Locality-Adapted Kernel Densities of Term Co-occurrences for Location Prediction of Tweets

Ozer Ozdikis[a,*], Heri Ramampiaro[a], Kjetil Nørvåg[a]

[a]*Norwegian University of Science and Technology, Trondheim, Norway*

Abstract

While geographical metadata referring to the originating locations of tweets provides valuable information to perform effective spatial analysis in social networks, scarcity of such geotagged tweets imposes limitations on their usability. In this work, we propose a content-based location prediction method for tweets by analyzing the geographical distribution of tweet texts using Kernel Density Estimation (KDE). The primary novelty of our work is to determine different settings of kernel functions for every term in tweets based on the location indicativeness of these terms. Our proposed method, which we call locality-adapted KDE, uses information-theoretic metrics and does not require any parameter tuning for these settings. As a further enhancement on the term-level distribution model, we describe an analysis of spatial point patterns in tweet texts in order to identify bigrams that exhibit significant deviation from the underlying unigram patterns. We present an expansion of feature space using the selected bigrams and show that it eventually yields further improvement in prediction accuracy of our locality-adapted KDE. We demonstrate that our expansion results in a limited increase in the size of feature space and it does not hinder online localization of tweets. The methods we propose rely purely on statistical approaches without requiring any language-specific setting. Experiments conducted on three tweet sets from different countries show that our proposed solution outperforms existing state-of-the-art techniques, yielding significantly more accurate predictions.

Keywords: Location prediction, Twitter, Kernel Density Estimation, Spatial point patterns

1. Introduction

Geographical information associated with user-generated content in social networks provides a valuable resource for a wide range of applications, including event detection, targeted advertisement, community detection, trend analysis and disaster management (Celik and Dokuz, 2018; Dredze et al., 2016; Ozdikis et al., 2016; Paule et al., 2018b). Twitter is one of the most popular of such social networks, which enables its users to post short text messages as tweets and share them with their followers. Tweets can automatically be geotagged with the actual location of the user at the time of posting if it is supported by the user's GPS-enabled device and its software. However, the amount of such geotagged tweets is reported to be only around 1-3% of the total number of tweets (Cheng et al., 2013; Graham et al., 2014; Paraskevopoulos and Palpanas, 2016). As a result, predicting tweet locations from their texts has recently received considerable attention in order to overcome this scarcity.

Problem Statement: The problem of location prediction for tweets, also referred to as *tweet localization*, can be defined as estimating the geographical origin where a tweet is posted from. Although numerous studies aim to estimate locations at the level of a country or city, the problem becomes more challenging if the objective is to make predictions at finer granularities, e.g., at the level of a street or building within a city, since the number of possible locations is usually much higher (Chong and Lim, 2018; Paraskevopoulos

*Corresponding author.
Email address: ozer.ozdikis@ntnu.no (Ozer Ozdikis)

and Palpanas, 2016; Paule et al., 2018b). In this work, we address the problem of fine-granular localization of tweets and propose a novel content-based method to predict their locations.

Our method treats each term in tweet texts as a separate source of geographical evidence. We train our prediction model according to a set of geotagged tweets from a region that is discretized as a grid, and estimate the most probable grid cell for a given non-geotagged tweet using probability distributions of its terms. The primary novelty of our method is to calculate the probability distributions using a locality-adapted setting of Kernel Density Estimation (KDE) (Silverman, 1986). Our hypothesis is that probability distributions of highly local terms (e.g., street names) should be concentrated around specific areas, whereas more common words (e.g., stop words) should have a more dispersed probability distribution over the entire region. The method we propose assigns a kernel bandwidth and a weight for each term according to its location indicativeness, which we measure using an information-theoretic metric, namely *information gain ratio*. These locality-adapted kernel bandwidths determine the level of concentration and dispersion in the probability distribution of each term without requiring a separate stage of parameter tuning. Term probability distributions over the grid cells are analyzed using tweets in a training set, and the location prediction for a new tweet is then performed according to a weighted combination of probability distributions of its terms.

We improve our model further by taking spatial relationships between co-occurring terms into account. We propose a method to evaluate spatial relationships in the form of *attraction* and *repulsion* between co-occurring terms in tweets based on an analysis of *spatial point patterns* (Diggle, 2003). Selected term pairs that exhibit statistically significant clustering or dispersion tendency with respect to the underlying unigram distributions are also included in the feature space to improve the accuracy of predictions. To explain our idea, consider an example where we want to predict the location of a tweet mentioning *heathrow*. The term *heathrow* can be considered to provide strong geographical evidence supporting the region around the Heathrow Airport in London. In this tweet, if *heathrow* is also followed by the term *terminal*, we could make even more precise estimations since the resulting bigrams would have a stronger clustering pattern compared to *heathrow* alone. On the other hand, if the tweet mentions *heathrow express* or *heathrow shuttle*, it is more likely to have been posted somewhere away from the airport, probably referring to the bus that rides to the airport. Such term pairs can have a dispersion effect and repel the geographical focus of the tweet to a region away from a specific geographical area. Therefore, as opposed to considering each term independently from each other, we first detect spatially significant bigrams in texts and extend our feature space accordingly in order to make more accurate predictions.

The contributions of this paper can be summarized as follows:

- The problem of tweet localization is investigated using probability densities of textual features in tweets, which include unigrams and bigrams that are selected based on their spatial attraction and repulsion patterns.
- We present kernel density estimators with settings determined according to the location indicativeness of texts.
- The proposed method is completely based on statistical analysis of tweet texts. It does not require any external data source, any separate stage of parameter tuning, or any assumption about the language of tweets.
- We perform an extensive comparative study to evaluate our approach. Our experimental results show that the proposed method can estimate tweet locations with significantly higher accuracy in comparison to the state-of-the-art baselines, including neural networks.

We investigated density estimators to model the probability distributions of terms in (Ozdikis et al., 2018a). The unigram model that we briefly explained in that paper yielded promising results. In this work, we enhance our density-based prediction approach by extending the feature space with bigrams selected by an analysis of spatial point patterns. In (Ozdikis et al., 2018b), we studied spatial point patterns of terms to improve the accuracy of Naive Bayes classifiers. In the current work, we describe its adaptation to our density-based tweet localization method, which eventually improved the accuracy of predictions

significantly. In addition to providing more detailed explanations and new examples about our techniques, we study alternative bigram selection techniques and conduct new experiments related to the size of feature space. We examine the usage of term pairs that co-occur in tweets irrespective of their arrangement, and demonstrate that using bigrams is a better alternative than using non-sequential co-occurrences. We also present our findings on the usage of bigrams in different prediction models for tweet localization. Evaluations and discussions have been particularly extended to compare our method with additional baselines, including a neural networks solution that is reported to achieve the best median distance error at the Twitter geolocation prediction shared task in W-NUT'16 (Han et al., 2016; Miura et al., 2016).

The rest of this paper is organized as follows: We present our research objectives in Section 2, which is followed by a review of the literature about location estimation in Section 3. Section 4 is devoted to our proposed tweet localization method, which includes our locality-adapted kernel density estimation and our analysis of spatial point patterns in bigrams. The results of our evaluations, comparisons with the baselines, and alternative settings of our methods are given in Section 5. We discuss our findings and possible enhancements in Section 6, and finally conclude the paper in Section 7.

2. Research Objectives

Based on the definition of tweet localization problem, this work aims to answer the following research question: *How to develop an effective method to predict the locations of tweets based on their textual contents?* Accordingly, the main objectives of this study can be summarized as follows:

1. **Fine-granular prediction accuracy:** Our main objective is to develop a content-based method that is able to estimate the posted locations for tweets at fine-granular level (e.g., within 1km) with highest possible accuracy.

2. **Analysis of spatial relationships in text:** We aim to develop statistical methods to distinguish specific patterns in geographical properties of term co-occurrences in tweet texts.

3. **Practicality and generalizability:** Our methods should be practical to apply on different tweet datasets without any assumption about the language of tweets and without any need for a gazetteer, an external location-based service or a data-specific parameter tuning.

4. **Online prediction support:** Considering wide range of possible locations in the search space in a fine-grained setting and the variety of textual content in tweets, we aim to maintain the applicability of our methods for online prediction.

We present a comprehensive literature review of related studies and the current state of the art in the following section. Recent studies on the field suggest that tweet localization is an active research area with various approaches proposed to solve the problem. These approaches can vary depending on the targeted prediction granularity, tweet attributes that are used in analysis, and practicality of the developed methods. Our methods differ from previous studies by applying purely statistical techniques on the tweet text to yield higher fine-granular prediction accuracies without any assumption about the language of tweets. Common strategies to improve accuracy in previous studies usually include utilizing external location-bases services or using additional tweet attributes (Bakerman et al., 2018; Paraskevopoulos and Palpanas, 2016; Schulz et al., 2013). An important research question that we aim to answer in this work is *whether it is possible to improve the accuracy by performing further analysis on tweet texts alone,* and the results of our experimental evaluations suggest *yes*. The reason for our focus on content-based approaches is that tweet texts can provide fine-granular evidence about the location even in the absence of any other geographical cues (Cheng et al., 2013). Although our methods can further be extended to include other sources of spatial evidence, we show that we can achieve significant accuracy improvement even if we use the tweet text alone, without creating any dependency on additional data.

Our research differs from previous studies also in the way we determine textual features and use them in our probabilistic analysis. Modeling probability distributions of features on a geographical area has

been experimented before in several previous studies (Bakerman et al., 2018; Hulden et al., 2015; Lichman and Smyth, 2014; Priedhorsky et al., 2014; Zhang and Chow, 2013). There may be variances in problem definitions, specific implementations and tweet attributes used in computations. However, we observe that a common challenge in these models is related to tuning of several parameters in the density estimations, such as the number of components in mixture models, covariances of variables and feature weights. Our method requires selecting values for two parameters, i.e., bandwidth of kernel functions and probability weights, and we describe how we determine these values automatically based on the location indicativeness of text features without performing any specific parameter tuning. We show that our proposed method is generalizable to different datasets yielding high prediction accuracy.

Bigrams have previously been used in various text classification and location estimation tasks (Doran et al., 2014; Flatow et al., 2015; Han et al., 2014; Joulin et al., 2017; Melo and Martins, 2017; Priedhorsky et al., 2014). We observe that including bigrams in location analysis without considering their geographical characteristics may not always improve the results. For example, Han et al. (2014) noted that their preliminary results with high order n-grams were disappointing in their Naive Bayes prediction model. In consistence with this reported comment, our experiments also revealed that using all bigrams in texts does not necessarily increase the prediction accuracy. Therefore, different from the previous studies, we examine spatial characteristics of term co-occurrences in texts and we show that an effective selection of bigrams, as we proposed in this paper, is beneficial to make better estimations. To the best of our knowledge, this is the first comprehensive analysis of spatial properties of term co-occurrences in tweets and their application in different prediction models for tweet localization.

3. Related Work

Numerous studies have recently been conducted to perform spatial analysis on Twitter and estimate the locations for tweets, users and even real-world events (Han et al., 2016; Ozdikis et al., 2016, 2017; Poulston et al., 2017; Yamaguchi et al., 2014; Zheng et al., 2018). Although some of the methods in these studies can be similar and used interchangeably, tweet localization aims to estimate the originating location for a single tweet, whereas the predictions for users and events are usually made based on a collection of tweets (Dredze et al., 2016). In this work, we address the problem of tweet localization. Since geographical origins of tweets provide a valuable resource to perform further information extraction from Twitter, it has been an increasingly active research area in recent years (Bakerman et al., 2018; Li et al., 2018; Ozdikis et al., 2018a,b; Paule et al., 2018a,b).

3.1. Tweet localization

Proposed methods for tweet localization can vary depending on the expected prediction granularity, selected tweet attributes and utilized data sources. For example, Schulz et al. (2013) described a multi-indicator approach that combines different tweet features, such as tweet text, timezone and user profile. The authors also used external data sources, such as DBpedia, Geonames, and Foursquare, in order to resolve toponyms in texts. A similar approach that used location based services and Geonames is presented in (Jayasinghe et al., 2016). Short and non-standart language in tweets makes it particularly difficult to find explicit references to place names in their texts, even if external gazetteers are used in analysis (Hoang and Mothe, 2018; Middleton et al., 2018; Schulz et al., 2013). Bakerman et al. (2018) combined tweet text with network data that is composed of previous tweets initiated by users' friends. Another hybrid method is given in (Rodrigues et al., 2016). In that work, the authors proposed a Markov random field probability model to infer users' locations based on the content of their tweets and their friendship networks. Compton et al. (2014) estimated users' locations by examining the locations of their friends and by searching for a network to minimize the total variation of distances between connected users. Ebrahimi et al. (2017) presented another network-based approach by using references to local celebrities as location indicators. Rout et al. (2013) applied a classification approach in order to estimate users' home locations in terms of city based on the locations of their friends and followers in Twitter. Chong and Lim (2019) focused on geolocating tweets that are posted by the same user and within a short time interval. They analyzed staying and visiting behaviors of users and proposed a model that performs query expansion on tweets.

