Noname manuscript No.
(will be inserted by the editor)

Modeling Locations with Social Media

Neil O'Hare · Vanessa Murdock

Received: date / Accepted: date

Abstract In this paper we focus on the locations explicit and implicit in users descriptions of their surroundings. We propose a statistical language modeling approach to identifying locations in arbitrary text, and investigate several ways to estimate the models, based on the term frequency and the user frequency. The geotagged public photos in Flickr serve as a convenient ground truth. Our results show that we can predict location within a 1 km by 1 km cell with 17 percent accuracy, and within a 3 km radius around such a 1 km cell with 40 percent accuracy, using only a photo's tags. This is significantly better than the state of the art. Further we examine several estimation strategies that leverage the physical proximity of places, and show that for sparsely represented locations, smoothing from the immediate neighborhood improves results. We also show that estimation strategies based on user frequency are much more reliable than approaches based on the raw term frequency.

Keywords language models · geographic context · geotagging · user-generated content · Flickr

1 Introduction

We lead a double life in parallel social systems. In our everyday experience, we have friends, family, events, and social connections that happen in the real world, and enrich and give meaning to our lives. We also have friends, family, events and social connections that exist primarily online, which also enrich and give meaning to our lives. For most of us, our online life and our offline life have points of intersection such as events

Neil O'Hare
Yahoo! Research
Barcelona, Spain
E-mail: nohare@yahoo-inc.com

Vanessa Murdock
Yahoo! Research
Barcelona, Spain
E-mail: vmurdock@yahoo-inc.com

that we arrange online, but that take place offline, or places that we visit and then photograph, discuss, and share with our online social community. In this paper we focus on location as an important link between our online and offline lives.

As GPS-enabled devices become ubiquitous, online social platforms increasingly leverage location as an important aspect of our online social context. Web 2.0 platforms such as Foursquare, Flickr, Facebook and Twitter connect people to each other and to their surroundings. Because of the availability of location-based services and advances in personal computing devices such as smart phones and tablet computers, people increasingly expect services such as search and advertising to be location-savvy as well. With our portable devices increasingly intelligent, we expect services to understand our geographic context without having to explicitly indicate our location, or even to enable GPS on our handheld devices.

Geographic context goes beyond geographic coordinates to include the user's current, previous, and future locations, the locations referenced in the user's information need, as well as the locations explicit and implicit in the user's interaction with the device. For example, if the application is local search, and the user is looking for a dry cleaner on the way to work, the system should understand the trajectory between the user's current location and their destination, as well as alternate routes in between. Ideally the user need not provide disambiguating locations. Instead he would query with "dry cleaner on the way to work" and be presented with a list of feasible options between home and work, within a minimal distance out-of-the-way. This type of natural interaction with a search system is still beyond the limits of the state of the art.

To imbue a search system with such refined location awareness, we must understand the user's present location. We might use cell-phone tower IDs, however, disclosing this type of private information to an external application may be considered a violation of the users' privacy, as they cannot control the use of this information. GPS coordinates may be available to the search engine via the user's mobile device, if the user chooses to opt in to allow this information to be known. If the user opts out, however, he still expects the search engine to return relevant results. This is a reasonable expectation. A user should be able to get satisfactory search results in a local context without giving up his privacy.

As the user's current location is only part of the puzzle, we must understand how the user himself describes his location. In order to recognize locations implicit in queries, or described in content that might be presented to the user, we must recognize the words people use to describe places. People may include location-specific terms such as "The Mission" (in San Francisco) or "Park Hill" (in Denver). They may include vernacular place names that are not included in the official representation of a place ("The Gherkin", "Ole Miss"), and they may include terms or phrases that imply a specific location, but are not any kind of location (such as "Gaudi" or "double-decker bus").

Using public geo-tagged photos from Flickr[1] as our testbed, we propose a statistical language modeling approach to capture relationships between language and location. These language models can then be used to estimate the geographic focus of any given text excerpt created by a user, serving as a useful enabling technology for location-based services. We investigate several ways to estimate these models, based on the term frequency and the user frequency. Although we are not specifically concerned with the

[1] www.flickr.com visited March 2011

task of placing images on a map, we use this framework to evaluate the quality of our approach, as the geotagged public photos in Flickr provide a convenient ground truth. Our results show that we can predict location within a 1 km by 1 km cell with 17 percent accuracy, and within a 3 kilometer radius around such a 1 km cell with 40 percent accuracy, using only a photo's tags. This is significantly better than the state of the art. Further, we examine several estimation strategies that leverage the physical proximity of places, and show that for sparsely represented locations, smoothing from the immediate neighborhood improves results.

The rest of the paper is organized as follows. In Section 2 we discuss recent work in modeling locations. In Section 3 we give a brief overview of the location models, including details about how we segment the globe into a grid of location candidates. We discuss the data and the experimental setup in Section 4. We present and discuss our experimental results in Section 5. In Section 6, we then present results of a subset of our experiments on a slightly smaller, but publicly available, dataset, to provide results that can be replicated by other researchers. In Section 7 we describe an application of these models to geo-locating new images from Getty. This paper is an extended version of the work of Serdyukov et al. (2009). Since some of our results and conclusions differ from theirs, we examine the reasons for these differences in Section 8. The final section presents a summary of our conclusions.

2 Related Work

We propose to model locations using data from Flickr, although such a system could be extended to other types of geo-tagged data from, for example, location-based services such as Twitter. As stated above, the work of Serdyukov et al. (2009) is discussed in Section 8. Here we present an overview of other related work.

Crandall et al. (2009) leverage image content and textual metadata to predict the location of a photograph at two levels of granularity: at the city level (approximately 100 km) and at the individual landmark level (approximately 100 m). While this seems at first glance to be comparable to our work, in fact the task is defined quite differently. They choose a fixed set of cities, and report their ability to predict which of ten landmarks in a given city is represented in a photograph. Their experiments are limited to a specific set of landmarks in a fixed set of cities, as there are no images in their test or training sets that represent places outside of this set of locations. By contrast our system seeks to predict which of any arbitrary grid cells an image originated from, anywhere in the world, which is a much more difficult task. Nonetheless our results at predicting arbitrary 100 km grid cells are comparable to their city-level classification over 10 candidate cities (roughly 60 percent accuracy for both systems). They find that for this task the addition of visual content features does not improve over the textual metadata. Rather, the visual information in combination with the textual information is helpful in predicting which of ten landmarks in a given city is represented in the photograph. This demonstrates that the visual characteristics of landmarks are distinct within a city. It also emphasizes that the SIFT visual image features (Lowe, 2004) are particularly powerful for object matching tasks like detecting specific instances of a landmark, but they are not as powerful for detecting higher-level semantic categories like specific cities.

Visual features for predicting geographic locations were also investigated in the work of Hays and Efros (2008). They use a nearest-neighbor classification method and

they evaluate their system on a subset of Flickr photos that have been identified in the tags with at least one location, and which do not have certain non-geographic tags such as "birthday", "concert", etc. Their system is able to classify 16% of images within 200 km. Although this does not come close to the accuracy of systems that also leverage textual metadata, it demonstrates that visual information is useful for this task.

Another approach to associating location data to images is to select tags to assign to them. Ahern et al. (2007) identify geographically related tags by finding dense areas using geodesic distances between images, and ranking all tags in these areas with a $tf.idf$-based metric. In follow-on work, they leverage tags that represent local events (Rattenbury et al., 2007). Similarly, Naaman et al. (2003) and Moxley et al. (2008) propose approaches for recommending tags to the user, given a known location for an image. It is reasonable to assume the location of an image is known, as many images are geotagged, and even more will be in the future. However, even if an image is geo-tagged, the coordinates represent a point on the globe, whereas the image may represent a much larger area. For example, a person may photograph a city view from the top of a tall building. The geo-coordinates will indicate the tall building, while the image and its tags actually represent the entire city. Another example would be a photograph of a street scene or landscape intended to represent an iconic image of a neighborhood, city or country. We wish to apply our models to data that does not have any geo-coordinates associated with it (such as search engine queries), thus in our experimental setup we do not assume that the location is known.

Yi et al. (2009) use language modeling to determine the locations implicit in queries. They sample a large corpus of search engine queries, and identify the location mentions in the queries using a proprietary tool specifically designed for this purpose. They create a test set of queries mentioning a city name, by removing any locations identified by their tool, to create a query in which the location is implicit. Their task is to recover the city referred to in the original query. The location identification tool identifies the single most important location in a query. Thus, in a query such as "restaurants near Times Square in New York City", the tool will identify the location "New York City", and after removing this the remaining implicit query is "restaurants near Times Square". The authors evaluate their method over approximately 1600 U.S. cities identified in their data, and they report extremely high accuracy at predicting the city a query refers to. We believe this is because the tool only identifies one location, in the case that more than one is referred to. Identifiers such as Times Square, or Disneyland, or the University of Denver, were included in the test set, and so the high performance of the models is likely due to the presence of points of interest, suburbs, or neighborhoods within a city, which uniquely identify the city. However, points of interest, suburbs, and neighborhoods are themselves locations and represent explicit mentions of locations in queries rather than implicit mentions of locations.

Jones et al. (2008b) characterize geographic modification in query rewrites. This work uses the same proprietary tool to identify the locations referred to in queries described by Yi et al. above, and then profiles the distance between the user (determined by his IP address) and the locations mentioned in the query. The aim is to discover a class of queries whose intention is local. They further analyze query sessions to characterize the modification of queries that mention geographic entities. Their paper provides an example application of location modeling, which is local intent discovery.

Other work relating locations to queries include Backstrom et al. (2008), who measure the geo-specificity of a query using the level of dispersion around the location of the query's highest frequency. With a similar goal in mind, Zhuang et al. (2008)

calculate the inverse correlation of a query's click distribution over locations with their populations. Vadrevu et al. (2008) use the probability of co-occurrence of a query term with place names from each region to determine queries that might be related to a given region.

Working at a smaller granularity, Hollenstein and Purves (2010) examine the use of tags indicating vernacular regions in Flickr. Vernacular regions are commonly referred to by people, although they may not represent official places. Examples of vernacular regions are "Downtown", "The Shops", or "The High Street". The authors determine that Flickr can be used as a resource for identifying vernacular regions in cities. Their work centers around case studies of six cities in the U.S. and Europe, and provides a characterization of the tagging of vernacular regions in Flickr. With reference to our work, vernacular regions will not always help us distinguish one city from another (most major U.S. cities have an area commonly referred to as "downtown"), but they may be a reliable indicator of regions within a given city. Furthermore, this is one of the few studies that attempts to identify regions smaller than a city.

Vague places include vernacular regions that are commonly understood, but not defined by any official boundary. An example would be "The Midlands" or "South Denver". It is possible that people differ in the precise boundaries of these locations, but the boundaries can be drawn in a fuzzy manner to allow for geographical information retrieval, for example, or other applications that do not rely on a precise boundary. Jones et al. (2008a) query the web for a small number of vague locations in the U.K., and then extract specific location mentions from the pages whose topic is the vague location. For example, they extract city names from documents about "The Highlands" (referring to the Scottish Highlands). They do Kernel Density Estimation to estimate the boundary of the vague region, according to the extents given by specific place mentions within the web documents.

We have focused on the previous work that seeks to predict locations in Flickr data and query logs, although there is a wealth of research on identifying and predicting locations in other types of data, such as building location topic models from blog data (Mei et al., 2006; Wang et al., 2007), finding the geographic focus of web pages (Ding et al., 2000; Amitay et al., 2004; Zong et al., 2005), and estimating the home location of twitter users (Cheng et al., 2010; Eisenstein et al., 2010).

