-indicative hashtags that have been assigned by their authors, e.g. '#chillicrab', '#naan' etc. For venues with only one such post, there can only be one distinct food entity each under the null model and comparison is not meaningful. As sampling is required for the null model, we conduct 10 runs and take the average expected food entity count for each venue. For further analysis, we also repeat a similar procedure for users to compare their actual and expected food entity count. The intuition is that users may possess the entity-focused characteristic as well due to food preferences or constraints e.g. vegetarian. The user statistics are computed over 2843 Instagram users and 218 Burpple users.



(a) CDF (Instagram venues)

(b) CDF (Instagram users)

(c) CDF (Burpple venues)

(d) CDF (Burpple users)

Figure 7.1: CDFs of actual and expected distinct food entities for venues and users. F(x) on y-axis is probability of venues or users with $\leq x$ distinct food entities.

Figure 7.1 plots the Cumulative Distribution Function (CDF) of distinct food entities for venues and users on both Instagram and Burpple, whereby distinct entity counts are on a per venue or user basis. In each graph, the blue line represents the actual count while the red line is for counts from the null model (averaged over 10 runs). For Figures 7.1(a) and (c) venues are shown to be focused around specific food entities such that on average, each venue has fewer distinct food entities than expected under the null model. For example in Figure 7.1(a), around 98% of the Instagram venues are associated with 10 distinct food entities or less in the actual data. In contrast, the null model has a corresponding proportion of around 91%. A similar trend can be observed for Burpple venues as shown in Figure 7.1(c). Thus, the entity-focused characteristic is clearly evident for the venues of both datasets.

Figures 7.1(b) and (d) plot for Instagram and Burpple users respectively. There is much less difference between the actual and null model count, as both the blue and red lines overlap substantially in both figures. Comparing the plots for venues and users, we conclude that users are relatively less focused on food entities when

compared to venues. These findings have implications for entity linking and should be considered when designing models. In particular, given a test post with both user and venue information, it may be easier to improve linking accuracy by exploiting other posts from the same venue rather than from the same user. In Section 7.3.2, we shall introduce a query expansion approach based on exploiting the entity-focused characteristic of venues.

7.3 Models

In this section, we present a series of models for IEL, culminating in a final best performing model. We start with the naive Bayes model. This can be regarded as a standard information retrieval baseline. Let **w** be the set of words in a post, where for notation simplicity, we assume each unique word $w \in \mathbf{w}$ occurs only once in the post. In our problem setting, we assume the entity probability $p(e)$ to be uniform as labeled posts are unavailable for estimation. The probability of food entity e given **w** is:

$$p(e|\mathbf{w}) \propto \prod_{w \in \mathbf{w}} p(w|e) = \prod_{w \in \mathbf{w}} \frac{f(e,w) + \gamma}{\sum_{w'} f(e,w') + W\gamma} \tag{7.1}$$

whereby $f(e,w)$ is the number of co-occurrences of word w with entity e, γ is the smoothing parameter and W is the vocabulary size. In the absence of labeled posts, the co-occurrences are estimated solely from the Wikipedia knowledge base. For each food entity e, we derive $f(e,w)$ by the count of w occurrences in the Wikipedia page of e and in Wikipedia text snippets around hyperlinks to e (refer Section 7.4.2). Finally entities are ranked by $p(e|\mathbf{w})$. The naive Bayes model is efficient and highly amenable to extensions.

7.3.1 Entity-Indicative Weighting (EW)

The naive Bayes model multiplies word probabilities without considering which words are more important for entity linking. Intuitively, some words are more indicative of food entities than others and should be assigned greater importance in

entity linking models. Formally, an entity-indicative word w has relatively high $p(e|w)$ for some entity/entities in comparison with other words, e.g. 'sushi' is more entity-indicative than 'dinner'.

An entity-indicative word is different from a high probability word given an entity. For example, a food entity e may have high probability of generating the word 'rice', i.e. $p(\text{'rice'}|e)$ is high. However if many other food entities are also related to rice, then the word may not indicate e with high probability i.e. low $p(e|\text{'rice'})$. If a post $\mathbf{w}$ mentions other more entity-indicative words, e.g. related to ingredients or cooking style, then such words should be assigned more importance when computing $p(e|\mathbf{w})$.

To capture the above intuition, we propose the entity-indicative weighting (EW) model. This assigns continuous weights to words and incorporates easily into the naive Bayes model. Let $\beta(w)$ be the *entity-indicative* weight for word w. This weight $\beta(w)$ is added as an exponent to the term $p(w|e)$ in Equation 7.1. By taking the log to avoid underflow errors, we obtain the EW model:

$$\ln p(e|\mathbf{w}) \propto \sum\nolimits_{w \in \mathbf{w}} \beta(w) \ln p(w|e) \tag{7.2}$$

Interestingly, Equation (7.2) is similar in form to the weighted naive Bayes model proposed in prior work [89, 28] for classification tasks. Here, we use it for IEL.[4]

To compute the weights $\beta(w)$, we apply the vector space model and treat entities as akin to documents. By definition, entity-indicative words are associated with fewer entities and have large inverse-document frequencies which can be used as weights. Formally given word w, we compute its weight as:

$$\beta(w) = \log(1 + E/df(w)) \tag{7.3}$$

where E is the number of distinct food entities considered and $df(w)$ counts entities with at least one occurrence of w.

[4]We have earlier used a similar model for geolocation. See Section 4.3.1.

7.3.2 Query Expansion with Same-Venue Posts

Based on the entity-focused characteristic, we expect that as a venue accumulates posts over time, its set of entities will be discussed repeatedly over different posts. This implies that for a test post discussing some entity e, there may exist other same-venue posts related to e. Hence if we augment the test post appropriately with words from other same-venue posts, we can potentially mitigate information sparsity in one post and improve entity linking. To achieve this, we treat each test post as a query and apply query expansion.

