Collaborative Language Models for Localized Query Prediction

YI FANG, Santa Clara University
ZIAD AL BAWAB, Microsoft
JEAN-FRANCOIS CRESPO, Google

Localized query prediction (LQP) is the task of estimating web query trends for a specific location. This problem subsumes many interesting personalized web applications such as personalization for buzz query detection, for query expansion, and for query recommendation. These personalized applications can greatly enhance user interaction with web search engines by providing more customized information discovered from user input (i.e., queries), but the LQP task has rarely been investigated in the literature. Although exist abundant work on estimating *global* web search trends does exist, it often encounters the big challenge of data sparsity when personalization comes into play.

In this article, we tackle the LQP task by proposing a series of collaborative language models (CLMs). CLMs alleviate the data sparsity issue by collaboratively collecting queries and trend information from the other locations. The traditional statistical language models assume a fixed background language model, which loses the taste of personalization. In contrast, CLMs are personalized language models with flexible background language models customized to various locations. The most sophisticated CLM enables the collaboration to adapt to specific query topics, which further advances the personalization level. An extensive set of experiments have been conducted on a large-scale web query log to demonstrate the effectiveness of the proposed models.

Categories and Subject Descriptors: H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Query log mining, trending topic recommendation, language models, generative models, discriminative models

ACM Reference Format:
Yi Fang, Ziad Al Bawab, and Jean-Francois Crespo. 2014. Collaborative language models for localized query prediction. ACM Trans. Interact. Intell. Syst. 4, 2, Article 9 (June 2014), 21 pages.
DOI: http://dx.doi.org/10.1145/2622617

1. INTRODUCTION

Web users interact with search engines mainly through queries. As people increasingly turn to the search engines for news and information, it is tempting to view search activity at any moment in time as a snapshot of the collective consciousness. Consequently, large search engines begin to offer the services about what people are currently

Author's addresses: Y. Fang (Contact Author), Department of Computer Engineering, Santa Clara University, 500 El Camino Real, Santa Clara, California; email: yfang@scu.edu; Z. A. Bawab, Microsoft, 1020 Enterprise Way, Sunnyvale, California; J.-F. Crespo, Google, 1600 Amphitheatre Parkway, Mountain View, California.

DOI: http://dx.doi.org/10.1145/2622617

Fig. 1. Trending Now Module on Yahoo! front page on 03/21/2013 at 5 PM PDT.

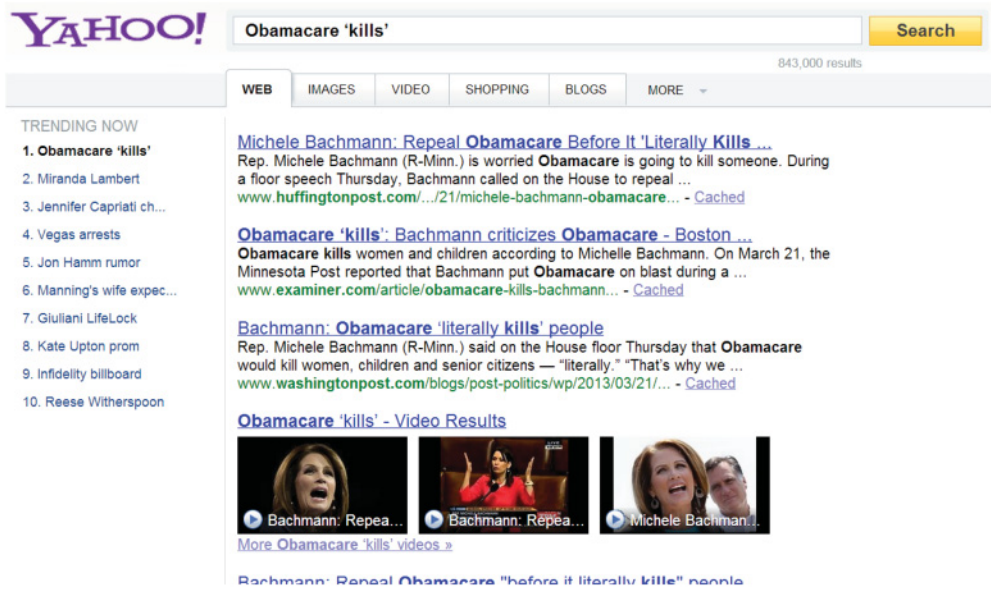

Fig. 2. The page shown after a user clicked on the trending query topic "ObamaCare kills."

searching, such as Yahoo!'s Trending Now,[1] Google Trends,[2] and Bing's Popular Now.[3] These services reflect the instantaneous interests, concerns, and intentions of the global population. They can be utilized to strengthen user interaction with search engines. For example, the Trending Now Module on Yahoo!'s front page, shown in Figure 1, is a trending topic recommendation system deployed on the web. The trending topics displayed were extracted from search query log, mainly globally trending and of interest to a wide user base. Every click on the trending query topics takes the user to a search result page where the topic is entered automatically as a query, as shown in Figure 2.

[1]http://www.yahoo.com/.
[2]http://www.google.com/trends.
[3]http://www.bing.com/.

Through the search result page, users can then find detailed information about the trending topic.

Moreover, search queries provide valuable insights about the users themselves and thus can be further exploited to enhance user interaction with search engines. In particular, users can get more customized web experience through web personalization, and consequently are more inclined to engage and interact with personalized web services. Web personalization has recently received tremendous attention from both industry and academia. Among many observable user attributes, user location is particularly simple for search engines to obtain and allows personalization even for a first-time web search user. This attribute becomes even more prominent with the rise of location-enabled mobile devices.

This article addresses the task of estimating web query trends for a specific location, which we call *localized query prediction* (*LQP*). LQP has many potential usages and applications. It can be utilized to detect buzz query topics that are locally more relevant to the users and thus may increase user engagement, because, for example, people in small towns may be more interested in what is happening to their local high school quarterbacks than to the national NFL champions. LQP can also be used for personalized search assist by suggesting to users potentially highly popular queries in that region. In addition, LQP has great promise to serve as a timely and sensitive surveillance tool to detect local outbreaks of diseases. In general, an understanding and prediction of local web search trends can be very useful for advertisers, marketers, economists, scholars, and anyone else interested in what are currently top-of-mind in the specific locations.

Although much work on estimating global web query trends exists, the personalized version has been rarely studied in the literature. In fact, when it comes to personalization (or localization in particular), the existing query estimation techniques often encounter big challenge due to data sparsity. We may not have sufficient queries from a particular location to accurately and robustly estimate the models or even some simple statistics. The problem becomes exacerbated along two dimensions: more granular time intervals and more segmented locations. In a small geographical region or/and within a short time span, the observed queries could be very scarce. Yet, these two dimensions are toward the directions of deep personalization and real-time information needs, which are very demanding from web users. Therefore, the LQP task in such scenarios is a very important but challenging problem.

LQP can be essentially boiled down to estimating the probability of a given query issued from a particular location. Thus, statistical language models can be a natural tool to tackle the task. To overcome the data sparsity issue in language models, a background language model is often used for smoothing and has become an indispensable part of any language model. Although a lot of progress has been made in the smoothing techniques, most of them assume a fixed background language model. However, the personalization in LQP and many other web applications requires the background language models to be able to adapt.

In this article, we tackle the LQP task by proposing a series of collaborative language models (CLMs). They alleviate the data sparsity issue by collaboratively collecting queries and trend information from the other locations. Our major contributions can be summarized as follows:

(1) We study an important but rarely investigated personalization task (i.e., LQP) in a principled approach.
(2) Unlike the traditional language models with fixed background language models for smoothing, our proposed CLMs are personalized language models with flexible background language models that are customized to specific locations.