3 Modeling Locations

There are many gazetteers available to developers that represent places according to a hierarchy (such as Geonames[2], and Placemaker[3]). Furthermore there is proprietary data gathered by companies such as Ordnance Survey[4], Navteq[5] and TeleAtlas[6], which determines place boundaries, hierarchies, and containment. There is also open source data such as OpenStreetMap[7]. Each of these systems represents places differently. Most common is to represent a place as a centroid and a bounding box. This is a simplification of the point-in-polygon approach, but gives a generally accurate approximation of the

[2] http://www.geonames.org/ visited January 2012
[3] http://developer.yahoo.com/geo/placemaker/ visited January 2012
[4] http://www.ordnancesurvey.co.uk/oswebsite/ visited January 2012
[5] http://www.navteq.com/ visited January 2012
[6] http://www.teleatlas.com visited January 2012
[7] http://www.openstreetmap.org/ visited January 2012

extent of a place. The difficulty is that many, if not most, of the places we care about for any application we might think of are represented differently by each system. In fact, the systems have different coverage (some represent cities especially well, others focus on roadways, or points of interest), but beyond that, the systems do not agree on bounding box extents, or centroids. In fact, finding a concordance between them in an automatic way is still an open question. Furthermore, while many places have official, agreed-upon, boundaries, many well-established places commonly understood by the citizens of that place either have no official boundary, or the boundary is not universally agreed upon by the people inhabiting the place. This is especially true for neighborhoods, but is also true for large regions, such as the Rocky Mountains. (Jones et al., 2008a)

As a complicating factor, there are officially defined geo-political areas, such as postal codes, which do not fit neatly into any hierarchy. Consider the town of Banksville, NY. It happens that Banksville is officially a city in New York in the U.S. The town itself spills over into Connecticut, and is part of the Connecticut zip code *06831*. So if you start at Banksville, and traverse the hierarchy upwards, you find that its parent is New York. However if you start at the state level, and traverse the hierarchy down, you find that Banksville is the child of both Connecticut and New York. This type of inconsistency is unavoidable because places are defined by people organically, without concern for administrative tidiness.

Because of these difficulties in representing places, and because we wish to be as generic as possible in designing a system that might be used for any number of applications, we seek a representation of places that is agnostic with regard to the place hierarchy and the official representation. We define the boundaries of places on a generic $n \times n$ grid, where n is a parameter that represents the length of the side along the longitude lines. (The length of the side along the latitude lines will vary by latitude. For example, the distance between two longitudinal lines at the equator will narrow as you travel away from the equator, to zero-width at the poles.)

We further represent each place with a model of the language people use to describe that place. This allows us to reflect common references to places even when they differ from official names. It also allows us to capture colloquial names, language variants, and additional characteristics of a place (such as typical activities that take place there, or other salient features). In this work we present a prototype based on Flickr, but any geo-tagged textual data could be used to model places. As more geo-tagged data becomes available, we expect the models of locations to be a richer representation of place.

3.1 Placing a Grid on the Globe

We represent the globe as a grid by quantizing the latitude/longitude values to create grid cells that are 100km, 10km and 1km along the longitude lines. There are other representations of locations that create equal area grid cells based on triangles, hexagons or pentagons (Toyama et al., 2003), but these are much more costly to compute and we do not investigate them in this work. It is not clear that this extra care in defining locations yields an improved performance for a given application. From here forward we refer to grid cells interchangeably as “locations”, “cells” or “grid cells”.

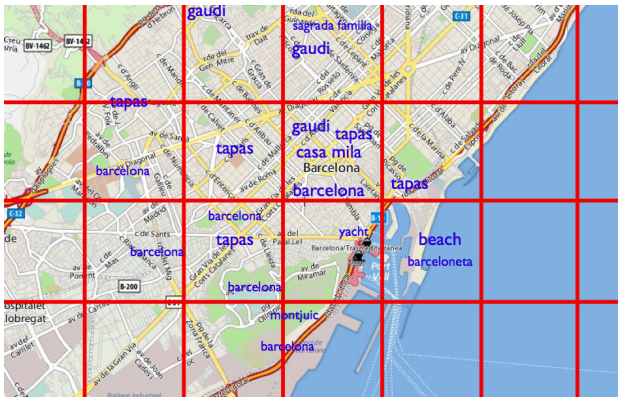

Fig. 1 Example grid-based representation of Barcelona.

3.2 Language Models for Location

We associate each geo-tagged image from Flickr with a unique grid cell. Thus, each location is associated with a set of photos and their textual metadata. In this work we use the photos' tags (short words or phrases provided by users to describe the image), as they are the most succinct representation of the content and context of the photo. Sigurbjornsson and van Zwol (2008) found that of the 52% of tags that they could classify using WordNet, 28% of them were classified as locations. Hollenstein and Purves (2010) found tag sets useful for modeling vernacular regions (in their case "downtown"), further suggesting that the tag sets are a valuable source of place data. Figure 1 shows an example of a grid-based representation of Barcelona, and some tags that one would typically expect to find in the cells of this grid. Although the tags will occasionally explicitly refer to a place name (e.g. "barcelona"), this is not always the case, and tags may be used which indirectly reflect the location (e.g. "tapas", "gaudi").

When users tag their photos in Flickr, they provide single words, or phrases, separated by commas. For example, a photo of the Sagrada Familia basilica in Barcelona might be described with "Barcelona, Gaudi, sagrada familia, church". In this case, the photo is described with four tags. We consider "Sagrada Familia" to be one tag because it represents a single concept identified by the user who chose not to separate the two words with a comma. Flickr normalizes the tags by lower-casing them, removing white space, and replacing commas with white space. So the Flickr normalization produces the four tags "barcelona gaudi sagradafamilia church". Normalizing the tag-sets in this way has the benefit that it allows phrase information to be retained. In this example, Sagrada Familia is treated as one entity, "sagradafamilia". Since we wish to apply our models to any source of text (such as queries, or free text, which do not have this neatly preserved phrase information) we use the raw tags, converting all tags to lowercase but maintaining whitespace. The tag 'Sagrada Familia' in this representation becomes 'sagrada famila' and the phrase information is not maintained, leading to a 'bag of words' approach, as opposed to a 'bag of tags' approach. We do, however, use the normalised tags in our baseline evaluation in Section 8, to facilitate comparison with the work of Serdyukov et al. (2009).

We build a language model from the textual representation for each location. Language Models have been used in speech recognition, optical character recognition and

machine translation (Manning and Schütze, 1999), and were originally proposed for information retrieval by Ponte and Croft (1998). We adopt the retrieval framework, treating each location as a pseudo-document, whose term distribution can be estimated, and then each location can be ranked against some query text according to its probability of having "generated" the query.

We estimate the likelihood of an individual term, given the language model of a location, using the maximum likelihood estimate (MLE), which maximizes the observed likelihood given the data:

$$P_{mle}(t|\theta_L) = \frac{c(t, L)}{|L|}, \tag{1}$$

where $c(t, L)$ is the term frequency of the term t in location L, and $|L|$ is the total number of terms in the location.

Given some arbitrary text, T, that we wish to locate, we rank candidate locations by their probability, given the text, $P(L|T)$:

$$P(L|T) = \frac{P(T|\theta_L)P(L)}{P(T)}, \tag{2}$$

where $P(T|\theta_L)$ is the probability of the text given the model of the location, θ_L. If we assume independence between terms, this can be calculated as the product of the probabilities of the individual terms:

$$P(T|\theta_L) = \prod_{i=0}^{|T|} P(t_i|\theta_L). \tag{3}$$

In Equation 2, $P(T)$, the prior probability of the candidate text can be ignored since it is a constant for all locations and does not affect the ranking. $P(L)$ is the prior probability of a location and is typically assumed to be uniform. This leads to a model where locations are ranked solely by $P(T|\theta_L)$, the probability that the location model created the query text, a ranking approach known as query likelihood.

Assuming that $P(L)$ is uniform makes sense if we assume that all locations have an equal prior probability of relevance. However, we may want to take advantage of the fact that some locations are inherently more popular than others. For example, we might want to take advantage of that fact that an island in the middle of the Pacific ocean has a very low prior, whereas locations in Manhattan, New York, have a much higher prior probability. Since we are building models from Flickr photos, we can easily compute the prior, $P(L)$, as the number of photos in a cell divided by the total number of photos in the collection, and then rank candidate locations based on $P(T|\theta_L)P(L)$.

3.3 User Frequency for Estimating Term Probabilities

In Equation 1 we estimate the probability of a term given a location with the maximum likelihood estimate. A problem with basing the estimate on the term frequency is that individual users who tag a lot of photos in a given location can come to dominate the textual representation of that location.

An alternative approach is to base the estimate on user frequency, the number of unique users who use the term in the location:

$$P_{user\_mle}(t|\theta_L) = \frac{c_{user}(t, L)}{|L|_{user}}, \tag{4}$$

where $c_{user}(t, L)$ is the number of unique users who use the term in the location. $|L|_{user}$ is calculated as the sum of the user frequency of all terms in the location:

$$|L|_{user} = \sum_{t_i \in L} c_{user}(t_i, L). \tag{5}$$

Estimating term probabilities based on user frequencies in this way alleviates bias caused by users who create a disproportionate number of tags in one location. It also reduces the effect of bulk-uploading. User frequency has previously been considered in other work for extracting representative tags for locations (Ahern et al., 2007; Kennedy et al., 2007), for finding similar locations (Clements et al., 2010) and suggesting relevant tags for photos (Moxley et al., 2008).

3.4 Smoothing

MLE estimates are problematic when dealing with sparse or missing data. In particular, terms not present in a location will have a zero probability, meaning that locations not containing all terms will have a probability of zero of "generating" a given candidate string. To deal with this, smoothing approaches allocate a portion of the probability mass to unseen events. Two common smoothing approaches are Jelinek Mercer smoothing and Dirichlet Smoothing, both of which smooth the probability distribution by combining the estimate with an estimate calculated from a background model that does not suffer from the same data sparseness problem. In the context of modeling locations, the background model we use is based on all locations on the globe.

Jelinek Mercer smoothing performs a simple linear interpolation between the location model and the background model:

$$P_{jm}(t|\theta_L) = (\lambda)P_{mle}(t|\theta_L) + (1 - \lambda)P_{mle}(t|\theta_G), \tag{6}$$

where λ must have a value between 0 and 1, $P_{mle}(t|\theta_L)$ is the MLE estimate of the probability of the term given the location, and $P_{mle}(t|\theta_G)$ is the probability of the term given the Global language model.

Dirichlet smoothing performs a similar linear interpolation, but varies the amount of smoothing according to the size of the candidate location, with larger locations (those containing more terms) smoothed less:

$$P_{dir}(t|\theta_L) = \frac{|L|}{|L| + \mu} P_{mle}(t|\theta_L) + \frac{\mu}{|L| + \mu} P_{mle}(t|\theta_G). \tag{7}$$

$|L|$ is the total number of terms for that location, and μ is the Dirichlet smoothing parameter, which can be understood as representing the strength of our belief in the background model.

3.4.1 Hierarchical Smoothing

It is likely that the terms used to describe the cells surrounding a given location will also be descriptive of that location. Thus when estimating the probabilities for terms in a given cell, we make use of the terms in neighboring cells. We illustrate this in Figure 2, where terms found in the neighboring cells are representative of the candidate cell. Also, there are situations in which tags specify an area larger than the size of the

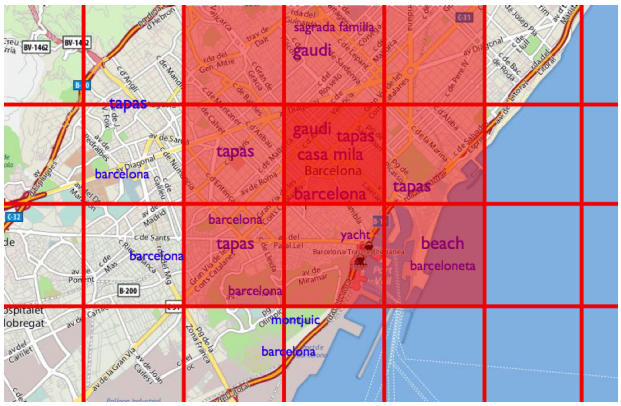

Fig. 2 Neighbouring Cells used in Hierarchical Smoothing

location representation. For example, tags that specify a country such as Ireland or a continent such as Europe specify areas much larger than that covered by 100km cells. In such cases, tags from neighboring cells can be considered descriptive of a location.

