Available online at www.sciencedirect.com
ScienceDirect
Int. J. Human-Computer Studies 68 (2010) 496–507
International Journal of Human-Computer Studies
www.elsevier.com/locate/ijhcs

Personalized blog content recommender system for mobile phone users

Po-Huan Chiu[a], Gloria Yi-Ming Kao[b,*], Chi-Chun Lo[a]

[a]Institute of Information Management, National Chiao Tung University, HsinChu 300, Taiwan, ROC
[b]Graduate School of Technological and Vocational Education, National Taiwan University of Science and Technology, Taipei 106, Taiwan, ROC

Received 30 December 2008; received in revised form 28 February 2010; accepted 25 March 2010
Available online 7 April 2010

Abstract

Compared to newspaper columnists and broadcast media commentators, bloggers do not have organizations actively promoting their content to users; instead, they rely on word-of-mouth or casual visits by web surfers. We believe the WAP Push service feature of mobile phones can help bridge the gap between internet and mobile services, and expand the number of potential blog readers. Since mobile phone screen size is very limited, content providers must be familiar with individual user preferences in order to recommend content that matches narrowly defined personal interests. To help identify popular blog topics, we have created (a) an information retrieval process that clusters blogs into groups based on keyword analyses, and (b) a mobile content recommender system (M-CRS) for calculating user preferences for new blog documents. Here we describe results from a case study involving 20,000 mobile phone users in which we examined the effects of personalized content recommendations. Browsing habits and user histories were recorded and analyzed to determine individual preferences for making content recommendations via the WAP Push feature. The evaluation results of our recommender system indicate significant increases in both blog-related push service click rates and user time spent reading personalized web pages. The process used in this study supports accurate recommendations of personalized mobile content according to user interests. This approach can be applied to other embedded systems with device limitations, since document subject lines are elaborated and more attractive to intended users.

Keywords: Recommender system; Mobile services; Blog; Content push; Recommendation

1. Introduction

In mid-2008, Chunghwa Telecom (Taiwan's largest mobile carrier) started offering a service that allows customers to use their mobile phones to track blog documents. Users can now subscribe to their favorite blogs and have new blog documents delivered to their mobile phones via the WAP Push message feature (Leu et al., 2006). Blog documents are presented in XHTML (to support both text and image displays) and reformatted to fit small screens (Baluja, 2006). Three months after the service was introduced, click rates went flat. Chunghwa identified the reason as too many blogs in the system—over 3000 and rising. Large numbers of new documents are generated everyday, and users cannot read them all (Luther et al., 2008). Filtering out less important or interesting content is clearly a requirement for such a service to succeed.

Current blog document recommendation mechanisms rely on human input. Employees are hired to choose a limited number of documents and to deliver them to all mobile phone users. In the absence of personalized recommendations, most Chunghwa users stop clicking on recommended content. Here we will describe a process for (a) classifying blog documents according to multiple themes, (b) analyzing user mobile phone reading behaviors to determine personal theme preferences (Eirinaki and Vazirgiannis, 2003), and (c) making recommendations for blog documents suitable for individual users. The overall goal is to increase click rates for this service.

1.1. Comparison of personal computer and mobile phone features associated with blog reading

The large majority of web content is aimed at PC surfers rather than mobile phone users, and most blog document

*Corresponding author.
E-mail addresses: gloriakao@cis.nctu.edu.tw, gloriakao@gmail.com (G.Y. Kao).

1071-5819/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijhcs.2010.03.005

Table 1
A comparison of personal computer and mobile phone features associated with blog-reading behaviors.

Comparison items	Personal computers	Mobile phones
Push or pull	Pull mode, user-initiated online browsing	Push mode, users receive WAP Push as content is updated
Document subject descriptions	Allows for greater ambiguity	Must be short and attractive
Ranking documents after reading	Easy ranking using star or point system	Difficult to input, ranking is not easy
Screen size	1024×768 pixels or more	320×240 pixels or less
Browser compatibilities	Internet explorer and Firefox have over 90% market share	Multiple vendors and phone models have different browser features
Cost	Free	Cost per packet
Anonymous browsing	Mostly anonymous	Phone numbers can be read
Suitable content	Ajax, Flash, video, rich content	Simple text or small images
Browsing duration	Mostly exceeding 10 min	Mostly less than 10 min

recommendation systems are customized for PC access (Hayes et al., 2006). A comparison of accessing methods is presented in Table 1. In this section we will review their respective characteristics.

1.1.1. Blog-reading behaviors

Push or pull: PC browsing behavior is almost completely user-initiated. Thus, when new blog documents are posted, users need to make a focused effort to go online and visit the blogger's web page. Mobile phones differ in that content providers can actively push updated documents to user devices. In Taiwan, mobile phone WAP Push technology can deliver approximately 100 Chinese characters of summary, plus a URL link. If users want to read the content, all they need to do is click on the link, thus saving them the effort of visiting the website on a regular basis.

Document subject descriptions: PC users are more likely to read blog posts, even on topics that they have little interest in. Web page designers usually give blog subject titles and document summaries on one side of the screen area, making it easy for users to decide whether or not they want to read specific entries. On mobile phones in Taiwan, WAP Push messages are generally limited to 100 Chinese characters; they must be shorter if the carrier wants to present a list of multiple blog document subjects. The writing and rewriting of summaries requires human effort.

Ranking documents after reading: PC users are accustomed to simple and useful blog document ranking systems involving stars or other marks. Ajax technology takes advantage of keyboard and mouse functions to speed up page refreshing actions for ranking. Doing this on mobile phones requires GUI design decisions that can produce more problems than benefits. A possible GUI compromise is to offer two close buttons: "close + I liked it" and "close + I did not like it".

Location-based service: Users occasionally want content related to their immediate locations—for example, reading comments about a feature film when they are approaching a theater. An advanced mobile recommender system can make recommendations based on user location information.

