International Journal of Computer Techniques -– Volume 3 Issue 5, Sep - Oct 2016

RESEARCH ARTICLE OPEN ACCESS

A LOCATION- AND DIVERSITY-AWARE NEWS FEED SYSTEM FOR MOBILE USERS

[1]Ms. Sangeetha .R, [2]Mrs. Sangeetha Lakshmi G.

[1]M.Phil Research Scholar, Department of Computer Science DKM College for Women (Autonomous), Vellore, TamilNadu, India.

[2] Assisant Professor, Department of Computer Science DKM College for Women (Autonomous), Vellore, TamilNadu, India.

--************************--

Abstract:

A location-aware news feed (LANF) system generates news feeds for a mobile user based on her spatial preference her current location and future locations) and non-spatial preference (i.e., her interest). Existing LANF systems simply send the most relevant geo-tagged messages to their users. Unfortunately, the major limitation of such an existing approach is that, a news feed may contain messages related to the same location (i.e., point-of-interest) or the same category of locations (e.g., food, entertainment or sport). We argue that diversity is a very important feature for location-aware news feeds because it helps users discover new places and activities. In this paper, we propose D-MobiFeed; a new LANF system enables a user to specify the minimum number of message categories (h) for the messages in a news feed. In D-MobiFeed, our objective is to efficiently schedule news feeds for a mobile user at her current and predicted locations, such that (i) each news feed contains messages belonging to at least h different categories, and (ii) their total relevance to the user is maximized. To achieve this objective, we formulate the problem into two parts, namely, a decision problem and an optimization problem. For the decision problem, we provide an exact solution by modeling it as a maximum flow problem and proving its correctness. The optimization problem is solved by our proposed three-stage heuristic algorithm.

Keywords— **A location-ware news feed(LANF), Points-of-interest, D-MobiFeed, database security.**

--************************--

I. INTRODUCTION

With the advance of wireless communications and the ubiquity of GPS-equipped smart phones, social network applications have become more prevalent and location-aware, as widely known as location-based social networks (LBSNs) (e.g., Facebook Places and Foursquare). A news feed is a common functionality of existing LBSNs. It enables mobile users to post geo-tagged messages and receive nearby user-generated messages as news feeds at anytime, anywhere. For example, "Bob can receive a news feed with 3 messages that are most relevant to him among the messages within 1 km from his location every 10 seconds". Figure 1a depicts an application scenario.

The geolocation of a message could be a point (e.g., m4), a circular region (e.g., m5), or the spatial region of a venue (e.g., m6 and m7 are spatially associated with restaurant R1). Besides, geotagged messages can be categorized by their underlying venues; for instance, m6 and m7 are posted from users at restaurant R1, so they are intuitively categorized to a "restaurant" category. Our previous work MobiFeed; the state-of-the-art location-aware news feed system schedules news feeds for mobile users. In MobiFeed, the relevance of a message m to Bob is measured by both the content similarity between m and Bob's submitted messages (i.e., a non-spatial factor) and the distance between m and Bob (i.e., a spatial factor). Modified is motivated by the fact that, if the news feeds are only computed based on a user's location at the query time (i.e., it does not consider the user's future locations, e.g., GeoFeed), the total relevance of news feeds is not optimized. For example, in Fig. 1a, there are 11 messages (i.e., m1 to m11) with their geo-location intersecting Bob's query regions (i.e., circular regions in Fig. 1a) at time t0, t1, and/or t2. Assume mi is more relevant to Bob than mj if i < j, and the number of messages per news feed (i.e., k) is 3. GeoFeed returns (m1, m2, m3) at t0, (m4, m6, m7) at t1, and (m5) at t2.

To improve the relevance of news feeds, given Bob's current location at t0, MobiFeed predicts two future locations for him at t1 and t2, and schedules news feeds by considering all three query regions at the same time, which results in a better solution with (m1, m2, m3), (m4, m8, m9), and (m5, m6, m7) at t0, t1 and t2, respectively. Thus, MobiFeed aims at maximizing the total relevance of news feeds by utilizing a location prediction technique. Unfortunately, relevance alone is unable to capture the broader aspects of user satisfaction. Although users expect to receive messages that are highly relevant to their interests, they may prefer a location-aware news feed with a certain level of diversity (i.e., the messages in a news feed belong to a certain number of categories). In conventional web search or recommender systems, topic diversification is a key method to improve user satisfaction.

This work considers a mobile environment that makes our location- and diversity-aware news feed system unique and more challenging. With the geographical distance between a message and a mobile user in a relevance measure model, the relevance of a message to a mobile user is changing as the user is moving. Such a dynamic environment gives us an opportunity to employ location prediction technique to improve the quality of news feeds and the system efficiency. Existing diversification problems focus on retrieving an individual list of items with a certain level of diversity. In contrast, with our location prediction techniques, we aim at improving the quality of news feeds by scheduling multiple location- and diversity-aware news feeds for mobile users simultaneously.