Let test post **w** be posted from venue v. The idea is then to score candidate words w' appearing in other posts from v's and whereby $w' \notin \mathbf{w}$. The expanded words w's aim to provide additional information for inferring the latent entity in **w**. Among the many scoring schemes in the literature, we adopt a relatively simple cosine similarity scheme from [21]. This scheme scores each candidate word w' by its average relatedness $0 \leq \Omega(w', \mathbf{w}; v) \leq 1$ to the test post as:

$$\Omega(w', \mathbf{w}; v) = \frac{1}{|\mathbf{w}|} \sum\nolimits_{w \in \mathbf{w}} \frac{d_v(w', w)}{\sqrt{d_v(w') d_v(w)}} \tag{7.4}$$

where $|\mathbf{w}|$ is the number of words in **w**, $d_v(w', w)$ is the count of v's posts containing both w' and w; and $d_v(w)$ is the count of v's posts with w. Intuitively, if w' co-occurs more with each word from **w** on average, then average relatedness is higher. However, relatedness can be over-estimated for common words. To mitigate this, Equation (7.4) includes in the denominator the product of word frequencies as the normalization term.

Following query expansion using same-venue posts, we combine two different word sets in a weighted naive Bayes model, which we refer to as QE(v)[5]:

$$\ln p(e|\{\mathbf{w}, \mathbf{w'}\}, v) \propto \sum\nolimits_{w \in \mathbf{w}} \ln p(w|e) + \sum\nolimits_{w' \in \mathbf{w'}} \Omega(w', \mathbf{w}; v) \ln p(w'|e) \tag{7.5}$$

[5]This is very similar to our earlier geolocation model based on query expansion along the user facet. See Section 4.3.2.

where **w'** is the set of added words for post **w** from venue v. Since $0 \leq \Omega(w', \mathbf{w}; v) \leq 1$, Equation (7.5) illustrates that the original query words $w \in \mathbf{w}$ have greatest importance in the model while the importance of newly added words $w' \in$ **w'** vary based on how related they are to the query.

In our experiments, we shall also compared against a model variant QE(u), which selects augmenting words from same-user posts. As conjectured in Section 7.2.2, this model may be less likely to improve linking accuracy.

7.3.3 Fused Model (EWQE)

We now combine the EW and QE(v) models to create a new fused model called EWQE[6]. Intuitively, we consider a word as important only when it is both entity-indicative *and* highly related to the test post. For example, if a word is not indicative of any entities, then it is less useful for entity linking even if it is present in the test post or is a highly related word based on Equation (7.4) . On the other hand, a non-related word may be indicative of some entity which is unrelated to the test post, such that test post augmentation with it introduces noise and lowers accuracy.

To model the conjunction logic of the discussed intuitions, we multiply the weights from entity-indicative weighting and query expansion to obtain the combined model EWQE(v):

$$\ln p(e|\{\mathbf{w}, \mathbf{w'}\}, v) \propto \sum_{w \in \mathbf{w}} \beta(w) \ln p(w|e) + \sum_{w' \in \mathbf{w'}} \beta(w') \Omega(w', \mathbf{w}; v) \ln p(w'|e) \quad (7.6)$$

We note that it is also possible to combine entity-indicative weighting with user-based query expansion. We refer to such a model as EWQE(u) and include it in our experiments.

[6]Recap that we have a very similar model for geolocation. See Section 4.3.3.

7.3.4 Venue-based Prior

In our final model, we augment the probabilistic generative process in Equation (7.6) with a venue-based prior distribution over entities $p(e|v)$. Let joint probability $p(e, \{\mathbf{w}, \mathbf{w'}\}, v)$ be factorized as $p(v)p(e|v)p(\{\mathbf{w}, \mathbf{w'}\}|e)$. We now need to compute $p(e|v)$ while $p(\{\mathbf{w}, \mathbf{w'}\}|e)$ can be computed as before with the EWQE(v) model. Assuming uniform venue probability $p(v)$ and incorporating a weighting term η ($0 \leq \eta \leq 1$), we have:

$$\ln p(e|\{\mathbf{w}, \mathbf{w'}\}, v) \propto \eta \ln p(e|v) + (1-\eta)\left(\sum_{w \in \mathbf{w}} \beta(w) \ln p(w|e) + \sum_{w' \in \mathbf{w'}} \beta(w')\Omega(w', \mathbf{w}; v) \ln p(w'|e)\right) \quad (7.7)$$

Basically $p(e|v)$ bias the entity score in a venue-specific manner, rather than a post-specific manner as prescribed by query expansion. Given a set of training posts labeled with food entities, $p(e|v)$ is computed trivially. However in our setting, we assume no labeled posts for training. Hence we compute Equation (7.7) in a 2-stage process as follows:

- Stage 1: With a desired IEL model, link the training posts. For each venue v, compute the aggregated entity scores $\tilde{p}(e|v)$, e.g. if using the EW model, we compute $\tilde{p}(e|v) = \sum_{\mathbf{w} \in v} p(e|\mathbf{w})$. Normalize $\tilde{p}(e|v)$ to obtain $p(e|v)$.
- Stage 2: Combine $p(e|v)$ with the scores from the EWQE(v) model as detailed in Equation (7.7) to derive the final entity scores for ranking.

7.4 Experiments

7.4.1 Setup

Our experiment setup is weakly supervised. Training posts are assumed to be unlabeled with respect to food entities. These training posts are used only for query expansion and for computing the venue prior over entities, but not for computing the

entity profile $p(w|e)$. The entity profile $p(w|e)$ and entity-indicative weights $\beta(w)$ are computed using only Wikipedia pages. However, we retain a small validation set of entity-labeled posts for tuning model parameters with respect to the ranking metrics. Also, all posts are already associated with posting venues, regardless of whether they are in the training, test or validation set.

For ease of discussion, denote posts with food entity hashtags e,g, '#chillicrab' as type A posts and post without such hashtags as type B posts. Type A posts are easily associated with Wikipedia food entities, which facilitates the construction of both test and validation sets. Our datasets contain a mixture of both post types. For Instagram, we have 18,333 type A vs 216,881 type B posts[7] whereas for Burpple, we have 1944 type A vs 200,293 type B posts. We conduct 10 experiment runs for each dataset, whereby in each run, we *mask the food entity hashtags* of type A posts and randomly assign 50% of them to the training set, 20% to the validation set and 30% to the test set. The type B posts are all assigned to the training set. Lastly, most of our type A posts contain only one food entity hashtag each, hence we use such single-entity posts for evaluation in our test set.