Table 2
Single day click counts for different device vendors.

Device vendor	Click times (users could click multiple times)
Sony Ericsson	15,449
Samsung	1175
Nokia	1276
PocketPC (Windows Mobile)	547
LG	395
Motorola	392

1.1.2. Device limitations

Screen size: Unlike PC screens, mobile phones have very small browsing areas. When blog document subjects are shown in list form on a mobile phone screen, items near the top get the best click rates—the more a user has to scroll down, the smaller the chances of an item being selected. PC users are more likely to browse lists and to skip over individual items to locate content they are most interested in.

Browser compatibilities: Internet explorer and Firefox currently dominate the web browser market (over 90% combined market share). In contrast, the list of mobile phone browsers includes Nokia, Sony Ericsson, Motorola, and Windows Mobile, among others. To match mobile phone capabilities, web servers need to identify browser type and version in order to provide suitable content (Chen and Kotz, 2000). It is possible to read browser type information from HTTP headers. We calculated click counts for different browser types over one day; our results are shown in Table 2.

The users of Sony Ericsson mobile phones, which have NetFront browsers developed by the ACCESS Company of Japan, were clearly the most active in this research. In many respects, NetFront browsers are incompatible with PC web browsers—for example, they have limited JavaScript capabilities, HTML support, and content presentation styles. We were therefore required to convert various blog content formats into plain text and small images. Character sets were also challenging: mobile phones can only accept UTF-8 encoding, and blog documents may contain mixes of Big5, GB2312, and ISO-2022-JP characters. Blog documents may contain illegal characters or coding mixes that are resolvable with an internet explorer (IE) browser; all of these must be deleted before sending documents to mobile phones.

1.1.3. General considerations

Cost: In Taiwan, most PC users access the web via ADSL, with Internet service providers charging monthly fees according to bandwidth. In contrast, mobile phone companies usually charge by traffic amount, with users paying for GPRS/3G wireless bandwidth according to transmission packets.

Anonymous browsing: Using a PC to visit web pages is mostly an anonymous activity. Although web servers can record the IP addresses of incoming connections (Srivastava et al., 2000), user identities are protected by firewalls and proxies (Reed et al., 1998). Since most blogs do not require logins, user preferences are not easy to record. Conversely, a mobile phone carrier's backend system can be used to retrieve user phone numbers in the HTTP header while they are browsing, making it easy to store personal browsing histories in databases. This provides sufficient information for a blog content recommender system based on analyses of user preferences. In a backend system, carriers use the dynamic IP addresses of mobile phones to retrieve phone numbers, which are inserted into HTTP headers via WAP gateways. Service providers can request permission from mobile carriers to read phone numbers from HTTP headers: when users browse web pages via a WAP gateway, they can extract phone numbers from web servers.

Suitable content: Internet bloggers use various content formats, including HTML tags, Ajax, Flash, and video. The formats of most original blog documents and web page layouts are not suitable for mobile phones, which cannot parse complex web pages. Although browsers on advanced mobile phones are becoming more powerful, extracting text and images from blogs and reformatting them for small screens is required to achieve maximum compatibility.

Browsing duration: Whereas PC users can surf the web for many hours at a time, mobile phone users are accustomed to reading very short documents. Reading long documents on small mobile phone screens is unusual, since it causes physical eyestrain and conflicts with the typical short-term tasks that mobile phones are used to perform.

1.2. Mobile phone blog recommendations by human experts

Prior to implementing a personalized recommender system for mobile phones, blog recommendations for Chunghwa Telecom customers were made entirely by human experts, who chose blog documents in blocks of twelve and used the WAP Push function to deliver messages to users. The experts were responsible for finding the latest blog documents of high interest and suitability for mobile phone users, rewriting their subject lines to make them shorter and more attractive, and distributing their recommendations without personalization. The subject appearance order they chose was important because only the first five list items could be viewed. Users then decided whether or not they wanted to click a URL link containing the WAP Push message, activate their mobile phone browsers and go online, browse the entire list of twelve blog topics in XHTML format, and get more information about one or more of the listed documents.

Human experts obviously do not have enough time to make personalized recommendations for documents for every mobile user. In some cases, mobile phone companies or independent researchers have experimented with systems focused on specific content recommendations—for example, news media headlines (Lee and Park, 2007). We believe a similar system can be used to recommend specific blog content.

2. Design

2.1. System goals

We suggest using a mobile content recommendation system (M-CRS) to achieve our stated objectives. Our proposed system consists of four elements:

1. Creating groups of users with similar preferences, and pushing blog content according to those preferences and user interests.
2. Adjusting recommendation accuracy according to user feedback and browsing histories (Chakrabarti, 2002; Smyth and Cotter, 2004). The larger the number of users involved in the recommendation process, the easier it will be to use collaborative filtering to make more accurate recommendations (Goldberg et al., 1992; Morita and Shinoda, 1994; Resnick et al., 1994).
3. Using the mobile carrier's backend system to determine user preferences for blog documents.
4. Limiting push messages to those users most likely to respond. This is especially important because the WAP Push system is considered a limited carrier resource that is used for other purposes. Our M-CRS can be used to identify users with no interest in reading blog documents on their mobile phones; these can be deleted from WAP Push message delivery lists.

For successful implementation, M-CRS must be capable of analyzing approximately 3000 blog sites and calculating the preferences of 20,000 users within a 2 h limit. Requirements for this task are a solid load balance architecture, fast algorithms, and appropriate cache technologies to match mobile carrier environment needs. Finally, our proposed system requires periodic click rate comparisons with human expert recommendations. A high-level M-CRS workflow diagram is shown in Fig. 1.