(3) The most sophisticated CLM integrates a discriminative component into the generative language models. It enables the collaboration to adapt to specific query topics, which further advances the personalization level.
(4) Most of the prior work on query trend prediction are temporal based by using previous queries as predictors. Our work can be viewed as spatial-based prediction by looking at the current queries in the other locations that might share similar characteristics with the location of interest. The breaking news or emerging trending queries can be more quickly captured by our approach.
(5) We conduct an extensive set of experiments on a large-scale web query log from Yahoo!. Various traditional methods are used as baselines for comparison and perplexity is used as the evaluation metric. The results show CLMs can improve the predictive performance over the baselines with a large margin, especially when more severe data sparsity is present. We also demonstrate an application of LQP in buzz query detection.

2. RELATED WORK

The study of web search trends has been an active area in both industry and academia. Numerous studies have been conducted on predicting the upcoming query trends. Liu et al. [2008] proposed to unify both periodicity and accidental factors with classical autoregression time series model for predicting the query frequency. Golbandi et al. [2013] also used a linear autoregression model to predict query counts for search trend detection. Adar et al. [2007] investigated the general trends for queries in several datasets of queries, blog posts, and news articles. Vlachos et al. [2004] focused on burst and periodic queries, representing them concisely using coefficients in a Fourier transform. Chien and Immorlica [2005] presented an efficient method for finding related queries by correlating queries with similar time series distributions. Identifying trends in queries has real applications. Shokouhi [2011] indicated that the seasonal nature of queries can be detected using time series analysis. Some recent work has demonstrated that web search volume can "predict the present" [Shimshoni et al. 2009; Choi and Varian 2009], meaning that it can be used to accurately track outcomes such as unemployment levels, auto and home sales, and disease prevalence in near real time.

The aforementined work focuses on characterizing the absolute query frequency. A closely related task, buzz query detection, is to discover the trending queries by looking at the change in query frequency. The services such as Yahoo!'s Trending Now, Google Trends, and Bing's Popular Now mainly fall into this category. Because of users' increasing need for time-sensitive information, this task has recently attracted much attention. The problem can be formulated as anomaly detection, which finds irregularities of the query such as a large divergence from the mean number of occurrences [Vlachos et al. 2004; Kleinberg 2003; Dong et al. 2010; Parikh and Sundaresan 2008]. Moreover, Kulkarni et al. [2011] identified several interesting features by which changes to query popularity can be categorized. Some work utilizes implicit user feedback. For example, Diaz [2009] determined the newsworthiness of a query by predicting the probability of a user clicking on the news display of a query. Konig et al. [2009] estimated the click-through rate for dedicated news search result with a supervised model.

Most of the aforementioned web query prediction works are based on temporal prediction, meaning they use the previous queries as the indicators to predict the future query trends. However, the web is such as a dynamic environment that every day there are many new queries issued that are never seen before. These emerging queries are largely caused by breaking news or trending subjects, which are of great interest to many web applications. The temporal prediction approach cannot even take into these emerging queries into account (if they do not appear in the current location), not to mention accurate estimation. In contrast, our approach can be regarded as spatial

prediction, which considers the current queries from the other locations. The breaking news or emerging trending queries can be more quickly captured by this approach.

In recent years, personalization has received tremendous attention from the research community because its great potential to improve the relevance of web services. Among many observable user attributes, approximate user location is particularly simple for search engines to obtain. Welch and Cho [2008] presented a method for automatic identification of a class of queries they defined as localizable from a web search engine query log. Yi et al. [2009] attempted to discover users' implicit geographic intention in web search. To the best of our knowledge, there is no prior work in the literature on personalized estimation or prediction of web query trends. A related work is Google's Insight for Search.[4] However, it can only show the search trends in the time span of 7 days at the finest level, which suffers much less from data sparsity, but could not satisfy users' ever increasing needs for the time-sensitive information. Another related website is Trendsmap,[5] but it identifies local trending topics based on tweets rather than queries. Moreover, the optional locations are all international big cities. Thus, it also suffers less from data sparsity, but it may be too coarse to have the taste of personalization. Our prior work [Bawab et al. 2012] investigated the task of finding trending local topics, and it has shown that many relevant topics may be missed due to data sparsity.

The LQP task can be essentially boiled down to estimate the statistical language models of search queries. The root of language modeling dates back to the beginning of the 20th century [Manning and Schutze 1999]. For many years, language models have been used primarily for automatic speech recognition. Since it was introduced to information retrieval in 1998 by Ponte and Croft [1998], it has sparked genuine interests in the research community. For a general survey, please refer to Zhai [2008]. In many language models, smoothing is a crucial component to address the data sparsity problem [Chen and Goodman 1996; Zhai and Lafferty 2001]. Various smoothing techniques have been proposed. However, most of them assume a fixed background language model, which is inadequate for many personalized applications such as LQP.

Another closely related body of work is collaborative filtering [Breese et al. 1998], which is an information filtering process using techniques involving collaboration among multiple entities. At the abstract level, CLMs are similar to user-based collaborative filtering in Recommender Systems, but there are several key differences: (1) the observed variables in LQP are not static user ratings but rather the statistical language models of queries (the collaboration in CLMs is between the language models of the queries) and (2) collaborative filtering techniques often compute the pairwise similarity just based on the two involved entities. In contrast, CLMs obtain the similarities by looking at multiple other locations and see how they can collaboratively generate the observed queries.

3. LOCALIZED QUERY PREDICTION

3.1. Data Sparsity

The web search trends can be characterized by the query generation probability. The higher the probability, the more likely that the query is popular. In consequence, to predict the local search trends, the LQP task can be boiled down to estimating the probability of producing the web query q from a particular location s at time t. This probability can be determined by the query language model $p(q|s,t)$. A statistical language model assigns a probability distribution over a sequence of linguistic units in a

[4]http://www.google.com/insights/search/.
[5]http://trendsmap.com/.

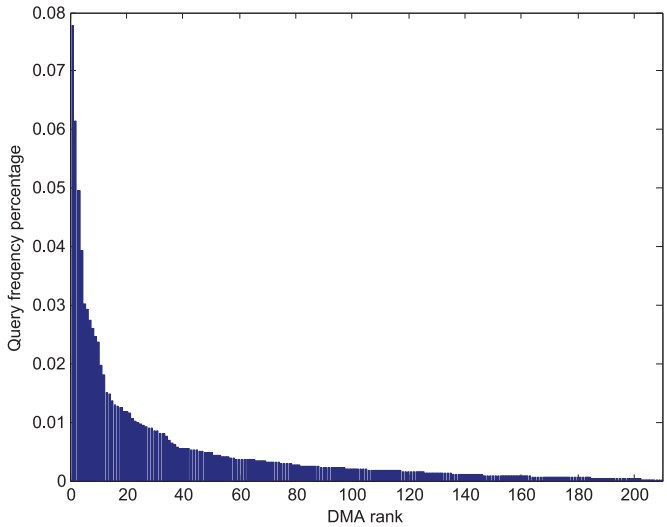

Fig. 3. Distribution of Yahoo! query volume over 210 designated market areas in the United States June 1–7, 2011. The DMAs are ordered by their query volume.

language [Zhai 2008]. It has been heavily used in a wide range of information retrieval and natural language processing applications. The language modeling approaches are appealing in both theory and applications because of their good empirical performance and great potential of leveraging statistical estimation methods.

In this article, the linguistic unit in the language models is queries, but the models can be easily generalized to n-gram models [Zhai 2008]. In addition, the location s represents one of the designated market areas (DMAs). DMAs are geographic areas defined by Nielsen media research company as a group of counties that make up a particular television market. There are 210 Nielsen DMAs in the United States. The reason why we use DMAs is that they are proven (by the TV industry) to be effective in targeting geographic audience with customized services.