7.4.2 Food Entities

We consider 76 food entities that are defined by Wikipedia as local cuisines of Singapore[8], as well as associated with distinct pages/descriptions. For each entity e, we construct its profile, i.e. $p(w|e)$ from its Wikipedia description page and Wikipedia text snippets with hyperlinks to e. For example, the Wikipedia page 'Pakistani_cuisine' contains many hyperlinks to the food entity 'Naan' [9]. When building the profile for 'Naan', we include the preceding and succeeding 10 words around each hyperlink.

[7]Filtering by vocabulary has been applied, hence the numbers sum to less than the total food-related posts in Section 7.2.1.

[8]https://en.wikipedia.org/wiki/Singaporean_cuisine

[9]oven-baked flatbread

7.4.3 Compared Models

We compare the following models:

- NB: The naive Bayes model from Equation (7.1).
- EW: Entity-indicative weighting as indicated in Equation (7.2).
- QE(v): Venue-based query expansion whereby each test post is augmented with words from other same-venue posts, as indicated in Equation (7.5).
- QE(u): User-based query expansion whereby each test post is augmented with words from other same-user posts.
- EWQE(v): Fusion of venue-based query expansion and entity-indicative weighting as shown in Equation (7.6).
- EWQE(u): Fusion of user-based query expansion and entity-indicative weighting.
- NB-EWQE(v): In stage 1, we compute $p(e|v)$ with the NB model, which is then combined with the EWQE(v) model in stage 2. See Equation (7.7).
- EW-EWQE(v): This follows the previous model except that in stage 1, we use the EW model.

For each model, we use the validation set to tune γ, the smoothing parameter for $p(w|e)$, based on the grid [0.01, 0.1, 1, 10]. For NB-EWQE(v) and EW-EWQE(v), γ is jointly tuned with η whereby η is varied in steps of 0.1 from 0 to 1.

For further comparison, we adapt EL models from [27, 26] such that they can be used for implicit entity linking. Without any adaptation, it is impossible for the vanilla models to link posts directly to entities. Our adaptations also aim to exploit the entity-focused characteristic of venues, or other related characteristic. Lastly, we also include a word embedding baseline [82] that does not require any adaptation. The baselines are:

- TAGME: In the TAGME model [27, 67] for EL, the candidate entities for a mention are voted for by candidate entities from other mentions in the same

post. Adapting the idea to IEL, candidate entities for a post are voted for by candidate entities from other posts in the same venue. Since a candidate entity gathers larger votes from the same or related entities, this voting process exploits the entity-focused characteristic of venues as well. Basically let $\mathbf{w}_{i,v}$ denote the i-th post from venue v. Then candidate entity e_i for $\mathbf{w}_{i,v}$ gathers a vote from $\mathbf{w}_{j,v}$ computed as:

$$vote(\mathbf{w}_{j,v} \rightarrow e_i) = \frac{\sum_{e_j : p(e_j|\mathbf{w}_{j,v})>0} sr(e_i, e_j) p(e_j|\mathbf{w}_{j,v})}{|e_j : p(e_j|\mathbf{w}_{j,v}) > 0|} \tag{7.8}$$

where $sr(e_i, e_j)$ is the Jaccard similarity of incoming Wikipedia links [67] between e_i, e_j, and $p(e_j|\mathbf{w}_{j,v})$ can be based on any implicit entity linking models. Finally for ranking entities, we compute the final score for entity e_i as $p(e_i|\mathbf{w}_{i,v}) \sum_j vote(\mathbf{w}_{j,v} \rightarrow e_i)$.

- LOC: Analogous to entity-focused venues, locations in the form of grid cells may be entity-focused as well. To exploit this, we implement the framework from [26]. For each grid cell, the distributions over entities are inferred via EM learning and integrated with implicit entity linking models. Unlike [26], we omit the dependency on posting time as our targeted posts include food reviews which are usually posted after, rather than during meal events. We tune grid cell lengths based on grid [200m, 500m, 1km, 2km].

- PTE: This is a graph embedding method [82] that learns continuous vector representation for words, posts and entities over a heterogeneous graph. The graph consists of word nodes, post nodes and entity nodes, connected via the following edge types: word-word, post-word and entity-word. For each test post, we compute its vector representation by averaging over the representations of its constituent words. We then compute the cosine similarities to entity representations for ranking. As in [82], we use an embedding dimension of 100. We set the number of negative samples to be 200 million.

For the baselines TAGME and LOC, we integrate the implicit entity linking models NB, EW and EW-EWQE(v). For each model, we replace the relevant mention-to-entity computations with post-to-entity computations. For example, TAGME(NB) computes $p(e_j|\mathbf{w}_{j,v})$ in Equation (7.8) using the NB model. Such integration leads to the baseline variants: TAGME(NB), TAGME(EW), TAGME(EW-EWQE(v)), LOC(NB), LOC(EW) and LOC(EW-EWQE(v)).

7.4.4 Metrics

Different from our work in Chapter 6, we are conducting IEL where noisy mention extraction is not an issue. In addition, we have access to substantial number of posts labeled with food entity hashtags. For these reasons, we use a ranking metric, the Mean Reciprocal Rank (MRR), instead of comparison-based evaluation proposed in Chapter 6.

We have previously used MRR and its macro-version in our geolocation tracks. For each test post, we are now ranking food entities, not venues. Hence for clarity, we redefine the metrics here. Given a post $\mathbf{w}_i$, let the rank of its food entity be $r(\mathbf{w}_i)$, where $r(\mathbf{w}_i) = 0$ for the top rank. Over the set of test cases $\mathbb{T}$, MRR is defined as:

$$\text{MRR}(\mathbb{T}) = \frac{1}{|\mathbb{T}|} \sum_{i=1}^{|\mathbb{T}|} \frac{1}{1 + r(\mathbf{w}_i)} \tag{7.9}$$

MRR is a micro measure. Hence in a sample of test posts, more popular food entities contribute more to MRR. For further analysis, we also consider treating all entities as equally important, regardless of their popularities. Thus we introduce Macro-MRR, the macro-averaged version of MRR. For all test posts pertaining to the same food entity, we compute the MRR of the food entity. We then average the MRRs over distinct food entities. Formally denote $\mathbb{T}_e$ as the set of test posts related to e. We compute:

$$\text{Macro-MRR}(\mathbb{T}_v) = \frac{1}{E} \sum\nolimits_{e=1}^{E} \text{MRR}(\mathbb{T}_e) \tag{7.10}$$

Table 7.2: MRR and Macro-MRR values averaged over 10 runs for each dataset. The best performing model is bolded.