In other studies that utilized different tweet features for localization, Priedhorsky et al. (2014) used a combination of message text, timezone, language, user profile and user description, and discussed which fields provide more useful location information. Mahmud et al. (2014) presented an ensemble of statistical and heuristic classifiers that use previous tweets of a user, tweet times and place names mentioned in tweets in order to estimate the home location of Twitter users in terms of city. Zubiaga et al. (2017) investigated the usefulness of eight tweet-inherent features for the country-level classification of tweets. The authors reported that the selection of an appropriate combination of features leads to an accuracy improvement, while the tweet content alone is identified as the most useful feature in their experiments with 25 countries. Utility of features may also depend on the expected prediction granularity. For example, tweet language and timezone may not provide much useful information if the objective is to localize a tweet inside a city. Similarly, location field in the user profile mostly provides coarse granular information, such as at the level of a country or city (Giridhar et al., 2015). Moreover, tweet metadata such as timezone or user profile can be incomplete and incorrect, which limits their use for fine-grained prediction (Bakerman et al., 2018; Hecht et al., 2011). Last but not least, querying external data sources or tweet histories of specific users can impose limitations due to dependencies, query rate limits, and online execution. Therefore, tweet text is mostly used as the primary resource of geographical evidence for tweet localization.

3.2. Content-based methods

Recent efforts for content-based geolocation of tweets apply various techniques from information retrieval, machine learning and natural language processing in order to improve the accuracy of predictions (Ajao et al., 2015; Jurgens et al., 2015; Melo and Martins, 2017; Zheng et al., 2018). A common approach for location prediction of texts is to treat the problem as a classification task. In this approach, the region of interest is modeled as a grid and the most probable grid cell for a document is predicted according to a training set of geotagged items. State-of-the-art methods mostly use Naive Bayes classifiers (Chi et al., 2016; Han et al., 2014; Hulden et al., 2015), discriminative models such as Kullback-Leibler divergence (Hulden et al., 2015; Roller et al., 2012; Wing and Baldridge, 2014), neural networks (Li et al., 2018; Miura et al., 2016; Rahimi et al., 2017) and geographic probability distributions (Bakerman et al., 2018; Cheng et al., 2013; Priedhorsky et al., 2014).

In other recent studies, Paule et al. (2018b) proposed a majority voting approach that uses the locations of most similar tweets as geolocation votes, which was later improved with a learning-to-rank method using several features in tweets (Paule et al., 2018a). Wing and Baldridge (2014) employed a logistic regression classifier for text-based geolocation of documents. In that work, the authors also discussed the scalability limitations of logistic regression for large number of classes, as is the case with fine-granular localization of tweets. Eisenstein et al. (2010) used topic models to analyze the relationships between geographical regions and latent topics in microblog posts. Krishnamurthy et al. (2015) proposed a knowledge based solution in order to predict the city of a Twitter user. In that work, the authors identified references to Wikipedia entities in user's tweets and calculated a localness score for each city according to the locations of detected entities. Miura et al. (2016) designed a content-based neural networks model and achieved the best median error distance in a shared task on tweet localization (Han et al., 2016). This neural networks solution is also among our baselines and we explain it in more detail in our experimental evaluations.

Paraskevopoulos and Palpanas (2016) proposed a fine-grained prediction method that analyzes the similarity of tf-idf vectors generated for keywords in tweets. The authors also presented an extension to that method by incorporating tweet time in their analysis. Flatow et al. (2015) described a method to identify phrases in tweets that are associated with small areas. Location estimation was then performed for tweets that included any of these phrases. In another study targeting fine-grained localization, Chong and Lim (2018) introduced a learning-to-rank framework in order to make estimations in terms of Foursquare venues. Estimations were made based on Foursquare check-ins of users, their tweet histories and temporal popularity of venues. Lee et al. (2014) extracted a list of venues from Foursquare check-ins in a city in order to build a high-quality location model for fine-grained location estimation of tweets. The authors built probabilistic language models for venues using text messages associated with the check-ins. They performed location prediction only for tweets that may be related to a location, which are determined according to a list of local keywords. Doran et al. (2014) presented an ensemble of language models for location prediction of social

media posts inside a city. They introduced a geo-smoothing function to capture the influence of language on neighboring regions.

Feature selection techniques that identify and prioritize strong location indicative terms have been proposed to improve accuracies of several probabilistic models (Han et al., 2014; Ozdikis et al., 2018b; Van Laere et al., 2014). For example, Cheng et al. (2013) determined local words according to an analysis of frequency and dispersion. In (Han et al., 2014), the authors experimented with numerous feature selection methods, such as information gain ratio, χ^2 statistic, geospreading, and Ripley's K function, and showed that the selection using information gain ratio performed best in terms of accuracy. In that work, location indicative features that are selected according to their information gain ratio are used in a Naive Bayes classifier. We present further information about their approach in Section 5.1.2, where we describe the baseline methods in our evaluation. Han et al. (2014) also demonstrated that models trained on geotagged tweets are also applicable to non-geotagged data. In a similar study, Van Laere et al. (2014) employed Ripley's K function in order to improve the performance of location estimation for Flickr photos, particularly when only few terms can be selected for prediction.

Ripley's K function is a spatial analysis method that is used to measure the deviation of a point pattern from spatial homogeneity (Diggle, 2003; Ripley, 1977). In addition to feature selection (Van Laere et al., 2014), it has also been used to study the interaction between two spatial point patterns and retrieve event-related Flickr images (Ruocco and Ramampiaro, 2015). In our previous study (Ozdikis et al., 2018b), we used Ripley's K function to improve the accuracy of Naive Bayes classifiers. In this work, we present its adaptation to our locality-adapted kernel density estimation and experiment it on datasets from three different regions of the world.

3.3. Probability density functions

Probability density functions are used to estimate the probability density of a continuous random variable at a given point based on previous observations (Chen, 2017; Silverman, 1986). Gaussian Mixture Models (GMM) and Kernel Density Estimation (KDE) are two mature and widely used techniques to estimate probability densities. GMM has been used to estimate users' home locations in (Chang et al., 2012) and to estimate tweet locations in (Priedhorsky et al., 2014). Lichman and Smyth (2014) evaluated it as a baseline method to model user check-ins. Recently, Bakerman et al. (2018) proposed a hybrid approach for location prediction that combines GMMs for text features and network features. However, determining the number of components and tuning the mixture weights in GMMs can pose additional challenges since the number of Gaussian components can vary considerably across different datasets. Therefore, KDE is usually considered a strong alternative to GMM in the estimation of probability densities.

KDE makes less rigid assumptions about the distribution of the observed data, which makes it more suitable for arbitrary data distributions (Silverman, 1986). Its advantages over GMMs are widely discussed in (Lichman and Smyth, 2014; Zhang and Chow, 2013). Lichman and Smyth (2014) developed a mixture-KDE approach to predict individuals' locations according to their activity history. The authors calculated a weighted combination of three density distributions, namely at individual-level, region level and population level. They tuned parameters in their methods using a validation data set and applied the selected values for all users. In our content-based tweet localization problem, where we have thousands of text features to combine, we determine kernel bandwidths and weights automatically according to the locality strength of each term separately. Zhang and Chow (2013) used KDE to model the geographical distribution of a user's visited locations in order to make recommendations in location-based social networks. In (Hulden et al., 2015), the authors addressed data sparsity problem in grid-based models for text localization, and applied KDE to smooth out the counts of documents and words over the region in order to improve the accuracy of estimations. Van Laere et al. (2014) proposed a feature selection technique using KDE to geotag Flickr photos and Wikipedia articles. In (Lu et al., 2015), kernel density estimation has been used to generate visualizations on the map.

Our approach differs from previous tweet localization studies by modeling the geographical distribution of textual features in tweets using locality-adapted kernel density estimators, where we define kernel settings according to the location indicativeness of each feature separately. In addition to unigram features in tweets, we also use bigrams that exhibit spatial clustering or dispersion tendency with significant deviation from

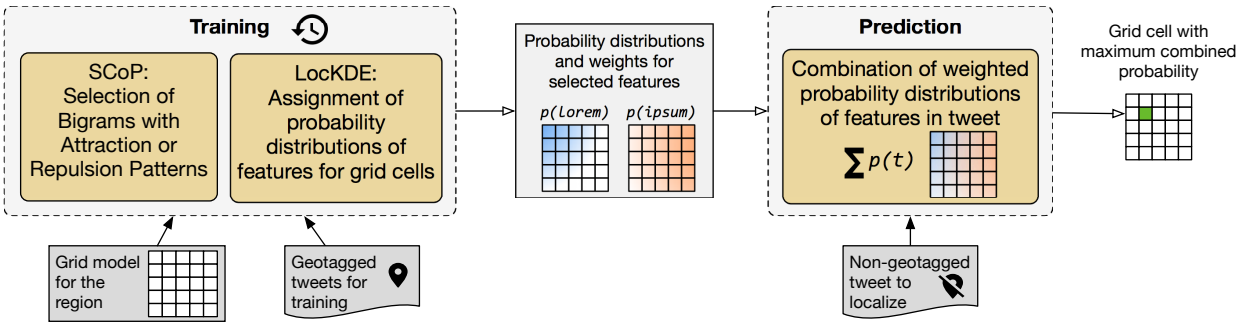

Figure 1: System Overview: Training and prediction stages in the proposed localization process.

the underlying unigram patterns. Probability distributions of selected features, which are calculated as integrated densities over the grid cells, are weighted and combined based on feature localities measured by information gain ratio. We determine kernel bandwidths and probability weights using statistical techniques, without requiring any parameter tuning. Finally, we note that although we used only the tweet text for prediction, our model can be extended to include the distributions of additional tweet features as well.

4. Tweet Localization

Following the common practices in the literature, our location estimation method models the region of interest as a grid and discretizes the geographical area into smaller uniform-size cells (Doran et al., 2014; Hulden et al., 2015; Wing and Baldridge, 2014). In this setting, predicting the location of a non-geotagged tweet is described as finding the most probable grid cell and assigning its geographical centroid in terms of latitude-longitude as the location for that tweet (Paule et al., 2018a; Roller et al., 2012). These predictions are made based on a given set of geotagged tweets for training.

A high-level overview of our localization process is depicted in Fig. 1. As shown in the figure, the training stage in our proposed method consists of two major steps, namely 1) selection of bigrams with attraction or repulsion patterns, and 2) calculation of probability distributions of features over the grid cells using KDE. In the remainder of this paper, we use the abbreviation *SCoP* to refer to our method for the selection of bigrams using spatial co-occurrence patterns. Our locality-adapted KDE that we propose for the second step of training will be denoted as *LocKDE*.

We first describe the calculation of probability distributions using LocKDE in Section 4.1. Then, we present SCoP in Section 4.2, since selection of bigrams using SCoP is an improvement over LocKDE by extending the feature space. The result of these two steps in the training stage is the probability distributions of selected features over the grid cells, as shown in Fig. 1. In the prediction stage, location for a non-geotagged tweet is estimated according to a weighted combination of probability distributions of the textual features in that tweet. We present how we combine probability distributions for the location prediction of a non-geotagged tweet in Section 4.3.

In the rest of the paper, we use c to represent a cell in grid C, which partitions a given region of interest. X_t denotes the set of tweets in our training set that include the term t in their texts. We use $x{=}\langle t_x, l_x\rangle$ to represent a geotagged tweet, where t_x denotes the list of terms in x, and l_x is its coordinates in terms of latitude-longitude. *Geographical distribution of a term* refers to the geographical distribution of tweets mentioning that term.

4.1. LocKDE: Assignment of probability distributions using locality-adapted KDE

Our prediction method is based on the hypothesis that a tweet mentioning a term t is more likely to have been posted from the spatial proximity of other tweets that also mentioned t. Accordingly, if we calculate the probability to observe a term in a grid cell, we can estimate the likelihood for a new tweet to be posted from

that cell by combining the probability distributions of its terms. The method we apply for the calculation of probability distributions is KDE. It is a statistical tool that is widely used to estimate the probability density function of a random variable based on a finite data sample (Chen, 2017; Lichman and Smyth, 2014; Silverman, 1986). In our content-based tweet localization problem, we calculate the probability density for a term t at a location l using the density function $\hat{f}$ in Eq. 1.

$$\hat{f}_t(l) = \frac{1}{|X_t|h} \sum_{x \in X_t} K\left(\frac{l - l_x}{h}\right) \tag{1}$$

In this equation, $K(\cdot)$ represents the *kernel function* and h denotes the *bandwidth* (also known as the *smoothing parameter*) controlling the sharpness-smoothness of the density distribution. We apply the Gaussian kernel given in Eq. 2, which has also been widely adopted in similar previous studies (Chen, 2017; Lichman and Smyth, 2014; Silverman, 1986; Zhang and Chow, 2013). In our implementation, we use the `gaussian_kde` class provided by the SciPy[1] library.

$$K\left(x\right) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \tag{2}$$

Selection of an optimal bandwidth plays a critical role in KDE, since it directly affects the smoothness of the density distribution. Higher bandwidth values result in smoother graphs, whereas the density estimations with a lower bandwidth would have sharper peaks. Several techniques can be employed to choose the bandwidth h in KDE (Silverman, 1986). One alternative is to assume a normal distribution for the data and derive a bandwidth value from the standard deviation of samples (Zhang and Chow, 2013). However, this approach is noted to perform well only if the data really is normally distributed, and it over-smooths the estimations in non-unimodal distributions (Silverman, 1986). Van Laere et al. (2014) also stated that an optimal value chosen by such methods might not always be appropriate since in some cases it may be more beneficial to have a higher level of smoothing than is inherent in the data. Another alternative is to search for an optimal bandwidth value by tuning it in a validation process (Hulden et al., 2015; Lichman and Smyth, 2014). However, considering that there may be thousands of distinct terms in training tweets, it may not be practical to perform this tuning for every term in the feature space. Moreover, this method is prone to suffer from data sparsity. If there are not sufficiently many samples for every term, it may not result in reliably optimal bandwidths.