We define hierarchical versions of Jelinek Mercer and Dirichlet Smoothing which successively smooth with more and more general models. Such hierarchical models have been well studied. They were used by Westerveld et al. (2003) in video retrieval, who created hierarchical Jelinek Mercer language models by smoothing the estimate for a shot in a video with the estimate for the scene containing the shot, which was smoothed by the estimate for the video containing the scene, and finally by the estimate for the collection. In other work, Mc Donald and Smeaton (2005) proposed hierarchical Dirichlet language models for video, while O'Hare and Smeaton (2009) used hierarchical language models for context-based person identification in personal photo collections. Hierarchical smoothing has also been used in sentence and passage retrieval (Murdock, 2006), where a sentence is smoothed first by the document containing it, and then by the collection. Hierarchical smoothing of location models is particularly intuitive because the models represent places that are physically near each other, and thus likely to be described with similar terms.

We define hierarchical versions of Jelinek Mercer and Dirichlet Smoothing that take into account estimates for the probability of a term given a neighborhood around a candidate location, $P(t|\theta_{N_d})$. The size of the neighborhood is specified by the parameter d, and consists of all cells within distance d from a location. For 3-level Jelinek Mercer smoothing, this gives:

$$P_{jm\_hier}(T|\theta_L) = \lambda_L P_{mle}(t|\theta_L) + \lambda_{N_1} P_{mle}(t|\theta_{N_1}) + \lambda_G P_{mle}(t|\theta_G), \tag{8}$$

where the parameters λ_L, λ_{N_1} and λ_G represent weights for the location, neighborhood and global models, and sum to one. In this work we also consider 4-level hierarchies, where a location is smoothed by its neighborhood at $d = 1$, by its neighborhood at $d = 2$, and then by the collection. It would also be possible to explore 3-level and 4-level hierarchies with different values of d. We leave this to future work.

We create similar hierarchical Dirichlet models by smoothing the neighborhood estimate with the background model, and replacing $P(t|\theta_G)$ in Equation 7 with the neighborhood estimate:

$$P_{dir\_hier} = P_{dir\_l}(P_{dir\_n}). \tag{9}$$

We can also recursively replace $P(t|\theta_G)$ with smoothed estimates from neighborhoods of varying sizes to create hierarchies of arbitrary depth.

To distinguish this approach from the cell-based hierarchical smoothing described below, we will refer to it as term-based hierarchical smoothing, as it smoothes estimates for individual terms.

3.4.2 Cell-based Smoothing

Rather than using hierarchical smoothing based on neighborhoods to improve the estimate of the probability of a term given a location, an alternative way to incorporate neighborhood information is to re-rank the results by considering the relevance score of neighboring cells (locations) when calculating the final score for a location. In this approach, term probabilities estimates are smoothed from the collection (as in equations 6 and 7), and an initial ranking is created. The final score is the combination of the score for a location and its neighbors:

$$P_{cell\_hier}(T|\theta_L) = \alpha P(T|\theta_L) + (1-\alpha) \sum_{L' \in N_d} \frac{P(T|\theta_{L'})}{|N|}, \tag{10}$$

where $|N|$ is the number of cells in the neighborhood of L (this would be eight neighboring cells for $d = 1$).

We refer to this approach as cell-based hierarchical smoothing, although it is not strictly hierarchical as the scores for all terms in the neighboring cells are already smoothed by the collection before being combined with the smoothed score for the candidate location. As such, this method is not a smoothing method at all but a re-ranking of global smoothing scores based on the scores of nearby locations.

3.4.3 Directional Relevance Progagation

For their cell-based smoothing models, Serdyukov et al. (2009) propose directional relevance propagation, where the relevance scores are only propagated from locations that have lower relevance scores than the locations to be smoothed. This will lead to only "local winners" in a neighborhood having their scores boosted.

We propose an analogous approach for term-based smoothing of term probabilities. For hierarchical smoothing, terms that appear in a location will be smoothed hierarchically with the neighborhood models. Terms that do not appear in a location will only be smoothed by the collection model. Thus only seen terms have their score estimates 'boosted' by the neighborhood estimate. This is similar to directional relevance propagation for cell-based smoothing, with the added advantage that the directional propagation takes place on a term by term basis.

We will refer to each of these approaches as 'directional' smoothing as they are analogous in how they prefer to boost the scores of locations which are already more probable, even though the implementation of each is quite different.

4 Experimental Setting

Our data set, which we refer to as the "10M Dataset", is a large scale data set made up of over 10 million public photos uploaded to Flickr from more than 320,000 users, after

Table 1 Statistics about the 100km, 10km and 1km models.

100km	
Training	
Unique Locations (Training)	15,428
Average Location Size (Term Fr.)	5,225
Average Location Size (User Fr.)	1,645
Average Photos per Location	525
Test	
Unique Locations	7,364
Unique Locations Correctly Predicted	3,644
Photos Not in Model	0.15%
10km	
Training	
Unique Locations (Training)	181,135
Average Location Size (Term Fr.)	445
Average Location Size (User Fr.)	167
Average Photos per Location	45
Test	
Unique Locations (Test)	58,180
Unique Locations Correctly Predicted	13,789
Photos Not in Model	2.94%
1km	
Training	
Unique Locations (Training)	886,685
Average Location Size (Term Fr.)	91
Average Location Size (User Fr.)	41
Average Photos per Location	9
Test	
Unique Locations	175,746
Unique Locations Correctly Predicted	16,295
Photos Not in Model	21.66%

a bulk upload filter has been applied and photos with empty tag sets were removed.[8] The purpose of this filter is to combat the effect of bulk uploads, where a user tags a number of photos with identical tags. When there are multiple photos from the same user with identical tags, we keep only one of these photos, randomly chosen. We partitioned this set into training, tuning, and test sets, giving approximately 8 million photos for building the models, and approximately 1 million photos each for tuning the model parameters and for evaluation. As we are interested in building general models of location that have applications beyond placing photos on a map (and so may use text from a variety of sources, not just normalized Flickr tags), we only use the raw tags representation when evaluating this dataset.

Table 1 describes some characteristics of the models and the test set, with respect to 100km, 10km and 1km cells. The average location size is based on the total number of tags found in each location, while the average location size for user frequency is based on the sum of the user frequencies for every term in each location. The table also shows the number of unique locations in the test set, along with the number of those locations in which at least one photo was correctly placed (from the user frequency Dirichlet model). We see that photos can be correctly placed in approximately half of

[8] Note that Flickr has a public API which allows members of the research community to download metadata and images from the public photos of users. http://www.flickr.com/services/api/ visited January 2012

the locations in the test set for the 100km models, whereas a much smaller proportion of the locations are correctly predicted for the 1km model. The number of photos from the test set whose locations are not represented in the training data are also shown in Table. Obviously, the model cannot locate these photos correctly because their correct locations are not represented in the model. For the 1km model, over 21% of test photo locations are not represented in the training data. Locations in North America and Europe are disproportionately represented in the data. Locations in Asia, Africa and South America have less coverage.

4.1 Evaluation Measures

We evaluate the quality of our models by their predictive ability in recovering the grid cell from which a photo originated. Our primary evaluation metric is accuracy (Ac), the percentage of photos that are located correctly by the model. Since these models could also be used in an interactive setting, where a system would suggest a list of potential locations for an arbitrary text string, and since we may also be interested in the ability of the model to predict locations geographically close to the true location, we also report results for the following metrics:

Accuracy within K Cells (Ac@K) Accuracy within K cells measures the ability of the model to predict the correct location within K cells of the correct location. So, for the 100km model *Ac@1* would measure the ability of the model to predict the correct location within a 300km cell (100km, plus 100km in each direction).

Parent Accuracy (PAc) Parent accuracy determines the ability of the model to accurately predict the correct location at the parent level in the location representation hierarchy (i.e. for the 1km model, does it predict the correct 10km cell? Does the 10km model predict the correct 100km cell?).

Mean Reciprocal Rank (MRR) The mean rank of the ground truth location in the result list measures the ability of the model to return the correct result toward the top of the result list. The mean reciprocal rank is favored as an evaluation measure because it can be interpreted without knowing the number of documents, with a value between 0 and 1. It is not severely influenced by target documents retrieved at low ranks. Also, in settings such as this where there is only one relevant document, the reciprocal rank is the same as average precision (Kantor and Voorhees, 1996).

H-Hit Rate H-Hit rate measures the percentage of photos correctly placed in the top H results in the list (Chen et al., 2003). It is different from precision at K since it is a binary measure when calculated for a single test case. It could be considered a generalization of accuracy, with the value h determining how high in the result list the correct location should be for a result to be considered correct. For example, considering 5-hit rate, if the correct location is in the top 5 results, the 5-hit rate is 1 regardless of whether it was found at rank one or rank five (precision at 5 would be 0.2).

The accuracy metric is quite strict because the task is to recover the exact cell the photo originated from, whereas the photo itself might be more generic. For locations in a city that do not fall into a single point, such as coastlines, avenues, or neighborhoods,

Table 2 Baseline Results on Raw Tags over a large data set of 10 million Flickr photos. The table shows the results for both Dirichlet and Jelinek-Mercer smoothing, with a location prior (Pr) and without a location prior. Dirichlet smoothing with a uniform location prior seems to have a slight performance advantage, although the difference is small. μ and λ indicate the optimal smoothing parameters: m=million, k=thousand

Method	Ac	MRR	Ac1	Ac2	Ac3	PAc	3hit	5hit
100km								
$\text{Dir}_{\mu:9\text{m}}$	**0.498**	0.564	**0.567**	**0.5895**	**0.605**	-	**0.606**	0.638
$\text{Dir/Pr}_{\mu:100\text{k}}$	0.474	0.543	0.545	0.569	0.589	-	0.583	0.620
$\text{JM}_{\lambda:0.95}$	0.383	0.463	0.462	0.489	0.508	-	0.508	0.552
$\text{JM/Pr}_{\lambda:0.15}$	0.494	**0.565**	0.561	0.582	0.599	-	0.605	**0.643**
10km								
$\text{Dir}_{\mu:250\text{k}}$	0.375	0.458	0.508	0.537	0.553	0.539	0.515	0.558
$\text{Dir/Pr}_{\mu:20\text{k}}$	0.365	0.445	0.495	0.523	0.539	0.526	0.497	0.538
$\text{JM}_{\lambda:0.95}$	0.285	0.371	0.411	0.448	0.471	0.458	0.421	0.472
$\text{JM/Pr}_{\lambda:0.1}$	**0.386**	**0.470**	**0.520**	**0.548**	**0.564**	**0.549**	**0.525**	**0.569**
1km								
$\text{Dir}_{\mu:14\text{k}}$	**0.162**	**0.226**	**0.293**	**0.343**	**0.375**	0.355	**0.256**	0.297
$\text{Dir/Pr}_{\mu:4\text{k}}$	0.160	0.222	0.289	0.340	0.371	0.351	0.252	0.292
$\text{JM}_{\lambda:0.95}$	0.122	0.180	0.229	0.277	0.309	0.297	0.205	0.243
$\text{JM/Pr}_{\lambda:0.15}$	0.158	0.225	0.289	0.342	0.374	**0.356**	0.256	**0.300**

we might consider accuracy within 3 or 5 km to be sufficient. Thus the results reported for accuracy at 2 or 3 are more representative of the metrics required in practice. When evaluating our approaches, we optimized the model parameter settings by performing a grid search of the parameter space using the tuning set of the corpus, choosing the parameter settings that optimized the accuracy evaluation measure.