	Instagram		Burpple	
Model	MRR	Macro-MRR	MRR	Macro-MRR
NB	0.344	0.218	0.335	0.259
EW	0.461	0.301	0.467	0.377
QE(v)	0.403	0.236	0.389	0.252
QE(u)	0.326	0.215	0.336	0.237
EWQE(v)	0.543	0.323	0.503	0.388
EWQE(u)	0.449	0.284	0.419	0.329
NB-EWQE(v)	0.543	0.323	0.500	0.389
EW-EWQE(v)	**0.593**	**0.340**	**0.537**	**0.401**
TAGME(NB)	0.368	0.233	0.344	0.259
TAGME(EW)	0.462	0.293	0.446	0.363
TAGME(EW-EWQE(v))	0.520	0.296	0.507	0.390
LOC(NB)	0.409	0.236	0.357	0.259
LOC(EW)	0.472	0.254	0.413	0.315
LOC(EW-EWQE(v))	0.520	0.271	0.467	0.333
PTE	0.288	0.216	0.291	0.274

where MRR($\mathbb{T}_e$) is MRR for set of test posts $\mathbb{T}_e$ and E is the number of distinct food entities.

7.4.5 Results

Table 7.2 displays the MRR and Macro-MRR values averaged over 10 runs for each dataset. In subsequent discussions, a model is said to perform better or worse than another model only when the differences are statistically significant at p-level of 0.05 based on the Wilcoxon signed rank test.

EW and QE(v) easily outperform NB, which affirms the utility of entity-indicative weighting and venue-based query expansion. EW also outperforms QE(v), e.g. EW's MRR is 0.461 on Instagram posts, higher than QE(v)'s MRR of 0.403. By combining both models together in EWQE(v), we achieve even better performance than applying EW or QE(v) alone. This supports EWQE(v)'s modeling assumption that a word is important if it is both entity-indicative and highly related to the test post.

While venue-based query expansion is useful, user-based query expansion is

less promising. Over the different datasets and metrics, QE(u) is inferior or at best on par with NB. This may be due to the entity-focused characteristic being weaker in users. This observation is consistent with our earlier empirical findings that users are less focused on food entities when compared to venues. Consequently user-based query expansion may augment test posts with noisy words less related to their food entities. Combining user-based query expansion with entity-indicative weighting also leads to mixed results. Although EWQE(u) outperforms QE(u), the former still underperforms EW. As the results are not promising, we omit further model variants that integrate user-based query expansion.

Our results also show that the venue-based prior distribution over entities is useful, but only if it is computed from a reasonably accurate linking model. Over all dataset-metric combination, the best performing model is EW-EWQE(v) which incorporates a prior computed using the EW model. Although NB-EWQE(v) incorporates a prior as well, it utilizes the less accurate NB model. For Instagram, the tuning procedure consistently indicates in each run that the optimal η is 0 for NB-EWQE(v), thus it is equivalent to the model EWQE(v). For Burpple, the optimal η is non-zero for some runs, but NB-EWQE(v) performs only on par with EWQE in terms of statistical significance.

The TAGME variants exploit the entity-focused characteristic of venues via a voting mechanism. Performance depends on the voting mechanism as well as the underlying entity linking models. Intuitively better underlying models should lead to higher ranking accuracies in the corresponding variants. For example, TAGME(EW-EWQE(v)) outperforms TAGME(EW) while TAGME(EW) outperforms TAGME(NB). However comparing the variants against their underlying models, we note that only TAGME(NB) consistently improves over NB, while TAGME(EW) and TAGME(EW-EWQE(v)) fails to outperform EW and EW-EWQE(v) respectively. The same observation applies to the LOC variants. LOC(NB) consistently outperforms NB. LOC(EW) only outperforms EW for MRR on Instagram and is inferior in other dataset-metric combination. LOC(EW-EWQE(v)) is also inferior to EW-EWQE(v).

Such mixed results of LOC variants may be due to grid cells being less entity-focused than venues. Lastly, PTE did not perform well in this task. We note that each entity has only one Wikipedia description page and are mentioned in a limited number of Wikipedia contexts. Hence the Wikipedia content of food entities may be overly sparse for learning good entity representations. There are also language differences between Wikipedia pages and social media posts. This may impact cross-linking if embeddings are trained on only one source, but not the other. In conclusion, our proposed model EW-EWQE(v) performs well, while maintaining a conceptually simple design.

7.4.6 Case Studies

In Tables 7.3 to 7.5, we illustrate different model aspects by comparing pairs of models on test posts from Instagram. Comparison is based on the ranked position of the ground truth food entity (under column e) for each post. The ranked position is denoted as r_X for model X and is 0 for the top ranked. The ground truth entities can be inspected by prepending the URL 'https://en.wikipedia.org/wiki/' to the entity name.

Table 7.3: Sample test posts to illustrate entity-indicative weighting. Words in larger fonts indicate larger weights under the EW model.

		e	r_{NB}	r_{EW}
S1	"#singapore we already ate claws ."	Chilli_crab	2	0
S2	"finally got to eat rojak !!!"	Rojak	5	0
S3	"#singapore #tourist "	Hainanese_chicken_rice	18	2

Entity-indicative Weighting. Table 7.3 compares the models NB and EW. For each test post, words with larger weights under the EW model are printed in larger fonts. For post S1 with food entity 'Chilli_crab'[10], the largest weighted word is 'claws', referring to a crab body part. This word is rarely mentioned with other food entities, but appears in the context around the 'Chilli_crab' anchor in the Wikipedia page for 'The_Amazing_Race_25', hence it is highly indicative of 'Chilli_crab'. By

[10]crabs stir-fried in chilli-based sauce

assigning 'claws' a larger weight, EW improves the entity ranking over NB, from a position of 2 to 0. For S2, the word 'rojak' is indicative of the food entity 'Rojak' [11]. While NB does well with a ranked position of 5, EW further improves the ranked position to 0 by weighting 'rojak' more relative to other words. For post S3, the food entity 'Hainanese_chicken_rice'[12] is described in the Wikipedia page 'Singaporean_cuisine' as the most popular dish for tourists in the meat category. Thus by assigning a larger weight to 'tourist', EW improves the linking of S3.