A straightforward way to compute $p(q|s,t)$ is by looking at all the queries issued from the region s over the time period $t-1$ to t. $p(q|s,t)$ is then essentially the relative query frequency with respect to these queries. $p(q|s,t)$ can be estimated accurately when there are sufficient queries observed. Unfortunately, this is not the case for many DMAs. Figure 3 shows the relative query volume/frequency during the period of June 1, 2011 to June 7, 2011, with respect to the 210 DMAs. The DMAs are ordered by their query volume. This figure shows that the distribution seems to follow the classic "long-tail" distribution (or Zipf's Law [Manning et al. 2008]), which usually implies data sparsity for the vast majority of ranks. We can see that the top 20 DMAs account for over half of the total queries, whereas the vast majority of the other DMAs each have less than 1% of the total query traffic. Furthermore, the bottom DMAs have much less query volume. This can be more clearly seen in Figure 4, which plots the data on logarithmic scales. The data satisfying Zipf's Law should roughly fit a straight line in the log-log plot [Manning et al. 2008]. From the figure, we can find that there is a "drooping tail" starting from around the 100th DMA. This droop tail indicates there are not enough queries for these bottom DMAs to support Zipf's Law, which means the DMAs suffer from even more severe data sparsity issue than Zipf's Law indicates. For the lowest ranked DMAs, very few queries can be observed in 1 day. This poses a big challenge of data sparsity to estimate the conditional probability $p(q|s,t)$. The situation becomes worse when the model needs more frequent updates (i.e., shorter

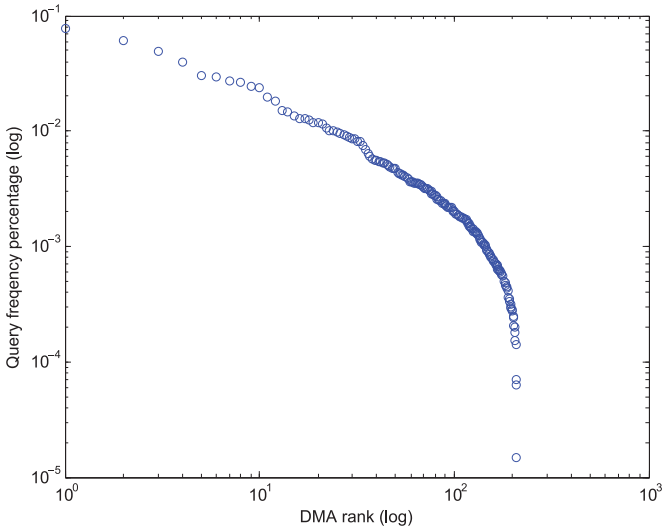

Fig. 4. Log-log graph of relative query volume.

time interval from $t-1$ to t) for real-time information needs. Even some large DMAs may encounter the data sparsity issue on an hourly basis.

3.2. Background Language Models for Smoothing

To overcome data sparsity in language models, a background language model is often used for smoothing. Background language models are usually constructed from the whole collection and thus suffers less from sparsity. Smoothing techniques try to balance the probability of observed queries with those unobserved ones. It discounts the probability mass assigned to the seen queries and distributes the extra probability to the unseen queries. Smoothing has become an indispensable part for any language model [Zhai 2008]. There exists different smoothing techniques, and in this article, we illustrate our approach with Dirichlet smoothing [Zhai 2008] as follows

$$p(q|s_i,t) = \frac{n^t_{qi} + \mu p(q|C)}{N^t_i + \mu}, \tag{1}$$

where n^t_{qi} is the number of queries q generated in target location s_i at time t, N^t_i is the total number of queries in s_i at t, and μ is the smoothing parameter and can be determined by cross-validation. $p(q|C)$ is the background or collection language model, as they are calculated based on the whole collection C (i.e., $p(q|C) = \frac{N_q}{N_C}$, where N_q is the number of query q observed in all the DMAs and N_C is the total number of queries in the whole collection). Dirichlet smoothing can adjust the amount of reliance on the observed queries according to their sizes. When there are not sufficient queries (i.e., N^t_i is small), the background language model will be critical to estimate the query language model $p(q|s_i,t)$.

Like most existing language models, Equation (1) uses a fixed background language model $p(q|C)$ that does not depend on location s_i. However, these models are not suitable for the LQP task because they are invariant with respect to different locations and lose the taste of personalization. Therefore, we propose the collaborative language models (CLMs) by using variable background language models denoted by $\hat{p}(q|s_i)$. These background language models are able to adapt to various locations based on

how the target location is correlated with the other locations, and then we can use the correlation to make personalized prediction. By replacing $p(q|C)$ in Equation (1) with $\hat{p}(q|s_i)$, the query language model is

$$p(q|s_i, t) = \frac{n_{qi}^t + \mu \hat{p}(q|s_i, t)}{N_i^t + \mu}. \tag{2}$$

4. COLLABORATIVE LANGUAGE MODELS

4.1. Basic CLM

The motivation of the collaborative language models is very similar to collaborative filtering. The underlying assumption of the CLM approach is that those locations that are correlated in the near past tend to correlate again in the near future. Therefore, once we figure out the correlation, we can utilize it to predict the queries at the target location by looking at the current queries in the other locations. Formally, to estimate the background language model $\hat{p}(q|s_i, t)$ in Equation (2), by probability chain rule and marginalization, we have:

$$\hat{p}(q|s_i) = \sum_{j \neq i} p(q|s_j, s_i) p(s_j|s_i) = \sum_{j \neq i} p(q|s_j) p(s_j|s_i), \tag{3}$$

where s_j is a different location from the target s_i and $p(q|s_j)$ is the query language model of s_j. $p(s_j|s_i)$ is the transition probability from s_i to s_j, which also measures the similarity between s_i and s_j. Since we look at the locations at the same time t, we drop the variable t in the conditions of all the probabilities in Equation (3) to simplify the notations. Equation (3) assumes q is independent of s_i given s_j. It depicts a generative process of how a query q in s_i is generated: we first randomly choose a location s_j based on the probability $p(s_j|s_i)$, and then we pick the query q (within s_j) based on the probability $p(q|s_j)$. Each query is conditionally independently generated given the location.

Equation (3) is the basic CLM (or BCLM). The basic idea is to discover how the queries in one location (target) can be collaboratively generated by the other locations (indicators) in the training phase and then use this collaboration (possibly in real-time) to predict the query trends for the target location that may not be able to figure out the trends by its own queries because of its small query volume. BCLM is a generative probabilistic model and the graphical model representation is shown in Figure 5 (top).

The generative process of BCLM is similar to the multinomial mixture model [Nocedal 1980]. One difference is that we can choose the location s_j so that it has sufficient queries and thus $p(q|s_j)$ can be assumed known. Therefore, the only parameters that need to estimate are $p(s_j|s_i)$, which can be obtained by maximum likelihood estimation (MLE) as follows:

$$\max_{p(s_j|s_i)} \prod_{k=1}^{T} \sum_{j \neq i}^{M} p\left(q_k^i|s_j\right) p(s_j|s_i), \tag{4}$$

where q_k^i denotes the queries observed in s_i and T is the total number of them. M is the total number of the indicator locations. This is a convex optimization problem where the global optional solution can be achieved. Specifically, $p(s_j|s_i)$ can be estimated by the Expectation and Maximization (EM) algorithm [Dempster et al. 1977] to iterate over the following two steps:

E-step:

$$p\left(s_j|q_k^i, s_i\right) = \frac{p(s_j|s_i) p\left(q_k^i|s_j\right)}{\sum_{j \neq i}^{M} p(s_j|s_i) p\left(q_k^i|s_j\right)} \tag{5}$$