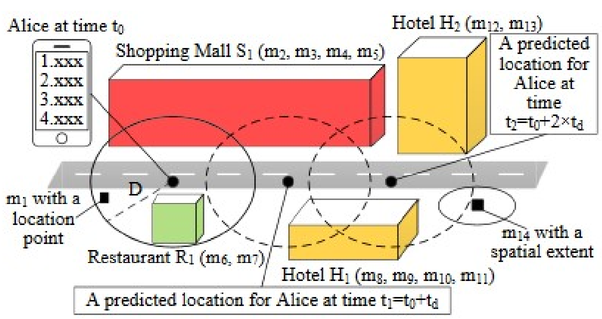

Fig : Location-aware news feed scheduling

u's required minimum message display time *td, u's* specified range distance *D, u's* requested number of messages per news feed, and a look-ahead steps *n,* the *location prediction* function estimates *n* future locations for *u* at times ti = to + td, t2 = to + 2 ***x*** td, ... , and *tn* = to + *n* ***x*** *td,* the *relevance measure* function calculates the relevance score of each candidate message with a geo-location intersecting any u's query region (i.e., a circular area centered at *u.location* or a predicted location with a radius D), and the *news feed scheduler* generates news feeds from the candidate messages for u's query regions at *t0,t1,... ,tn* with the best total relevance score.

II. BACKGROUND OF SQL INJECTION ATTACK

This h-diversity constraint checking is referred to as a decision problem. Finally, the news feed scheduler generates n + 1 news feeds satisfying the h-diversity constraint, and their total relevance score is maximized. The problem of maximizing the total relevance score is referred to as an optimization problem. The computed n + 1 news feeds are sent to u. u's mobile device immediately displays the first news feed (i.e., the one generated for the query region at t0), and then displays each of the remaining news feeds one by one for every td. The terminology of h-diversity constraint is borrowed from the l-diversity principle proposed for privacy-preserving data publishing.

Basically, this principle is used to generalize nonsensitive attributes (e.g., zip codes 13053 and 13068 are generalized to "130**" and ages 28, 29, and 21 are generalized to "< 30") in a class of records such that the sensitive attribute contains at least l different values, in order to protect the privacy of published data.

In this work, we focus on a different problem because D-MobiFeed aims to maximize the relevance of news feeds for mobile users while news feeds satisfy the h-diversity constraint (i.e., the messages in each news feed belong to at least h categories). To the best of our knowledge, this is the first study to incorporate both relevance and diversity for scheduling location aware news feeds for mobile users in LBSNs. In general, the key contributions of this work can be summarized as follows:

1. We extend our previous model MobiFeed (i.e., the state-of-the-art location-aware news feed system) to consider both the relevance and diversity of news feeds when generating newsfeeds for mobile users.
2. We model the decision problem as a maximum flow problem to find the minimum total diversity of a set of n + 1 news feeds for a user based on the user-specified h-diversity constraint.
3. We propose a three-stage heuristic approach to solve the optimization problem. The first stage solves a minimum cost flow problem to guarantee the minimum total diversity in a set of n + 1 news feeds. The second stage addresses a replenish-upto- k problem to maximize the total relevance of these news feeds. The last stage simply sorts the messages in each news feed.
4. We not only conduct a user study, but also evaluate the performance of D-MobiFeed using a real LBSN data set crawled from Foursquare through extensive experiments. The rest of the paper is organized as follows.

Social shopping service users create personal profiles to collect information on different items they find. Instead of simply updating their status on other social networks with a description or link of their purchases, users download software that allows them to grab images of those products to post on their own shopping lists. Some social shopping sites form affiliate relationships with merchants, who often pay percent commissions on sales that come as a result of their products being featured on other sites. Sites have gone so far as to allow users to add their credit card number so their purchases are automatically checked in. Some fashion corporations have invested in sensors placed in their stores and dressing rooms so users on social shopping applications have to physically be in their store or trying something on in order to gather points. This increases participation and encourages customers to try on other clothes.

1*H*-DIVERSITY CONSTRAINT CHECKING

The h-Diversity Constraint Checking (DCC) problem is nontrivial because a brute-force method has to try all possible combinations of news feeds for n + 1 news feeds to find an exact solution. Such a brute-force method is too costly for our online scheduling problem. To this end, we model the DCC problem as a maximum flow problem and prove the exactness and correctness of the model. Finally, the DCC problem returns the minimum total diversity (Definition 2) as an input for the scheduling step (see Fig.). In the next section, we will describe how the scheduling step computes a set of news feeds that satisfy the minimum total diversity and have the maximum total relevance score.