5 Results and Discussion

As mentioned previously, Flickr tags have a raw form and a normalized form. In initial experiments, we compared building models from the normalized tags to building the models from the raw tags. We found that the accuracy of the models built from raw tags was slightly lower than models built from normalized tags. This is to be expected as the normalized tag vocabulary is more specific to a location. For example, treating "newyork" as a unique tag eliminates uncertainty about whether the tag set refers instead to "york". However, it represents a type of over-fitting specific to Flickr, as the normalized tags are unlikely to be found in other types of data, such as query logs or web pages. For this reason we focus on models based on the raw tags and, unless otherwise specified, the results presented in this paper refer to models built from the raw tags.

The collection smoothing results and, in particular, the Jelinek Mercer smoothing approach without the location prior, are shown as baseline results, as these are standard approaches in information retrieval. Basic Jelinek Mercer smoothing does not have any bias towards popular locations, and uses the background model to compensate for zero-score estimates that are often returned by a Maximum Likelihood Estimate model, which performs too poorly to be considered a realistic baseline. These baseline results for the raw tags, smoothed from the collection, shown in Table 2, illustrate that it is possible to locate 16% of photos correctly within a 1km cell, 39% in a 10km cell and 50% in a 100km cell.

The parent accuracy for 10km cells performs better than the 100km model, which is an interesting result that warrants further investigation. Similarly the results for accuracy at 1, accuracy at 2 and accuracy at 3 for the 10km model, which equate to 30km, 50km and 70km cells, all outperform the 100km model. Since the 10km representation can be considered to roughly correspond to the city level, this might suggest that users are particularly good at creating tags that are descriptive of smaller areas such as cites, to such an extent that these models are more effective at describing a larger region than models specifically for those regions. Although the results in Table 2 do not show similar ability of 1km models to improve prediction results at the 10km level, we will show in Section 5.3 that with user frequency term estimates the 1km models do offer such an improvement, suggesting that tags are more often descriptive of the neighbourhood level rather than the city level.

The finer quantization used in the smaller cells may also be a factor in this better performance, as a finer-grained quantization is closer to the raw data, and the arbitrary positioning of the cell boundaries is less likely to impact the evaluation measures. To understand why, consider that accuracy at 3 (representing a 70km cell) may outperform its parent accuracy (a larger, 100km, cell) because the 70km cell is centered around the target location, whereas the parent cell is not. A location could be adjacent to another 10km cell but could be contained within a different 100km parent cell.

5.1 The Effect of the Location Prior

Intuitively, we can imagine that the prior probability of a location is not uniform. We might want to adjust the prior probability of a location to account for the coverage bias in Flickr, or the fact that certain locations simply contain more salient sub-parts. We examine the effect of using the prior probabilities of locations, $P(L)$, when calculating $P(L|T)$. Table 2 shows global smoothing results on the 10M dataset for Dirichlet and Jelinek Mercer smoothing, with and without the location prior. The best results are given by the Dirichlet approach without the location prior, and by the Jelinek Mercer approach with the location prior, while the performance of Jelinek Mercer smoothing with no location prior is well below that of all the other approaches. It is also noteworthy that the location prior harms performance for Dirichlet smoothing while giving a very large improvement for Jelinek Mercer smoothing.

We believe that the reason for this is related to the way in which the Dirichlet prior smooths longer documents less, a property of Dirichlet smoothing discussed at length by Smucker and Allan (2005). Locations associated with more terms will be smoothed less by the Dirichlet model. For documents that contain a term, the document probability will be higher than the collection probability, so smoothing these terms less will boost their score. For our location models, this behavior is exacerbated because the size of a location (the total number of terms in the photos it contains) is directly related to the prior probability (the number of photos in the location). In our setting a longer document corresponds to a location that has been photographed and tagged more often. In effect, the length of our location pseudo-documents is a proxy for the location's popularity. For Dirichlet smoothing the number of terms in a location is acting as a proxy for the prior probability of the location, and boosts the scores of locations with a high prior without using the prior directly. Since Jelinek Mercer does not make use of document length, including the prior probability for this model gives

a huge improvement. The performance of Jelinek Mercer smoothing with the location prior is broadly equivalent to that of the Dirichlet approach without the location prior.

Using the location prior explicitly with the Dirichlet model actually harms performance slightly, possibly due to the fact that the model is effectively taking the prior probability into account twice. This means that popular locations will have their already high prior overestimated (or overemphasized), adversely effecting performance. The Jelinek Mercer smoothing approach without the location prior is the only model that does not consider, either implicitly or explicitly, the prior probability of relevance for locations, and that would explain why there is such a large difference in the results for this model, achieving accuracy results approximately 25% below the other models for all cell sizes.

As shown in Table 2, the optimal results for Dirichlet smoothing are achieved with a very large Dirichlet prior parameter, which is an order of magnitude greater than the average documents sizes reported in Table 1. The optimal value μ= 9 million for 100km cells compares with an average location size of 5,225, and is also larger than largest individual location, which has almost 2.5 million terms. Similarly, the optimal values for μ for the 10km and 1km cells are much larger than the average document length. These high values for the smoothing parameter emphasise the behaviour of Dirichlet smoothing discussed above. For Dirichlet smoothing to penalise smaller locations in comparison with larger documents, the Dirichlet prior would have to be much larger than the document length of less popular locations, and comparable to the size of more popular locations. Since the most popular locations have a huge number of photos and a very large document length, the Dirichlet prior will also need to be very high to boost the scores of popular locations and achieve optimal performance. Whether this behavior is desirable or not depends on the application. For the placing task, recognizing that certain locations are more popular is beneficial to performance. For recommending off-the-beaten-path points of interest in a city, we may want a system that penalizes popular locations.

Figure 3 shows the results of the parameter search for the Jelinek Mercer models, as carried out on the tuning set of the corpus. A λ value of 0 (from Equation 6) indicates that the location model estimates for a candidate location are ignored, and only the global language model is considered when ranking locations. A value of 1, on the other hand, implies that only the location model is used and that the background model is ignored. Apart from the steep drops in performance as we approach the extreme values of 0 and 1, the performance is quite stable across parameter values, particularly when the location prior is used, a reflection of the fact the the prior probability has a strong influence on the ranking.

Ranking based on the document estimates only (λ=1), which corresponds to ranking by the maximum likelihood estimate (MLE) without any smoothing, gives accuracy of 0.204 without the prior, and 0.249 with the prior. This is surprisingly high, in fact, since MLE models are known to perform very poorly for information retrieval, mainly because they give a score of 0 to candidate documents that do not contain all of the query terms. Jelinek Mercer smoothing improves the performance by insuring candidate locations that do not contain all terms in a photo's tag set have a non-zero score. The smoothing will also naturally decrease the influence of terms that appear in many locations and thus do not carry any geographic information. This feature of Jelinek Mercer smoothing has been discussed at length previously (Hiemstra, 1998). Thus, the Jelinek Mercer model improves performance by assigning a non-zero score to missing terms and by down-weighting non-discriminative terms, but it does not have any direct

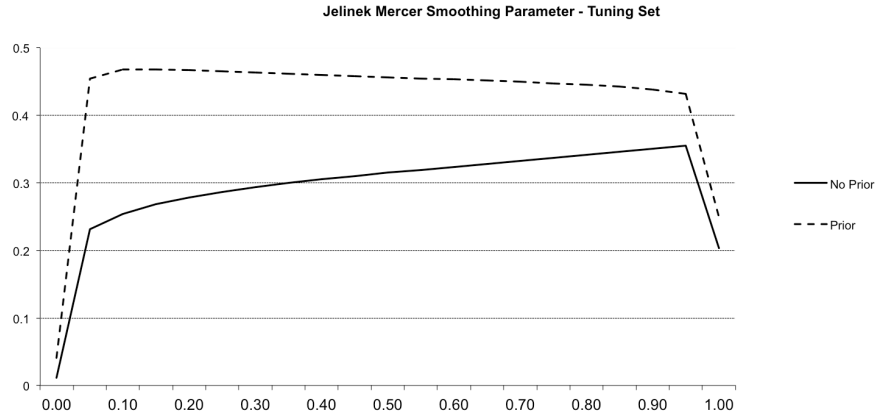

Fig. 3 Accuracy results for 100km model for different values of the Jelinek Mercer smoothing parameter, λ, with and without the location prior.

bias towards more popular locations. The reason why the MLE (λ=1) model works well for 100km cells relates to the size of the vocabulary. The large size of these cells and the fact that they are populated with many tags from many users, means most of the query vocabulary overlaps with the training vocabulary, so there will be fewer query photos with a score of 0.

The Jelinek Mercer model with λ=0 scores candidate locations (i.e. those containing at least one query term) based on the global model only. Without the location prior, this means that all candidate locations will have the same score for a given photo, resulting in very poor performance. Using the location prior re-ranks these candidate locations by their prior probability which, as shown in Figure 3, still results in very poor performance. This shows that a naive strategy of ranking locations by their popularity is ineffective.

5.2 Hierarchical Smoothing

Hierarchical smoothing gives a negligible improvement in performance, which confirms the results presented previously (Serdyukov et al., 2009). We look at the 3-level and 4-level term-based Dirichlet hierarchical smoothing results for 100km, 10km and 1km cells in Table 3. The results show a very slight improvement in performance for the 100km (0.7%) and 10km cells (1.1%). For 1km cells, there is a very slight decrease in performance in terms of accuracy, although MRR, Ac@k and H-hit rate performance all improve slightly.

Table 4 shows the hierarchical Dirichlet smoothing results for 100km cells, comparing term-based with cell-based smoothing, and directional with non-directional smoothing approaches. The directional approaches help for both the cell-based and the term-based approaches, although the difference is very small.

Table 3 Dirichlet Hierarchical Smoothing results over a large dataset of 10 million Flickr photos. In general, taking into consideration the term statistics of surrounding locations helps, although the difference is small when the dataset is large, and the locations are already well-represented. Gl: Global smoothing, 3-h: 3-level hierarchical smoothing, 4-h: 4-level hierarchical smoothing.

Method	Ac	MRR	Ac1	Ac2	Ac3	PAc	3hit	5hit
100km								
Gl.$_{\mu:9m}$	0.498	0.564	0.567	0.589	0.605	-	0.606	0.638
3-h$_{\mu:15,50(m)}$	0.501	0.567	0.573	0.595	0.612	-	0.611	0.641
4-h$_{\mu:20,100,150(m)}$	**0.502**	**0.568**	**0.574**	**0.596**	**0.613**	-	**0.612**	**0.642**
10km								
Gl.$_{\mu:250k}$	0.375	0.458	0.508	0.537	0.553	0.539	0.515	0.558
3-h$_{\mu:250k,50m}$	0.375	0.459	0.509	0.538	0.554	0.540	0.516	0.559
4-h$_{\mu:750k,100m,5m}$	**0.380**	**0.466**	**0.518**	**0.548**	**0.564**	**0.549**	**0.527**	**0.571**
1km								
Gl.$_{\mu:14k}$	**0.162**	0.226	0.296	0.343	0.376	0.355	0.256	0.297
3-h$_{\mu:20k,2m}$	0.162	0.226	0.292	0.343	0.374	0.355	0.257	0.298
4-h$_{\mu:20k,2m,2m}$	0.1616	**0.227**	**0.293**	**0.344**	**0.376**	**0.356**	**0.257**	**0.300**

Table 4 Directional term weight propagation, with 3-level Hierarchical Smoothing Results for 100km 10M Dataset. For the cell hierarchy approaches, the μ parameter represents the global smoothing parameter for the initial dirichlet model, while the λ parameter represent the amount of smoothing with neighbouring cells.