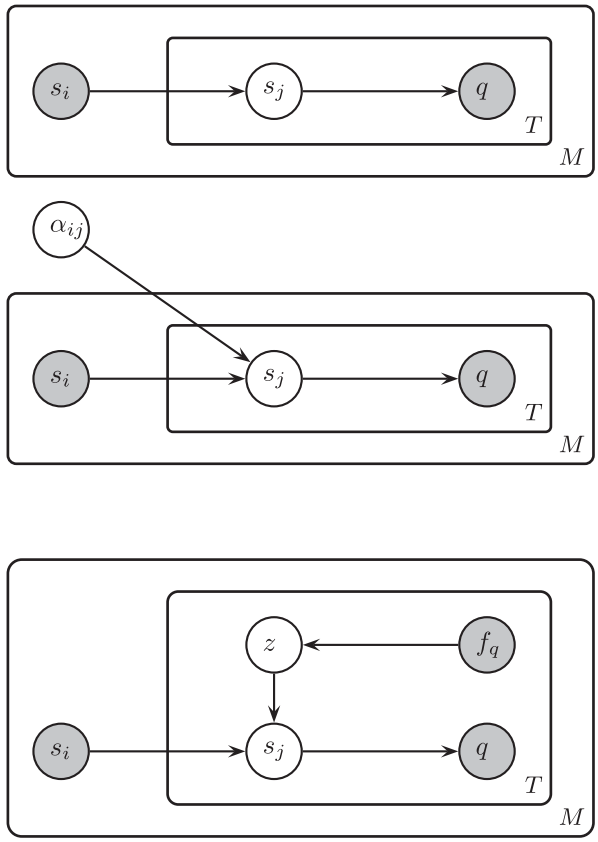

Fig. 5. Graphical model representation of collaborative language models. Top: Basic CLM. Middle: Dirichlet prior CLM. Bottom: Topic-dependent CLM. The shaded nodes are observed variables and unshaded are latent variables.

M-step:

$$p(s_j|s_i) = \frac{1}{T}\sum_{k=1}^{T} p\left(s_j \mid q_k^i, s_i\right) \tag{6}$$

Another more straightforward way to estimate $p(s_j|s_i)$ is to directly compute the similarity between the historical queries in s_j and s_i. For example,

$$p(s_j|s_i) = \frac{1}{Z(Q_i)} cosine(Q_j, Q_i), \tag{7}$$

where $Q_j = (w_1, w_2, \ldots, w_n)$ is a vector representation of all the queries in s_j and w_q is the query weight (e.g., tf-idf weighting) for the query q. *cosine* is the cosine similarity [Manning et al. 2008] between Q_j and Q_i. $Z(Q_i) = \sum_{j\neq i}^{M} cosine(Q_j, Q_i)$ is the normalization term to ensure $\sum_{j\neq i}^{M} p(s_j|s_i) = 1$. Any other similarity measures such as Pearson correlation can also be used to compute the similarity between Q_j and Q_i. This method is analogous to the ones in user-based collaborative filtering for Recommender Systems [Breese et al. 1998]. They calculate the pairwise similarity between two users and produce a prediction for the user by taking the weighted average of all the ratings.

In contrast, CLMs obtain the similarities by looking at *multiple* other locations and see how they can collaboratively generate the observed queries.

Whereas most query trend prediction approaches are temporal based by using the previous queries as predictors, the CLM approach is spatial-based prediction by collaboratively using the current queries from the other locations. Even without the data sparsity issue, it is still often desirable for many personalized applications to identify the collaboration in order to make personalized recommendation.

4.2. Dirichlet Prior CLM for Encoding Proximity Information

Equation (4) provides a probabilistic framework to estimate $p(s_j|s_i)$, which is essentially a normalized weight to measure the similarity between s_j and s_i. In many cases, we already have prior knowledge about these weights. For example, if s_j is geographically close to s_i, the weight should tend to be large. Within the probabilistic framework, we can encode this prior knowledge by a prior distribution over $p(s_j|s_i)$. Specifically, we use Dirichlet prior as follows:

$$Dir(p|\alpha_1, \ldots, \alpha_M) = \frac{1}{Z(\alpha)} \prod_{j \neq i}^{M} p_j^{\alpha_j - 1}, \tag{8}$$

where $Z(\alpha) = \frac{\prod_{j \neq i}^{M} \Gamma(\alpha_j)}{\Gamma(\sum_{j \neq i}^{M} \alpha_j)}$ is a normalization constant [Bishop 2006]. The reason to use Dirichlet prior is that it is the conjugate prior to the multinomial distribution $p(q|s_j)$, which leads to computational convenience for parameter estimation [Bishop 2006]. We refer this model as to Dirichlet prior CLM (or DPCLM).

We use the hyperparameter α in Equation (8) to encode our prior knowledge about geographical proximity. Specifically, α_j is inversely proportional to the distance between s_j and s_i (i.e., the larger the distance, the smaller the weight):

$$\alpha_j = \beta \frac{\frac{1}{d_{ij}}}{\sum_{j=1}^{M} \frac{1}{d_{ij}}}, \tag{9}$$

where d_{ij} is the distance between s_i and s_j. β is the parameter to control how confident we are about the prior knowledge. The graphical model representation of DPCLM is shown in Figure 5 (middle).

Once the prior is incorporated into Equation (4), the maximum a posteriori (MAP) estimate for $p(s_j|s_i)$ can be obtained by maximizing the posterior distribution. A similar EM algorithm can be derived as follows (see Rigouste et al. [2007] for the details).

E-step:

$$p\left(s_j|q_k^i, s_i\right) = \frac{p(s_j|s_i)p\left(q_k^i|s_j\right)}{\sum_{j \neq i}^{M} p(s_j|s_i)p\left(q_k^i|s_j\right)}$$

M-step:

$$p(s_j|s_i) = \frac{\alpha_j - 1 + \sum_{k=1}^{T} p\left(s_j|q_k^i, s_i\right)}{\sum_{j \neq i}^{M}(\alpha_j - 1) + T} \tag{10}$$

From Equation (10), we can see that when we do not have sufficient queries (i.e., T is small), the prior knowledge α will play a big role in the estimation of $p(s_j|s_i)$. When T becomes large, the observed queries will dominate the estimation. Therefore, this model uses the prior geographical information to guide the parameter estimation, which could be especially useful in the case of data sparsity.

4.3. Topic-Dependent CLM

Both BCLM and DPCLM models assume the same fixed weights/collaboration for all kinds of queries. However, in many cases, this is too rigid. For example, in Los Angeles, the queries about basketball may have great correlation with those from Dallas during the week of May 2, 2011, because they were the opponents of NBA playoffs. On the other hand, this may not be the case for entertainment-related queries and instead Los Angeles may share large similarity with New York on that topic. Another example could be San Jose and Seattle share great similarity on issuing information technology–related queries, but not so much on other topics. Therefore, the best collaboration/weighting strategy for a query is not necessarily the best one for other queries. We could benefit from developing a query-dependent model in which we can choose the collaboration strategy individually for each query. Because it is not realistic to determine the proper collaboration strategy for every query, we can classify queries into one of several topic classes. The queries within the same topic class shares the same strategy, and the queries with different topics could have different collaboration strategies.

In this section, we present the topic-dependent CLM (or TDCLM) by introducing an intermediate latent class layer to capture the query topic information. Specifically, we can use a multinomial variable $z \in K$ to indicate which query topic class the similarity weight is drawn from. Under different topics, the similarities are different, which are thus denoted by $p(s_j|s_i, z)$ (instead of $p(s_j|s_i)$). The choice of z depends on the query q. By marginalizing out the hidden variable z, the corresponding probabilistic model can be written as

$$\hat{p}(q|s_i, f_q) = \sum_{z=1}^{K} \sum_{j \neq i}^{M} p(q|s_j, z) p(s_j|s_i, z) p(z|f_q), \tag{11}$$

where K is the total number of the latent topics. z is determined by $p(z|f_q)$, which measures the probability of q belonging to the topic z. It is noticeable that this is a soft version of "query categorization," which leads to a probabilistic membership assignment of queries to latent query classes. Specifically, each query q is denoted by a bag of query features $f_q = (f_1, \ldots f_R)$, where R is the number of query features. $P(z|f_q)$ can be modeled by a soft-max function $\frac{1}{Z_q} \exp(\sum_{m=1}^{R} \lambda_{zm} f_m)$, where Z_q is the normalization factor that scales the exponential function to be a proper probability distribution (i.e., $Z_q = \sum_{z=1}^{K} \exp(\sum_{m=1}^{R} \lambda_{zm} f_m)$). $p(s_j|s_i, z)$ measures the similarity between s_i and s_j under the topic z. $p(q|s_j, z)$ measures the probability of q generated by s_j under the topic z. We assume the construction of the language model given s_j is conditionally independent of the topic z. In other words, $p(q|s_j, z) = p(q|s_j)$, which is assumed already known. The probabilities that need to estimate are $p(s_j|s_i, z)$ and $p(z|f_q)$. By plugging the soft-max function and $p(q|s_j)$ into Equation (11), we can get

$$\hat{p}(q|s_i, f_q) = \sum_{z=1}^{K} \sum_{j \neq i}^{M} p(q|s_j) p(s_j|s_i, z) \frac{1}{Z_q} \exp\left(\sum_{m=1}^{R} \lambda_{zm} f_m\right). \tag{12}$$