III. System Over view:

an overview of the MobiFeed framework. MobiFeed stores geo-tagged user-generated messages in a database. It interacts with the *location prediction* and *relevance measure* functions to select a collection of messages from the database as a news feed for a mobile user at a particular location.

Geo-tagged messages. A geo-tagged user-generated message is defined as a tuple *(MessagelD, SenderlD, Con-tent, Timestamp, Spatial),* where *MessageID* and *SenderID* are the message identifier and its sender's identifier, respectively. *Content* is the message content. *Timestamp* is the message submission time, and *Spatial* specifies the message's spatial extent. As depicted in Figure 1, the spatial extent of a message can be a point location (e.g., *m1),* a user-specified spatial region (e.g., *m14),* or the spatial region of a venue (e.g., the spatial extent of *m2* is the shopping mall *S1).*

System users. A mobile user *u* at location *u.location* equipped with a GPS-enabled mobile device is able to (a) post a new message with a spatial extent, and (b) receive at most *u.k* messages within u's specified range distance *u.D* (i.e., the query region of a news feed) at a particular time as a news feed. MobiFeed computes a news feed for *u* selecting messages based on their relevance to u and u's movement. Each selected message must be displayed on u's mobile device without any interruption for at least u's spec-ified minimum display time u.td. u reports its location to the system at every time period u.tu. After receiving u's lo¬cation update, a news feed is computed for u. In MobiFeed, we set u.tu = u.td. If u.tu < u.td, newly selected messages cannot be displayed until previously selected messages have been displayed for u.td. On the other hand, if u.tu > u.td, the system may not be able to provide an accurate news feed for u as it does not know u's exact location. Altogether, u.k, u.D and u.td constitute u's preferences for MobiFeed.

Quality measure. Given a user ut and a message mj, the relevance measure function returns a relevance score relevance Score(ui,mj). Without loss of generality, we as¬sume the relevance score is between zero and one. In information retrieval, query-relevance ranking algorithms usually display a document that is more relevant to a user's query at a higher position in a result list [1]. To this end, Mob¬iFeed supports different weights for different slots in a news feed result list, i.e., a higher weight is given to a message displayed at a higher slot because it would be easier to draw a user's attention. In this paper, we use a simple weighting scheme. Given a result list with k slots, the weight of the first slot is k, the weight of the second slot is k — 1, and so on. In general, the weight of a message mj at the top j-th slot (1 < j < k) is displayWeight(j,k) = **k** — (j — 1). Thus, the relevance score of a news feed ft with k messages m1, m2,.. . , mk displayed at the j-th position in a result list for a user ut is calculated as:

relevanceScore(fi) = relevanceScore(ut, mj) j=1 x displayWeight(j, k). (1)

Problem definition. Figure 3 depicts an example of location-aware news feed scheduling, where a user u sends a query with her location to MobiFeed at the current time to. The location prediction function predicts u's location at each of the next two (i.e., n = 2) minimum display times td, i.e., t1 = to + td and t2 = to + 2 x td. There are totally 11 candidate messages for the three news feeds at the times t0, t1, and t2. m4 and m5 are tagged with a spatial region and a point location, respectively. {m1,m2,m3}, {m6,m7,m8,mg], and {m10,m11} are associated with venues A, B, and C (represented by rectangles), respectively. The lifetime of each message with its relevance score for u at t0, t1, and t2 is shown on a timeline chart. Note that the lifetime of m5 is broken from t1 to t2; however, most existing scheduling algorithms assume tasks with a continuous lifetime. Our scheduling problem can be formulated as follows: Given a user u's location-aware news feed query and a look-ahead step n, MobiFeed predicts u's locations at each of the next n minimum display times, and schedules at most k messages for the news feed at each location (i.e., one reported and n predicted locations), such that the total relevance score of the generated news feeds is maximized. Since our problem focuses on online scheduling, it requires efficient query processing.

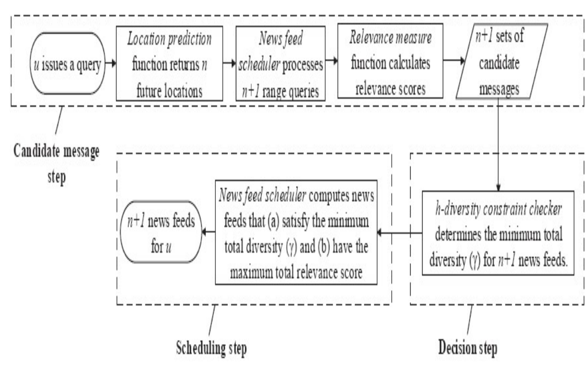

Fig. 5: The work flow of D-MobiFeed.