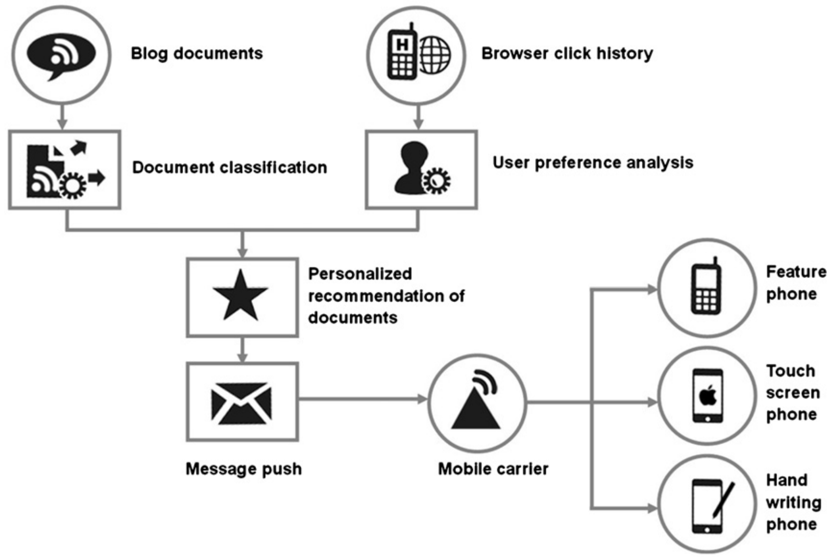

Fig. 1. M-CRS high-level workflow.

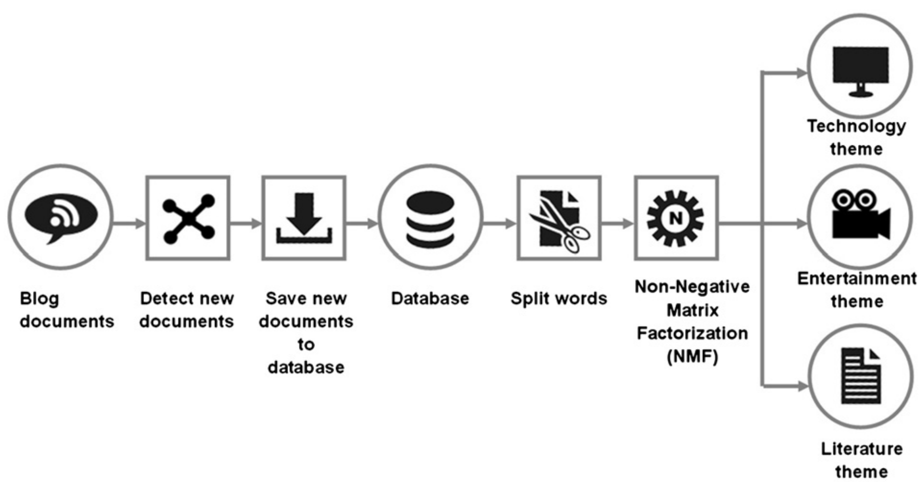

Fig. 2. Document classification steps.

2.2. Document classification

Making suitable recommendations for mobile phone users requires categorizing thousands of new documents into themes that can be matched with user preferences. Blog sites generally contain multiple documents on various themes, thus making one document the smallest possible theme unit. Our proposed system entails three document classification steps: detecting new documents and saving them in a database, splitting words, and matrix factorization (Fig. 2).

2.2.1. Detecting new documents and saving them in a database

New blog posts can be detected using RSS feeds. Since they are structured in XML data format, our proposed M-CRS can easily parse them to identify new items. M-CRS can be set to scan all registered blog URLs once per hour, with new posts launching a retrieval process. RSS feeds usually contain partial content or basic summaries; M-CRS must fetch complete documents in unstructured HTML format and purge irrelevant tags, Java scripts, network ads, and so on before parsing usable fields such as subject, author, content, and posting date. The parsing task is made more difficult by the various HTML layout formats used by bloggers. Parsed and structured information is stored in a database for later use. To improve server efficiency, many blog websites block software robots from fetching HTML content. For this reason, M-CRS must present itself as a normal IE browser and simulate all IE protocols in order to gain website access.

2.2.2. Split words

Since Chinese words in sentences are not separated by white spaces, they are more difficult to split than English words (Salton and Buckley, 1988). A Taiwanese word-splitting project called Chinese Knowledge Information Processing (CKIP) (Chien, 1997) provides HTTP web service interfaces to developers. After registering on the CKIP website, developers can send documents to the system via HTTP protocols for the purpose of splitting words. We feel that network overhead is a very serious CKIP performance issue: at normal network speeds, the system requires approximately 8 s to process a single document. This means that 3000 documents will require more than 6 h for processing—an unrealistic amount of time for a mobile carrier environment. We therefore decided to use our own Chinese word-splitting algorithm on a local computer to reduce these processing costs.

Chinese words can consist of two or more characters. To train computers to find meaningful Chinese words, we assume that certain combinations of characters occur more frequently and have more potential to represent meaning. Traditionally, Chinese dictionaries have been used to split phrases into meaningful word combinations. We decided against using this approach because of the large number of new words that have yet to appear in Chinese dictionaries. Instead, we propose using a hash table-based algorithm to split words on a large scale for multiple blog documents.

Our proposed process involves three algorithms: gram count, log likelihood ratio (LLR), and term frequency-inverse document frequency (TF-IDF) (Jurafsky and Martin, 2008). Gram count is based on finding character combinations. For example, a Chinese phrase may be split into 1-ideograph, 2-ideograph, etc. combinations. The frequency of each ideograph is calculated in terms of its number of appearances in a document. Split Chinese characters are inserted into hash tables as the keys while frequencies as the values. Next, the LLR algorithm is used to determine which 2-ideograph Chinese character combinations are meaningful, based on whether its LLR value exceeds a threshold. The TF-IDF is used to evaluate the importance of a word to a document in a corpus. (Salton and McGill, 1983; Shardanand and Maes, 1995).