Different from traditional language models, which are typically generative, Equation (12) incorporates a discriminative component (the soft-max function) into a generative model. It can be viewed as a hybrid of generative and discriminative models [Bishop 2006]. One of its big advantages over the fully generative models is that TDCLM is able to handle the queries that are not seen in the training phase. Specifically, because λ_{zm} is associated with each query feature instead of each training query, once λ_{zm} is estimated from the training data, it can be applied to any unseen queries, as long as they have the query features f_q. This yields a great advantage when

dealing with breaking news or trendy queries, which may be never seen before in many cases. At the same time, TDCLM still holds the great advantage of generative models. Because of the query *generative* process, TDCLM does not need manual labeling of training data, which could be quite expensive in many applications of fully discriminative models. In the experiments, we utilize over 200 million observed queries in training without any labeling effort. The graphical model representation of TDCLM is shown in Figure 5 (bottom).

The parameters can be determined by maximizing the following data log-likelihood function:

$$\begin{aligned} L &= \sum_{k=1}^{T} \log\left(p\left(q_k^i \middle| f_q^k, s_i\right)\right) \\ &= \sum_{k=1}^{T} \log\left(\sum_{j\neq i}^{M}\sum_{z=1}^{K} p\left(q_k^i|s_j,z\right)p(s_j|s_i,z)p\left(z\middle| f_q^k\right)\right) \\ &= \sum_{k=1}^{T} \log\left(\sum_{j\neq i}^{M}\sum_{z=1}^{K} p\left(q_k^i|s_j\right)p(s_j|s_i,z)\frac{1}{Z_q}\exp\left(\sum_{m=1}^{R}\lambda_{zm}f_m^k\right)\right) \end{aligned}$$

We can use the EM algorithm to estimate the parameters. The E-step can be derived as follows by computing the posterior probability of s_j and z given query q_k^i and its query features f_q^k,

$$\begin{aligned} p\left(s_j,z|q_k^i,\, f_q^k, s_i\right) &= \frac{p\left(q_k^i|s_j,z\right)p\left(s_j,z|s_i,\, f_q^k\right)}{\sum_{j\neq i}^{M}\sum_{z=1}^{K} p\left(q_k^i|s_j,z\right)p\left(s_j,z|s_i,\, f_q^k\right)} \\ &= \frac{p\left(q_k^i|s_j\right)p(s_j|s_i,z)p\left(z\middle| f_q^k\right)}{\sum_{j\neq i}^{M}\sum_{z=1}^{K} p\left(q_k^i|s_j\right)p(s_j|s_i,z)p\left(z\middle| f_q^k\right)}. \end{aligned}$$

By optimizing the auxiliary function [Dempster et al. 1977], we can derive the following M-step update rules,

$$\max_{\lambda} J(\lambda) = \sum_{k=1}^{T}\sum_{j\neq i}^{M}\sum_{z=1}^{K} p\left(s_j,z|q_k^i,\, f_q^k, s_i\right)\log\left(\frac{1}{Z_q}\exp\left(\sum_{m=1}^{R}\lambda_{zm}f_m^k\right)\right) \tag{13}$$

$$p(s_j|s_i,z) = \frac{\sum_{k=1}^{T} p\left(s_j,z|q_k^i,\, f_q^k, s_i\right)}{\sum_{j\neq i}^{M}\sum_{k=1}^{T} p\left(s_j,z|q_k^i,\, f_q^k, s_i\right)} \tag{14}$$

Equation (13) can be optimized by any gradient descent method. In particular, we use the L-BFGS method [Nocedal 1980] due to its efficiency. The method requires us to compute the derivative of J with respective to $\lambda_{\hat{z}m}$ as follows:

$$\frac{\partial J}{\partial \lambda_{\hat{z}m}} = \sum_{k=1}^{T}\sum_{j\neq i}^{M}\sum_{z=1}^{K} p\left(s_j,z|q_k^i,\, f_q^k, s_i\right)\left(\delta_{\hat{z}z} - \left(\frac{1}{Z_q}\exp\left(\sum_{m=1}^{R}\lambda_{\hat{z}m}f_m^k\right)\right)\right)f_m^k, \tag{15}$$

where $\delta_{\hat{z}z} = 1$ if $\hat{z} = z$, otherwise $\delta_{\hat{z}z} = 0$. Equation (15) can be conveniently derived based on the derivatives of the soft-max function [Jordan and Xu 1995]. The number of latent topics K can be obtained by maximizing the sum of log-likelihood and some model selection criteria. In the experiments, we choose Bayesian information criterion

Table I. Testbed Statistics in Order of Magnitude
"M" denotes the order of million. "k" denotes the order of thousand.

# of queries in training set	200M
# of distinct training queries	15M
# of queries in test set	40M
# of distinct test queries	2.5M
Target DMA	Houston, TX Greenville, SC St. Joseph, MO
# of test queries in Houston	1M
# of distinct test queries in Houston	200k
# of test queries in Greenville	200k
# of distinct test queries in Greenville	70k
# of test queries in St. Joseph	10k
# of distinct test queries in St. Joseph	3.5k

Table II. The Nine Category Features for TDCLM

1) Sports	2) Travel	3) Entertainment
4) Politics	5) Technology	6) Places
7) Religion	8) Education	9) Finance

(BIC) [Schwarz 1978] as the selection criterion, which is a measure of the goodness of fit of an estimated statistical model, defined as $\max 2L - r\log(T)$, where r is the number of parameters in the statistical model.

5. EXPERIMENTAL SETUP

5.1. Data

We use a large industrial-scale real-world query log for this study, by collecting the data from the Yahoo! web search log over the period of June 1, 2011 to June 8, 2011. We use the queries from June 1 to June 7 as the training data to build the models and then use the queries from June 8 as the test data to evaluate their predictive performance. A 1-week training period is arguably a reasonable time span to discover the current correlations among DMAs. A longer time span may not be able to capture the quick shift of web user interests especially in breaking news. A shorter one may tend to overfit the limited observations. We filter out the queries that are generally considered not of interest according to a predefined blacklist. The queries are then preprocessed by some standard normalization techniques such as converting the upper case to lower case and removing repetitive spaces [Manning et al. 2008]. Three DMAs are chosen as the targets for evaluation based on their representativeness of size. The ranks of their query volumes on the training data are 10th (Houston, TX), 40th (Greenville, SC), and 200th (St. Joseph, MO), respectively. They represent large, medium, and small DMAs, respectively. The data statistics in order of magnitude[6] is shown in Table I.

For the topic-dependent CLM, we select a total of 10 query features. One feature is the length of the query. The other nine features are the probabilities that the query belongs to a predefined set of categories. We use an ensemble method of Conditional Random Fields [Lafferty et al. 2001] and Maximum Entropy [Ratnaparkhi et al. 1996] to obtain the features. All these query features can be automatically extracted from the given query through natural language processing techniques. Table II includes the nine category features in the experiments, which cover a diverse range of topics.