Query Expansion. Table 7.4 illustrates posts where the QE(v) model improves over the NB model. While S4 mentions dinner, the food entity is not evident. However the word 'dinner' co-occurs with more informative words such as 'chicken' and 'rice' in other posts from the same venue. Such words are retrieved with query expansion and used to augment the post. The augmented post is then linked more accurately by the QE(v) model. For S5, query expansion augments the post with 6 words of which 5 words share similar weights. Out of the 5 words, the word 'pakistani' is indicative of the food entity 'Naan', helping to improve the ranked position further from 1 to 0.

Table 7.4: Sample test posts with added words (in brackets) from query expansion (QE(v) model). The top 5 added words with largest weights are listed.

		e	r_{NB}	$r_{QE(v)}$
S4	"last night dinner at #singapore #foodporn" (rice,0.25),(chicken,0.23),(late,0.21),(food,0.21),(to,0.20)	Hainanese_ chicken_rice	19	3
S5	"indian feast #daal #palakpaneer #mangolassi @rebekkariis du vil elske det!" (pakistani,0.17),(cuisine,0.17)(buffet,0.17)(lunch,0.17)(team,0.17)	Naan	1	0

Venue-based Prior. Table 7.5 compares EWQE(v) and EW-EWQE(v). S6 is posted from a food venue which serves 'Mee_pok' [13] as one of its food entities. This food entity is mentioned explicitly in other same-venue posts. Hence on applying the EW model, we infer this venue as having a high prior probability for this entity. In fact if we rank food entities by the venue prior $p(e|v)$ alone, 'Mee_pok' is ranked at position 0. Integrating the prior distribution with other information as

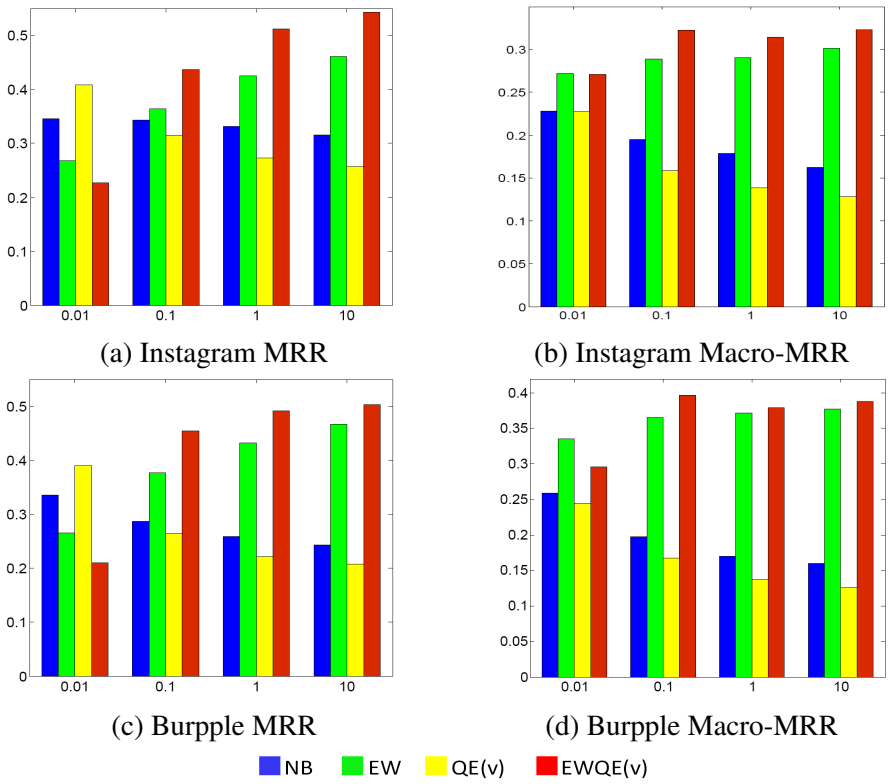
[11] a traditional fruit and vegetable salad dish
[12] roasted or steamed chicken with rice cooked in chicken stock
[13] a Chinese noodle dish

done in EW-EWQE(v), the same rank position of 0 is obtained. For S7, the ingredient black pepper sauce is mentioned, which is indicative to some extent of 'Black_pepper_crab' [14]. However EWQE(v) manages only a ranked position of 9. From other same-venue posts, the venue prior is computed and indicates the food entity to be highly probable at S7's venue. Subsequently, EW-EWQE(v) integrates the venue prior and improves the ranked position to 2.

Table 7.5: Sample test posts for comparing models EWQE(v) and EW-EWQE(v). $r_{p(e|v)}$ corresponds to ranking with the venue prior $p(e|v)$.

		e	$r_{p(e\|v)}$	$r_{EWQE(v)}$	$r_{EW-EWQE(v)}$
S6	"life's simple pleasures. #gastronomy"	Mee_pok	0	56	0
S7	"the black pepper sauce is robust and quite spicy, one of my favourite in singapore."	Black_ pepper_crab	1	9	2

7.4.7 Parameter Sensitivity



(a) Instagram MRR (b) Instagram Macro-MRR

(c) Burpple MRR (d) Burpple Macro-MRR

Figure 7.2: Model performance (Y-axis) with different γ values (X-axis).

For models which have γ as the sole tuning parameter, we compare their sen-

[14] crabs stir-fried in black pepper sauce

sitivity with respect to γ. Figure 7.2 plots the performance of NB, EW, EWQE and EWQE(v), averaged over 10 runs for different values of γ. It can be seen that EWQE(v) outperforms NB over most of the applied γ values, i.e. 0.1, 1 and 10. Although EWQE(v) is simply a product combination of the EW and QE(v) models, it easily outperforms its constituent models, validating our combination approach. This trend is consistent across both metrics and datasets. We also note that in the absence of a validation set for tuning, a natural option is to use Laplace smoothing, i.e. $\gamma = 1$. In this perfectly unsupervised setting, it is reassuring that EWQE(v) remains the best performing model. Lastly when γ is very small at 0.01, EW and EWQE(v) appears under-smoothed and perform worse than NB. In this setting where smoothing is limited, QE(v) outperforms all other models, possibly because augmenting each test post with additional words is analogous to additional smoothing for selected words.