Fig : the Work flow of Dmob-feeds

The DCC problem is equivalent to the following reduced problem in terms of the result (i.e., given the same input, if the original problem has a positive (respectively, negative) answer, the reduced problem also has a positive (respectively, negative) answer, and vice versa): Definition

4. Reduced h-Diversity Constraint Checking (RDCC) Problem: Given n + 1 sets of candidate messages for u's query regions at t0, t1, t2, . . . , and tn(i.e., CandidateMsgi, where 0 ≤ i ≤ n), along with their category, D-MobiFeed decides whether it could schedule n + 1 news feeds such that each news feed contains exactly h messages, and each message belongs to adistinct category. The RDCC problem can be modeled as the maximum flow problem. Consider our example in Figure 3, where n = 2, h = 3, and three sets of candidate messages are available for u's query region at t0, t1, and t2. This RDCC problem is represented by a flow graph depicted in Figure 6.

VI. LOCATION PREDICTION AND MES¬SAGE RELEVANCE MEASURE

The location prediction function can use any existing lo-cation prediction algorithm if it can predict a user's location at a specified future time in a road network. We here de-scribe how to incorporate the path prediction algorithm [4] into MobiFeed. Given a user w's current location, w's his-torical trajectories, the road map, and a future time t, the path prediction algorithm estimates w's location at t. Let G = (V, E) be a graph model of a road network, where E is a set of road segments and V is a set of intersections of road segments that are represented by circles and lines, re-spectively. The algorithm performs two steps to predict w's direction and speed.

Step 1. Direction prediction. When a user w is moving on an edge ei, this step predicts which adjacent edge ej of ei w will go based on w's historical trajectory set TU. This step has three ways to predict a user's path [4]. (1) Given two edges ei and ej incident to a vertex v, the transition probability of w turning from ei to Cj is defined as P(ei,v,ej) = ('~T(T''~e--jek), where T(TU,ei — ej) is the number of trajectories in Tu that turn from ei to ej and ek is an adjacent edge of ei incident to v. For each adjacent edge ej of ei incident to v, this step calculates P(ei, v, ej) and predicts that w will turn to ej with the largest probability. (b) However, if T(Tu,ei — ej) is empty, the notion of reverse mobility statistics P(ei,v, ej) = ^^^^lle-) is used. (c) In case that both T(TU, ei — ej) and T(TU, ej — ei) are empty, we select the adjacent edge of ei incident to v with the smallest deviation angle from w's current travel direction, which is derived from w's initial location at the query time and current location.

Step 2. Speed prediction. This step estimates w's travel speed S(e) on an edge e. The basic idea is to compute S(e) by the average historical travel speeds of e from w's Tu [4]. If e does not exist in TU, we use a heuristic method that computes S(e) = A(e) x a, where A(e) is the speed limit of e and a is a system parameter. In MobiFeed, after we find that w moving on an edge ei will enter an edge ej at t' from a vertex vs and stay at ej at t. Let (xs,ys) denote the location of vs and (xe,ye) denote the the location of the other vertex of ej. The predicted location of w at t is calculated as (Mxye+^ys) where Ai = (t — t') x S(ej) and X2 = L(ej) — Xi (where L(ej) is the length of ej).

3.2 Message Relevance Measure

MobiFeed only requires the relevance measure function to return a score to indicate the relevance of a message mj to a user wi, i.e., relevance Score(wi,mj). We present three rele-vance measure methods, and then describe how to combine them to implement the relevance measure function. Message categories. We group messages into categories based on their geo-tagged locations or keywords. For exam-ple, in Four Square, each message can be categorized by its one or more associated venues, e.g., restaurant, stadium, and museum. We maintain a user-category matrix where each entry cij is the ratio of the number of a user wi's messages associated with the