There are simpler rule of thumb methods for bandwidth selection, some of which are also available in the SciPy library (Chen, 2017; Silverman, 1986; Zhang and Chow, 2013). Although they may not be strictly accurate for every distribution, they are practical to apply with any number of tweets without requiring a separate parameter tuning stage. The default setting of the `gaussian_kde` in SciPy library uses one of these rule of thumb methods, namely the Scott's rule, which assigns $h_{Scott}(t)=|X_t|^{-1/6}$ in a bivariate setting. In this work, we first evaluate our location prediction method using Scott's rule, and propose an enhancement for term-specific bandwidth selection in order to improve the prediction accuracy. Since our method is based on an evaluation of location indicativeness of terms, we call it *locality-adapted bandwidths*.

4.1.1. Locality-adapted kernel bandwidths

Location indicativeness of textual features in documents is mainly studied in the context of feature selection (Chang et al., 2012; Han et al., 2014; Van Laere et al., 2014). These studies reveal that different terms in tweets can have different locality strengths. In other words, while some words can be very descriptive in predicting the geographical origin of a tweet (e.g., street names, venues), some may have almost no geographical indication (e.g., stop words). Based on this observation, our claim is that the prediction accuracies can be improved if the bandwidth of the kernel function is adapted according to the location indicativeness of terms. Specifically, kernel function of a term with strong locality should be given a lower bandwidth so that its density distribution concentrates on the local neighborhood of observation points,

[1] https://pypi.python.org/pypi/scipy/0.19.1

while weakly local terms should have a higher bandwidth to have less peaky density distribution over the entire region. Our hypothesis is that such an adaptation that is inversely proportional to the spatial strength of each term would improve the accuracy of our KDE-based predictions.

The method we propose to obtain locality-adapted bandwidths uses an information-theoretic metric, namely the *information gain ratio (IGR)*, which has been found to be an effective feature selection metric to obtain location indicative terms in a document corpus (Chi et al., 2016; Han et al., 2014; Melo and Martins, 2017; Ozdikis et al., 2018b). It is measured as the ratio of information gain (IG) of a term t to its intrinsic entropy. Calculation of IG is given in Eq. 3, where $P(t)$ and $P(\bar{t})$ denote the probabilities of presence and absence of a term t, respectively, and $P(c)$ represents the ratio of tweets in c to the number of all training tweets.

$$IG(t) = P(t)\sum_{c\in C} P(c|t)\log P(c|t) + P(\bar{t})\sum_{c\in C} P(c|\bar{t})\log P(c|\bar{t}) - \sum_{c\in C} P(c)\log P(c) \tag{3}$$

Eq. 4 shows the calculation of the IGR for a given term t. The denominator in this equation represents the intrinsic entropy of t over the region.

$$IGR(t) = \frac{IG(t)}{-P(t)\log P(t) - P(\bar{t})\log P(\bar{t})} \tag{4}$$

We calculate IGR for every distinct term in the training set in order to evaluate their location indicativeness. The reason for selecting IGR is twofold. Firstly, previous studies show that IGR yields the most accurate results in location estimation among other feature selection techniques, such as IG, χ^2 and geospreading (Han et al., 2014; Ozdikis et al., 2018b). Secondly, IGR values range between 0 and 1, where a more location indicative term is expected to have a higher IGR value. In practice, we observe that the most local terms in our training sets are assigned the IGR value of 1, whereas the least local term has an IGR value around 0.05.

We introduce the setting in Eq. 5 to adapt the kernel bandwidth of a term t according to its location indicativeness. This setting adapts the value assigned by Scott's rule (i.e., h_{Scott}) specifically for t as being inversely proportional to the locality of t represented by its IGR value.

$$h_{IGR}(t) = h_{Scott}(t) \times (1 - IGR(t) + \lambda) \tag{5}$$

In this equation, λ represents the minimum IGR value found in the training set, i.e., $\lambda{=}min_{t'\in T} IGR(t')$, which is practically around 0.05 as mentioned above. As a result of this setting, the locality-adapted bandwidth $h_{IGR}(t)$ for the most local term would be decreased by a ratio of λ, and for the least local term, it would be equal to $h_{Scott}(t)$ without any change. The λ value in Eq. 5 also ensures non-zero bandwidth for terms having IGR=1. We present the improvement in accuracy obtained by this setting and provide illustrative examples about the effect of this tuning in our evaluations in Section 5.3.

4.1.2. Integration of densities

The locality-adapted bandwidths are used in probability density function $\hat{f}_t$ given in Eq. 1. Density functions find the density of probability at a given point. In order to calculate the probability of observing a term t in a grid cell c, the density values must be integrated over the area of c, such that $p_t(c) = \iint_c \hat{f}_t(lat, lon)\, d_{lat}\, d_{lon}$ (Lichman and Smyth, 2014; Silverman, 1986). Accordingly, once the probability density functions are initialized for each term, the next step in our training is to assign probability masses to grid cells by applying the `integrate_box` function provided in the `gaussian_kde` class. As a result, using the boundary coordinates of grid cells, we obtain $p_t(c)$ values for grid cells for every term in our feature space.

It could also be an alternative to use density values at midpoints of grid cells instead of integrating densities over the cell areas (Hulden et al., 2015; Van Laere et al., 2014). However, in our location prediction problem, selecting a single point inside a cell may not actually represent the real probability mass that should be assigned for that area, since every point inside a grid cell can have a different density value. Moreover, that approach would also be highly sensitive to the granularity of the grid, as larger grid cells would result

in fewer sample points to calculate densities. Therefore, rather than using densities at certain points, we calculate integrated densities over the cells to obtain aggregated values. As a result, this approach provides us more reliable probability masses that lie within the range of [0,1], which are also more interpretable compared to the density values.

4.2. SCoP: Selection of bigrams based on spatial co-occurrence patterns

In Section 4.1 above, we explained the assignment of probability distributions to grid cells for a feature space that consists of distinct terms in tweets. In this section, we investigate the extension of feature space with a selection of spatially significant bigrams in tweets. Our hypothesis is that even if a term is highly location-indicative, another term that precedes or succeeds it may change its geographical interpretation. Therefore, considering these two terms together as a bigram and including it in the calculation of probability distributions should improve the accuracy of predictions. In Section 4.2.1, we present our method to detect spatially significant bigrams, and in Section 4.2.2, we describe how we use them in the enhancement of feature space for location prediction. As we show in our evaluations, this extension improves the accuracy of predictions without incurring remarkable increase in the size of feature space.

4.2.1. Analysis of spatial point patterns in bigrams

The method we propose for the selection of bigrams relies on an analysis of spatial point patterns using Ripley's K-function, a widely adopted tool to analyze the distribution patterns of objects in two-dimensional space (Ripley, 1977; Ruocco and Ramampiaro, 2015; Van Laere et al., 2014). The function calculates a value that is proportional to the number of point pairs that lie within a distance of δ to each other. The K_δ function that we use in our implementation is given in Eq. 6. In this equation, $area(C)$ represents the area of our grid, $d(x_i, x_j)$ is the distance between two tweets x_i and x_j according to their coordinates, δ is a numerical value that enables the evaluation of spatial relationships at different distance scales, and e_i represents the edge correction coefficient for x_i. Larger values of K_δ are obtained if tweets in X_t have higher concentration around a small region, which can be considered as an attraction pattern (i.e., clustering tendency). Lower values would result in a distribution where tweets are distant from each other, which can be interpreted as a repulsion pattern (i.e., dispersion tendency).

$$K_\delta(X_t) = area(C) \times \frac{\sum_{x_i \in X_t} (e_i \times |\{x_{j \neq i} \in X_t \mid d(x_i, x_j) < \delta\}|)}{|X_t|^2} \tag{6}$$

As discussed in (Goreaud and Pélissier, 1999), K_δ values may not capture high concentrations at the boundaries of a study area. Therefore, we use edge correction coefficient e_i to account for tweets that are posted from points closer than δ to the boundary of our grid. In the calculation of e_i for a tweet x_i, we consider a square region b_i that is centered at l_{x_i} and has an edge of length 2δ. Then we calculate e_i as the ratio $area(b_i)/area(b_i \cap C)$. As a result, if part of b_i falls outside the grid C, the area of intersection $area(b_i \cap C)$ takes a smaller value than the area of b_i, resulting in $e_i > 1$. In practice, there are few tweets located near the boundary of C, and therefore in most cases, the region b_i falls completely inside the grid and e_i becomes equal to 1.

We employ Ripley's K-function to compare the spatial distributions of bigrams and unigrams, so that we can identify bigrams whose distributions significantly deviate from the underlying unigram patterns. More specifically, given that X_{t_i} represents the tweets mentioning a unigram t_i, our objective is to find if its subset $X_{t_i t_j}$ (i.e., tweets that contain the bigram $t_i t_j$) has a significantly different spatial distribution pattern than X_{t_i}. In order to evaluate the clustering and dispersion tendency of $X_{t_i t_j}$ with respect to the underlying distribution of X_{t_i}, we execute a stochastic process, namely Monte Carlo simulation (Diggle, 2003; Ripley, 1977). The simulation consists of taking random samples from X_{t_i}, calculating K_δ for these samples, and forming a confidence envelope with upper and lower bounds. A bigram with K_δ value that is above the upper bound indicates clustering tendency (attraction), whereas a K_δ value below the lower bound is interpreted as dispersion (repulsion) with respect to t_i.

The steps of our method to select spatially significant bigrams using Monte Carlo simulation is presented in Algorithm 1. For the set of unigrams T in training set, the algorithm finds bigrams $t_i t_j$ such that the

Algorithm 1 Find bigrams with attraction or repulsion with respect to the first term in the bigram.

1: **Input1:** T: Set of distinct unigrams in training tweets
2: **Input2:** δ: Distance range for Ripley's K-function
3: **Input3:** m: Number of Monte Carlo simulations to execute
4: **Output:** B_{SCoP}=$\{t_i t_j \mid X_{t_i t_j}$ has clustering or repulsion tendency with respect to $X_{t_i}\}$
5: **for each** term t_i in T **do**
6: **for each** co-occurring term t_j in T **do**
7: Find the set of tweets $X_{t_i t_j}$ for which t_i is followed by t_j in the tweet text
8: Apply K-function in Eq. (6) on $X_{t_i t_j}$ to get $K_\delta(X_{t_i t_j})$
9: **for** i=1...m **do**
10: Randomly sample n tweets from X_{t_i}, where n=$|X_{t_i t_j}|$
11: Let $X^i_{t_i}$ denote this sample, apply K-function on $X^i_{t_i}$ to find $K_\delta(X^i_{t_i})$
12: **end for**
13: Calculate upper (u) and lower (l) boundaries of envelop using $K_\delta(X^i_{t_i})$ values with 0.05 confidence interval
14: **if** $K_\delta(X_{t_i t_j}) > u$ **or** $K_\delta(X_{t_i t_j}) < l$ **then**
15: Insert $t_i t_j$ to the set of selected bigrams B_{SCoP}
16: **end if**
17: **end for**
18: **end for**

co-occurrence of t_j following t_i exerts an attraction or repulsion influence on t_i. If the tweets that include $t_i t_j$ has significantly higher K_δ value compared to tweet samples mentioning t_i alone, $t_i t_j$ is regarded to have a clustering tendency in relation to t_i. Similarly, if the $K_\delta(t_i t_j)$ value is below the lower boundary, $t_i t_j$ is selected as a repulsion bigram for t_i. Significant bigrams that are identified using this algorithm are returned in a set denoted by B_{SCoP}. The steps in Algorithm 1 describe the analysis of bigrams with respect to the first term in the bigram. Similar procedures are followed to identify spatially significant bigrams with respect to the second term as well. Ordering of terms in bigrams should be taken into consideration since we examine the distribution of a bigram conditioned on the distribution of a unigram.

As stated in (Van Laere et al., 2014), K_δ for two sets of objects with different number of elements would not be comparable, since the larger set is more likely to obtain higher K_δ values by chance. We do not observe this issue in our algorithm. Each iteration of the simulation takes exactly n=$|X_{t_i t_j}|$ samples from X_{t_i} (line 10), which satisfies comparable K_δ values for bigrams and their corresponding unigrams. This strategy is also advantageous in terms of computational cost. In fact, except for a few term pairs that co-occur very frequently (e.g., *United Kingdom*), we observe that n=$|X_{t_i t_j}|$ is remarkably lower than $|X_{t_i}|$, which means that K_δ values are computed for small samples from X_{t_i}. Moreover, in order to enable quick lookup of nearest neighbors in our implementation, we transform latitude-longitude coordinates of tweets into three-dimensional Euclidean coordinates and index them in a k-d tree[2] (Roller et al., 2012; Van Laere et al., 2014). We also parallelize the computation of K_δ values, since there is no dependency or sequential relationship in the spatial analysis of bigrams. These settings provided us noticeable performance improvement in the execution of our algorithm. We provide examples for bigrams detected by SCoP in Section 5.5. The next section explains how we use these detected bigrams in the extension of feature space.