2.2.3. Non-negative matrix factorization algorithm

Lee and Seung's (1999) non-negative matrix factorization (NMF) algorithm has a strong performance reputation for problems such as determining facial features from photograph collections. An article matrix consists of one row for each article and one column for each keyword. To factorize such a matrix, the algorithm finds two smaller matrices that can be multiplied to reconstruct the original. It attempts to reconstruct the original as accurately as possible by calculating its features and weights (Shahnaz et al., 2006). Our proposed system uses the NMF algorithm to categorize blog documents according to theme and to generate theme-based keyword lists. It is capable of identifying multiple themes across all blog documents. Higher relation scores are given to documents belonging to the same theme.

The goal of the NMF algorithm is to find two smaller feature and weight matrices that can be multiplied together to construct a large article matrix. The features matrix shown in Table 3 has one row for each feature and one column for each word; values indicate how important each word is to a feature. Each feature can represent a theme emerging from a set of articles. The weights matrix maps features to an articles matrix. Each row is one article and each column one feature. Values represent how relevant a feature is to an article. A features matrix has one column for every word; each row contains a list of word weights. Since each row is a feature consisting of a combination of words, reconstructing an articles matrix is a matter of combining rows in different amounts. The weights matrix example in Table 4 has one column for every feature and one row for every article.

Table 3
Features matrix sample.

	Keyword 1	Keyword 2	Keyword 3	Keyword 4
Feature 1	5	1	7	0
Feature 2	1	0	0	3
Feature 3	0	2	0	1

Table 4
Weights matrix sample.

	Feature 1	Feature 2	Feature 3
Article 1	12	0	0
Article 2	0	10	3
Article 3	2	5	0

If the number of features equals the number of articles, the best situation is to have one feature that perfectly matches each article. However, the purpose of matrix factorization is to reduce large sets to smaller sets that capture their most common features. Ideally, a smaller set can be combined with different weights to reproduce the original dataset. This is very unlikely in practice, therefore the algorithm aims at reproducing the original dataset as closely as possible.

2.3. User preference analysis

Our proposed system uses browsing histories as raw data to determine user interest scores for individual themes, and a collaborative filtering algorithm to predict interest scores for new themes. Collaborative filtering algorithms examine large groups of individuals, identify sets of people with similar tastes, and create ranked lists of suggestions. The process consists of the four steps shown in Fig. 3. In the first step, when a mobile phone user reads a blog document, M-CRS records the event in a database consisting of three fields: mobile phone number, browsing time, and blog document URL. To determine user interest in a keyword, it is necessary to count the number of times the word appears in one or more documents.

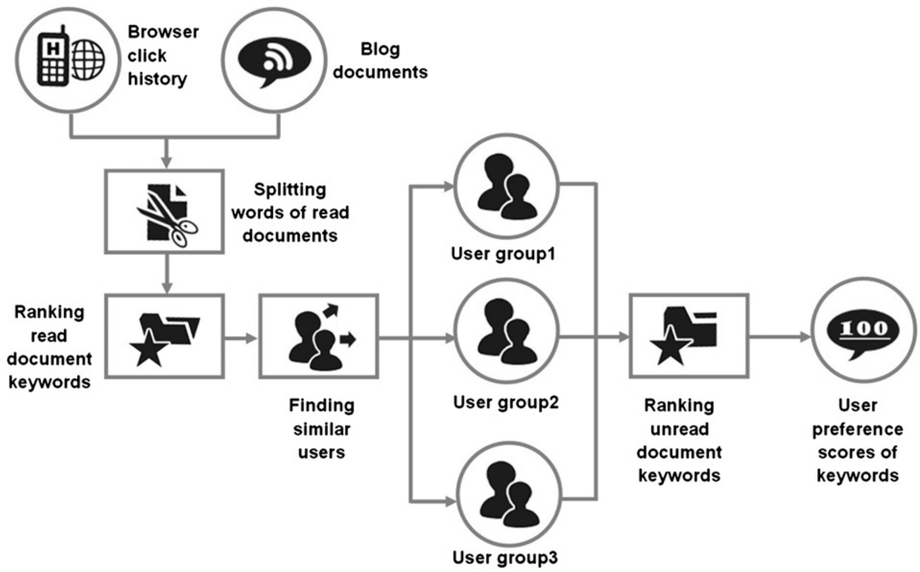

Fig. 3. User preference analysis steps.

Table 5
Fictitious user's preference scores for five keywords.

Keyword	Score
Fresh	10
Flavor	6
Soup	4
Dish	3
Noodle	2

In the second step, scores are given to words that appear frequently in user-read documents; the data are used to construct user-specific keyword interest score tables. A word-splitting process is used to determine keyword frequencies in user-read documents. Longer documents usually contain more keywords. To determine user interest, the TF-IDF algorithm calculates keyword frequencies in browsing histories. A high frequency is considered a sign of interest, but the algorithm ignores excessively high frequencies that are usually attached to such common words as prepositions, definite articles, or different forms of the "be" verb.

In step three, interest score tables are used to perform Pearson correlation analyses (which produce better results for data that are not well normalized). Last, grouping individuals according to similar interests makes it possible to predict keyword interest scores for unread documents, based on the assumption that two users in the same group will be interested in the same topics and documents. We used a hierarchical clustering algorithm (Segaran, 2007) to construct user groups.

A theme is a keyword representing a document category. To determine user interests in specific themes, the recommender system uses browsing histories to calculate keyword appearance frequencies in read documents. After ranking read document keywords, the system produces preference scores. Table 5 shows preference scores for five keywords for an imaginary user. For a theme T containing the keywords "drink", "fresh", "summer", "soup", and "noodle", the imaginary user's interest score is $0+10+0+4+2=16$. A CF approach is required to determine preference scores for keywords in documents that have not yet been read. For example, zero interest was expressed for the keyword "drink". If a second user with similar browsing characteristics has a preference score of 7 for that word, then it can be added to the first user's preference table with score of less than 7 points, suggesting that the two users may share an interest in the same keyword, but at different strengths.