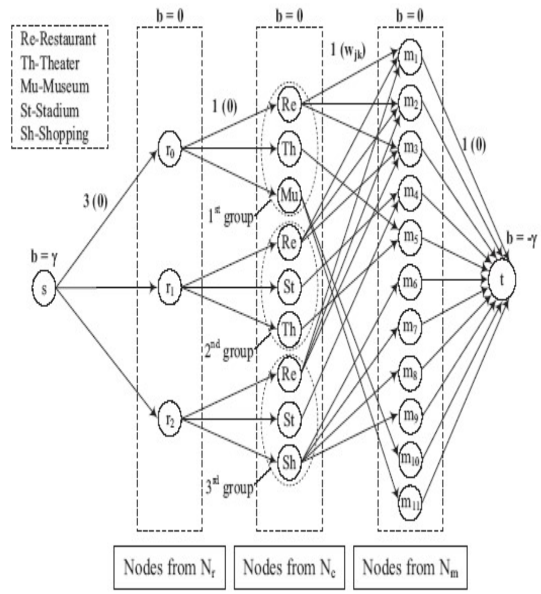

Fig: Flow graph representation of the RDCC problem

(Definition 4) and the -Selection problem (Definition 5). All the edges in the graph have capacity 1 except the edges between source node s and nodes from Nr. The value in parentheses indicates the cost of edges for the representation of the -Selection problem. r0 links to the nodes representing 'Restaurant', 'Theater', and 'Museum' in the first group of Nc), (3) an edge e(cj,mk) for every category cj∈Nc (cj is assumed in (i + 1)-th group of Nc) linking to message mk∈CandidateMsgi which belongs to category cj , with capacity 1 (e.g., in Figure 6, for the category 'Restaurant' in the first group of Nc, it connects to m1, m2, and m3), and (4) an edge e(mk, t) for every message ∈ Nm, with capacity 1. In Figure 6, the label on the top of each set of edges indicates their edge capacity. Given the above graph, the maximum flow problem is to assign

An integer flow value x(vi, vj) ∈ [0, c(vi, vj)] for each edge e(vi, vj) ∈ E such that for every node v ∈ V , the following :

$$\sum_{e(v,v_i)\in E} x(v, v_i) - \sum_{e(v_i,v)\in E} x(v_i, v) = b(v),$$

and the value of the flow (i.e.,) is maximized, that is, we want to route as much flow as possible from s to t. Solution. Apart from the classical algorithms, many others, such as binary blocking flow algorithm and push-relabel method, have been proposed to efficiently solve the maximum flow problem. We employ the relabel-to-front algorithm, which is a particular implementation of the pushrelabel method. This algorithm runs in time O(V 3), where V is the number of nodes in the flow graph.

V. LOCATION-AWARE NEWS FEED SCHEDULER

In this section, we present a n-look-ahead location-aware news feed scheduling algorithm for MobiFeed, where *n* is a system parameter to control the number of locations predicted for a mobile user. In general, *n* should be larger, if the location prediction algorithm can provide more accurate locations, and thus, different values of *n* would be assigned to different users and areas in a road network (e.g., $P(e_i,v, ej)$ should be larger than a certain threshold). Since our online scheduling algorithm has to be efficient and scalable, we design a heuristic scheduling algorithm for MobiFeed.

Data structure. In MobiFeed, a spatial grid structure is used to index all geo-tagged messages. Given a user u's query, a range query is issued to the grid index to retrieve the geo-tagged messages, which are not generated by *u,* associated with a location point, a spatial extent, or a venue region intersecting the query region.

Algorithm. After a mobile user *u* issues a location-aware news feed query to MobiFeed, MobiFeed calls the *location prediction* function to return *n* future locations for u. Its scheduler then finds a set of candidate messages for each of *n* + 1 locations and calls the *relevance measure* function to filter out all candidate messages that do not belong to any top-*5* categories and determine the relevance of each remaining candidate message to u. The scheduler finally returns a news feed for each location such that the total relevance score is maximized. Our heuristic scheduling algorithm consists of two main steps.

Step 1. Candidate message step. Given u's query at time to, the *location prediction* function predicts *n* locations for *u* at times ti, $t_2, ..t_n$, where $t_i = t_0 + u.t_d \times i$ and $u.t_d$ is u's specified message minimum display time. For each of *n* + 1 locations, a range query with a circular region centered at the location with a radius of *u.D* is issued to retrieve the messages intersecting the query region as a set of candidate messages $CandidateMsg_i$ ($0 < i <$ n). Then, the *relevance measure* function filters out all messages that do not belong to any top *5* categories from each $CandidateMsg_i$. For each remaining candidate message *m,* a relevance score for *m,* i.e., *relevance Score(u,m),* is calculated to indicate the relevance of *m* to *u.* Finally, the messages in each $CandidateMsg_i$ are sorted by their relevance score in non-increasing order. To break ties, precedence will be given to a message with a more recent post time.