7.5 Concluding Remarks

We have proposed several novel yet well principled models to conduct implicit food entity linking in social media posts. Our best model exploits the characteristic that food venues are typically focused around a limited set of food entities, and the intuition that entity-indicative words should be assigned larger weights. We have also shown our proposed model to outperform more complex state-of-the-art models. In future work, we intend to explore IEL in non-geotagged social media posts, where posting venues are unknown.

Chapter 8

Conclusion

8.1 Dissertation Summary

This section summarizes the dissertation work and highlights the main contributions. With the growing popularity of LBSN, this dissertation focuses on mining information from LBSN content to answer the questions of where the user is posting from and what he is posting about. These two questions respectively lead to the problem of recovering the venue and semantic contexts.

Venue Context Recovery. In venue context recovery, we link tweets to their posting venues, a task which we denote as fine-grained geolocation. We formulate this task as a ranking problem whereby for each tweet, we rank candidate venues such that high ranking venues are more likely to be the posting venue. Based on our empirical analysis of the data, we uncover various user and tweet usage scenarios which led to the three geolocation tracks covered in Chapters 3, 4 and 5. Along with each track, we surface user behavior that are useful for the geolocation task.

Firstly, in Chapter 3, we geolocate tweets posted by users who have location history in the form of geocoded tweets. We show that users are spatially focused in being more likely to visit venues near where they have visited in the past. Our geolocation model exploits this characteristic and other characteristics, i.e. venues near each other having more similar content and dependency of venue popularity

on time of the day. In extensive experiments, our proposed model outperforms competitive baselines.

In Chapter 4, we geolocate tweets posted by another class of users, i.e. those without any location history. In the absence of location history, our model exploits user content history and other intuitive ideas. Specifically, we highlight that users tend to make repeat visits to the same or similar venues and also that users with more similar tweet content history are more similar in their venue visitation history. To exploit such behavior, our geolocation model utilizes query expansion and collaborative filtering.

Finally, in Chapter 5, we geolocate tweets in sequences whereby tweets in the same sequence are posted by the same user within a short time interval. The intuition is that given a tweet targeted for geolocation, other tweets in the same sequence may provide useful information. This is a common tweet posting scenario, but to our knowledge, not previously explored. We propose a novel model that combines different query expansion approaches and a HMM model. In particular, our model includes temporal query expansion whereby a tweet is augmented with words from other tweets in the same sequence. This exploits the user tendency to stay at the same venue or visit nearby venues within a short time interval. We show our model to be more robust and accurate than baselines in comprehensive experiments.

Semantic Context Recovery. We explore semantic context recovery by framing it as the task of entity linking. We explore two variants of entity linking: namely Explicit Entity Linking (EL) and Implicit Entity Linking (IEL).

We conduct EL in Chapter 6, to link mentions of named entities in tweets to the entities in a referent knowledge base, which in our case is Wikipedia. We formulate a collective linking approach which exploits information from multiple tweets posted close in time and space. Due to the effects of geography and events, such tweets are more likely to mention entities that are semantically more related. We also propose comparison-based evaluation which mitigates challenges from the lack of annotated data, noisy mention extraction and missing entities in the targeted

knowledge base. Based on comparison-based evaluation, we show our collective linking approach to outperform competitive EL approaches.

Lastly, in Chapter 7, we conduct IEL to link entire posts to the referent knowledge base entities. IEL does not require mention-extraction and can process posts both with and without named entity mentions. We use IEL to link food-related Instagram and Burpple posts to the correct food entities in Wikipedia. Firstly via empirical analysis, we surface the characteristic that food venues are focused around a limited set of food entities each. Next we propose an IEL model that exploits this entity-focused characteristic and other intuitions such as emphasizing entity-indicative words more. Our IEL model outperforms state-of-the-art baselines, including EL models adapted to the IEL task.

8.2 Future Work

To conclude this dissertation, we discuss potential future work.

Firstly, our models in different geolocation tracks have utilized different features and aspects of user behavior. Some features are in fact cross-applicable across the different tracks. For example, it is possible to modify the model in Chapter 3 to also incorporate the query expansion and collaborative filtering aspects from Chapter 4. Likewise, tweet posting time can be considered as well in the models of Chapters 4 and 5. It remains to be explored how much geolocation performance can be improved by a more feature-comprehensive model.

Secondly 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 direction 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 [78, 55, 79].

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, we can't 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.

Our existing models have exploited certain user characteristics such as being spatially focused near previously visited venues, repeat visitation etc. In future work, one can explore other user characteristics. Previously in Sections 3.6.8.3 and 5.4.4.2, we have highlighted cases where the existing models perform less well. For example, users can deviate significantly from their usual visitation behavior due to novelty seeking[90] or lifestyle-driven changes, e.g. change of workplace, shifting of houses etc. Users may also exhibit cyclical visitation patterns and periodically visit certain venues. Thus future work can explore how the various user aspects of novelty seeking, cyclical behavior and behavior evolution can be integrated into our models. In particular, behavior evolution may require the development of incremental models that can be updated dynamically as new user data stream in.

Finally, there is much room for future work in Implicit Entity Linking. Our current model exploits the posting venue information for IEL. The availability of such information differs across platforms. For example, posting venues are readily observed in Foursquare posts but usually not in pure tweets. It remains to be explored in future work how one can exploit the entity-focused characteristic of venues in such cases. In another direction, it is also useful to consider the linking of more general entities beyond just food entities. This may constitute a more challenging problem for IEL.

Bibliography

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

[2] Farzindar Atefeh and Wael Khreich. A survey of techniques for event detection in twitter. *Computational Intelligence*, 31(1), 2015.

[3] David Blei, Thomas Griffiths, and Michael Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *Journal of the ACM*, 57(2), 2010.

[4] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, (3):993–1022, 2003.