Method	Ac	MRR	Ac1	Ac2	Ac3	3hit	5hit
Dirichlet - no Location Prior							
Gl.$_{\mu:9m}$	0.498	0.564	0.567	0.589	0.605	0.606	0.638
Term Hierarchy							
Non-directional$_{\mu:15m,50m}$	0.500	0.566	0.574	0.596	0.613	0.609	0.638
Directional$_{\mu:15m,50m}$	0.501	**0.567**	0.573	0.595	0.612	**0.611**	**0.641**
Cell Hierarchy							
Non-directional$_{\mu:9m,\alpha:0.6}$	0.499	0.565	0.568	0.590	0.607	0.607	0.640
Directional$_{\mu:9m,\alpha:0.7}$	**0.502**	0.563	**0.575**	**0.599**	**0.618**	0.605	0.633

It is instructive to look at the hierarchical smoothing results on a smaller dataset of 550,000 images (shown in Table 5).[9] The size of the improvement given by hierarchical smoothing is much larger for models built from less data. Table 5 shows the results for 3-level and 4-level term-based hierarchical smoothing on 100km cells on the smaller dataset, showing a larger improvement from the hierarchical smoothing approaches. The best hierarchical smoothing method improves accuracy for 100km cells from 0.4613 to 0.4747, a relative improvement of almost 3%. This demonstrates that it is possible to gain an added improvement by going beyond a 3-level hierarchy. We would expect to see additional gains in performance by adding extra layers to the hierarchy, but we do not explore this here as the effect on a large-scale data set is likely to be minimal.

The fact that these improvements can not be replicated on the larger dataset suggests that the hierarchical smoothing is merely compensating for sparse data. When we build a model based on a richer dataset, the benefits of this approach all but disappear. Where the location is sparsely represented, the hierarchical models using hierarchies of an arbitrary depth give useful performance gains, and we may consider using this method of smoothing for locations that have less coverage in the data. Although the

[9] We do not present the complete set of results for the small dataset for all hierarchical smoothing approaches here (for brevity), but the relative performance of the different approaches is the similar to those in Table 4.

Table 5 Hierarchical smoothing results for a small dataset of 550,000 photos for 100km cells. The advantage of smoothing from the surrounding locations has a clear benefit for these models, suggesting that hierarchical smoothing is beneficial for sparsely represented locations.

Method	Ac	MRR	Ac1	Ac2	Ac3	3hit	5hit
Gl.$_{\mu:1.25m}$	0.461	0.530	0.542	0.571	0.595	0.573	0.597
3-h$_{\mu:12,10(m)}$	0.468	0.533	0.560	0.585	0.612	0.578	0.610
4-h$_{\mu:10,10,10(m)}$	**0.475**	**0.544**	**0.575**	**0.606**	**0.632**	**0.589**	**0.625**

Table 6 User Frequency Results - Dirichlet Smoothing

Method	Ac	MRR	Acc1	Ac2	Ac3	PAc	3hit	5hit
100km								
Term Freq.$_{\mu:9m}$	0.498	0.564	0.567	0.586	0.605	-	0.606	0.638
User Freq.$_{\mu:500k}$	**0.587**	**0.652**	**0.670**	**0.698**	**0.718**	-	**0.695**	**0.725**
10km								
Term Freq.$_{\mu:250k}$	0.375	0.458	0.508	0.537	0.553	0.539	0.515	0.558
User Freq.$_{\mu:6k}$	**0.394**	**0.459**	**0.516**	**0.546**	**0.563**	**0.566**	**0.538**	**0.581**
1km								
Term Freq.$_{\mu:14k}$	0.162	0.226	0.293	0.343	0.375	0.355	0.256	0.297
User Freq.$_{\mu:2k}$	**0.172**	**0.239**	**0.311**	**0.364**	**0.397**	**0.377**	**0.272**	**0.314**

performance gains diminish (and almost disappear) when we migrate to a large-scale dataset, these results show that our proposed modifications to the hierarchical smoothing approaches do improve performance over the previous models, in particular we show the benefits of using a hierarchy deeper than three levels, which smooths based on cells further away than immediate neighbors, unlike results presented previously.

5.3 User Frequency for Term Probability Estimates

The results reported so far have estimated the parameters of the language models based on the frequency of terms in a location. An alternative, discussed in Section 3.3, is to estimate the language model parameters based on the user frequency of a term in the location. That is, the term statistics are given by the number of users who have used that term, as opposed to the total number of occurrences of the term in the location. This means that terms will only have a high probability for a location if many users have tagged images in that location with that term.

The user frequency global smoothing results on the 10M dataset are shown in Table 6. User frequency gives a large improvement over all evaluation measures, with an 18% (0.498 to 0.587) improvement in accuracy for 100km cells, a 5% improvement for 10km cells, and a 6% improvement for 1km cells.

It is somewhat counterintuitive that the biggest improvement is gained for the largest cell size, as we might expect smaller cells that contain fewer photos to be more prone to bias by individual users. Smaller cells, however, are less likely to contain many photos by the same user. For example, if a user takes a large number of photos while on a trip to a city and its surrounding areas, it is likely that most of these photos will fall within the same 100km cell. If the cell does not contain photos from many other users, the textual metadata will be biased by that one user. It is less likely that all or most of the photos from single trip will lie within the same 1km cell.

A single prolific user with a small vocabulary has much more influence over the term frequency distribution for a cell. This is a particular problem in Flickr when users

assign the same tag set, or nearly the same tag set, to large numbers of photos. This problem of large numbers of photos being assigned identical tag sets can be addressed by filtering for duplicate tag sets. A larger problem is near-duplicate tag sets, where one or two terms distinguish photos from each other, but the other 5 to 7 tags are identical. These tag sets are harder to filter as it is not a matter of simply identifying duplicate sets. Estimating the term weights by user frequency solves this problem entirely, as only one instance of the same tag, for a given user, is counted. The 100km cells are more prone to bias by near-duplicate tags by individual users, and so these larger cells benefit more from term probability estimates based on user frequency rather that raw term frequency.

We previously discussed, with reference to the term frequency results, the fact that the 10km models performed better at 100km prediction that the 100km models. For user frequency models, the same can be said of 1km models with respect to their parent 10km cells. Although the improvement in the predictive power of the 1 km cell over the 10 km cell is quite small (0.397 compared to 0.394), the accuracy at 3 for the 1km user frequency models, which represents a 7km cell, also outperforms the accuracy of the 10km model. This seems to indicate that the tags users provide are often specific to a neighborhood within a city.

The user frequency results represent the best performing language model for locations, predicting the correct location almost 59% of the time for 100km cells and including the correct location in the top 5 predictions 72% of the time. Similarly, for 10km cells, we predict the correct location 39% of the time and suggest the correct location in the top five 58% of the time, while for 1km cells we correctly place 17% of photos and suggest the correct location in the top 5 in 31% of cases. If we expand to a radius of 3km around a 1km cell, we have an accuracy of almost 40%. Expanding by a radius of 30km around a 10km cell gives an accuracy of 56%. These results represent an improvement of 85% for 100km cells, 93% for 10km cells and 132% for 1km cells over previously published results, which represents a significant improvement over the state of the art.

5.4 The Bias towards Popular Locations

In Section 5.1, we discussed the effect of the location prior on performance, and the manner in which Dirichlet smoothing biases the results toward popular locations. To explore this further, we partitioned the test set into 20 bins, each containing the same number of photos from the test collection. They were partitioned based on the sum of the user frequencies for each term in the vocabulary for each cell (the quantity $|L|_{user}$ from Equation 5), so the first bin contained the 5% of the test photos belonging to the most popular location, on so on. We bin the test photos based on their location size in this way to evaluate locations which are popular separately from those which are less popular. Figure 4 shows the average accuracy for each of these partitions. The x-axis represents partitions of the test set from the most popular to the least popular, where each partition represents 5% of the photos from the test collection.

As expected, the model is much more accurate when locating photos in more popular locations. More than 77% of photos are correctly placed in the most popular locations. As with any language-modeling approach, locations that are sparsely represented in terms of vocabulary are likely to show poor performance relative to locations that are richly represented, although the performance of photos in the less popular

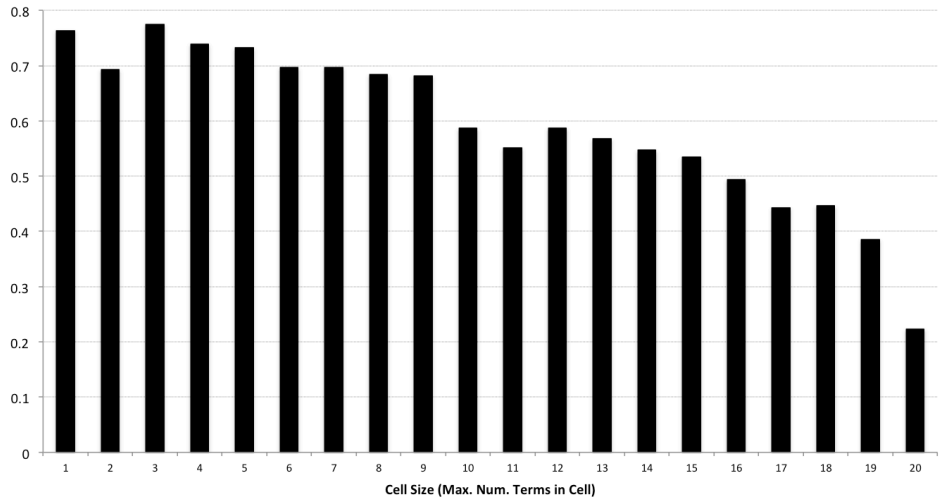

Fig. 4 Accuracy results for the 100km User Frequency model, partitioned into 20 bins, each containing the same number of test photos. The partitions are created based on the size of the photos' location, in terms of the sum of the user frequencies of the terms in the vocabulary of the cell (the quantity $|L|_{user}$ from Equation 5). The labels on the x-axis represent the rank of the bin, with bin 1 containing test photos from the largest cells.

locations is still respectable. Note that in the second smallest partition, at rank 19 (representing cells of size 1042-2313), 38.5% of photos are located in the correct cell. This partition corresponds to the average cell size for the 100km models (1,645, see Table 1). For the 100km models, 90% of the test photos are in locations that are more popular than this average cell size.

6 Evaluation on the CoPhIR Dataset

The results reported in this study are based on experiments conducted on a very large dataset of over 10 million images (after bulk upload filtering). Since we cannot make this dataset publicly available, it is not possible for others to replicate these experiments. To make our core findings reproducible, we report here the results of a subset of our experiments on the CoPhIR dataset (Bolettieri et al., 2009), which is the largest publicly available corpus of Flickr images. The entire corpus consists of almost 106 million images, 8,655,289 of which are geotagged. We removed one of the 106 archive files from the corpus due to known character encoding problems[10]. This left 8,578,936 geotagged images, which was reduced to 2,800,069 after bulk upload filtering and the removal of photos with empty tagsets.

To make the experimental setup reproducible, we performed the bulk upload filtering and corpus partitioning in a repeatable way. For bulk upload filtering, given a set of photos with identical tags, we choose the one with the minimum value for *photo id*. In order to facilitate a repeatable partition between training, tuning and test sets,

[10] Personal communication with the creators of the CoPhIR dataset. The removed archive was sapir_id_1_xml_r.tgz

Table 7 Baseline Results on the CoPhIR Dataset.