4.2.2. Extending the feature space

The extension of feature space is performed by using bigrams from B_{SCoP} as additional features in tweets. The extension operation is executed as follows: Let x represent a tweet with n terms in its text, denoted by $[t^1_x, t^2_x, ...t^n_x]$. If a bigram $t^i_x t^{i+1}_x$ is found to have a significantly different distribution with respect to t^i_x, we remove t^i_x and insert the bigram $t^i_x t^{i+1}_x$ as a new feature in tweet. We follow the same steps for the second term, i.e., we check the relationship of the bigram with t^{i+1}_x and replace it if the bigram has significantly different distribution with respect to t^{i+1}_x. Finally, the bigrams that are included in the extension of tweets are also included in the feature space for training. Our rationale in this operation is that if a term changes

[2]https://docs.scipy.org/doc/scipy-0.19.1/reference/generated/scipy.spatial.cKDTree.html

the spatial interpretation of its preceding or succeeding term, considering these two terms together as a new feature provides more reliable spatial evidence and improves the accuracy of predictions.

We exemplify the extension operation on a hypothetical tweet with terms [a,b,c,d]. Assume that the bigram ab was found to have a significantly different spatial pattern with respect to the distribution of a using Algorithm 1. In this case, applying extension on this tweet results in [b,c,d,ab]. If B_{SCoP} also includes the bigram bc with respect to c, the unigram c is similarly replaced with bc to yield [b,d,ab,bc]. We would like to point out that we employ a replacement strategy, which is different from (Ozdikis et al., 2018b), where bigrams were inserted without replacement. The replacement of features as described above resulted in higher accuracies in our experiments.

For a non-geotagged tweet to be localized, we first apply the similar feature extension operation using the bigrams in B_{SCoP}. Location prediction for that tweet is then performed by combining the probability distributions of its features and selecting the grid cell with maximum combined probability. The following section describes the details of this prediction stage.

4.3. Prediction by the combination of probabilities

Location prediction for a new tweet is performed by combining the probability distributions of its features (i.e., unigrams and selected bigrams) and finding the grid cell maximizing the cumulative probability. Since different features can have different spatial representative strength, we adopt a weighted sum approach and apply Eq. 7 for combination (Lichman and Smyth, 2014; Priedhorsky et al., 2014; Zhang and Chow, 2013). For a tweet x with features t_x, the $p(c|x)$ function returns the cumulative probability for a grid cell c according to the probability distributions of features calculated in the training stage. Finally, $argmax_{c \in C} p(c|x)$ is selected as the estimated grid cell and its centroid is assigned as the estimated coordinates for tweet x.

$$p(c|x) = \sum_{t \in t_x} w_t \times p_t(c) \tag{7}$$

In this equation, w_t represents the weight that we assign for feature t. One option in the selection of w_t's is to use uniform weighting (e.g., w_t=1 for every feature in the feature space). On the other hand, because of differences in the geographical characteristics of features, higher prediction accuracy can be achieved if weights are determined for each feature separately. Applying an optimization algorithm such as Expectation Maximization using a validation dataset could be an alternative for their tuning (Bakerman et al., 2018; Lichman and Smyth, 2014; Zhang and Chow, 2013). However, in our case where we have thousands of distinct features and thus thousands of weights to tune, this approach would not be practical and would require a considerable number of tweets for validation. The method that we propose in this work for the selection of weights is to use IGR values that we have already calculated for training. Since location indicative features are expected to have higher IGR, their effect on the combined results would be directly proportional to these values. In our evaluations, we demonstrate the improvement obtained by using IGR-based weighting over the uniform weights.

5. Evaluation and Discussion

In this section, we present the evaluation results of our method applied to tweet sets from three major cities in the world. First we describe our evaluation methodology, including the description of our tweet datasets and baseline prediction methods. Then, we examine the results of our method in comparison to the baselines, present our experiments with alternative settings, and discuss our findings.

5.1. Evaluation methodology

We evaluate our method on three datasets that are composed of geotagged tweets from London, Paris and Berlin, collected for two months between October and December in 2015 using the Twitter Streaming API. These three regions and distributions of collected tweets are shown in Fig. 2. We model each region as a 100×100 grid, where a cell covers an area of approximately 0.5km^2. We did not apply any restriction

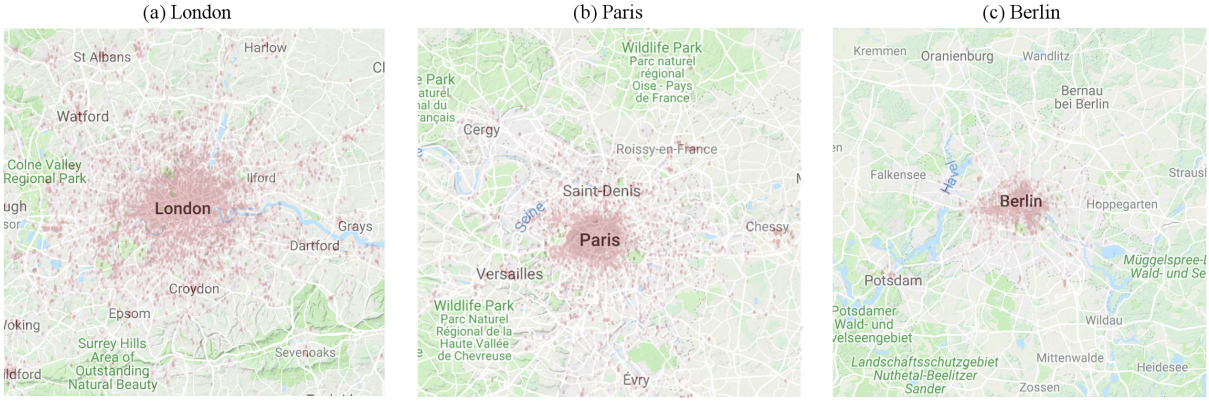

Figure 2: Geographical distribution of tweets from (a) London, (b) Paris, and (c) Berlin.

on the language of tweets, and collected all geotagged tweets with a latitude and longitude provided by the stream.

Following the common practices in the literature for data cleaning and spam removal, we first excluded exact duplicate tweets and Foursquare check-ins from our datasets (Dredze et al., 2016; Eisenstein et al., 2010; Han et al., 2014). Spam removal mainly aims to filter out tweets from possible spammers, promoters, marketers and other automated-script-style Twitter accounts. Such tweets have almost the same text and they are usually posted from the same places (e.g., periodically posted weather forecasts, job announcements or advertisements) (Ozdikis et al., 2018b). We observe that if there are many tweets posted regularly from the same account (e.g., more than two tweets per day), they are likely to be auto-generated tweets from such Twitter accounts. Therefore, similar to the spam removal procedures in previous studies, we also excluded tweets from users with more than 1000 friends or followers or who posted more than two tweets per day (Cheng et al., 2013; Eisenstein et al., 2010; Lee et al., 2010; Ozdikis et al., 2018a,b). The data cleaning process yields more than 300K tweets posted by around 75K different users in our London dataset.

Tweet texts are tokenized using the Twokenize[3] library, and tokens that appear in less than five tweets, hyperlinks, and single characters are excluded in training to reduce data sparsity. In our experiments, we randomly selected 95% of tweets in each dataset for training, and used the remaining 5% for test. The number of tweets, distinct users and distinct terms in training tweets after data cleaning are presented in Fig. 2.

5.1.1. Evaluation metrics

The performance of location prediction is measured using the following three metrics from the literature (Han et al., 2014; Paule et al., 2018b; Zheng et al., 2018):

Median Error Distance (*MED*): This is the median of distances between the predicted locations and the actual locations for test tweets. The predicted location in terms of latitude-longitude for a test tweet is the centroid of the estimated grid cell, while its actual location is the coordinates associated with that tweet as received from Twitter. Since this is a distance-based metric that measures the error, lower median error distance indicates higher effectiveness.

Accuracy (*Acc*): This is a token-based metric where predicted locations and expected locations are compared in terms grid cells. Accuracy is calculated as the proportion of test tweets for which the actual grid cell is correctly predicted. Therefore, higher ratios of accuracy are more desirable.

[3]https://github.com/brendano/ark-tweet-nlp/

Accuracy at n kilometers (*Acc@n*): *Acc* metric described above expects a prediction to be exactly the same grid cell as the actual grid cell for a tweet. However, predictions can still be acceptable if they are in the spatial proximity of the actual locations. Therefore, in order to capture close-enough predictions, *Acc@n* is defined as the proportion of tweets in the test set for which the estimated location is at most n kilometers away from the actual location of a tweet.

5.1.2. Evaluated methods

We compare our proposed prediction method with state-of-the-art techniques that are widely applied in the literature, which include Naive Bayes classifiers, similarity measure using Kullback-Leibler divergence, and neural network models. We also implemented several enhancements that have previously been employed to improve these methods. The baselines are selected among the content-based methods in the literature that were shown to yield high accuracies. Moreover, we also considered their applicability for multilingual tweets without any language-dependent setting and their practicality to be executed for large number of classes in our fine-granular grid setting. Prediction methods that we implemented in our evaluation are as follows:

Class Prior (CP): This method basically returns the grid cell with maximum number of tweets in the training set without performing any analysis of tweet text. This is used to show that assigning all test tweets simply to the most populous grid cell does not yield useful results.

Naive Bayes classifier (NB): Multinomial Naive Bayes classifier is a simple and scalable generative model that is widely applied for location estimation. It incorporates class priors in prediction and it is reported to perform well even on scarce training data (Han et al., 2014; Hulden et al., 2015). Therefore, we implemented a Naive Bayes classifier with additive smoothing and trained it on the training tweets for each dataset. We calculated posterior probabilities for test tweets using their distinct terms, which performed slightly better than keeping repeating terms in texts. Probability of c for a given test tweet x is computed using Eq. 8. In this equation, $|X_c|$ represents the number of training tweets posted from c, $f_{t,c}$ is the frequency of a term t in grid cell c, and α has the value $1/|C|$ in additive smoothing. The predicted grid cell is then selected as the one that maximizes this posterior probability.

$$p(c|x) = \frac{|X_c|}{|X|} \prod_{t \in t_x} \frac{f_{t,c} + \alpha}{\sum_{t_i \in T} f_{t_i,c} + \alpha|T|} \tag{8}$$

NB$_{IGR}$: The basic setting of Naive Bayes classifier in our evaluation uses all terms in training tweets. Previous studies showed that selecting the location indicative terms in tweets and using only these selected terms in classification can yield higher accuracies (Han et al., 2014; Van Laere et al., 2014). Among a wide range of feature selection techniques, information gain ratio was shown to outperform others in the location prediction task. Therefore, we extended NB by selecting the top-n terms with highest information gain ratio and used these features in the training of Naive Bayes classifier. Since different datasets can achieve their highest accuracy using different n values, we applied 10-fold cross validation on each training set to determine their optimum top-n ratios. This prediction method is denoted by NB$_{IGR}$ with a specific value of $n \leq 1$. For example, NB$_{IGR}$ with n=0.3 refers to NB where top 30% of highest IGR unigrams are used in training. The tuned n values for each dataset will be given in the footnotes of Table 1.

NB$_{IGR+SCoP}$: As a further extension to NB$_{IGR}$, this method extends the feature space with the bigrams selected by SCoP, as described in (Ozdikis et al., 2018b). As we demonstrate in Section 5.2, this enhancement yields accuracy improvements for every dataset in our experiments, which is consistent with our previous findings. We present a detailed evaluation of SCoP later in Section 5.4.

Kullback-Leibler divergence (KL): Kullback-Leibler divergence is used to find the grid cell whose term distribution matches the distribution of terms in a test tweet (Melo and Martins, 2017; Roller et al., 2012; Wing and Baldridge, 2014). By applying Eq. 9 following the description in (Hulden et al., 2015), we calculate kl divergence of every grid cell $c \in C$ for a given tweet x and select the grid cell with minimum divergence

as the location for that tweet. In this equation, $f_{t,x}$ and $f_{t,c}$ represent the frequency of a term t in tweet x and in grid cell c, respectively. The smoothing constant α=1/$|C|$.

$$kl(c|x) = \sum_{t \in t_x} \frac{f_{t,x}}{|t_x|} \log \left(\frac{\frac{f_{t,x}}{|t_x|} \times \sum_{t_i \in T} f_{t_i,c} + \alpha|T|}{f_{t,c} + \alpha} \right) \tag{9}$$

KL$_{IGR}$: Similar to the enhancement that we applied for NB using feature selection, in this method we use only the terms with highest information gain ratio in the calculation of KL-divergence. In order to determine their optimum top-n ratios, we applied 10-fold cross validation on each training set.

Neural Networks (NN): Miura et al. (2016) proposed a neural networks model for content-based tweet localization using the FastText[4] library. FastText is stated to have close performance to deep complex models while being much faster and scalable, making it suitable for our large-scale prediction task that estimates among thousands of possible locations using large tweet sets (Joulin et al., 2017). The solution in (Miura et al., 2016) is also reported to achieve the best median distance error at Twitter Geolocation Prediction Shared Task in W-NUT'16 (Han et al., 2016). Therefore, we include it in our evaluated methods.