Step 2. Online scheduling step. As depicted in the running example (see Figure 3), some candidate messages are included in multiple sets of candidate messages. For example, m_1 is included in $CandidateMsg_0$, $CandidateMsg_1$ and $CandidateMsg_2$, so m_1 can be scheduled to one of these query regions or none. This step aims at scheduling at most $k \times (n + 1)$ candidate messages to the *n* + 1 query regions such that the total relevance score of these query regions is maximized. The input of this step is *n* + 1 sets of sorted candidate messages for *n* + 1 query regions. For each query region q_i, we calculate a score for its candidate message *mj* with the highest relevance score by *relevanceScore(u, mj)* **x** *displayWeight(j, k)* (see Equation 1), where *u* is the querying user and *k* is the highest available position in *qi* s result list. The message with the highest score, denoted as *BestMsg,* is selected. *BestMsg* is assigned to the query region giving the highest score, and it is no longer a candidate message for any query region. To break a tie, *BestMsg* is assigned to the query region where the first message in its candidate message set has the smallest relevance score. The reason is that other query regions have a higher chance to put a message with a larger relevance score to the same slot in the result list. Candidate messages are repeatedly selected to appropriate query regions until each query region has *k* messages or its candidate message set becomes empty. Whenever *k* messages have been assigned to a query region, its corresponding candidate message set is discarded. The computed *n* + 1 news feeds are sent to *u.* u's mobile device immediately displays the first news feed, i.e., the query region at $t0$, and then displays each of the remaining news feeds one by one for every td.

Correctness Proof

After solving the above maximum flow problem, we obtain the optimal value of flow (i.e.,). We interpret this value as follows1:

Theorem 1.

(i) If = (n+1)×h, h-diversity constraint can be satisfied for a news feed query.

(ii) If < (n + 1) × h, h-diversity constraint cannot be satisfied for a news feed query. However, indicates the minimum total diversity for these n + 1 news feeds, as defined in Definition 2. Proof. Given the graph modeling of the RDCC problem (see Figure 6), since the capacity c(mk, t) for every mk∈ Nm is set 1. The value of cannot be larger than (n + 1) × h according to the modelling of the flow graph in Section 5.1. to 1, it ensures that every message can be scheduled at most once. With the edges between nodes in Nc and nodes in Nm, every message is associated with its corresponding category. Besides, since the capacity c(ri, cj) with ri∈ Nr and cj∈Nc is set to 1, it guarantees that for each news feed, D-MobiFeed can select at most one message for each category. Finally, because we set the capacity of source edges (i.e., c(s, ri) for every ri∈ Nr) as h, it ensures that when a news feed already contains messages with h distinct categories, scheduling more messages with new categories does not contribute to the optimal value of the flow (i.e.,). For the case (i), if the optimal value of flow (i.e.,) equals (n + 1) × h, all the source edges are saturated (i.e., x(s, ri) = h for each edge (s, ri)). It means that for each of n +1 news feeds, D-MobiFeed could schedule exactly h messages, with each one belonging to a distinct category. Therefore, h-diversity constraint can be satisfied for current news feed query. For the case (ii), if < (n + 1) × h, one or more source edges are not saturated; it corresponds to the fact that at least one news feed cannot satisfy the h-diversity constraint. However, as we exactly solve the maximum flow problem formulated in Section 5.1, the value of is maximized. That is, the largest value of the minimum total diversity for the n + 1 news feeds (Definition 2).

	Message	Category	Relevance score to u
news feed at t_0	m_2	Restaurant	0.58
	m_5	Theater	0.53
	m_{10}	Museum	0.55
news feed at t_1	m_1	Restaurant	0.68
news feed at t_2	m_3	Restaurant	0.1
	m_4	Stadium	0.53
	m_8	Shopping	0.5

Table: Assignment Results of News Feeds

(a) The assignment results of news feeds at t0, t1, and t2 after Stage One. (b) The assignment results of news feeds at t0, t1, and t2 after Stages Two and Three.

$$\begin{aligned}\mathcal{G}(x) &= \sum_{e(v_i,v_j)\in E} w(v_i,v_j)\times x(v_i,v_j)\\ &= \sum_{x(c_j,m_k)=1} w(c_j,m_k)\\ &= \sum_{x(c_j,m_k)=1} (1-\mathit{relevanceScore}(u,m_k))\\ &= \gamma - \sum_{0\le i\le n} \mathit{unweightedRelevanceScore}(f_i),\end{aligned}$$

In Stage One, we have scheduled messages satisfying the minimum total diversity (). However, some news feeds may not be full, and some candidate messages remain unscheduled. In our example, we can schedule one more message for the news feed at t0 after Stage One (Figure 7a). Therefore, in this stage, we select messages from remaining candidate messages, such that each of n + 1 news feeds accommodates (at most) k messages. Keeping the objective of the DCS Problem (Definition 3) in mind, we also maximize the sum of relevance scores of newly selected messages in this stage. Definition 6.Replenish-Up-To-k Problem: Given the results of the -Selection Problem, D-MobiFeed selects messages from remaining candidate messages sets and adds them to n + 1 news feeds, such that (O1) in each news feed, there are at most k messages, and (O2) the sum of relevance scores of newly selected messages is maximized.