2.4. *Personalized document recommendations*

After M-CRS completes its categorizing task, individual themes will contain many more items than most mobile phone users want to deal with. Therefore, the next task is to recommend three documents that most closely match user interest in each theme according to the steps shown in Fig. 4. We have five reasons for using human experts for document selection instead of methods such as page ranking or link click rates (Baeza-Yates and Castillo, 2001):

1. Human experts are better able to determine document meaning and quality. Some documents attract visitors for non-quality reasons (e.g., extreme views or exaggerations). Also, page-ranking algorithms can be fooled by cross-posting robot software. We believe that the extra time required for human input is reasonable when the task is limited to making three recommendations.
2. Humans can reflect perspectives that differ from those of the mainstream media. Human experts can monitor a short-term "hot" topic and locate unique opinions on that topic. Such documents may not have high click

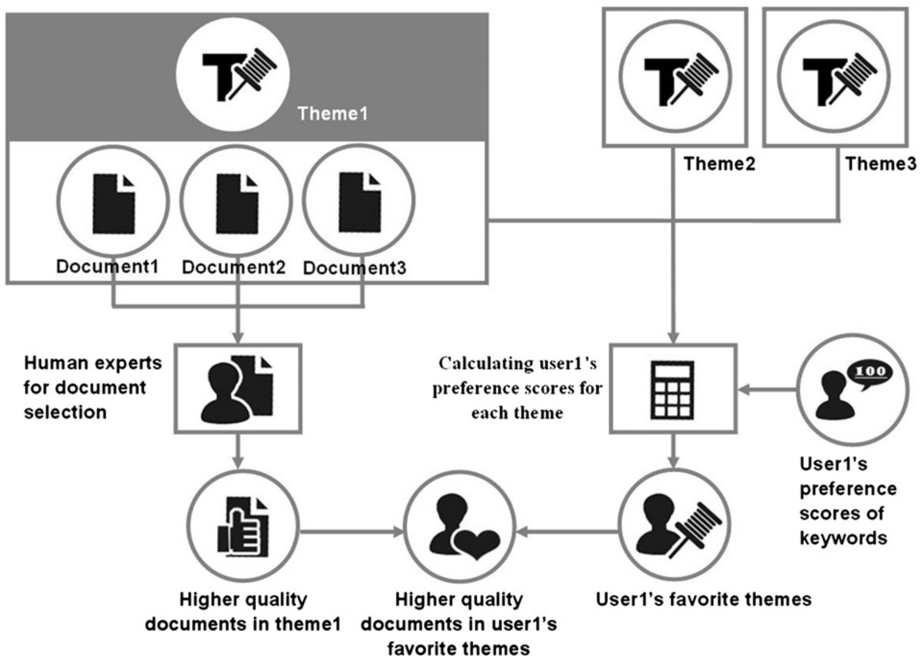

Fig. 4. Personalized document recommendation steps.

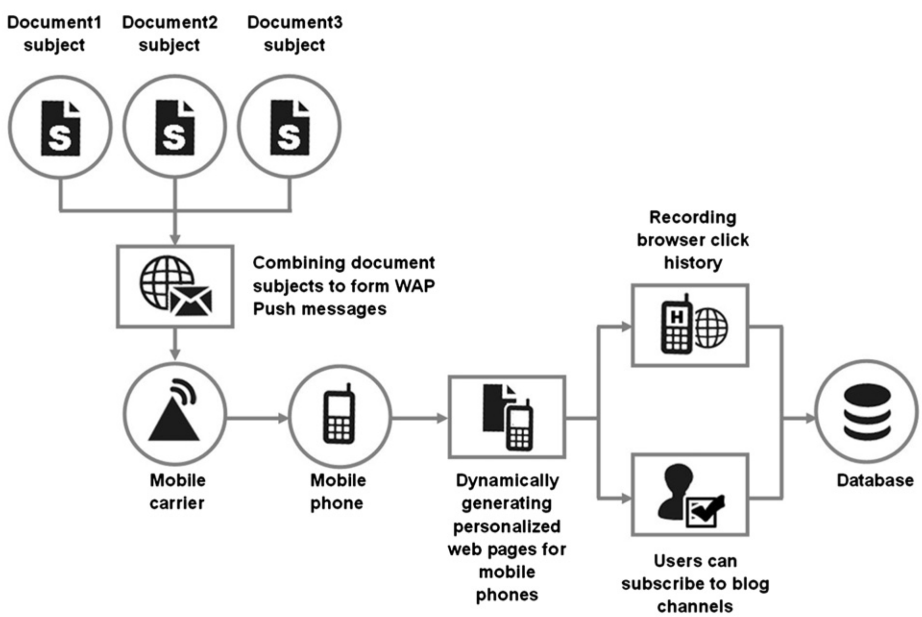

Fig. 5. Message push steps.

rates or page rank scores, yet still represent opinions that should be heard.

3. As stated earlier, the small size of mobile phone screens requires more succinct writing—that is, the key ideas of 5 subjects must be expressed in 100 or fewer Chinese characters. This task requires the skills of a newspaper headline writer.
4. Web pages containing Ajax, Flash, and videos (e.g., blogs with one line of text and a link to YouTube) are not suitable for mobile phone use. A human expert is needed to filter out images that cannot be recognized by mobile phone browsers after resizing.
5. Human experts are better equipped to analyze blog documents according to such intangible qualities as "fun".

In order to select quality documents for individual themes, human experts need to review an average of approximately 900 documents, a task requiring approximately 8 h for one individual. Since WAP Push messages are sent to users 3 times per week, that equals 24 h of effort to select high quality documents.