Following the descriptions in (Miura et al., 2016), we first applied a pretraining of word embeddings using the skipgram algorithm. Then, using the pretrained word vectors, we trained the supervised classifier in FastText library. These two steps required a selection of values for the following parameters in FastText: learning rate (*lr*), size of the context window (*ws*), number of negatives samples (*neg*), number of epoch (*epoch*), maximum length of word ngram (*wordngrams*), and size of word vectors (*dim*). We initially used the values provided by Miura et al. (2016) for these parameters. However, since every dataset can have different properties, we performed further tuning by a 10-fold cross validation in order to optimize the settings and get higher accuracies with FastText. We present the tuned values for each dataset in the footnotes of Table 1.

LocKDE: This is our locality-adapted KDE prediction method that uses only the unigrams in tweets (i.e., without using bigrams selected by SCoP). It uses our locality-adapted bandwidth in the computation of kernel densities (given in Eq. 5) and IGR-based weighting in the calculation of combined probabilities (given in Eq. 7). In our experiments, for a test tweet that does not include any term in the training set, we selected the grid cell with the highest class prior. We remind that our algorithm did not require any parameter tuning and thus, we applied the same settings in the training of all three datasets.

LocKDE$_{SCoP}$: This represents our proposed solution[5] that extends the unigram feature space with bigrams selected by SCoP and applies locality-adapted KDE on the extended feature space. In our experiments, we used δ=0.5km, which is approximately the size of a grid cell, and m=500 for the number of Monte Carlo simulations in Algorithm 1. We discuss the performance of our method under different settings in the following sections.

5.2. Evaluation results

The evaluation results of LocKDE$_{SCoP}$ in comparison to the baselines for each of the three datasets are given in Table 1. The table shows the performance of each method in terms of median error distance, exact grid cell accuracy, and accuracies with a tolerance of 0.5km, 1.0km and 5.0km. The best results for each of these evaluation metrics are marked in bold.

These results show that the most accurate estimations in terms of median error distance are obtained by our LocKDE$_{SCoP}$ method in the rightmost column. For London tweets, which is the largest tweet set in our experiments, we can make estimations that are remarkably close to the actual tweet locations, yielding a median error distance of 0.693km. The second lowest median error distance for London dataset is achieved by NN as 0.887km. NN appears to be a strong alternative among the baselines in most cases. It even yields slightly better results than LocKDE$_{SCoP}$ in terms of the accuracy of exact grid cell (Acc). However, as we increase the error tolerance, we observe that predictions of LocKDE$_{SCoP}$ are in fact not very distant from

[4]https://fasttext.cc/

[5]The source code is available at: https://github.com/oozdikis/tweet-localization-LocKDE

Table 1: Comparison of LocKDE$_{SCoP}$ and the baselines on three datasets.

Dataset	Evaluation Metric	Evaluated Methods								
		CP	NB	NB$_{IGR}$	KL	KL$_{IGR}$	NB$_{IGR+SCoP}$	NN	LocKDE	LocKDE$_{SCoP}$
London[a] *306K tweets*	MED	4.048	2.054	1.155	2.823	1.556	0.912	0.887	0.957	**0.693**
	Acc	0.071	0.353	0.386	0.336	0.319	0.414	**0.424**	0.346	0.422
	Acc@0.5	0.084	0.371	0.409	0.353	0.342	0.436	0.446	0.403	**0.467**
	Acc@1.0	0.179	0.438	0.485	0.412	0.446	0.507	0.512	0.505	**0.555**
	Acc@5.0	0.548	0.630	0.670	0.585	0.659	0.678	0.692	0.718	**0.739**
Paris[b] *153K tweets*	MED	3.576	1.475	0.768	2.444	1.549	0.597	0.588	0.742	**0.528**
	Acc	0.169	0.420	0.466	0.371	0.424	0.478	**0.483**	0.420	0.455
	Acc@0.5	0.175	0.435	0.481	0.384	0.438	0.492	0.494	0.465	**0.496**
	Acc@1.0	0.205	0.467	0.512	0.417	0.466	0.522	0.525	0.536	**0.566**
	Acc@5.0	0.668	0.694	0.735	0.630	0.657	0.740	0.739	0.786	**0.796**
Berlin[c] *38K tweets*	MED	2.811	1.954	1.595	2.581	1.966	1.415	1.067	0.975	**0.766**
	Acc	0.133	0.372	0.416	0.328	0.359	0.433	**0.456**	0.372	0.418
	Acc@0.5	0.135	0.388	0.428	0.344	0.373	0.445	**0.470**	0.447	0.466
	Acc@1.0	0.163	0.418	0.462	0.374	0.404	0.475	0.496	0.503	**0.526**
	Acc@5.0	0.761	0.734	0.783	0.654	0.728	0.788	0.777	**0.836**	0.835

[a] NB$_{IGR}$: n=50%, KL$_{IGR}$: n=30%, FastText: (lr=0.15, ws=5, neg=5, epoch=5, wordngrams=1, dim=300)
[b] NB$_{IGR}$: n=40%, KL$_{IGR}$: n=40%, FastText: (lr=0.2, ws=5, neg=5, epoch=5, wordngrams=2, dim=1100)
[c] NB$_{IGR}$: n=40%, KL$_{IGR}$: n=30%, FastText: (lr=0.275, ws=8, neg=5, epoch=5, wordngrams=2, dim=700)

the true tweet locations. Starting from 0.5-1km, we obtain the highest accuracy values in terms of Acc@n using LocKDE$_{SCoP}$. A similar pattern is also observed in LocKDE, i.e., it makes close predictions to the actual tweet locations even if they are not exactly the expected grid cells. This is probably achieved due to the spatial smoothing of probability distributions provided by KDE. In other words, since occurrence of a term in a grid cell also influences its probability distribution in the neighboring cells, an incorrect estimation made by our locality-adapted KDE method can still be in the proximity of the actual location of a tweet. On the other hand, NN calculates probabilities for grid cells independent from each other, without taking their spatial proximity into account. Therefore, an incorrect estimation made by NN has higher likelihood to be in a distant place than the actual location of a tweet.

Among other baselines, the results of CP show that despite high concentration of tweets at city centers (see Fig. 2), assigning the most populous grid cell to tweets without making any text analysis is not a viable alternative. Our second baseline, NB, yields noticeably better results than KL for every dataset. Moreover, selection of terms according on their IGR improves the accuracies for both of these methods. Since NB$_{IGR}$ performed better than KL$_{IGR}$, we experimented our SCoP extension on it and obtained even more accurate predictions in NB$_{IGR+SCoP}$. A similar improvement is also observed when we extended LocKDE with our SCoP analysis to obtain LocKDE$_{SCoP}$. These results suggest that an effective analysis of bigrams, as we proposed in this paper, is beneficial to make better estimations.

As a result of these experiments, we observe that our locality-adapted KDE method along with our analysis of spatial co-occurrence patterns performed better than the baselines and produced the lowest median error distances. Moreover, the accuracy values in Table 1 show that our method has higher accuracy for every dataset in comparison to NN if the accuracy is measured with an error tolerance of 1km or more. In order to evaluate the significance of improvement, we analyzed the difference in error rates between LocKDE$_{SCoP}$ and NN for Acc@1.0km and Acc@5.0km by employing McNemar's[6] test. The results indicated statistically significant improvement for every dataset in our experiments ($p<0.001$).

5.3. Locality-adapted bandwidths and weights

In order to explore the improvement that is achieved by our locality-adapted kernel bandwidths and probability weights, we examined the results of LocKDE and LocKDE$_{SCoP}$ under four different alternative settings. The results of our experiments are presented in Table 2.

In the simplest setting, denoted as (h=*Scott*, w=1.0) in Table 2, we do not apply any locality adaptation to kernel bandwidths and use the default rule of thumb, which is Scott's rule. This first setting also assigns

[6] https://www.statsmodels.org/dev/generated/statsmodels.sandbox.stats.runs.mcnemar.html

Table 2: Results of LocKDE and LocKDE$_{SCoP}$ with and without the proposed locality-adapted settings.

Dataset	Evaluation Metric	LocKDE under different settings				LocKDE$_{SCoP}$ under different settings			
		h=*Scott* w=1.0	h=*Scott* w=*IGR*	h=*IGR* w=1.0	h=*IGR* w=*IGR*	h=*Scott* w=1.0	h=*Scott* w=*IGR*	h=*IGR* w=1.0	h=*IGR* w=*IGR*
London	MED	1.356	1.090	1.068	0.957	0.869	0.757	0.716	**0.693**
	Acc	0.300	0.312	0.339	0.346	0.398	0.403	0.420	**0.422**
	Acc@0.5	0.356	0.372	0.394	0.403	0.445	0.452	0.465	**0.467**
	Acc@1.0	0.467	0.490	0.493	0.505	0.541	0.549	0.553	**0.555**
	Acc@5.0	0.691	0.713	0.709	0.718	0.724	0.737	0.735	**0.739**
Paris	MED	1.075	0.818	0.818	0.742	0.737	0.670	0.578	**0.528**
	Acc	0.360	0.378	0.410	0.420	0.411	0.421	0.448	**0.455**
	Acc@0.5	0.407	0.428	0.455	0.465	0.455	0.465	0.490	**0.496**
	Acc@1.0	0.494	0.525	0.522	0.536	0.544	0.557	0.558	**0.566**
	Acc@5.0	0.764	0.783	0.779	0.786	0.787	0.794	0.792	**0.796**
Berlin	MED	1.282	1.101	1.141	0.975	0.972	0.966	0.894	**0.766**
	Acc	0.357	0.364	0.377	0.372	0.398	0.399	0.414	**0.418**
	Acc@0.5	0.406	0.419	0.431	0.447	0.444	0.447	0.462	**0.466**
	Acc@1.0	0.469	0.485	0.484	0.503	0.507	0.512	0.517	**0.526**
	Acc@5.0	0.820	0.834	0.828	**0.836**	0.831	**0.836**	0.834	0.835

uniform weights to term probabilities (see Eq. 7), rather than using term-specific weights according to their locality. In the second and third settings, we replace h=*Scott* and w=1.0 with our locality-adapted enhancements for bandwidth and weighting, namely h=*IGR* and w=*IGR*, respectively. We observe that each of these enhancements improves the prediction accuracies for both LocKDE and LocKDE$_{SCoP}$, even when they are applied separately. The final setting uses locality-adapted kernel bandwidths and term-specific weights, i.e., (h=*IGR*, w=*IGR*). The rightmost columns for LocKDE and LocKDE$_{SCoP}$ in Table 2 reveal that applying locality-adapted bandwidths and probability weights together in this setting yields the lowest error distances for all datasets.

We exemplify the effect of applying our locality-based enhancement on the selection of kernel bandwidths in Fig. 3 and Fig. 4. Fig. 3(a) and 3(b) are generated for the unigram *olympiastadion* using training tweets in our Berlin dataset. Colored circles on the upper maps indicate the actual locations of tweets that include *olympiastadion*, and red shadings on the lower maps correspond to the calculated probability distributions using different settings in KDE. The distribution in (a) is obtained by using the default bandwidth (h_{Scott}) in KDE, while the distribution in (b) is generated by our locality-adapted kernel bandwidth (h_{IGR}). Since the term *olympiastadion* is a strong location indicative term for Berlin, it has a high information gain ratio, which results in a lower bandwidth value. As a result of lower kernel bandwidth, we observe a more concentrated probability distribution in Fig. 3(b) compared to (a), as expected. On the other hand, Fig. 3(c) depicts the distribution for *the*, a very common stop word in tweets. In fact, it is found as the least local term with lowest information gain ratio among the unigrams in our Berlin dataset. The term occurs in numerous tweets from different parts of the city. Accordingly, its probability distribution exhibits a more dispersed pattern compared to *olympiastadion*. Moreover, since it is the least local term, its locality-adapted bandwidth is not different from the default setting and it results in a dispersed probability distribution over the entire region.

Fig. 4 presents a similar example using bigrams in our Paris dataset. Fig. 4(a) and 4(b) show the distribution of tweets and their probabilities for the bigram *parc astérix*, which concentrates around a touristic park named Asterix in the north-eastern part of the city. The bigram is among the highly local features and therefore its locality-adapted kernel bandwidth results in a strong concentration around the park, as shown in Fig. 4(b). In another example, we present the distribution of tweets mentioning the bigram *#paris #france* in Fig. 4(c). Our analysis on bigrams assigned it a low information gain ratio, as these two terms together do not provide much geographical evidence about any specific place inside the city. Therefore, its locality-adapted bandwidth becomes almost equal to its default bandwidth ($h_{IGR} \approx h_{Scott}$) and it persists a dispersed probability distribution. We would like to remark that our statistical approach can capture relationships between different types of textual features, including words with language-specific characters, hashtags and even emojis in tweets.