VI. EXPERIMENTAL RESULT

Android applications are written in java programming language. Android is available as open source for developers to develop applications which can be further used for selling in android market. There are around 200000 applications developed for android with over 3 billion+ downloads. Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. For software development, Android provides Android SDK (Software development kit). Read more about open source software

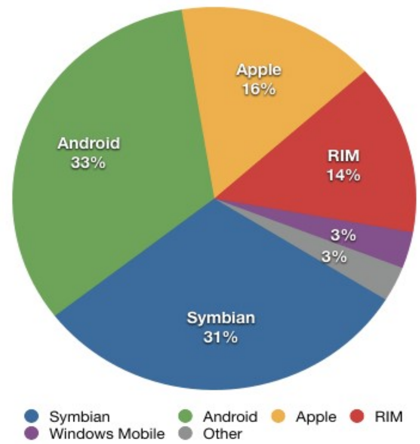

Fig: Operating System

To let you model and test your application more easily, the emulator utilizes Android Virtual Device (AVD) configurations. AVDs let you define certain hardware aspects of your emulated phone and allow you to create many configurations to test many Android platforms and hardware permutations. Once your application is running on the emulator, it can use the services of the Android platform to invoke other applications, access the network, play audio and video, store and retrieve data, notify the user, and render graphical transitions and themes. The emulator also includes a variety of debug capabilities, such as a console from which you can log kernel output, simulate application interrupts (such as arriving SMS messages or phone calls), and simulate latency effects and dropouts on the data network.