[5] Chris Buckley, Amit Singhal, Mandar Mitra, and Gerard Salton. New retrieval approaches using smart: Trec 4. In *TREC*, pages 25–48, 1996.

[6] Bokai Cao, Francine Chen, Dhiraj Joshi, and Philip S. Yu. Inferring crowdsourced venues for tweets. *2015 IEEE International Conference on Big Data (Big Data)*, 2015.

[7] Hau-Wen Chang, Dongwon Lee, Mohammed Eltaher, and Jeongkyu Lee. @phillies tweeting from philly? predicting twitter user locations with spatial word usage. *2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2012.

[8] Chen Cheng, Haiqin Yang, Irwin King, and Michael R. Lyu. Fused matrix factorization with geographical and social influence in location-based social networks. *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI)*, pages 17–23, 2012.

[9] Zhiyuan Cheng, James Caverlee, and Kyumin Lee. You are where you tweet: a content-based approach to geo-locating twitter users. *Proceedings of the 19th ACM international conference on Information and knowledge management (CIKM)*, pages 759–768, 2010.

[10] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. Friendship and mobility: User movement in location-based social networks. *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, pages 1082–1090, 2011.

[11] Wen-Haw Chong, Bing Tian Dai, and Ee-Peng Lim. Not all trips are equal: Analyzing foursquare check-ins of trips and city visitors. *Proceedings of the 2015 ACM on Conference on Online Social Networks (COSN)*, pages 173–184, 2015.

[12] Wen-Haw Chong, Bing Tian Dai, and Ee-Peng Lim. Prediction of venues in foursquare using flipped topic models. In *37th European Conference on Information Retrieval (ECIR)*, 2015.

[13] Wen-Haw Chong and Ee-Peng Lim. Collective entity linking in tweets over space and time. *39th European Conference on Information Retrieval (ECIR)*, 2017.

[14] Wen-Haw Chong and Ee-Peng Lim. Exploiting contextual information for fine-grained tweet geolocation. In *Eleventh International AAAI Conference on Web and Social Media (ICWSM)*, 2017.

[15] Wen-Haw Chong and Ee-Peng Lim. Tweet geolocation: Leveraging location, user and peer signals. *Proceedings of the 26th ACM international conference on Information and Knowledge Management (CIKM)*, 2017.

[16] Wen-Haw Chong and Ee-Peng Lim. Exploiting user and venue characteristics for fine-grained tweet geolocation. *ACM Transactions on Information Systems (TOIS)*, 36(3), Mar 2018.

[17] Konstantina Christakopoulou and Arindam Banerjee. Collaborative ranking with a push at the top. *Proceedings of the 24th International Conference on World Wide Web (WWW)*, pages 205–215, 2015.

[18] Justin Cranshaw, Raz Schwartz, Jason I. Hong, and Norman M. Sadeh. The livehoods project: Utilizing social media to understand the dynamics of a city. *Sixth International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2012.

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

[20] Silviu Cucerzan. Large-scale named entity disambiguation based on Wikipedia data. *EMNLP-CoNLL*, 2007.

[21] Ronan Cummins. The evolution and analysis of term-weighting schemes in information retrieval (doctoral dissertation). 2008.

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

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

[24] Jacob Eisenstein, Brendan O'Connor, Noah A. Smith, and Eric P. Xing. A latent variable model for geographic lexical variation. *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1277–1287, 2010.

[25] Géraud Le Falher, Aristides Gionis, and Michael Mathioudakis. Where is the soho of Rome? measures and algorithms for finding similar neighborhoods in cities. *Ninth International AAAI Conference on Web and Social Media (ICWSM)*, 2015.

[26] Yuan Fang and Ming-Wei Chang. Entity linking on microblogs with spatial and temporal signals. *Transactions of the Association for Computational Linguistics*, 2:259–272, 2014.

[27] Paolo Ferragina and Ugo Scaiella. Tagme: On-the-fly annotation of short text fragments (by Wikipedia entities). *Proceedings of the 19th ACM international conference on Information and knowledge management (CIKM)*, pages 1625–1628, 2010.

[28] J.T.A.S. Ferreira, D.G.T. Denison, and D.J. Hand. Weighted naive bayes modelling for data mining. Technical report, Department of Mathematics, Imperial College, 2001.

[29] Víctor Fresno, Arkaitz Zubiaga, Heng Ji, and Raquel Martínez-Unanue. Exploiting geolocation, user and temporal information for natural hazards monitoring in twitter. *Procesamiento del Lenguaje Natural*, 54, 2015.

[30] Jerome H. Friedman, Jon Louis Bentley, and Raphael Ari Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)*, 3(3):209–226, Sep 1977.

[31] Huiji Gao, Jiliang Tang, and Huan Liu. Exploring social-historical ties on location-based social networks. *Sixth International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2012.

[32] Marc G. Genton. Classes of kernels for machine learning: A statistics perspective. *Journal of Machine Learning Research*, 2:299–312, 2001.

[33] Mehmet Gönen and Ethem Alpaydin. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12:2211–2268, 2011.

[34] Bo Han, Paul Cook, and Timothy Baldwin. Text-based twitter user geolocation prediction. *Journal of Artificial Intelligence Research*, 49(1):451–500, 2014.

[35] Xianpei Han and Le Sun. An entity-topic model for entity linking. *2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 105–115, 2012.

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

[37] Neil Houlsby and Massimiliano Ciaramita. Scalable probabilistic entity-topic modeling. *arXiv preprint*, (arXiv:1309.0337), 2013.

[38] Hongzhao Huang, Yunbo Cao, Xiaojiang Huang, Heng Ji, and Chin-Yew Lin. Collective tweet wikification based on semi-supervised graph regularization. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 380–390, 2014.

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

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

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

[42] David Jurgens. That's what friends are for. inferring location in online social media platforms based on social relationships. *Seventh International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2013.

[43] Sheila Kinsella, Vanessa Murdock, and Neil O'Hare. "I'm eating a sandwich in Glasgow": modeling locations with tweets. *Proceedings of the 3rd international workshop on Search and mining user-generated contents (SMUC)*, 2011.

[44] Kisung Lee, Raghu K. Ganti, Mudhakar Srivatsa, and Ling Liu. When Twitter meets Foursquare: tweet location prediction using Foursquare. *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous)*, 2014.