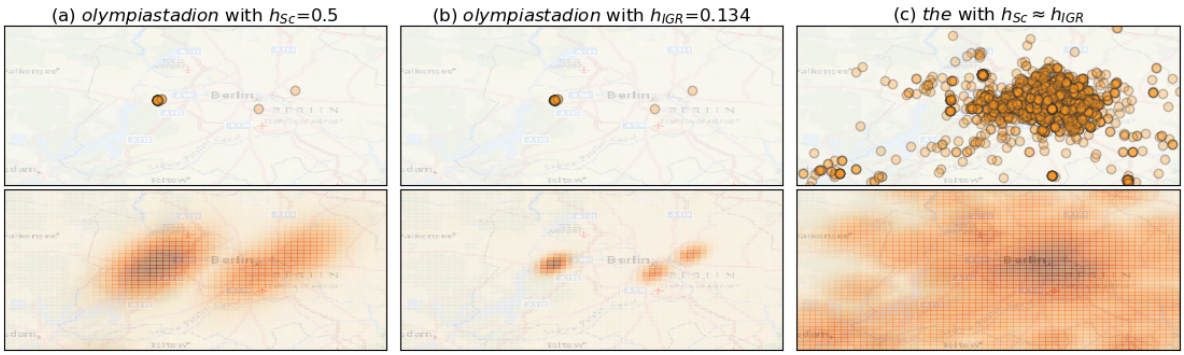

Figure 3: Probability distribution of example unigrams in Berlin area using default bandwidth and locality-adapted bandwidth.

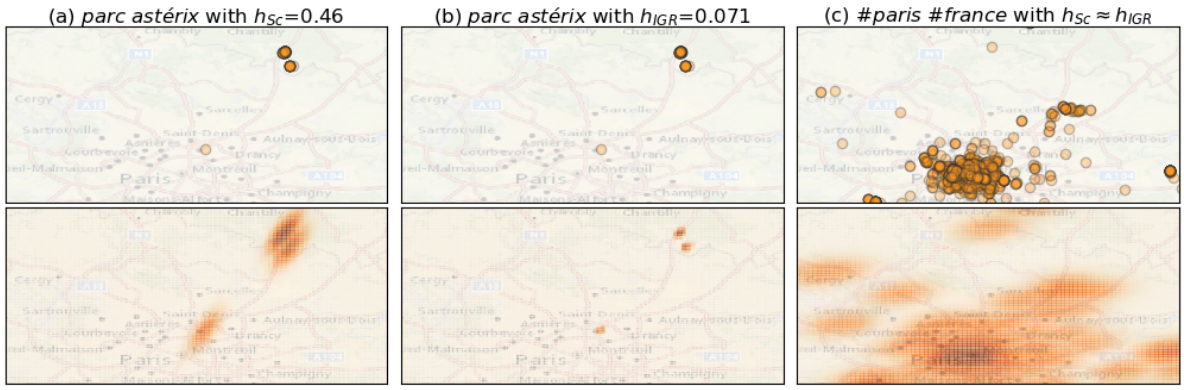

Figure 4: Probability distribution of example bigrams in Paris area using default bandwidth and locality-adapted bandwidth.

5.4. Bigram selection using SCoP

The results in Table 1 showed that extending the feature space using bigrams by analyzing their spatial co-occurrence patterns improves the accuracies for both NB_{IGR} and LocKDE predictors. In this section, we investigate alternatives for bigram selection and compare the results with our proposed SCoP method.

Effective analysis of bigrams is important not only to achieve higher accuracies but also to prevent any redundant increase in the size of feature space. In our London dataset, there are more than 64K bigrams in total that appear in at least five tweets. We performed our experiments with NB_{IGR} and LocKDE using all of these bigrams without making any specific selection. We note that using all bigrams in addition to the unigrams in the training set resulted in an increase of approximately 150% in the size of feature space. Moreover, as an alternative to SCoP, we also implemented a bigram selection method that selects top-n of the bigrams according to their information gain ratio. The evaluation results for the bigram-extended settings of NB_{IGR} and LocKDE are shown in Fig. 5(a) and 5(b), respectively. In Fig. 5(a), $\text{NB}_{IGR+IGR}$ denotes the Naive Bayes classifier that uses unigrams and bigrams that are selected by IGR. Similarly, LocKDE_{IGR} in (b) is used as an abbreviation for our LocKDE method that is extended by top-n bigrams with the highest IGR values. We compare these methods with their SCoP counterparts, namely $\text{NB}_{IGR+SCoP}$ and LocKDE_{SCoP}.

In Fig. 5(a), both NB settings use the same set of unigrams (i.e., 50% of unigrams with the highest IGR, as we had found earlier for London dataset). Our SCoP analysis revealed that around 6K bigrams had significantly different spatial patterns than these unigrams, which led to the extension of feature space with these bigrams in $\text{NB}_{IGR+SCoP}$. Since SCoP does not require any additional parameter, error distance of $\text{NB}_{IGR+SCoP}$ is shown as a single horizontal line which corresponds to 0.912km. Accuracy of $\text{NB}_{IGR+IGR}$, on the other hand, changes depending on the ratio of bigrams included in training. It achieved its lowest

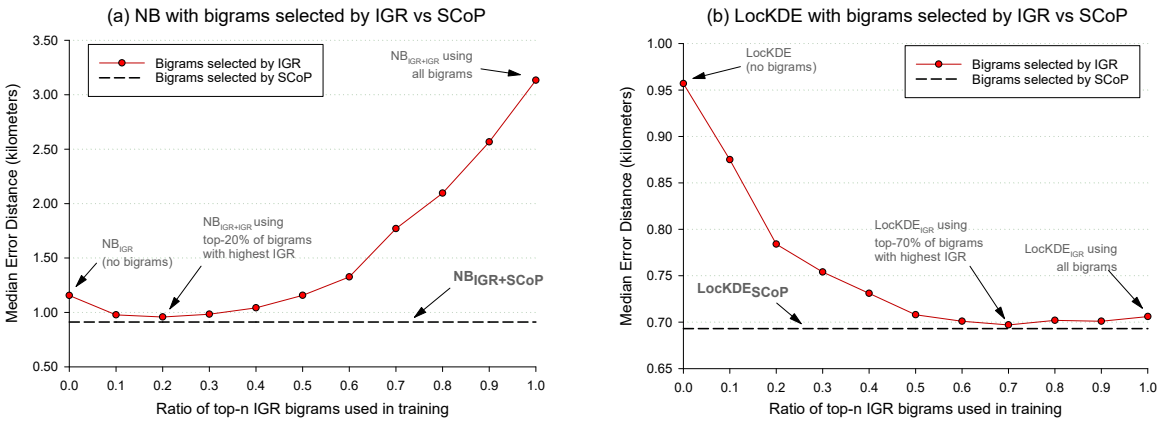

Figure 5: Evaluation of (a) NB and (b) LocKDE on London dataset using different methods for bigram selection.

median error distance as 0.958km when top-20% of bigrams (more than 12K) are used, which is twice the amount that is selected by our SCoP analysis. As we increase n and include more bigrams in our feature space, the results begin to deteriorate. That means, having more features does not necessarily improve the performance of Naive Bayes predictor. These experiments indicate that $\text{NB}_{IGR+SCoP}$ has higher accuracy than the best result that can be obtained by $\text{NB}_{IGR+IGR}$. We observed similar results with our Paris and Berlin datasets, and thus, we conclude that SCoP achieved higher accuracy with fewer bigrams and without any parameter tuning.

Fig. 5(b) presents a similar comparison between LocKDE_{SCoP} and LocKDE_{IGR}. The median error distance of predictions made by LocKDE_{SCoP} is 0.693km, whereas the setting with the highest accuracy that can be obtained by LocKDE_{IGR} yields an error distance of 0.697km using 70% of all bigrams. Although the accuracies are very close, the predictions made by LocKDE_{SCoP} uses only 21K bigrams, which is approximately half the number of bigrams used in LocKDE_{IGR}. It is noteworthy that unlike the pattern in Fig. 5(a), the accuracy of LocKDE does not necessarily deteriorate as we use more bigrams. This is probably the result of our locality-adapted probability assignment in LocKDE. Since we handle each feature differently according to their locality strength, weakly local features have weaker influence on combined probabilities, and thus they do not cause much distortion in estimations. We performed these analyses on our smaller datasets from Paris and Berlin as well, and noticed that we can obtain slightly better accuracy using IGR. However, these accuracies are also achieved only with a remarkable increase in feature space. As a result, we conclude that extending the feature space with bigrams improves the accuracies both for NB and LocKDE predictors. Making a selection of significant bigrams using our SCoP analysis is advantageous since it does not require any data-specific parameter tuning and yields high accuracies without incurring remarkable increase in the size of feature space.

5.5. Example bigrams detected by SCoP

In this section, we present two illustrative examples in Fig. 6 and Fig. 7 regarding the attraction and repulsion patterns detected by our SCoP analysis. In these figures, the colored circles represent the locations of tweets mentioning a specific unigram/bigram, and shadings in red are generated by KDE with its default setting in order to visualize tweet densities. Fig. 6(a) and 6(b) show the locations of tweets mentioning the unigrams *british* and *museum* in London area, respectively. Fig. 6(a) indicates a relatively scattered distribution over the city, mostly focusing on highly populated areas at the city center. This can be expected since *british* is a relatively common word around the London area. The term *museum* is also mentioned in different parts of the city, although it exhibits higher concentration in several specific places. If these two terms are considered together as a bigram, we observe a significant clustering tendency as shown in Fig. 6(c). In other words, the density distribution of the tweets mentioning the bigram *british museum*

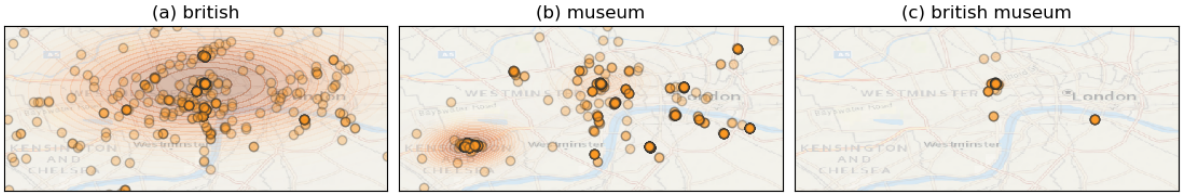

Figure 6: Distribution of tweets mentioning (a) *british*, (b) *museum*, (c) *british museum* in London area.

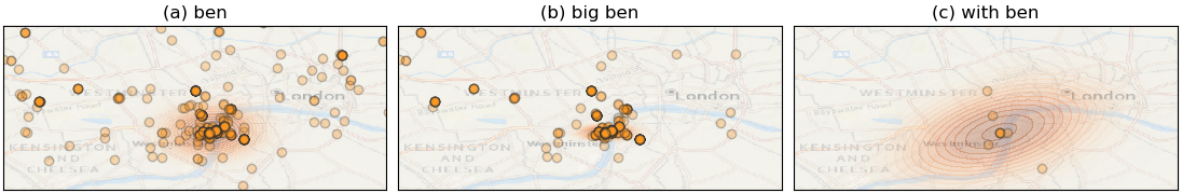

Figure 7: Distribution of tweets mentioning (a) *ben*, (b) *big ben*, (c) *with ben* in London area.

noticeably focuses on a specific place when compared to the distributions of each of these unigrams. This pattern has been successfully detected by our SCoP analysis, and the bigram is included in the feature space.

Fig. 7 presents another example for the results of our SCoP analysis, which displays an attraction pattern in (b) and a repulsion pattern in (c) with respect to the distribution of unigram *ben* in (a). The distribution of bigram *big ben* in Fig. 7(b) exhibits a stronger concentration around a specific place compared to the distribution of tweets mentioning *ben* alone. On the other hand, when *ben* is preceded with the term *with*, it has a sparser distribution and has a more dispersed shading, as shown in Fig. 7(c). Therefore, these two bigrams are included in the feature space since they have significantly different spatial patterns with respect to the unigram *ben*. We would like to point that, although the relationships in some bigrams are not directly obvious and they may not even exist in dictionaries, our SCoP analysis can identify those spatial patterns as well since it uses purely statistical methods.

5.6. Selection of term pairs: bigrams vs co-occurrences

In this section, we analyze the relationship between term pairs that do not necessarily appear adjacently as in the form of a bigram. In other words, we perform our SCoP analysis on term pairs that co-occur in a tweet, irrespective of their order and other terms that may appear between them. We find those co-occurring term pairs with significant deviation from the single-term patterns, and use them in the extension of feature space for the training of our LocKDE predictor. We present the results of using such co-occurrences in comparison to 1) the unigram model and 2) our proposed method that uses bigrams.

Median error distances that we obtained by LocKDE and its extensions using two different selection methods for term pairs (i.e., bigrams vs. co-occurrences) is depicted in Fig. 8. The bars in the figure indicate the sizes of their feature space, with red bars showing the amount of increase when we use term pairs in the form of bigrams or co-occurrences. Among these three settings, LocKDE has the smallest feature space since it uses the unigrams only. When compared to LocKDE, we observe remarkable improvement in accuracy using bigrams at the cost of a slight increase in the number of features. On the other hand, the rightmost bars in the figures show that the analysis of co-occurrences results in a significant increase in feature space. This can be expected since a tweet with n terms can have n^2 co-occurring term pairs, while it can have at most $n-1$ bigrams. Despite this increase in the size of feature space, using co-occurrences instead of bigrams does not necessarily improve the accuracy of predictions. Therefore, we consider bigram analysis using SCoP a more preferable option since it yields high accuracies with limited increase in the number of features used for training.