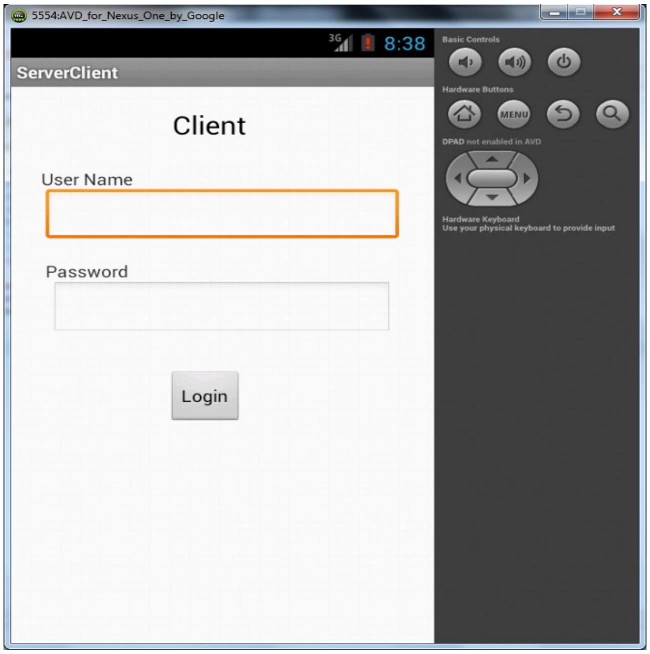

Fig: Login Page

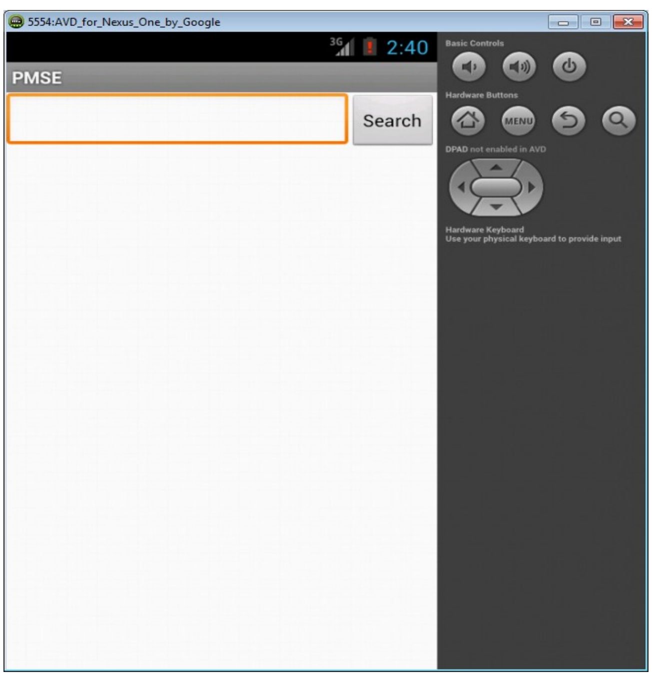

Fig : Search Page

CONCLUSION

In this project, we designed D-MobiFeed; location-aware news feed framework takes the relevance and diversity of news feeds into account when scheduling news feeds for moving users. D MobiFeed users can specify the minimum number of categories in a news feed as an h-diversity constraint, and it aims at maximizing the total relevance of generated news feeds and satisfying the h-diversity constraint. We focus on two key problems in D-MobiFeed, namely, decision and optimization problems. The decision problem is modeled as a maximum flow problem and enables D-MobiFeed to decide whether it can fulfill the h-diversity constraint for a news feed. For the optimization problem, we design an efficient three-stage heuristic algorithm to maximize the total relevance of news feeds under the h-diversity constraint. Experimental results based on a real social network data set crawled from Foursquare and a real road network show that D-MobiFeed can efficiently provide location- and diversity-aware news feeds when maintaining their high quality in terms of relevance. Our future direction is to measure the dissimilarity of pair wise messages in terms of their category information and study a new multi-objective optimization problem of finding a set of news feeds, in which each news feed satisfies the h-diversity constraint and the dissimilarity of the messages in each news feed is maximized while maximizing the total relevance of a set of n+1 news feeds for mobile users (where n is the look-ahead step).

FUTURE SCOPE

In the new era of 2.5G, 3G and 4G, Location Based Services have been recognized as one of the fastest growing areas for novel service provision in the telecommunications sector with great revenue potential. What differentiates them from traditional services is their ability to offer highly personalized, context sensitive and timely information to users anytime anywhere. However they have not matured enough yet in order to provide the so much anticipated 'killer application', mainly due to technical, business and ethical challenges, that have not yet been adequately addressed. All the participants in the LBS provision market should first understand and fix their roles within the value chain, then provide the essential guarantees for protecting user privacy and finally develop new intelligent ways to manipulate and present location information in order to increase user convenience and satisfaction.

REFERENCES

[1] R. A. Baeza-Yates and B. A. Ribeiro-Neto. Modern Information Retrieval. ACM Press/Addison-Wesley, 1999.
[2] J. Bao, M. F. Mokbel, and C.-Y. Chow. GeoFeed: Location-aware news feed system. In IEEE ICDE, 2012.
[3] C.-Y. Chow, J. Bao, and M. F. Mokbel. Towards location-based social networking services. In ACM SIGSPATIAL LBSN, 2010. [4] H. Jeung, M. L. Yiu, X. Zhou, and C. S. Jensen. Path prediction and predictive range querying in road network databases. VLDB Journal, 19(4):585-602, 2010.

[4] B. Chandramouli, J. Yang, P. K. Agarwal, A. Yu, and Y. Zheng. ProSem: scalable wide-area publish/subscribe. In SIGMOD , 2008.
[5] R. Cheng, Y. Xia, S. Prabhakar, an d R. Shah. Change Tolerant Indexing for Constantly Evolving Data. In ICDE , 2005.
[6] Facebook. http://www.facebook.com/.
[7]Facebook Statistics. http://www.facebook.com/press/info.php?statistics, 2010.
[8] FourSquare. http://www.foursquare.com/.
[9] D. Freni, C. R. Vicente, S. Mascetti, C. Bettini, and C. S. Jensen. Preserving location and absence privacy in geo-social networks. In CIKM , 2010.
[10] Google Buzz Moblie. http://www.google.com/buzz.
[11] Google GeoCoding. http://code.google.com/apis/maps/documentation/geocoding/.
[12] iGoogle. http://www.google.com/ig.
[13] A. Khoshgozaran and C. Shahabi. Private Buddy Search: Enabling Private Spatial Queries in Social Networks. In SIN , 2009.
[14] Loopt. http://www.loopt.com.
[15] M. F. Mokbel, X. Xiong, and W. G. Aref. SINA: Scalable Incremental Processing of Continuous Queries I n Spatio-temporal Databases. In SIGMOD , 2004.
[16] My Yahoo! http://my.yahoo.com/.
[17] M. Park, J. Hong, and S. Cho. Location-based recommendation system using bayesian users preference model in mobile devices. UIC , 2007.
[18] PostgreSQL . www.postgresql.org.
[19] S. Prabhakar, Y. Xia, D. V. Kala shnikov, W.G.Aref,andS.E.Ham- brusch. Query Indexing and Velocity Constrained Indexing: Scalable Techniques for Continuous Queries on Moving Objects. IEEE Trans. Computers , 51(10), 2002

BIOGRAPHIES

1.Ms. Sangeetha R., M.phil Research Scholar, Department of computer Science DKM College for Women's (Autonomous), Vellore, TamilNadu, India

2. Mrs. Sangeetha Lakshmi G., Asst. Prof Department of Computer Science DKM College for Women (Autonomous), Vellore, TamilNadu, India