Method	Ac	MRR	Ac1	Ac2	Ac3	PAc	3hit	5hit
100km								
$Dir_{\mu:4m}$	**0.506**	0.574	**0.577**	**0.598**	**0.616**	-	0.619	0.651
$Dir/Pr_{\mu:50k}$	0.476	0.550	0.552	0.576	0.597	-	0.594	0.637
$JM_{\lambda:0.96}$	0.398	0.481	0.481	0.509	0.530	-	0.529	0.576
$JM/Pr_{\lambda:0.12}$	0.498	**0.575**	0.571	0.592	0.610	-	**0.621**	**0.663**
10km								
$Dir_{\mu:200k}$	0.353	0.435	0.493	0.518	0.534	0.518	0.492	0.534
$Dir/Pr_{\mu:9k}$	0.336	0.415	0.476	0.501	0.515	0.499	0.467	0.508
$JM_{\lambda:0.98}$	0.270	0.353	0.409	0.444	0.466	0.449	0.402	0.450
$JM/Pr_{\lambda:0.08}$	**0.354**	**0.439**	**0.499**	**0.526**	**0.541**	**0.523**	**0.498**	**0.541**
1km								
$Dir_{\mu:8k}$	**0.133**	**0.190**	**0.254**	**0.309**	**0.343**	**0.330**	**0.215**	**0.252**
$Dir/Pr_{\mu:2k}$	0.131	0.186	0.250	0.305	0.338	0.322	0.211	0.247
$JM_{\lambda:0.98}$	0.101	0.151	0.202	0.253	0.287	0.277	0.172	0.204
$JM/Pr_{\lambda:0.2}$	0.128	0.187	0.248	0.303	0.338	0.321	0.213	**0.252**

Table 8 User Frequency Results on the CoPhIR Dataset (Dirichlet Smoothing, no location prior)

Method	Ac	MRR	Ac1	Ac2	Ac3	PAc	3hit	5hit
100km								
Term $Freq_{\mu:4m}$	0.506	0.574	0.577	0.598	0.616	-	0.619	0.651
User $Freq._{\mu:500k}$	**0.570**	**0.635**	**0.653**	**0.681**	**0.701**	-	**0.680**	**0.712**
10km								
Term $Freq._{\mu:200k}$	0.353	0.435	0.493	0.518	0.534	0.518	0.492	0.534
User $Freq._{\mu:50k}$	**0.369**	**0.452**	**0.519**	**0.544**	**0.560**	**0.544**	**0.512**	**0.554**
1km								
Term $Freq._{\mu:8k}$	0.133	0.190	0.254	0.309	0.343	0.330	0.215	0.252
User $Freq._{\mu:4k}$	**0.142**	**0.202**	**0.272**	**0.329**	**0.364**	**0.345**	**0.230**	**0.269**

we take the *user id* of the photo owner, and take the modulo of this by 100, which extracts the last two digits of the user id. We then assign each photo to a corpus partition based on this value (0-79: training, 80-89: tuning, 90-99: test). This process results in a pseudo-random, but repeatable, partition of the corpus, which also ensures that photos from the same user are in the same partition, and gives a split of approximately 80% (2,240,696) training, 10% (285,304) tuning and 10% (274,069) test photos.

We process the data as before, using the raw tags representation, and we repeat the baseline experiments exploring alternative smoothing approaches and the effect of the location prior, in addition to the user frequency evaluation. We do not repeat the hierarchical smoothing evaluation, as we have already shown that the effect of hierarchical smoothing is dependent on the collection size.

The results, which are consistent with the main experimental results, are shown in Tables 7 and 8; these results repeat the experiments presented in Table 2 and Table 6 in the main evaluation.

7 Example Application: Geo Locating News Images

The vast majority of images uploaded to Flickr are taken by common users or amateur photographers. The textual metadata associated with the image often serves as a reminder of the context of the image for the photographer and his social circle (Nov et al.,

2010; van House, 2007). They are typically not intended to serve as a description of the image for the general public, and are not usually intended to serve as an illustration for other content such as news content. By contrast, the images associated with news content depict events or people in the news, for which location is a relevant piece of context, but not the dominating feature of the image. They are typically created by professional photographers, and serve to illustrate content produced by a journalist. An example of this type of image is embodied in the Getty Images[11], a stock photo agency that provides images for news, and other commercial interests.

Nonetheless, for the Getty collection, in order to show news images relevant to a particular location, we must translate the textual description of the location, assigned to the image by the journalist or editor, to a set of geographic coordinates. The geographic coordinates, in turn, can be used by an application to personalize the image search experience, or to provide additional image data for a location-based mobile application, or to illustrate news content. Using geographic coordinates, rather than textual representations of locations, is a form of normalization, which ensures that multiple names for the same place are conflated to the same physical location.

Using the models described above, with one-kilometer grid cells and user-frequency term weights computed over the Flickr data described in Section 4, we predict the geographic coordinates for the textual locations in a sample of Getty images. We compare the predictive ability of the language model approach to a publicly available state-of-the-art location identification system, built upon curated data from data providers such as Navteq.

While in the experimental setup and in the application described below we focus on localizing image data, in fact the location field associated with the Getty Images is quite generic. Other applications include localizing mentions of points of interest from social network streams such as status updates from Facebook[12], or Twitter[13], which could be used in a recommender system or a local search application. The information could also be used to determine the points of interest mentioned in a user's search query history for the purpose of personalization or behavioral targeting. In this case, having a very accurate granularity (one kilometer or less) would be necessary.

7.1 The Data

The Getty images are labeled with a location, a title, and a caption in the metadata. We select the location field from a sample of this data, and exclude all locations that are not points of interest. The yields 73 unique location labels, each corresponding to a point of interest, from approximately 6000 images. The locations used in this experiment are listed in Appendix A. Note that the location field may contain more than one location, in cases where this was deemed relevant by the journalist or editor creating the data. To establish the ground truth coordinates of the points of interest, we took the coordinates listed on the official website for the POI when it was available, or located it directly on a map. The ground truth coordinates are listed in Appendix A. Note that although there are multiple mentions of the same POI, they each have a different textual representation. For example, the Roland Garros tennis stadium in

[11] http://www.gettyimages.com/ visited January 2012
[12] www.facebook.com visited January 2012
[13] www.twitter.com visited January 2012

Paris is listed as "Japan France Paris Taipei Roland Garros" and "France Italy USA Paris Roland Garros" among other variants in our data. While the editor providing the metadata indicated multiple locations as relevant to the image, the event depicted (a tennis tournament) took place at Roland Garros, presumably involving participants from other countries. We did not exclude examples that contain multiple place mentions because this is exactly as the data appears in the Getty Images. Part of the challenge of the application is to determine which locations are salient among multiple place mentions.

7.2 The Evaluation

We evaluate the distance from the true location, rather than the accuracy with which we are able to predict the correct cell. This allows us to compare with Yahoo! Placemaker[14], which is the state-of-the-art system for identifying locations in text. We report the mean and median distance from the predicted location to the true location for each example. The distance metric is Vincenty Distance (Vincenty, 1975), which takes into account the curvature of the earth between two latitude/longitude coordinates.

Not all locations mentioned in the location field in the metadata are present in the curated data accessed by Placemaker. We refer to points of interest that are identifiable by Placemaker as the "PM POIs" and other points of interest as "Other POIs". Placemaker identified 23 of the 73 points of interest as POIs. The other 50 examples were identified by Placemaker at a coarser granularity, as other types of locations such as cities, states, provinces or countries.

We compared the geo coordinates for the location field as given by Placemaker, with the ground truth location. For the 23 POIs that are present in Placemaker, the median distance from our ground truth location was 233 meters (the mean was 488 meters). Because the data available in Placemaker represents curated data, which is gathered by a surveyor who visits the location and records the coordinates, we assume this is as accurate as can reasonably be expected from any application. Still, surveyors can be incorrect, or may disagree on what is the centroid of a POI that covers a large land area (such as a University Campus or Airport). This disparity between our ground truth and the curated Placemaker locations suggests that a human surveyor can identify a location within 488 meters of its true location on average (or 233 meters if we prefer the median, which is less likely to be influenced a few large distances). So, in this setting, we can be confident that the accuracy of the ground truth is finer grained than the size of the 1KM grid cells that we are using.

We find that, although Placemaker is highly accurate for the points of interest it finds, there are many POIs that it does not identify, including well-known places such as the MGM Grand Garden Arena in Las Vegas, Nevada. For locations not identified by Placemaker, the median distance from the true location is 1.75 km (an average of nearly 170 km), as shown in Table 9.

For the language model approach we used the best performing language model described above. That is, using the 10M Flickr collection described in Table 1, we estimated term weights with user frequency as described in Section 5.3 with Dirichlet smoothing ($\mu = 10000$), and one kilometer grid cells. The language model approach is significantly better than the state-of-the-art, with a median distance of 470 meters (a

[14] http://developer.yahoo.com/geo/placemaker visited January 2012

Table 9 Median and Average distance (in kilometers) of predicted geographic coordinates from the true location of points of interest from news images.

	All Data		**PM POIs**		**Other POIs**	
Num examples	74		23		51	
	Median	Mean	Median	Mean	Median	Mean
Placemaker	7.0	115.8	**0.233**	**0.488**	1.75	167.9
LM (User Freq)	0.469	14.8	0.522	6.4	0.397	18.6
Cascade Model	**0.322**	**11.0**	**0.233**	**0.488**	**0.330**	**15.7**

mean of roughly 15 km) over all examples, and a median of 330 meters (mean 18.6 km) over the examples that Placemaker did not identify as POIs, as shown in Table 9. Even for the locations that correspond to POIs that Placemaker could identify, the language model approach performed comparably if we consider the median distance (522 meters vs 233 meters).

Because Placemaker is extremely accurate with the geo coordinates it assigns to the POIs it identifies, it makes sense to use those when possible. To take advantage of this, we also evaluate a cascade approach that combines the two models. If Placemaker identifies a POI, we take the coordinates of the POI from Placemaker and do not consider the grid cells. When Placemaker identifies a location as a city or state or country, we use the bounding box for this location to constrain our search over grid cells. Thus, in the case of the MGM Grand Garden Arena, in Las Vegas, NV, although Placemaker did not identify this as a POI, it did identify the city of Las Vegas. Therefore we choose the top-ranked grid cell returned by the location model that falls within the bounding box of the city of Las Vegas. The results of this cascade approach are reported in Table 9 in the rows labeled "Cascade". The results are comparable to a human surveyor identifying the exact location of a point of interest, if we consider the median values, and they are significantly better than the current state-of-the-art on the average.

7.3 Discussion

In the application described above, we know the ground truth coordinates. In fact, the state of the art location prediction system (Yahoo! Placemaker) was shown to be accurate within 500 meters on the points of interest in its place repository. Based on this, predicting a one kilometer grid cell is a useful task, as it attempts to geo-locate a point of interest as accurately as the state of the art. In fact, the language model approach, with user frequency term estimates, proves to be much better than the state-of-the-art system overall, as it does not rely on a repository of places, and can make use of other information such as events, people, or other characteristics associated with places. Using the two in a cascade architecture provides the best combination of both sources of information, significantly improving over the state-of-the-art. These results also confirm that language models created using a Flickr dataset can be used to geo-tag media items coming from a completely different source, and they can do this to a level of accuracy better than a kilometer.

Table 10 Baseline Global Smoothing Results 550K Dataset - with Normalised Tags

Cell Size	Ac	MRR	Ac1	Ac2	Ac3	PAc	3hit	5hit
$100km_{\mu:900k}$	0.478	0.537	0.559	0.585	0.606	-	0.575	0.607
$10km_{\mu:50k}$	0.280	0.349	0.407	0.430	0.449	0.437	0.402	0.436
$1km_{\mu:2k}$	0.089	0.129	0.189	0.233	0.264	0.249	0.147	0.173

8 Comparison with the previous work

This paper is an extension of the work of Serdyukov et al. (2009), starting from a higher baseline. We initially suspected that the improvements were due to using a larger dataset to build the models. After seeking to verify this, we found that it did not account for all of the performance gains. In this Section we attempt to explain these performance differences. We also clarify a minor discrepancy in the conclusions drawn from experiments with hierarchical smoothing.