2.5. Message pushing

For mobile phones, the three most commonly used message-pushing methods are short message service (SMS), multimedia messaging service (MMS), and WAP Push. SMS supports the delivery of 70 Chinese characters without images, and MMS supports the delivery of approximately 150 KB of mixed text, images, audio, and video data. Neither system is capable of recording browsing behavior. As stated earlier, WAP Push supports 100 Chinese characters and a URL link for online browsing. Its primary disadvantage is its slow GPRS transmission speed—a full web page requires up to 35 s to load. Since browsing speed is much better on 3G mobile phones, we chose WAP Push for message delivery (Fig. 5).

The 100-character limitation means that only 5 of 12 document recommendations can be shown at the same time. Since blog authors generally do not consider mobile phone screen issues, we feel it is best to use human experts to rewrite document subject lines for WAP Push messages (Fig. 6). As explained in an earlier section, item order significantly affects user reading behavior—an idea that must be kept in mind when estimating user interest in subjects, themes, and documents (Fig. 7). It also underscores the need to program web servers to record mobile phone user click behaviors plus data on phone number, click time, device model, browser type, and URL address. Finally, note that our proposed M-CRS supports blog channel subscriptions, monitoring for new posts, the transformation of documents to mobile phone-readable XHTML format, and user notification using the WAP Push feature.

3. M-CRS architecture and components

3.1. M-CRS components

There are six important elements in the M-CRS architecture (Fig. 8):

1. *Data sources*: Content is gathered from over 3000 blog sites. M-CRS users can add their favorite RSS or ATOM feeds as a means of continuously updating lists of high-traffic blog sites from web portals.
2. *Mobile carrier*: Chunghwa Telecom. PC server: two IBM 1U servers (to achieve load balance). WAP Push messages

Fig. 6. A WAP Push message on a mobile phone screen.

Fig. 7. A subject list on a mobile phone screen.

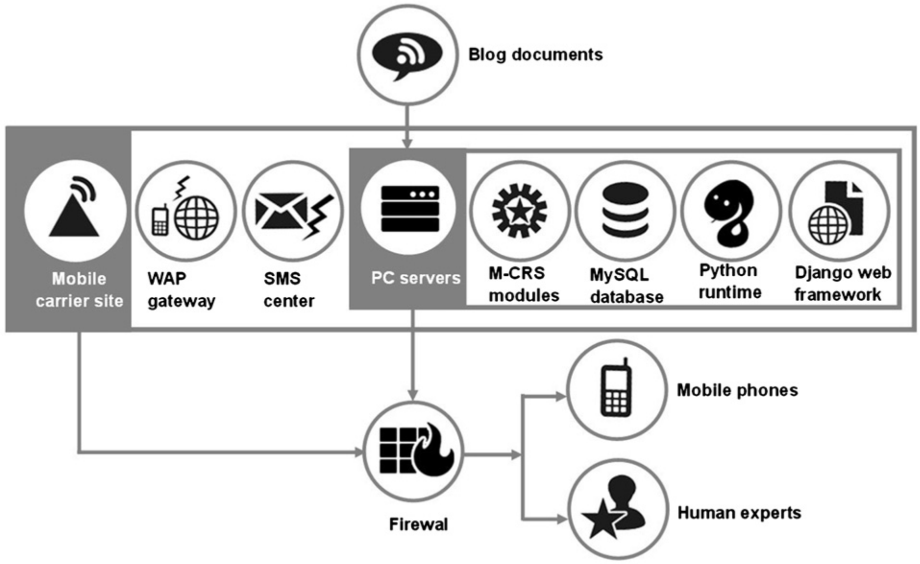

Fig. 8. M-CRS architecture.

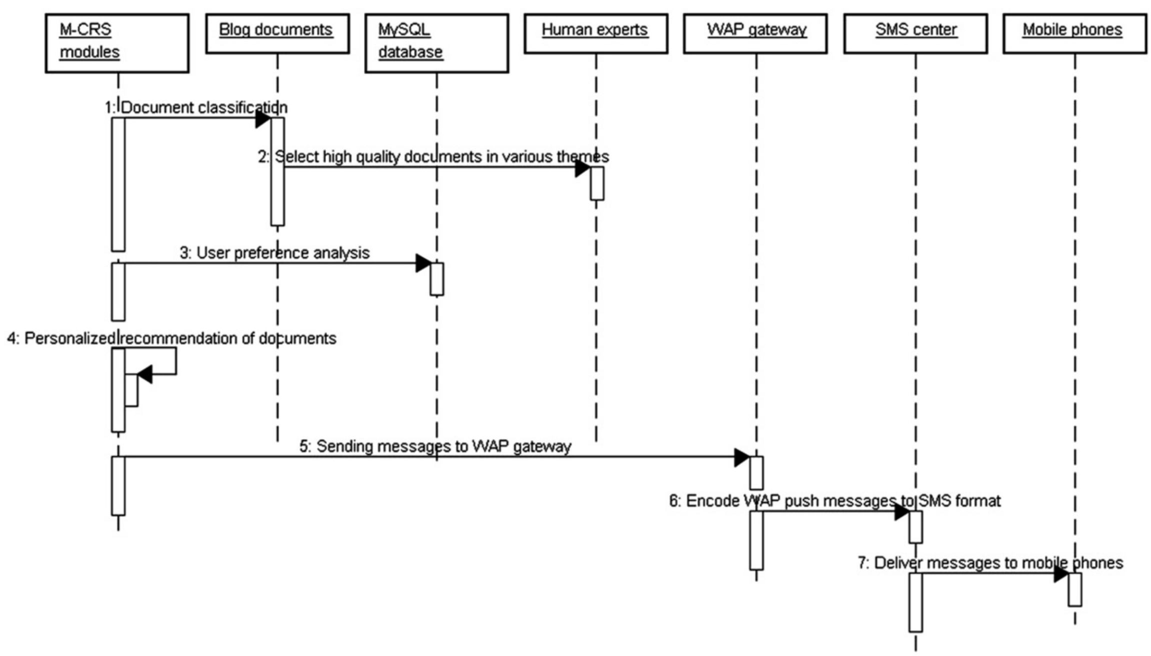

Fig. 9. Sequence diagram of steps for sending a WAP Push message from a backend system to mobile phones.