[45] Kenneth Wai-Ting Leung, Dik Lun Lee, and Wang-Chien Lee. Clr: a collaborative location recommendation framework based on co-clustering. *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval (SIGIR)*, pages 305–314, 2011.

[46] Chenliang Li and Aixin Sun. Fine-grained location extraction from tweets with temporal awareness. *Proceedings of the 37th international ACM SIGIR conference on Research and development in information retrieval (SIGIR)*, pages 43–52, 2014.

[47] Wen Li, Pavel Serdyukov, Arjen P. de Vries, Carsten Eickhoff, and M. Larson. The where in the tweet. *Proceedings of the 20th ACM international conference on Information and knowledge management (CIKM)*, pages 2473–2476, 2011.

[48] Xutao Li, Gao Cong, Xiaoli Li, Tuan-Anh Nguyen Pham, and Shonali Krishnaswamy. Rank-geofm: A ranking based geographical factorization method

for point of interest recommendation. *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 433–442, 2015.

[49] Moshe Lichman and Padhraic Smyth. Modeling human location data with mixtures of kernel densities. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, pages 35–44, 2014.

[50] Xiao Ling, Sameer Singh, and Daniel S. Weld. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, 3:315–328, 2015.

[51] Xiaohua Liu, Yitong Li, Haocheng Wu, Ming Zhou, Furu Wei, and Yi Lu. Entity linking for tweets. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1304–1311, 2013.

[52] Zhi Liu and Yan Huang. 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 (CIKM)*, pages 1949–1952, 2016.

[53] Xuelian Long, Lei Jin, and James Joshi. Understanding venue popularity in foursquare. *9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, 2013.

[54] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[55] Kathryn Mazaitis, Richard. C. Wang, Frank Lin, Bhavana Dalvi, Jakob Bauer, and William. W. Cohen. A tale of two entity linking and discovery systems. *Knowledge Base Population Text Analysis Conference.*, 2014.

[56] Edgar Meij, Wouter Weerkamp, and Maarten de Rijke. Adding semantics to microblog posts. *Proceedings of the fifth ACM international conference on Web search and data mining (WSDM)*, pages 563–572, 2012.

[57] Rada Mihalcea and Andras Csomai. Wikify!: linking documents to encyclopedic knowledge. *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management (CIKM)*, pages 233–242, 2007.

[58] David Milne and Ian H. Witten. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. *AAAI Conference on Artificial Intelligence (AAAI)*, 2008.

[59] David M. Mimno and Andrew McCallum. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 411–418, 2008.

[60] Zhao-Yan Ming and Tat-Seng Chua. Resolving local cuisines for tourists with multi-source social media contents. *Multimedia Systems*, 22(4):443–453, 2016.

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

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

[63] Neil O'Hare and Vanessa Murdock. Modeling locations with social media. *Information Retrieval*, 16(1), 2013.

[64] Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpely, Nathan Schneider, and Noah A. Smith. Improved part-of-speech tagging for online conversational text with word clusters. *NAACL*, 2013.

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

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

[67] Francesco Piccinno and Paolo Ferragina. From tagme to wat: a new entity annotator. *Proceedings of the first international workshop on Entity recognition and disambiguation (ERD)*, 2014.

[68] Tatiana Pontes, Gabriel Magno, Marisa A. Vasconcelos, Aditi Gupta, Jussara M. Almeida, Ponnurangam Kumaraguru, and Virglio A. F. Almeida. Beware of what you share: Inferring home location in social networks. *2012 IEEE 12th International Conference on Data Mining Workshops*, 2012.

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

[70] Reid Priedhorsky, Aron Culotta, and Sara Y. Del Valle. Inferring the origin locations of tweets with quantitative confidence. *Proceedings of the 17th ACM conference on Computer supported cooperative work and social computing (CSCW)*, pages 1523–1536, 2014.

[71] Yonggang Qiu and Hans-Peter Frei. Concept based query expansion. *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR)*, pages 160–169, 1993.

[72] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, Feb 1989.

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

[74] Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. Local and global algorithms for disambiguation to Wikipedia. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1 (ACL)*, pages 1375–1384, 2011.

[75] Injong Rhee, Minsu Shin, Seongik Hong, Kyunghan Lee, Seong Joon Kim, and Song Chong. On the levy-walk nature of human mobility. *IEEE/ACM Transactions on Networking (TON)*, 19(3), 2011.

[76] Stephen Roller, Michael Speriosu, Sarat Rallapalli, Benjamin Wing, and Jason Baldridge. Supervised text-based geolocation using language models on an adaptive grid. *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1500–1510, 2012.

[77] Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. Liege: Link entities in web lists with knowledge base. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, pages 1424–1432, 2012.

[78] Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. Linden: Linking named entities with knowledge base via semantic knowledge. *Proceedings of the 21st international conference on World Wide Web (WWW)*, pages 449–458, 2012.

[79] Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. Linking named entities in tweets with knowledge base via user interest modeling. *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, pages 68–76, 2013.

[80] Valentin I. Spitkovsky and Angel X. Chang. A cross-lingual dictionary for english Wikipedia concepts. *LREC*, 2012.

[81] Shiliang Sun. A survey of multi-view machine learning. *Neural Computing and Applications*, 23(7–8):2031–2038, 2013.

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

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

[84] Benjamin Wing and Jason Baldridge. Simple supervised document geolocation with geodesic grids. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, pages 955–964, 2011.

[85] Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR)*, pages 160–169, 1996.

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

[87] Jing Yuan, Yu Zheng, and Xing Xie. Discovering regions of different functions in a city using human mobility and pois. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, pages 186–194, 2012.

[88] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat-Thalmann. Time-aware point-of-interest recommendation. *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval (SIGIR)*, pages 363–372, 2013.

[89] Nayyar A. Zaidi, Jesús Cerquides, Mark James Carman, and Geoffrey I. Webb. Alleviating naive bayes attribute independence assumption by attribute weighting. *The Journal of Machine Learning Research*, 14(1):1947–1988, 2013.

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

[91] Justin Zobel, Alistair Moffat, and Kotagiri Ramamohanarao. Inverted files versus signature files for text indexing. *ACM Transactions on Database Systems (TODS)*, 23(4):453–490, 1998.