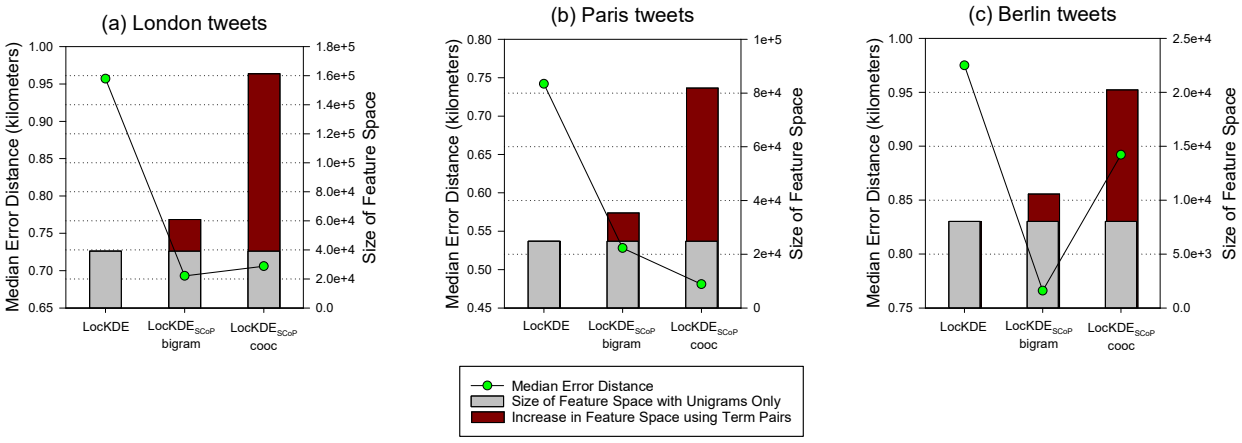

Figure 8: Error distances and sizes of feature space using different methods for the selection of term pairs in LocKDE.

5.7. Performance evaluation

In addition to the improvement that we obtain in the accuracy of predictions using LocKDE$_{SCoP}$, it is also worth discussing the computational efficiency of SCoP and LocKDE. In this section, we elaborate on the execution time of our algorithm and its utility for online predictions. We analyze the time and space complexity of training and prediction stages of our method, and compare it with the performance of NN, i.e., the baseline that uses FastText.

5.7.1. Complexity analysis of SCoP

Our bigram selection using SCoP given in Algorithm 1 has quadratic complexity with respect to the number of distinct unigrams in the training set. Considering the sampling process in Monte Carlo simulation and the calculation of pairwise distances between tweets, its execution time also depends on the number of iterations m in sampling and the frequencies of bigrams n. As a result, the time complexity of Algorithm 1 is $\mathcal{O}(r^2mn^2)$, where $r{=}|T|$ represents the number of distinct unigram features. In our experiments, the complete SCoP analysis for our London dataset took nearly 8 hours on a server with 2.6GHz 16-core (32 hyperthreads) CPU.

The amount of space needed for our SCoP analysis depends on the largest tweet set that includes the same bigram. In the unlikely case of having a bigram t_it_j in every tweet in the training set, all tweet coordinates would be needed for the calculation of K_δ at line 8. Therefore, Algorithm 1 has linear worst case space complexity with respect to the number of tweets in the training set. However, the number of tweets with a bigram is usually much lower than the number of all tweets in the training set (e.g., even the most frequent bigram is mentioned in only 3% of tweets in our London dataset). Therefore, in practice, the execution of SCoP does not require all tweet coordinates to be available in memory.

5.7.2. Complexity analysis of LocKDE

As noted in Section 4.1, we use the SciPy library to calculate the probability distributions of features over grid cells. Although the execution times may change depending on the implementation, theoretically KDE is reported to have quadratic time complexity with respect to the sample size $|X_t|$ (Qahtan et al., 2017). Since we perform the density estimation for every feature in the feature space, calculation of probability distributions has computational complexity $\mathcal{O}(fg^2)$, where f is the number of features and $g{=}|X_t|$ represents the number of tweets that include a feature t. As a result, having the same feature in many tweets has a quadratic effect on the execution time of LocKDE.

The result of training is the probability distributions of features on grid cells. Therefore, the number of features in the training set has a linear effect on the amount of space needed for training. In a $100{\times}100$ grid setting, maintaining a small feature space is particularly important since we would store $f{\times}10^4$ values.

In our experiments, the calculation of probability distributions in the training of LocKDE (using only the unigrams without applying our SCoP extension) took around 48 minutes for 39K unigrams. Our experiments with LocKDE$_{SCoP}$ revealed that how we utilize the detected bigrams in the extension of feature space plays an important role in the training time of LocKDE$_{SCoP}$. In other words, the training time can change depending on how the bigrams are used to modify the text features in tweets (i.e., replacing unigrams with bigrams or inserting bigrams as additional features to tweets). We explained in Section 4.2.2 that we adopted a replacement strategy, where a unigram t_x^i in a tweet x is replaced with the bigram $t_x^i t_x^{i+1}$ if that bigram has an attraction or repulsion pattern with respect to t_x^i. This replacement strategy has two types of influence on the number of KDE computations. Firstly, the average number of tweets that include a feature decreases since the frequency of a bigram can not be higher than the frequency of its unigrams, i.e., $|X_{t_i t_j}| \leq \max(|X_{t_i}|, |X_{t_j}|)$. Considering the complexity $\mathcal{O}(fg^2)$ given above, this decrease in g makes a quadratic effect on the execution time. The second influence of our replacement strategy is on the size of tweet features. If the SCoP analysis detects that the bigram $t_x^i t_x^{i+1}$ has an attraction or repulsion pattern with respect to both t_x^i and t_x^{i+1}, then both of these unigrams are replaced with the bigram, which makes a shrinking effect on the dimension of tweet features. That results in fewer data samples to be used in the calculating of KDE.

Consequently, in consistence with the complexity analysis given above, calculation of probabilities for all features (unigrams and bigrams detected by SCoP) in our London dataset using LocKDE$_{SCoP}$ took around 35 minutes on our server. We note that this improvement is achieved by maintaining a small feature space, i.e., by storing the probability distributions of 21K bigrams in addition to the unigrams. Our further investigation revealed that if we had adopted an insertion strategy, the training process would have taken 53 minutes along with a slight decrease in prediction accuracy.

5.7.3. Performance of prediction

After the training of our prediction model, location prediction for a test tweet took much shorter time that would not hinder online processing. Average prediction time for a tweet using LocKDE (without using bigrams) was 65ms. Because of the feature replacement strategy explained above, prediction time of LocKDE$_{SCoP}$ was remarkably lower (57ms). Moreover, after minor enhancements in our code using more efficient data structures, we managed to obtain an average prediction time of 28ms using LocKDE$_{SCoP}$.

We also compare the execution time performance of LocKDE$_{SCoP}$ and NN, namely our baseline that uses FastText library. After the tuning of six parameters for FastText, which were listed in Section 5.1.2, training of FastText on our London dataset took approximately 5 minutes using its multithreading feature on our server with 32 hyperthreads. Once the model is trained, we measured its prediction time as 2.5ms per tweet. That means, training and test times of FastText were remarkably lower compared to our method. However, we consider three important factors that has to be taken into consideration while making this comparison. Firstly, our reported training time for FastText does not include the time spent for its parameter tuning. A tuning stage for six parameters using a 10-fold cross validation can take several hours depending on the number of combinations for parameter values. An advantage of our method is that it does not require any parameter tuning. Secondly, we observed that the execution time of FastText depends on the values of its parameters. For example, doubling the size of word vectors in FastText (i.e., changing *dim* from 300 to 600) resulted in a two-fold increase in its training and prediction times. Finally, and probably more importantly, FastText is a precompiled C++ program, whereas LocKDE$_{SCoP}$ is implemented in Python. Therefore, we believe that the actual execution time of our solution can further be improved by using a compiled programming language. In the next section, we also discuss various alternatives to improve the performance and prediction accuracy of our methods.

6. Discussion and Future Work

As we demonstrated in previous sections, our prior analysis of spatial co-occurrence patterns in bigrams is beneficial to improve the accuracies significantly at the cost of an only 30-50% increase in the size of feature space. Maintaining a small feature space is particularly critical both for the space and time complexity of

our algorithms. As noted in Section 5.7, the time complexity of SCoP is $\mathcal{O}(r^2mn^2)$. That means, apart from the number of unigrams in the training set, there are two important factors that affect the time to analyze spatial co-occurrence patterns in SCoP, which are 1) frequencies of bigrams, and 2) the number of Monte Carlo simulations m. Referring to the sampling strategy in line 10 of Algorithm 1, high-frequency bigrams, i.e., large number of tweets that include a specific bigram, can negatively affect the execution time. For example, the bigram *United Kingdom* appears in around 10K tweets in our London dataset. Taking that many samples from corresponding unigrams and calculating their K_δ in every iteration of the Monte Carlo simulation took more than 14% of the time spent to analyze all bigrams in our training data. As an improvement to the current setting, alternative sampling strategies can be devised to reduce the simulation time for such cases. For example, sampling in line 10 of the algorithm can also be applied for unigrams. That means, fewer (but still equal) number of samples can be taken both from the bigrams and the unigrams, and statistical significance of the difference between two envelopes can be computed.

Regarding the second performance factor in SCoP, we used m=500 as the number of Monte Carlo simulations in our experiments, but we note that different values of m can be used in different studies (Ripley, 1977). In order to analyze the effect of using alternative m values, we experimented our SCoP analysis using m=200 and m=1000. The results showed that each of these experiments made a change only for 0.6% of the bigrams that were identified with m=500. Therefore, we selected m=500 since it has produced satisfactory results in reasonable time for our datasets. Considering that Monte Carlo simulations can be executed independently from each other, we also implemented our code in a way that executes the simulations on multiple cores in parallel.

The second step in our training calculated probability distributions of features using LocKDE. Challenges involved at this step were mainly due to the computation of integrated densities over the grid. Since probability calculations are independent from each other, our first solution to speed-up the training was to perform density estimations in multiple parallel processes. Moreover, as a further performance improvement, we applied a pruning of grid cells at a level higher than 100×100 granularity. In other words, we built a discretization of the grid at a resolution of 10×10, calculated probability distributions at this higher level first, and discarded those cells having near-zero probability without any drill down. This pruning strategy provided nearly 74% decrease in probability computations at 100×100 level. As a result of these enhancements, training of LocKDE$_{SCoP}$ for our London dataset took around 35 minutes on our server.

As presented in Section 3, effective analysis of tweet text to improve the accuracy of content-based predictions has been the primary motivation in numerous recent studies. Although a tweet can have several other attributes in addition to its text, experiments with these attributes indicated that tweet content is one of the most useful features for classifying tweets (Zubiaga et al., 2017). This is probably because the tweet text is actually written at the time of a post, whereas other tweet attributes are not necessarily updated every time a user posts a tweet. That means, tweet metadata can be outdated, incorrect or even incomplete. Moreover, tweet text is available in every tweet and can be collected from the Twitter Streaming API, which makes content-based methods more suitable for a real-time prediction scenario, in comparison to the methods that require additional querying for historical data. Apart from its higher accuracy and availability, another advantage of using tweet content in location prediction is generalizability across different social networking platforms (Cheng et al., 2013). In other words, prediction methods that do not depend on Twitter-specific metadata can be adapted to and applied for other platforms (e.g., Flickr, Facebook, Instagram). Because of these reasons, we adopted a content-based approach and used only the tweet text in our prediction method. We showed that even if we use only the tweet text, we can still improve the prediction accuracy by analyzing specific geographical patterns in term co-occurrences in texts.

Despite the aforementioned advantages, content-based methods can also have several limitations in the location estimation of tweets. For example, as noted in (Lee et al., 2014), some tweet texts may not include any useful hint about their originating fine-grained location (e.g., '*have a good day!*'). In other words, predictions for tweets that do not refer to a local keyword may not have a high reliability, which may result in lower recall in predictions. Lee et al. (2014) proposed a method to detect and filter out such tweets from estimation. Although that approach can improve the overall accuracy, it can exclude most of the tweets from the location prediction process. On the other hand, there are alternative solutions to improve the accuracy without excluding tweets from prediction. These solutions can utilize additional tweet attributes

and previous tweets of users, or analyze the locations of a user's friends in Twitter Bakerman et al. (2018); Dredze et al. (2016); Lichman and Smyth (2014); Zubiaga et al. (2017). However, these methods can also have various limitations and challenges to overcome. For example, querying tweet history or friend-follower network of a user may not be practical in a real-time context (Zubiaga et al., 2017). It can also be restricted by the query rate limits of Twitter. Tweet attributes that are retrieved along with the tweet text can provide strong evidence at the city or country level, whereas their usefulness for fine-grained prediction is usually very limited (Giridhar et al., 2015; Zubiaga et al., 2017). Different tweet attributes usually include different types of data. Some attributes can be strings (e.g., user name), some can be numeric (e.g., tweet time), and some can be categorical (e.g., language). Processing different data types may require different analysis techniques. Moreover, tweet features other than tweet text are not necessarily available in every tweet. Therefore, how to handle missing or outdated attributes can be another challenge. Because of such differences between the characteristics of tweet text and tweet metadata, there can be several ways to combine them in a single classification process. For example, Mahmud et al. (2014) created an ensemble of statistical and heuristic classifiers, whereas Zubiaga et al. (2017) appended multiple features into a single vector for the training of a classifier. As a result, there can be various alternatives to use additional features in Twitter, and the results may change depending on the selected approach and selected attributes. Therefore, we consider investigating the use of additional tweet features in a combined solution for fine-grained localization among the next steps of our research.