are transformed into GSM SMS format and sent to mobile phones via a SMS center. A WAP gateway is used to compress and encode web pages for mobile phone screens.
3. *Core modules*: M-CRS modules must be set up in the mobile carrier's IDC center. A sequence diagram is presented in Fig. 9.
4. *Development environments*: A Django web framework is used as an application server to provide dynamic content. MySQL, an open source database, is used for backend storage (Widenius et al., 2002). Python is used as a development programming language.
5. *Devices*: M-CRS supports over 90% of all mobile phone models sold by vendors in Taiwan, including Feature phone, Symbian, Windows Mobile, and Linux phone.
6. *Human resources*: Used to select high quality documents in each theme.

3.2. M-CRS sequence diagram

Input/output values for important components are described in Table 6.

Table 6
Input/output values for the M-CRS WAP Push process.

Step	Input	Output
Document classification	*M-CRS input*: Blog documents from the internet	*M-CRS output*: Classified blog documents across different themes
Select high quality documents across themes	*Human expert input*: Thematically classified blog documents	*Human expert output*: High quality documents across various themes
User preference analysis	*M-CRS input*: User browsing history (including themes)	*M-CRS output*: User preference scores for individual themes
Personalized document recommendations	*M-CRS input*: User preference scores for individual themes and high quality documents sorted by theme	*M-CRS output*: Personalized document recommendations
Send messages to WAP gateway	*WAP gateway input*: Mobile phone numbers, text messages for users, and an URL link to the recommended documents list	*WAP gateway output*: Encoded WAP Push message in binary format
Encode WAP Push messages into SMS format	*SMS center input*: Binary WAP Push messages, encoded into three SMS format messages	*SMS center output*: Results for sending messages to mobile phones (successful or failed)
Deliver messages to mobile phones	*Mobile phone input*: WAP Push messages encoded in SMS format	*Mobile phone output*: Text message and clickable URL link to launch browser

4. Case study

Our experiment involved 20,000 mobile phone users who were either taking part in a 30-day free trial period or already subscribed to the blog delivery service. Chunghwa Telecom launched several marketing events to promote the service; distributed advertisements via PC web portals, mobile phone portals, and with billing statements; and sent direct mail advertisements. Blog categories included technology, food, photos, pets, music, travel, medicine, games, literature, art, sports, movies, finance, celebrities, families, fortunes, jobs, school, knowledge, news, and beauty. The reading behaviors of existing subscribers were recorded by our proposed system, and the data were used to determine personal preferences. New users without browsing histories were given non-personalized content.

We were concerned about annoying our users with too many messages sent over a short period of time, but we had to balance that with our goal of gathering sufficient results within two weeks. As a compromise, we used the WAP Push system to send five messages to subscribers without using M-CRS—in other words, during these five runs, all recommendations were made by human experts without personalization. Users received the same list of topics based on the human experts' observations of current trends. In parallel, the M-CRS system was used to personalize recommendations distributed via WAP Push for another groups of users (also five runs). Each run involved 2000 users for a total of 20,000 users. Our data has shown:

- Number of blogs: 2945.
- Maximum, minimum, and average number of browsing history records per user: 242, 16, and 23.95, respectively.
- Maximum, minimum, and average numbers of themes for each run: 247, 63, and 132, respectively.
- Maximum, minimum, and average numbers of documents in all themes for each run: 1,013, 728, and 905, respectively.
- Average number of user groups in each run: 169.

Click count data for the 5 runs under the 2 conditions are shown in Table 7. Click count rates for the M-CRS condition were higher at statistically significant levels in 3 of the 5 runs. We separated our data on "time spent browsing web pages" into 8 categories (Table 8). The non-personalized row presents the percentages of users reading non-personalized blog articles, and the personalized row shows the percentages of users spending specific lengths of time reading a recommended document. We assume that the stronger the user interest in recommended articles, the greater the time spent reading personalized web pages.

To make improvements in the blog pushing service, we sent questionnaires to 1200 randomly selected users in the form of WAP Push messages. Users could respond to certain items (e.g., cost) but ignore other parts of the questionnaire. We received 86 usable responses; results are presented in Table 9.

Current issues of concern that we believe will become less important in the near future include:

- Bandwidth problems, due to the number of new 3G wireless networks in Taiwan. We believe users in the near future will be more willing to use mobile phones to access the internet.
- The growing popularity of large-screen mobile phones such as iPhones, which support more complex JavaScript syntaxes.
- The appearance of touch screens, which will help solve input problems such as ranking documents and providing feedback.

Two issues we believe will remain for several years are

- Costs tied to sending WAP Push messages on a large scale. To reduce WAP Push expenses, independent service providers will be limited to using pull-based mobile services.
- Personalized content recommendations will require more precise algorithms to predict user preferences.

Table 7
Results from t-tests for click count comparisons using/not using M-CRS.

Experiment run	M-CRS		Human expert		t	p
	Mean	Standard deviation	Mean	Standard deviation		
1	1.35	2.27	0.81	1.19	2.040	< 0.05
2	1.29	2.28	1.00	1.32	1.089	ns
3	1.24	2.37	0.76	1.21	1.772	< 0.05
4	1.46	2.49	0.40	0.83	3.952	< 0.001
5	1.11	2.18	0.87	1.31	0.888	ns

Table 8
User time spent browsing web pages.

Categories (minute spent) (%)	**0–1 (%)**	**1–5 (%)**	**5–10 (%)**	**10–20 (%)**	**20–30 (%)**	**30–40 (%)**	**40–50 (%)**	**50–60 (%)**
Non-personalized	29.0	39.6	10.9	10.7	4.5	2.3	2.1	0.9
Personalized	18.4	36.5	14.8	17.4	4.9	4.3	1.6	2.0

Table 9
Questionnaire responses for improving blog-pushing service.