[6]Due to the company policy, we could not disclose the specific statistics.

Table III. The Methods and Their Acronyms in the Experiments

FB	Fixed background model ($p(q\|C)$, baseline)
IGD	Inverse geographical distance (Equation (9))
EW	Equal weights (i.e., $p(s_j\|s_i)$ is uniform)
CS	Cosine similarity (Equation (7))
BCLM	Basic CLM (Section 4.1)
DPCLM	Dirichlet prior CLM (Section 4.2)
TDCLM	Topic-dependent CLM (Section 4.3)
TP-FB	Temporal-based prediction with FB as background model (Equation (1))
TP-BCLM	Temporal-based prediction with BCLM as background model
TP-TDCLM	Temporal-based prediction with TDCLM as background model

Each latent query class in TDCLM can be viewed as a linear combination of the query features.

5.2. Baselines

Table III summarizes the methods the we compare in the experiments. FB denotes the traditional background (collection) language model (i.e., $p(q|C)$ in Section 3) by using all the queries equally without weighting them for different DMAs. This method serves as the baseline for comparison. IGD denotes the method that directly uses Equation (9) as the weights without looking at the queries. "CS" is the cosine similarity method shown in Section 4.1. Similar to "tf-idf" weighting [Manning et al. 2008], we use "query frequency - inverse DMA frequency" as the query weighting scheme (by treating queries as words and DMAs as documents). For BCLM, DPCLM, and TDCLM, smoothing is also needed to build the language models $p(q|s_j)$ for the indicator DMAs s_j. We choose Dirichlet smoothing with the parameter $\mu = 5{,}000$, which shows good empirical performance in other applications [Zhai and Lafferty 2001]. All the query processing and language model building are done on the Yahoo! Hadoop cloud computing infrastructure.

All the CLMs focus on the estimation of background language model $\hat{p}(q|s_i, t)$ in Equation (2). They utilize the queries from the other locations s_j at the current time. In the experiments, we do not directly evaluate the query language model $p(q|s_i, t)$ in Equation (2) because we have to hold the queries (q_i^t) from the current location at the current time (as the ground truth) for evaluation. On the other hand, we can evaluate the model that replaces n_{qi}^t and N_i^t in Equation (2) by n_{qi}^{t-1} and N_i^{t-1}. This model is essentially temporal-based prediction smoothed by $\hat{p}(q|s_i, t)$, that is, $p(q|s_i, t) = \frac{n_{qi}^{t-1}+\mu\hat{p}(q|s_i,t)}{N_i^{t-1}+\mu}$. It utilizes the queries in both temporal and spatial dimensions: (1) n_{qi}^{t-1} and N_i^{t-1} comes from the queries at the current location s_i but from the previous time $t-1$; and (2) $\hat{p}(q|s_i, t)$ comes from the queries at the other locations s_j but at the current time t. Thus, we use TP-FB to denote this kind of method with FB as the background language model. Similarly, TP-BCLM uses BCLM, and TP-TDCLM uses TDCLM, as the background model, respectively.

5.3. Research Questions

An extensive set of experiments are designed to address the following questions:

—Can CLMs improve predictive performance over the traditional fixed background language models? (Section 6.1)

Table IV. The Perplexity Results of Different Methods
"FB" is used as the baseline to compute the "Reduction" in perplexity.

	Houston	Reduction	Greenville	Reduction	St. Joseph	Reduction
FB	90,836	—	93,659	—	95,884	—
IGD	98,139	–8.04%	96,762	–3.31%	93,427	2.56%
EW	103,023	–13.42%	101,818	–8.71%	107,326	–11.93%
CS	90,102	0.81%	89,542	4.4%	87,892	8.34%
BCLM	86,073	5.24%	83,075	11.30%	80,874	15.65%
DPCLM	86,032	5.29%	82,464	11.95%	79,145	17.46%
TDCLM	82,823	8.82%	80,019	14.56%	77,535	19.41%

—Does the increasing sophistication of the series of CLMs lead to increasing predictive performance? (Section 6.1)
—Does the size of the target DMA affect the performance of CLMs? (Section 6.1)
—Do the spatial-based CLMs have advantages over temporal-based prediction? (Section 6.3)
—How do CLMs perform for the specific applications of LQP such as buzz query detection? (Section 6.4)

In the experiments, we use perplexity [Manning and Schutze 1999] as the criterion for model evaluation. Perplexity is a quantitative measure for comparing language models and is often used to compare the predictive performance of language models. The value of perplexity reflects the ability of a model to generalize to unseen data. A lower perplexity score indicates better generalization performance. In our case, perplexity reflects the ability of a model to predict queries for a specific location. The perplexity is algebraically equivalent to the inverse of the geometric mean of per-query likelihood. Formally, the perplexity for a set of test query $Q(s_i, t)$ in s_i at time t is calculated as follows:

$$perplexity(Q(s_i, t)) = \exp - \frac{\sum_{k=1}^{|Q(s_i,t)|} \log\left(\check{p}(q^k|s_i, t)\right)}{|Q(s_i, t)|}, \tag{16}$$

where $|Q(s_i, t)|$ is the total number of test queries in s_i at t and $\check{p}(q^k|s_i, t)$ is estimated from training data by different methods in Table III.

6. EXPERIMENTS

6.1. CLMs versus Baselines

Table IV shows the perplexity results of different methods. We can see that CLMs achieve better results than the other methods for all the three DMAs. The gaps are more prominent for the smaller DMAs. Furthermore, more sophisticated CLMs gain more reduction in perplexity. TDCLM achieves the largest improvement on St. Joseph with 19.41% perplexity reduction over FB. The difference between BCLM and DPCLM is more noticeable in smaller DMAs. EW generates the worst results, which shows the importance of appropriate weighting of locations. For all the three DMAs, TDCLM achieves the best performance. This validates the assumption of TDCLM that the correlations between DMAs are not immutable for different queries and they should depend on the query topics.

In general, we can find that the small DMAs exhibit quite different patterns from the large DMA in the results. This may be explained by the fact that large DMAs may be more representative for the whole United Sates, and thus their language models are likely closer to the collection language model FB. The large DMAs may tend to be more correlated with national trends, and the geographical proximity may not matter much. To further investigate this hypothesis, for each target DMA, we rank the other

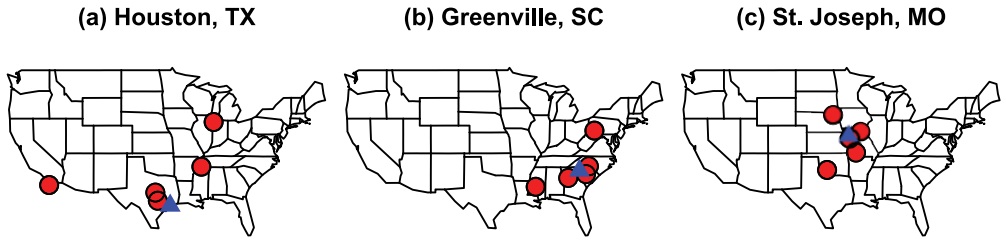

Fig. 6. The top five similar DMAs for three target DMAs, respectively, computed by DPCLM. The triangle in each plot denotes the target DMA and the circles denote its five most similar DMAs.

DMAs in the descending order of $p(s_j|s_i)$ calculated by DPCLM. Figure 6 plots the top five (most similar) DMAs for each target. We can see that the similar DMAs tend to be centered around the smaller DMA (e.g., St. Joseph), and they are more scattered for the large DMA (e.g., Houston). Moreover, Houston is highly correlated with other large DMAs such as San Diego and Chicago, while St. Joseph is more correlated with the small DMAs in its neighborhood.