We distinguish fine-grained localization from coarse-grained localization since they essentially pose different challenges. Solutions for these two problems can vary considerably, most notably due to the differences in the number of possible locations, usefulness of tweet features, and sizes of geographical regions. For example, although user profile and tweet language may not be very useful for fine-grained analysis, they can provide strong evidence at the city or country level (Giridhar et al., 2015; Zubiaga et al., 2017). We observe a similar distinction in previous studies. The proposed solutions mostly focus on a specific granularity in order to improve the prediction accuracy (Compton et al., 2014; Han et al., 2014; Mahmud et al., 2014; Paraskevopoulos and Palpanas, 2016; Zubiaga et al., 2017). In our work, we focused on fine-grained location prediction of tweets and performed experiments with tweets posted from three different cities. Accordingly, in a practical application of location prediction, if the originating city of a tweet is not known in advance, a coarse-grained prediction may need to be applied prior to the fine-grained localization (Zubiaga et al., 2017). In the future we also plan to investigate alternative solutions for coarse-grained localization of tweets.

7. Conclusion

In this paper, we proposed LocKDE$_{SCoP}$ as a probabilistic content-based location prediction method for tweets. Our method uses kernel density estimators where kernel bandwidths and probability weights are determined according to the location indicativeness of terms. It also includes an analysis of spatial co-occurrence patterns in tweet texts in order to identify statistically significant bigrams to be included in the feature space. Our proposed method uses purely statistical methods, and it does not require a separate stage of parameter tuning, any language-specific setting or any data source other than tweet texts. To evaluate our method, we compared it with different settings of widely-used Naive Bayer classifiers, Kullback-Leibler divergence measures and neural networks. The experimental evaluations conducted on three datasets from different countries in the world showed that LocKDE$_{SCoP}$ yields statistically significant improvement in prediction accuracy in comparison to the state-of-the-art baselines. We also demonstrated that the analysis of spatial co-occurrence patterns of bigrams improves the accuracy of unigram models without incurring an excessive increase in the size of feature space.

References

Ajao, O., Hong, J., and Liu, W. (2015). A survey of location inference techniques on Twitter. *Journal of Information Science*, 41(6):855–864.

Bakerman, J., Pazdernik, K., Wilson, A., Fairchild, G., and Bahran, R. (2018). Twitter geolocation: A hybrid approach. *ACM Transactions on Knowledge Discovery from Data*, 12(3):34:1–34:17.

Celik, M. and Dokuz, A. S. (2018). Discovering socially similar users in social media datasets based on their socially important locations. *Information Processing & Management*, 54(6):1154–1168.

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

Chen, Y.-C. (2017). A tutorial on kernel density estimation and recent advances. *Biostatistics & Epidemiology*, 1(1):161–187.

Cheng, Z., Caverlee, J., and Lee, K. (2013). A content-driven framework for geolocating microblog users. *ACM Transactions on Intelligent Systems and Technology*, 4(1):2:1–2:27.

Chi, L., Lim, K. H., Alam, N., and Butler, C. J. (2016). Geolocation prediction in Twitter using location indicative words and textual features. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (W-NUT'16)*, pages 227–234.

Chong, W.-H. and Lim, E.-P. (2018). Exploiting user and venue characteristics for fine-grained tweet geolocation. *ACM Transactions on Information Systems*, 36(3):26:1–26:34.

Chong, W.-H. and Lim, E.-P. (2019). Fine-grained geolocation of tweets in temporal proximity. *ACM Transactions on Information Systems*, 37(2):17:1–17:33, doi:10.1145/3291059.

Compton, R., Jurgens, D., and Allen, D. (2014). Geotagging one hundred million Twitter accounts with total variation minimization. In *Proceedings of 2014 IEEE International Conference on Big Data*, pages 393–401.

Diggle, P. (2003). *Statistical analysis of spatial point patterns.* Edward Arnold. 2nd edition.

Doran, D., Gokhale, S. S., and Dagnino, A. (2014). Accurate local estimation of geo-coordinates for social media posts. In *Proceedings of the 26th International Conference on Software Engineering and Knowledge Engineering*, pages 642–647.

Dredze, M., Osborne, M., and Kambadur, P. (2016). Geolocation for Twitter: Timing matters. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'16)*, pages 1064–1069.

Ebrahimi, M., ShafieiBavani, E., Wong, R., and Chen, F. (2017). Exploring celebrities on inferring user geolocation in Twitter. In *Advances in Knowledge Discovery and Data Mining - Proceedings of the 21st Pacific-Asia Conference (PAKDD'17), Part I*, pages 395–406.

Eisenstein, J., O'Connor, B., Smith, N. A., and Xing, E. P. (2010). A latent variable model for geographic lexical variation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP'10)*, pages 1277–1287.

Flatow, D., Naaman, M., Xie, K. E., Volkovich, Y., and Kanza, Y. (2015). On the accuracy of hyper-local geotagging of social media content. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining (WSDM'15)*, pages 127–136.

Giridhar, P., Abdelzaher, T., George, J., and Kaplan, L. (2015). On quality of event localization from social network feeds. In *IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, pages 75–80.

Goreaud, F. and Pélissier, R. (1999). On explicit formulas of edge effect correction for Ripley's K-function. *Journal of Vegetation Science*, 10(3):433–438.

Graham, M., Hale, S. A., and Gaffney, D. (2014). Where in the world are you? geolocation and language identification in Twitter. *The Professional Geographer*, 66(4):568–578.

Han, B., Cook, P., and Baldwin, T. (2014). Text-based Twitter user geolocation prediction. *Journal of Artificial Intelligence Research*, 49(1):451–500.

Han, B., Rahimi, A., Derczynski, L., and Baldwin, T. (2016). Twitter geolocation prediction shared task of the 2016 workshop on noisy user-generated text. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (W-NUT'16)*.

Hecht, B., Hong, L., Suh, B., and Chi, E. H. (2011). Tweets from Justin Bieber's heart: The dynamics of the location field in user profiles. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'11)*, pages 237–246.

Hoang, T. B. N. and Mothe, J. (2018). Location extraction from tweets. *Information Processing & Management*, 54(2):129–144.

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

Jayasinghe, G., Jin, B., Mchugh, J., Robinson, B., and Wan, S. (2016). CSIRO Data61 at the WNUT Geo Shared Task. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (W-NUT'16)*, pages 218–226.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2017). Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL'17)*, pages 427–431.

Jurgens, D., Finethy, T., McCorriston, J., Xu, Y., and Ruths, D. (2015). Geolocation prediction in Twitter using social networks: A critical analysis and review of current practice. In *International AAAI Conference on Web and Social Media (ICWSM'15)*, pages 188–197.

Krishnamurthy, R., Kapanipathi, P., Sheth, A. P., and Thirunarayan, K. (2015). Knowledge enabled approach to predict the location of Twitter users. In *European Semantic Web Conference (ESWC'15)*, pages 187–201.

Lee, K., Caverlee, J., and Webb, S. (2010). Uncovering social spammers: Social honeypots + machine learning. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'10)*, pages 435–442.

Lee, K., Ganti, R. K., Srivatsa, M., and Liu, L. (2014). When Twitter meets Foursquare: Tweet location prediction using Foursquare. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MOBIQUITOUS'14)*, pages 198–207.

Li, P., Lu, H., Kanhabua, N., Zhao, S., and Pan, G. (2018). Location inference for non-geotagged tweets in user timelines. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, doi:10.1109/TKDE.2018.2852764 (Early Access).

Lichman, M. and Smyth, P. (2014). Modeling human location data with mixtures of kernel densities. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'14)*, pages 35–44.

Lu, Y., Hu, X., Wang, F., Kumar, S., Liu, H., and Maciejewski, R. (2015). Visualizing social media sentiment in disaster scenarios. In *Proceedings of the 24th International Conference on World Wide Web (WWW'15)*, pages 1211–1215.

Mahmud, J., Nichols, J., and Drews, C. (2014). Home location identification of Twitter users. *ACM Transactions on Intelligent Systems and Technology - Special Section on Urban Computing*, 5(3):47:1–47:21.

Melo, F. and Martins, B. (2017). Automated geocoding of textual documents: A survey of current approaches. *Transactions in GIS*, 21(1):3–38.

Middleton, S. E., Kordopatis-Zilos, G., Papadopoulos, S., and Kompatsiaris, Y. (2018). Location extraction from social media: Geoparsing, location disambiguation, and geotagging. *ACM Transactions on Information Systems*, 36(4):40:1–40:27.

Miura, Y., Taniguchi, M., Taniguchi, T., and Ohkuma, T. (2016). A simple scalable neural networks based model for geolocation prediction in Twitter. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (W-NUT'16)*, pages 235–239.

Ozdikis, O., Oğuztüzün, H., and Karagoz, P. (2016). Evidential estimation of event locations in microblogs using the Dempster-Shafer theory. *Information Processing & Management*, 52(6):1227–1246.

Ozdikis, O., Oğuztüzün, H., and Karagoz, P. (2017). A survey on location estimation techniques for events detected in Twitter. *Knowledge and Information Systems*, 52(2):291–339.

Ozdikis, O., Ramampiaro, H., and Nørvåg, K. (2018a). Locality-adapted kernel densities for tweet localization. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'18)*, pages 1149–1152.

Ozdikis, O., Ramampiaro, H., and Nørvåg, K. (2018b). Spatial statistics of term co-occurrences for location prediction of tweets. In *Proceedings of the 40th European Conference on Information Retrieval (ECIR'18)*, pages 494–506.

Paraskevopoulos, P. and Palpanas, T. (2016). Where has this tweet come from? Fast and fine-grained geolocalization of non-geotagged tweets. *Social Network Analysis and Mining*, 6(1):89.

Paule, J. D. G., Moshfeghi, Y., Macdonald, C., and Ounis, I. (2018a). Learning to geolocalise tweets at a fine-grained level. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM'18)*, pages 1675–1678.

Paule, J. D. G., Sun, Y., and Moshfeghi, Y. (2018b). On fine-grained geolocalisation of tweets and real-time traffic incident detection. *Information Processing & Management*, doi:https://doi.org/10.1016/j.ipm.2018.03.011.

Poulston, A., Stevenson, M., and Bontcheva, K. (2017). Hyperlocal home location identification of Twitter profiles. In *Proceedings of the 28th ACM Conference on Hypertext and Social Media (HT'17)*, pages 45–54.

Priedhorsky, R., Culotta, A., and Del Valle, S. Y. (2014). Inferring the origin locations of tweets with quantitative confidence. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing (CSCW'14)*, pages 1523–1536.

Qahtan, A., Wang, S., and Zhang, X. (2017). KDE-Track: An efficient dynamic density estimator for data streams. *IEEE Transactions on Knowledge and Data Engineering*, 29(3):642–655, doi:10.1109/TKDE.2016.2626441.

Rahimi, A., Cohn, T., and Baldwin, T. (2017). A neural model for user geolocation and lexical dialectology. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'17)*, pages 209–216.

Ripley, B. D. (1977). Modelling spatial patterns. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(2):172–212.

Rodrigues, E., Assunção, R., Pappa, G. L., Renno, D., and Meira Jr., W. (2016). Exploring multiple evidence to infer users' location in Twitter. *Neurocomputing*, 171(C):30–38.

Roller, S., Speriosu, M., Rallapalli, S., Wing, B., and Baldridge, J. (2012). Supervised text-based geolocation using language models on an adaptive grid. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL'12)*, pages 1500–1510.

Rout, D., Bontcheva, K., Preoţiuc-Pietro, D., and Cohn, T. (2013). Where's @wally?: A classification approach to geolocating users based on their social ties. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media (HT'13)*, pages 11–20.

Ruocco, M. and Ramampiaro, H. (2015). Geo-temporal distribution of tag terms for event-related image retrieval. *Information Processing & Management*, 51(1):92–110.

Schulz, A., Hadjakos, A., Paulheim, H., Nachtwey, J., and Mühlhäuser, M. (2013). A multi-indicator approach for geolocalization of tweets. In *Proceedings of the International AAAI Conference on Web and Social Media (ICWSM'13)*, pages 573–582.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London.

Van Laere, O., Quinn, J., Schockaert, S., and Dhoedt, B. (2014). Spatially aware term selection for geotagging. *IEEE Transactions on Knowledge and Data Engineering*, 26(1):221–234.

Wing, B. and Baldridge, J. (2014). Hierarchical discriminative classification for text-based geolocation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*, pages 336–348.

Yamaguchi, Y., Amagasa, T., Kitagawa, H., and Ikawa, Y. (2014). Online user location inference exploiting spatiotemporal correlations in social streams. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM'14)*, pages 1139–1148.

Zhang, J.-D. and Chow, C.-Y. (2013). iGSLR: Personalized geo-social location recommendation: A kernel density estimation approach. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL'13)*, pages 334–343.

Zheng, X., Han, J., and Sun, A. (2018). A survey of location prediction on Twitter. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1652–1671.

Zubiaga, A., Voss, A., Procter, R., Liakata, M., Wang, B., and Tsakalidis, A. (2017). Towards real-time, country-level location classification of worldwide tweets. *IEEE Transactions on Knowledge and Data Engineering*, 29(9):2053–2066, doi:10.1109/TKDE.2017.2698463.