To investigate the role of the size of the data set with regard to model performance, we started with a smaller set of 1.5 million photos, randomly selected from Flickr. After filtering for bulk uploads, we were left with about 550K photos. As before, we partitioned the data into a training set (80%), a tuning set (10%), and a test set (10%). Although this dataset is based on the same initial dataset as reported by Serdyukov et al. (2009), they sampled this data to give a subset of 397K images, which was reduced to 140K photos after bulk upload filtering. We were unable to replicate this sampling, as it was random, so it was not possible to compare our models to theirs using an identical dataset. Instead, we report results on a dataset of comparable size.

Since Serdyukov et al. (2009) created their models using the tags as normalized by Flickr, to facilitate comparison we report results using normalized tags in this section. Table 10 shows the results of baseline approaches, using Dirichlet smoothing without a location prior, on the 550K dataset using the normalized tags, or 'bag of tags', representation. These results show a large improvement over those previously reported by Serdyukov et al. (2009), for identical models. Compared with their baseline approaches, our results show an improvement of 65% in accuracy in placing photos at the 100km level, 55% at the 10km level and 31% at the 1km level.

In Figure 5 we show the learning curve for 100km cells as the size of the training set is increased. The size of the training set reported by Serdyukov et al. (2009) is best approximated by the 30% (125,956 photos) point on the learning curve, where we achieve an accuracy of 0.424, which is 47% better than their baseline result of 0.288. Thus the amount of data used to build the models only partly explains the performance gains of our system.

One implementation detail not documented in the prior work of Serdyukov et al. is that in the retrieval system implementation of the model, the posting lists were truncated for efficiency reasons. The posting list is a list of documents containing a given term, and truncating it to size N means that only the first N documents containing this term are considered. Occurrences of the term in other documents are effectively ignored. The motivation for doing this is that the scores for terms that do not appear in many documents (i.e. high *idf* terms) will not be affected as their postings lists are already small. Only low *idf* terms will be affected, and they do not contribute as significantly to the document score. Considering a term in the scores for some documents but not others will have an effect on the ranking, and possibly reduce the accuracy of the results.

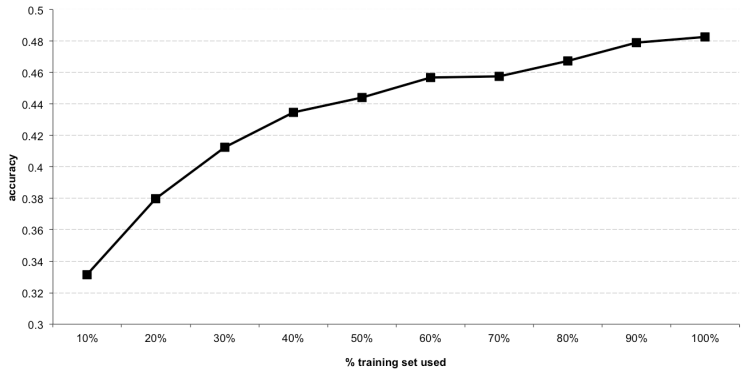

Fig. 5 Results for 100km cells, with the 550K data set, showing the effect of increasing the amount of data used to populate the language models.

Another factor that may explain the higher baseline in this work is the unusually high value of the optimal Dirichlet prior parameter. At μ=2,000, for example, we achieved an accuracy of 0.394 on the tuning partition from the 550K dataset, while the performance levels off at approximately 0.48 accuracy with a Dirichlet prior around μ=250,000. The results are sensitive to the parameter values in this setting because of the large variance in the size of the vocabulary representing the locations; Serdyukov et al. (2009) do not report their optimal parameter settings or the range of their parameter, and a narrower parameter search could account for the performance differences.

The implementation of the hierarchical models reported in Serdyukov et al. (2009), differ from those reported in this paper in the following ways. First, they only considered 3-level hierarchies (location, neighborhood, global) and did not allow for hierarchies of arbitrary depth. They specify the neighborhood depth with a parameter d, but this specifies the size of the *single* neighbor in the hierarchy, whereas we consider neighborhoods larger than size one (in a 4-level hierarchy). Also, the hierarchical smoothing method is defined in their work as a hybrid between Dirichlet and Jelinek Mercer smoothing (Equation 4 in their paper), which is different from the Hierarchical Dirichlet model described in this work in Equation 9. We believe these differences in the models explain the better performance of the hierarchical smoothing in this work, they also explain the improvement as we consider neighborhoods larger than size one (in a 4-level hierarchy). As noted in Section 5, however, we found that the improved results from hierarchical smoothing only held for a smaller dataset.

9 Conclusions and Future Work

Some locations are much better represented than others. This is not only true in the Flickr data, but in life as well. Some locations are very densely populated, and have a long human history. Some locations are very sparsely populated, and have almost no human history. Depending on the application, we might prefer a model biased toward populous (or popular) places over one that is biased toward more remote (or off-beat) places. This study explores a bias toward popular locations that is not always explicitly

addressed in work based on user-generated content and social media. We quantify the effect of that bias on the accuracy of the resulting models.

We have shown that Dirichlet smoothing relies more on the collection statistics for locations represented by a smaller number of terms, and gives more weight to the specific location statistics when they are sufficiently represented in the data, which effectively boosts the scores of more popular places. Thus, there is no need to incorporate a location-based prior in the model when Dirichlet smoothing is employed. Should we wish to control the bias toward popular locations, we might instead use Jelinek Mercer smoothing and bias our estimate of the location prior to explicitly represent our preference for certain types of locations. We leave this investigation to future work. We do conclude, however, that when using Jelinek Mercer smoothing, estimating the location prior from the popularity of the location in the data is equivalent to using Dirichlet smoothing with no explicit location prior.

On the topic of popular vs. sparsely represented locations, we showed that hierarchical smoothing from the neighborhoods surrounding a location improves the estimates for sparsely represented locations. That is, people describe a place in a manner similar to its surrounding places. This benefit is lost if the place in question is already sufficiently represented in the data. This suggests that more sophisticated modeling is unnecessary when working with very large data sets. We also quantified the extent of the bias towards popular locations, showing that photos in the most popular locations can be located with twice the accuracy of photos in average locations. Over 90% of photos are associated with locations that are more popular than the average location.

One surprising conclusion is that city-level and neighbourhood-level descriptions in the data are particularly informative. The results for the 10km cell width were better at predicting the location of their 100km parent cells, than the parent cells themselves. Similarly, the 1km cells were better at predicting the location of their 10km parents. This is likely because within larger areas, there will be "hot spots" that contain a lot of information. Imagine, for example, the 100km cell in Wyoming that contains Devil's Tower. Within that 100km, the cells that do not contain Devil's Tower will be sparsely represented in the data. Thus, the only cell that is really necessary to identify the surrounding 100km, is the 10km cell that contains Devil's Tower. Similarly, within 10km cells, there will be hotspots within certain 1km cells. We do note, however, that the finer quantization used by the 10km and 1km models could also be partly responsible for this performance difference, as missclassified locations near the boundary of a cell will have less impact on the prediction of parent cells.

We evaluate on the Flickr data, as this is a clean experimental setup, assuming that the geographic coordinates associated with the images are accurate. There are multiple ways to associate geo coordinates with images in Flickr, some of which are more accurate than others. Unfortunately, the data about how a specific image was geo-tagged is not available, so we cannot know which images were geo-tagged by hand, and which were device geo-tagged. Flickr does provide a number from one to 16 which describes the relative accuracy of the geotagging (one being the least accurate, and 16 being the most accurate) but it is derived from the zoom level of the map on which the image was placed, or from a setting the user specifies when they upload their images. Unfortunately, the highest accuracy, 16, is also the default accuracy, so we cannot know whether a user was very conscientious and zoomed into the street level before placing their image, or whether they ignored the setting entirely, and left the accuracy at its default value. Furthermore, it is not clear what physical distance corresponds to a given accuracy level. This means that, for the main Flickr evaluation, we cannot

be sure that the ground truth is more accurate than the granularity to which we are attempting to locate the images (particularly for 1km cells). This is an artifact of this particular data set. Other data, such as data from Twitter, which is geo-tagged with a GPS-enabled smart phone would not have his issue. We find, (in Section 7), that for a smaller set of images, for which we know the ground truth is highly accurate, these same Flickr models can locate those images to an accuracy of less then 1km. Thus for many applications, the 1 km cells are appropriate.

The final conclusion from this work is that estimating the term weights from the user frequency, rather than the term frequency, improves results dramatically. This follows from the intuition that if multiple people agree on the characterization of a place, this is more reliable than a single person's label of a place. Language model approaches benefit from the non-location terms used to describe a place, so if multiple people label a place with the same contextual terms (terms such as "double-decker bus" or "Gaudi"), then we have more confidence that those terms are predictive of that location. This also alleviates the effect of tags that are meaningful to a specific user in a given location, but are not representative of the location itself (for example tags such as "John, Mary, Christmas, 2007").

Although we evaluated these systems with Flickr data, the motivation behind this work is to create general models of location from user generated content, which can then be used to understand the geographic focus of any user generated text. We have used Flickr as a testbed for developing and understanding these models, and in our future work we will apply the lessons learned from this by incorporating multiple sources of geo-tagged data.

In this work we partition the globe into grid cells because of the relative efficiency of the scheme. This partition avoids the computational complexity introduced by partitionings based on triangles or hexagons, or overlapping cells. Because of this, it is practical for a variety of web applications where efficiency in terms of time and space are important. With a more complex partitioning, we might expect a trade-off between efficiency and performance. If the performance gains merit a more complex partitioning then the partitioning would be justified. As the performance of the hierarchical smoothing models show, however, the benefit to be gained from more sophisticated approaches is lost when the dataset is large enough. The investigation of the costs and benefits of a more complex representation of spatial relationships should be a direct follow-on to the current work.

In addition to further sources of data and a more complex spatial representation, we expect to continue this work by examining applications of location modeling. By modeling hyperlocal places we can improve the results for users of applications, in particular mobile applications, which depend on understanding how users describe the places in their immediate vicinity. This would allow us to provide geographically relevant images and videos by understanding when the user's query pertains to a specific location, and matching that to the locations implicit in the metadata associated with multimedia artifacts. Another application of this type of modeling is to determine the geographic scope of documents, which would allow the search engine to improve results for queries that have a specific geographic intent. A final piece of the puzzle is to use the models described in this paper to predict the location of the user based on their query histories, so that search results can be tailored to their geographic context without the user having to reveal their exact geographic coordinates.

Leveraging public geo-tagged social media data allows us to understand how users describe the places that surround them, how they move through them, and what they

think, see, and feel while they are there. We, as scientists, are granted an opportunity to understand human-centric geographies, thanks to the willingness of users to be transparent about their locations and share their interactions. We, as people, users of social media, are cartographers for a new kind of map of the world, whose boundaries represent our understanding of a place, expressed with a language that reflects the richness of human experience.