By comparing TDCLM with DPCLM, we can see that TDCLM achieves more perplexity reduction for larger DMAs. This may come from the fact that large DMAs usually have more topically diverse queries than small DMAs have. Therefore, by adding the topic layer on top of the basic CLMs, the collaboration scheme is more flexible to adapt to the heterogeneous queries. By looking at the parameters that are learned from the topic-dependent CLM, we can find some interesting observations. For Houston, one set of learned feature weights (λ_{zm}) associated with a topic z have two relatively large values on the category features "religion" and "politics" and have small values on the other category features. Therefore, this latent topic z is mainly a combination of religious and political subjects. Under this topic, the most similar DMA (by $p(s_j|s_i, z)$) is San Francisco, CA, which is not on the top five similar DMA lists from either BCLM or DPCLM. The high correlation on this topic may come from the breaking news in that week that Texas Governor Rick Perry chose the association AFA, which advocates against gay rights, to host a big Christian event in Houston.[7] Many San Franciscans may be also interested in this news, as San Francisco is generally considered as the center of the gay right movement in the United States. From this case study, we can see TDCLM's capability of identifying topic-dependent similar DMAs, which could be very useful in many personalized web applications.

6.2. Effect of Parameters

In this experiment, we investigate the effect of β on DPCLM for the three DMAs. β controls the confidence level of the prior knowledge. Figure 7(a) shows the results. We can see that DPCLM is generally not very sensitive to β, especially for large DMAs. For small DMAs, large β can gain perplexity reduction (e.g., $\beta = 100$ vs. $\beta = 5{,}000$ for St. Joseph). In all the other experiments, we choose $\beta = 5{,}000$ for DPCLM. On the other hand, a sensible β can help the EM algorithm converge faster. Figure 7(b) shows the likelihood results of the three different CLMs over different EM iterations for DMA Greenville, SC. We can see that it only takes about 20 iterations for DPCLM to converge to the maximum likelihood, while takes about 35 iterations for BCLM to converge. This computational reduction is important for some real-time applications requiring updating the models very frequently. In addition, TDCLM takes more iterations to

[7]http://www.chron.com/life/houston-belief/article/Perry-s-Houston-prayer-summit-blurs-lines-between-1683179.php.

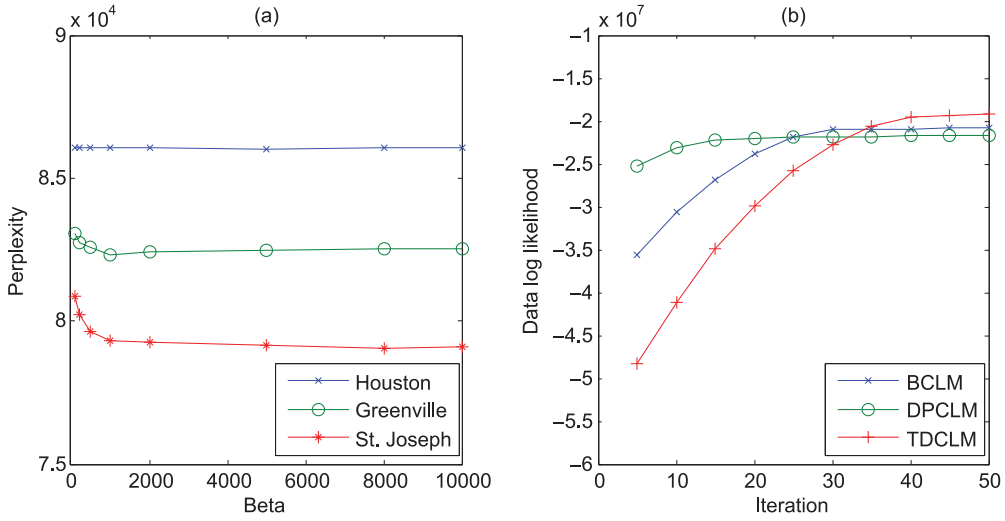

Fig. 7. (a) The perplexity results of DPCLM on varying β for the three DMAs. (b) The data log-likelihood of BCLM, DPCLM and TDCLM over different EM iterations for Greenville, SC.

Table V. The Perplexity Results of Temporal-Based Prediction Smoothed by Different Background Language Models

	Weekly			Daily		
	Houston	Greenville	St. Joseph	Houston	Greenville	St. Joseph
FB	90,836	93,659	95,884	90,836	93,659	95,884
TP-FB	75,372	79,451	83,546	73,139	81,551	88,640
BCLM	86,073	83,075	80,874	86,073	83,075	80,874
TP-BCLM	73,042	77,355	75,941	72,136	79,649	77,431
TDCLM	82,823	80,019	77,535	82,823	80,019	77,535
TP-TDCLM	71,857	76,279	74,982	70,766	78,327	76,331

converge and it also takes much more time for each iteration than BCLM and DPCLM (see Section 4.3).

6.3. Temporal-Based Prediction

As pointed out in Section 5.2, TP-FB, TP-BCLM, and TP-TDCLM exploit both historical queries at the current location and the current queries from the other locations to build the language models. In this experiment, we investigate the predictive performance of these three models. Table V shows the experimental results for two scenarios. In the "Weekly" scenario, we use the queries from the current location over a 1-week of period (i.e., June 1 to June 7) to compute n_{qi}^{t-1} and N_i^{t-1} in Equation (2). In the "Daily" scenario, we only use the queries from June 7 for the temporal component. The background models in Equation (2) are computed by FB and TDCLM, respectively, on the day of June 8, 2011.

By comparing TP-FB versus FB, TP-BCLM versus BCLM, and TP-TDCLM versus TDCLM, we can see that the predictive performance is boosted by incorporating the historical queries. The improvement is more noticeable for Houston. Moreover, TP-TDCLM and TP-BCLM perform better than TP-FB, especially with larger gaps for St. Joseph. By adding the time component, TP-TDCLM performs the best among all the cases. When the scenario moves from "Weekly" to "Daily," we can see that the performance of the temporal-based models gets worse for Greenville and St. Joseph

Table VI. The Top 10 Buzz Queries on June 8th of 2011 for the Three Target DMAs, Computed Based on Eqn. (17) with FB and TDCLM as the Background Language Model, Respectively
The queries are ordered by their buzziness scores (Eqn. (17)). The different queries between FB and TDCLM for each DMA are highlighted in yellow.

Houston	FB	luke bryan, solar flare, stl cardinals, harris metro, houston weather, texas lottery, schlitterbahn, cato, kid rock, belmont stakes
	TDCLM	stl cardinals, luke bryan, solar flare, harris metro, houston weather, texas lottery, schlitterbahn, cato, belmont stakes, six flags over texas
Greenville	FB	belk, gunbroker, myrtle beach, luke bryan, craigslist greenville, shrek forever after, aerolite, solar flare, identity theft, belmont stakes
	TDCLM	myrtle beach, belk, gunbroker, luke bryan, sc lottery, craigslist greenville, ncesc, belmont stakes, shrek forever after, aerolite
St. Joseph	FB	luke bryan, solar flare, crossfit, belmont stakes, joplin tornado, dex, bonnaroo, missouri river levels, sentinel event, schlitterbahn
	TDCLM	missouri river levels, kc royals, federal flood insurance, crossfit, luke bryan, edline, schlitterbahn, joplin tornado, sentinel event, iowa workforce

because they may suffer from data sparsity by the move. In contrast, the temporal-based predictions for Houston perform better in "Daily" than in "Weekly." These experimental results along with those in Section 6.1 indicate that CLMs have great advantage over the existing methods for small DMAs and short time spans. This advantage is very useful in many real-world web applications because personalization and real-time information needs are two very desirable features for web users.

6.4. Application to Local Buzz Query Detection

In this section, we apply CLMs to detect the buzz queries that are potentially of interest to specific locations. This task is motivated by the Yahoo! Trending Now module, which displays 10 buzz queries that are currently trending globally. Our goal is to personalize this module based on user location. The problem can be formulated as anomaly detection [Dong et al. 2010]. Specifically, once we have a query language model $p(q|s,t)$ conditional on s and t, we can compute the buzziness score by looking at the difference between $p(q|s,t)$ and $p(q|s,t-1)$ as follows:

$$buzz(q|s,t) = \log(p(q|s,t)) - \log(p(q|s,t-1)) \tag{17}$$

Other more sophisticated anomaly detection algorithms can be applied, but they all rely on accurate estimation of $p(q|s,t)$. In the experiment, $p(q|s,t-1)$ are obtained by computing Equation (1) over the queries from June 1 to June 7. $p(q|s,t)$ are computed over the June 8 queries, based on Equation (1) (with FB as background model) and Equation (2) (with TDCLM as background model), respectively. Table VI shows the top 10 buzz queries (ordered by buzziness score) for the three target DMAs. The results can be viewed as the "local buzz of the day" of June 8, 2011.

Similar to the perplexity results, the identified buzz queries are more different (between FB and TDCLM) in smaller DMAs. For Houston, there is only one different query: "six flags over texas" versus "kid rock" "ix flags over texas" seems more relevant, as it is a major amusement park in Texas. In addition, "stl cardinals" is at the top in TDCLM while it is ranked at the third in FB. This query is likely relevant because there were MLB games between St. Louis Cardinals and Houston Astros on June 8 and June 9, 2011. For Greenville, TDCLM's "sc lottery" and "ncesc" (The Employment Security Commission of North Carolina) also seem at least more geographically relevant than FB's "solar flare" and "identity theft." For St. Joseph, there are four different queries and the ranking is also quite different. TDCLM's "kc royals," "federal flood insurance,"

and "iowa workforce" seem more relevant, considering St. Joseph is around the border of Missouri, Kansas, and Iowa. In general, the aforementioned qualitative judgments demonstrate that TDCLM produces more relevant buzz queries for the local web users.

7. CONCLUSIONS AND FUTURE WORK

In this article, we study the LQP task in a principled approach. This problem subsumes many interesting personalized web applications that can greatly enhance user interaction with web search engines by providing more customized information discovered from user queries. A series of collaborative language models are proposed to tackle the data sparsity issue and at the same time to promote personalization. The most sophisticated CLM (i.e., TDCLM) enables the personalization to adapt to latent query topics. It can be viewed as a novel hybrid probabilistic model of generative and discriminative models, which gains benefits from both of them. We conduct an extensive set of experiments on a large-scale web query log. The results show that CLMs can substantially improve predictive performance over the existing methods, especially when more severe data sparsity is present. In addition, we show that TP-TDCLM achieves the best results by combining both temporal and spatial predictions. We also demonstrate an application of LQP to local buzz query detection.

Although in this article CLMs are only illustrated by the localization application with the segments of DMAs, they can be readily applied to much smaller segments, or to age, gender groups, or the combination of them. In fact, CLMs are well suited for deep personalization tasks in which the target segment could be extremely small. As long as the segment has any previous observations even over a long period of time, CLMs are able to learn the correlation/collaboration and utilize it to estimate or predict the local query trends. In this sense, CLMs bear great similarity with collaborative filtering (CF), but the collaborative entities in CLMs are statistical language models of queries rather than static ratings in CF. In the future work, it is very interesting to apply CLMs to deeply personalized web applications, such as those targeting at the zip code level or even at the user/IP level. Moreover, although this article is focused on the task of localized query prediction, the proposed models are a general approach that is well suitable to many personalization applications when data sparsity is a severe issue. It is worth exploring more applications of the proposed models, for instance, in personalized recommendations, targeted advertising, location-based social networks, and so on.

ACKNOWLEDGMENTS

We thank George Mills and Anlei Dong of Yahoo! Labs, Zhiheng Huang and Fernando Diaz of Microsoft, (while they were all with Yahoo! Labs) for the helpful discussions of the work.

REFERENCES

E. Adar, D. S. Weld, B. N. Bershad, and S. S. Gribble. 2007. Why we search: Visualizing and predicting user behavior. In *Proceedings of the 16th International Conference on World Wide Web*. ACM, 161–170.

Z. A. Bawab, G. H. Mills, and J.-F. Crespo. 2012. Finding trending local topics in search queries for personalization of a recommendation system. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 397–405.

C. M. Bishop. 2006. *Pattern recognition and machine learning*. Springer.

J. S. Breese, D. Heckerman, C. Kadie, and others. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 43–52.

S. F. Chen and J. Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*. ACL, 310–318.

S. Chien and N. Immorlica. 2005. Semantic similarity between search engine queries using temporal correlation. In *Proceedings of the 14th International Conference on World Wide Web*. ACM, 2–11.

H. Choi and H. Varian. 2009. Predicting the present with Google trends. *Google Technical Report* (2009), 1–23.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B* (1977), 1–38.

F. Diaz. 2009. Integration of news content into web results. In *Proceedings of the 2nd ACM International Conference on Web Search and Data Mining*. ACM, 182–191.

A. Dong, Y. Chang, Z. Zheng, G. Mishne, J. Bai, R. Zhang, K. Buchner, C. Liao, and F. Diaz. 2010. Towards recency ranking in web search. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining*. ACM, 11–20.

Nadav Golbandi Golbandi, Liran Katzir Katzir, Yehuda Koren Koren, and Ronny Lempel Lempel. 2013. Expediting search trend detection via prediction of query counts. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*. ACM, 295–304.

M. I. Jordan and L. Xu. 1995. Convergence results for the EM approach to mixtures of experts architectures. *Neural networks* 8, 9 (1995), 1409–1431.

J. Kleinberg. 2003. Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery* 7, 4 (2003), 373–397.

A. C. Konig, M. Gamon, and Q. Wu. 2009. Click-through prediction for news queries. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 347–354.

Anagha Kulkarni, Jaime Teevan, Krysta M. Svore, and Susan T. Dumais. 2011. Understanding temporal query dynamics. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*. ACM, 167–176.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*. 282–289.

N. Liu, J. Yan, S. Yan, W. Fan, and Z. Chen. 2008. Web query prediction by unifying model. In *Proceedings of the IEEE International Conference on Data Mining Workshops*.

C. D. Manning, P. Raghavan, and H. Schutze. 2008. *Introduction to information retrieval*. Cambridge University Press.

C. D. Manning and H. Schutze. 1999. *Foundations of statistical natural language processing*. Vol. 59. MIT Press.

J. Nocedal. 1980. Updating quasi-Newton matrices with limited storage. *Math. Comp.* 35, 151 (1980), 773–782.

N. Parikh and N. Sundaresan. 2008. Scalable and near real-time burst detection from eCommerce queries. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 972–980.

J. M. Ponte and W. B. Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 275–281.

A. Ratnaparkhi and others. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 133–142.

L. Rigouste, O. Cappé, and F. Yvon. 2007. Inference and evaluation of the multinomial mixture model for text clustering. *Information Processing & Management* 43, 5 (2007), 1260–1280.

G. Schwarz. 1978. Estimating the dimension of a model. *The Annals of Statistics* (1978), 461–464.

Y. Shimshoni, N. Efron, and Y. Matias. 2009. On the predictability of search trends. *Google Inc* 2 (2009).

Milad Shokouhi. 2011. Detecting seasonal queries by time-series analysis. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1171–1172.

M. Vlachos, C. Meek, Z. Vagena, and D. Gunopulos. 2004. Identifying similarities, periodicities and bursts for online search queries. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*. ACM, 131–142.

M. J. Welch and J. Cho. 2008. Automatically identifying localizable queries. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 507–514.

X. Yi, H. Raghavan, and C. Leggetter. 2009. Discovering users' specific geo intention in web search. In *Proceedings of the 18th International Conference on World Wide Web*. ACM, 481–490.

C. X. Zhai. 2008. Statistical language models for information retrieval a critical review. *Foundations and Trends in Information Retrieval* 2, 3 (2008), 137–213.

C. Zhai and J. Lafferty. 2001. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 334–342.

Received March 2013; revised August 2013; accepted September 2013