Appendix A: Points of Interest from News Images

Point of Interest	Centroid Coordinates
France Cannes Festivals et des Congres	43.550861, 7.01725
Australia - Australasia Melbourne - Australia Stadium	-37.816389, 144.9475
New Zealand Wellington Christchurch Cathedral	-43.531, 172.637
India USA Texas Fort Worth Colonial Country Club	32.718997, -97.370828
Geographical Locations USA Florida - USA Miami American Airlines Arena	25.781389, -80.188056
Belgium Spain Casares Finca Cortesin Golf Club	36.39923, -5.226145
Dublin - Republic of Ireland Temple Bar - Dublin Ireland	53.345556, -6.262778
Japan France Paris Taipei Roland Garros	48.847222, 2.246389
USA Texas Fort Worth Colonial Country Club	32.718997, -97.370828
Czech Republic France Paris Roland Garros	48.847222, 2.246389
Belgium Spain Casares Ito Finca Cortesin Golf Club	36.39923, -5.226145
Geographical Locations Australia - Australasia Melbourne - Australia Docklands Stadium	-37.816389, 144.9475
USA Brazil Indianapolis Indianapolis Motor Speedway	39.798333, -86.232778
Ukraine France Serbia Paris Roland Garros	48.847222, 2.246389
USA Iowa Iowa Speedway Newton - Iowa	41.677778, -93.014444
Australia - Australasia Sydney Bondi Beach	-33.89102, 151.277726
England Germany Spain Casares Finca Cortesin Golf Club	36.39923, -5.226145
USA Texas Frisco Pizza Hut Park	33.154444, -96.835278
France Italy USA Paris Roland Garros	48.847222, 2.246389
USA New York City Madison Square Park	40.742169, -73.987985
Australia - Australasia Melbourne - Australia Docklands Stadium	-37.816389, 144.9475
Russia France Paris Australia - Australasia Roland Garros	48.847222, 2.246389
France nes Palais des Festivals et des Congres	43.550861, 7.01725
Australia - Australasia Benalla Winton Motor Raceway	-36.518333, 146.0875
USA Nevada Las Vegas MGM Grand Garden Arena	36.104808, -115.168614
France Finland Spain Paris Roland Garros	48.847222, 2.246389
Germany Potsdam Sanssouci Park	52.401974, 13.033583
France Spain USA Paris Roland Garros	48.847222, 2.246389
USA Florida - USA Miami American Airlines Arena	25.781389, -80.188056
Russia Slovakia France Paris Roland Garros	48.847222, 2.246389
New Zealand Auckland Waitakere Trusts Stadium	-36.86, 174.6362
Australia - Australasia Perth Subiaco Oval	-31.944444, 115.83
USA Indianapolis Indianapolis - Motor Speedway	39.798333, -86.232778
Romania Russia France Paris Roland Garros	48.847222, 2.246389
England Spain Casares Finca Cortesin Golf Club	36.39923, -5.226145
Belgium England Spain Casares Finca Cortesin Golf Club	36.39923, -5.226145
USA Nevada Las Vegas Garden Arena	36.104808, -115.168614
Russia France Paris Roland Garros	48.847222, 2.246389
Slovakia France Paris Roland Garros	48.847222, 2.246389
USA New Jersey Hamilton Farm Golf Club Gladstone - New Jersey	40.719729, -74.681266
Geographical Locations USA California San Jose	

- California HP Pavilion	37.332778, -121.901111
Dublin - Republic of Ireland Bank Of Ireland Ireland	53.344806, -6.260131
Germany Spain Casares Finca Cortesin Golf Club	36.39923, -5.226145
France Paris Roland Garros	48.847222, 2.246389
France UK Paris Roland Garros	48.847222, 2.246389
USA New Zealand Indianapolis Indianapolis	
- Motor Speedway	39.798333, -86.232778
USA New York City Landmark Sunshine Theater	40.723258,-73.989873
Germany Potsdam Sanssouci Park Prussia	52.401974, 13.033583
USA Washington DC RFK Stadium	38.889722, -76.971667
France Switzerland Paris Roland Garros	48.847222, 2.246389
USA Commerce City Dick's Sporting Goods Park	39.805556, -104.891944
Spain USA Canada New Zealand Indianapolis	
- Indianapolis Motor Speedway	39.798333, -86.232778
USA Canada Indianapolis Indianapolis Motor Speedway	39.798333, -86.232778
Australia - Australasia Sydney Sydney Cricket Ground	-33.891667, 151.224722
Asia USA Nevada Las Vegas MGM Grand Garden Arena	36.104808, -115.168614
USA California San Jose - California HP Pavilion	37.332778, -121.901111
USA Portland - Oregon PGE Park	45.521389, -122.691667
Bulgaria France Paris Australia - Australasia Roland Garros	48.847222, 2.246389
France Cannes Palais des Festivals et des Congres	43.550861, 7.01725
France Germany Paris Roland Garros	48.847222, 2.246389
New Zealand Auckland Waitakere Trusts Stadium Ellerslie	-36.86, 174.6362
Spain Casares Finca Cortesin Golf Club	36.39923, -5.226145
Europe France Cannes Anatolia Palais des	
- Festivals et des Congres	43.550861, 7.01725
USA California City Of Los Angeles Cedars	
Sinai Hospital Century Plaza	34.075198, -118.380676
USA Washington DC White House	38.89767, -77.03655
USA Texas Australia - Australasia Fort Worth	
- Colonial Country Club	32.718997, -97.370828
Czech Republic France Paris Australia	
- Australasia Roland Garros	48.847222, 2.246389
USA North Carolina Charlotte Walk Of Fame	35.221599, -80.843277
France Spain Paris Roland Garros	48.847222, 2.246389
Norway USA New Jersey Hamilton Farm	
- Golf Club Gladstone - New Jersey	40.719729, -74.681266
Canada Quebec Montreal Le Centre Bell	45.496111, -73.569444
Israel France Spain Paris Roland Garros	48.847222, 2.246389
Israel USA Washington DC White House	38.9051, -77.023

Acknowledgements This work was supported by the European Commission under contract FP7-248984 GLOCAL. The authors would also like to acknowledge Adrian Popescu for his helpful discussions about this work.

References

Ahern, S., Naaman, M., Nair, R., and Yang, J. (2007). World Explorer: Visualizing aggregate data from unstructured text in geo-referenced collections. In *JCDL '07*.

Amitay, E., Har'El, N., Sivan, R., and Soffer, A. (2004). Web-a-where: geotagging web content. In *SIGIR '04*, pages 273–280, New York, NY, USA. ACM.

Backstrom, L., Kleinberg, J., Kumar, R., and Novak, J. (2008). Spatial variation in search engine queries. In *WWW '08*.

Bolettieri, P., Esuli, A., Falchi, F., Lucchese, C., Perego, R., Piccioli, T., and Rabitti, F. (2009). CoPhIR: a test collection for content-based image retrieval. *CoRR*, abs/0905.4627v2.

Chen, L., Hu, B.-G., Zhang, L., Li, M., and Zhang, H. (2003). Face Annotation for Family Photo Album Management. *International Journal of Image and Graphics*, 3(1):81–94.

Cheng, Z., Caverlee, J., and Lee, K. (2010). You are where you tweet: a content-based approach to geo-locating twitter users. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 759–768, New York, NY, USA. ACM.

Clements, M., Serdyukov, P., de Vries, A. P., and Reinders, M. J. T. (2010). Finding wormholes with flickr geotags. In *ECIR'10*, pages 658–661.

Crandall, D., Backstrom, L., Huttenlocher, D., and Kleinberg, J. (2009). Mapping the world's photos. In *Proceedings of the 18th International Conference on World Wide Web*, pages 761–770. ACM.

Ding, J., Gravano, L., and Shivakumar, N. (2000). Computing geographical scopes of web resources. In *VLDB '00*, pages 545–556, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

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, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hays, J. and Efros, A. A. (2008). im2gps: estimating geographic information from a single image. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Hiemstra, D. (1998). A linguistically motivated probabilistic model of information retrieval. In *Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries*, ECDL '98, pages 569–584, London, UK. Springer-Verlag.

Hollenstein, L. and Purves, R. (2010). Exploring place through user-generated content: using Flickr to describe city cores. *Journal of Spatial Information Science*, (1).

Jones, C. B., Purves, R. S., Clough, P. D., and Joho, H. (2008a). Modelling vague places with knowldge from the web. *International Journal of Geographical Information Science*, 22(10):1045–1065.

Jones, R., Zhang, W., Rey, B., Jhala, P., and Stipp, E. (2008b). Geographic intention and modification in web search. *International Journal of Geographical Information Science*, 22(3):229–246.

Kantor, P. B. and Voorhees, E. M. (1996). Report on the trec-5 confusion track. In *TREC-5*, pages 65–74, Gaithersburg, Maryland, USA.

Kennedy, L., Naaman, M., Ahern, S., Nair, R., and Rattenbury, T. (2007). How flickr helps us make sense of the world: context and content in community-contributed media collections. In *Proceedings of the 15th international conference on Multimedia*, MULTIMEDIA '07, pages 631–640, New York, NY, USA. ACM.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60:91–110.

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts.

Mc Donald, K. and Smeaton, A. F. (2005). A Comparison of Score, Rank and Probability-based Fusion Methods for Video Shot Retrieval. In *CIVR 2005 - International Conference on Image and Video Retrieval*, pages 61–70, Singapore.

Mei, Q., Liu, C., Su, H., and Zhai, C. (2006). A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *WWW '06*.

Moxley, E., Kleban, J., and Manjunath, B. S. (2008). Spirittagger: a geo-aware tag suggestion tool mined from flickr. In *Proceeding of the 1st ACM international conference on Multimedia information retrieval*, MIR '08, pages 24–30, New York, NY, USA. ACM.

Murdock, V. (2006). *Aspects of Sentence Retrieval*. PhD thesis, University of Massachusetts.

Naaman, M., Paepcke, A., and Garcia-Molina, H. (2003). From where to what: metadata sharing for digital photographs with geographic coordinates. *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, pages 196–217.

Nov, O., Naaman, M., and Ye, C. (2010). Analysis of participation in an online photo-sharing community: A multidimensional perspective. *Journal of the American Society for Information Science and Technology*, 61(3).

O'Hare, N. and Smeaton, A. F. (2009). Context-aware person identification in personal photo collections. *IEEE Transactions on Multimedia, Special Issue on Integration of Context and Content for Multimedia Management*.

Ponte, J. M. and Croft, W. B. (1998). A Language Modeling Approach to Information Retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Re-*

search and Development in Information Retrieval (SIGIR '98), pages 275–281, Melbourne, Australia.

Rattenbury, T., Good, N., and Naaman, M. (2007). Towards automatic extraction of event and place semantics from flickr tags. In *SIGIR '07*.

Serdyukov, P., Murdock, V., and van Zwol, R. (2009). Placing Flickr Photos on a Map. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 484–491. ACM.

Sigurbjornsson, B. and van Zwol, R. (2008). Flickr tag recommendation based on collective knowledge. In *proceedings of the 17th International World Wide Web Conference (WWW 2008)*, Beijing, China.

Smucker, M. D. and Allan, J. (2005). An investigation of dirichlet prior smoothing's performance advantage. Technical report, The Center for Intelligent Information Retrieval, The University of Massachusetts.

Toyama, K., Logan, R., and Roseway, A. (2003). Geographic Location Tags on Digital Images. In *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, pages 156–166, New York, NY.

Vadrevu, S., Zhang, Y., Tseng, B., Sun, G., and Li, X. (2008). Identifying regional sensitive queries in web search. In *Proceedings of WWW '08*.

van House, N. (2007). Flickr and public image-sharing: Distance closeness and photo exhibition. In *Extended Abstracts CHI*.

Vincenty, T. (1975). Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. *Survey Review*, XXIII(176).

Wang, C., Wang, J., Xie, X., and Ma, W.-Y. (2007). Mining geographic knowledge using location aware topic model. In *GIR '07*.

Westerveld, T., de Vries, A. P., Westerveld, A. T., de Vries, A. P., and van Ballegooij, A. R. (2003). CWI at the TREC-2002 Video Track. In *The Eleventh Text REtrieval Conference (TREC-2002)*, pages 207–216, Gaithersburg, MD.

Yi, X., Raghavan, H., and Leggetter, C. (2009). Discovering users' specific geo intention in web search. In *WWW '09: Proceedings of the 18th International Conference on World Wide Web*, pages 481–490, New York, NY, USA. ACM.

Zhuang, Z., Brunk, C., and Giles, C. L. (2008). Modeling and visualizing geosensitive queries based on user clicks. In *LocWeb '08*.

Zong, W., Wu, D., Sun, A., Lim, E.-P., and Goh, D. H.-L. (2005). On assigning place names to geography related web pages. In *JCDL '05*, pages 354–362, New York, NY, USA. ACM.