Service weakness	**Numbers of users**
I think the service fee is too high (about USD2.50 per month)	26
It is hard to read content on mobile phones	24
I cannot find blog content that I want to read on a mobile phone	16
I received too many WAP Push messages in a week	12
I did not receive the latest content in real time. (The service looked at all blog websites in the system 12 times a day to detect new documents.)	8

5. Future work

This study was limited to analyzing user preferences based on mobile phone browsing histories. We did not address keyword search behaviors that can be used to enhance user preference parameter accuracy. We also did not address differences between heavy and light users, including the need to provide large numbers of more detailed single-theme blog documents for heavy users. Light users may prefer a broader range of documents across multiple themes. Such information can support M-CRS parameter refinement.

Our plans include experimenting with techniques for determining document quality without human input; for deleting documents that are too long, that contain videos, or that have excessive graphics; and for finding keywords that match "hot" topics. We hope to reference blog document click counts on websites to determine popularity, but this will require overcoming firewall restrictions on web bots. Such efforts may benefit from Li and Chen's (2009) work with neural networks, trust models, social relations, and semantic analyses for blog ranking.

6. Conclusion

In this paper we described our proposed M-CRS system for recommending personalized blog documents for mobile phone users, discussed differences in blog reading habits between PC and mobile phone users, and described difficulties tied to designing recommender systems, especially for mobile phone carriers. We also explained our rationale for using a mix of an automated blog retrieval system and human input to provide mobile phone users with more narrowly defined and interesting content. By categorizing blog documents according to different themes and analyzing mobile phone users' browsing histories, it is possible to increase the odds of sending blog documents of strong interest to individual consumers. With the support of a mobile carrier, we were able to test our M-CRS system with over 20,000 users in an actual commercial environment. Our findings indicate that the M-CRS system triggered increases in click rates.

References

Baeza-Yates, R., Castillo, C., 2001. Relating web characteristics with link based web page ranking. In: Proceedings of SPIRE 2001, IEEE CS Press, Laguna San Rafael, Chile, pp. 21–32.

Baluja, S., 2006. Browsing on small screens: recasting web-page segmentation into an efficient machine learning framework. In: Proceedings of the 15th International Conference on World Wide Web, Edinburgh, UK, pp. 33–42.

Chakrabarti, S., 2002. Mining the Web: Analysis of Hypertext and Semi Structured Data. Morgan Kaufmann, New York, NY.

Chen, G., Kotz, D., 2000. A survey of context-aware mobile computing research, Dartmouth Computer Science Technical Report TR2000–381, Hanover, NH.

Chien, L.F., 1997. PAT-tree-based keyword extraction for Chinese information retrieval. In: Proceedings of the 1997 ACM SIGIR, Philadelphia, PA, USA, pp. 50–58.

Eirinaki, M., Vazirgiannis, M., 2003. Web mining for web personalization. ACM Transactions on Internet Technologies (ACM TOIT) 3 (1), 1–27.

Goldberg, D., Nichols, D., Oki, B.M., Terry, D., 1992. Using collaborative filtering to weave an information tapestry. Communications of the ACM 35 (12), 61–70.

Hayes, C., Avesani, P., Veeramachaneni, S., 2006. An analysis of bloggers and topics for a blog recommender system. In: Workshop on Web Mining (WebMine), Berlin, Germany.

Jurafsky, D., Martin, J.H., 2008. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice Hall, Upper Saddle River, NJ.

Lee, D.D., Seung, H.S., 1999. Learning the parts of objects by nonnegative matrix factorization. Nature 401, 788–791.

Lee, H.J., Park, S.J., 2007. MONERS: a news recommender for the mobile web. Expert Systems with Applications 32 (1), 143–150.

Leu, J.S., Chi, Y.P., Shih, W.K., 2006. Design and implementation of blog rendering and accessing instantly system (BRAINS). Journal of Network and Computer Applications 30 (1), 296–307.

Li, Y.M., Chen, C.W., 2009. A synthetical approach for blog recommendation: combining trust, social relation, and semantic analysis. Expert Systems with Applications 36 (3), 6536–6547.

Luther, M., Fukazawa, Y., Wagner, M., Kurakake, S., 2008. Situational reasoning for task-oriented mobile service recommendation. The Knowledge Engineering Review 23 (1), 7–19.

Morita, M., Shinoda, Y., 1994. Information filtering based on user behaviour analysis and best match text retrieval. In: Proceedings of the 17th ACM Annual International Conference on Research and Development in Information Retrieval (SIGIR'94), Dublin, Springer-Verlag, Ireland, pp. 272–81.

Reed, M.G., Syverson, P.F., Goldschlag, D.M., 1998. Anonymous connections and onion routing. IEEE Journal on Selected Areas in Communications 16 (4), 482–494.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J., 1994. GroupLens: An open architecture for collaborative filtering of netnews. In: CSCW'94: Proceedings of the Conference on Computer Supported Cooperative Work, New York, NY, pp. 175–186.

Salton, G., McGill, M.J., 1983. Introduction to Modern Information Retrieval. McGraw-Hill, New York, NY.

Salton, G., Buckley, C., 1988. Term-weighting approaches in automatic text retrieval. Information Processing and Management 24 (5), 513–523.

Segaran, T., 2007. Programming Collective Intelligence: Building Smart Web 2.0 Applications. O'Reilly Media, Inc., Sebastopol, CA.

Shahnaz, F., Berry, M.W., Pauca, V.P., Plemmons, R.J., 2006. Document clustering using nonnegative matrix factorization. Information Processing & Management 42 (2), 373–386.

Shardanand, U., Maes, P., 1995. Social information filtering: Algorithms for automating ''word of mouth.'' In: Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems, Denver, CO, pp. 210–217.

Smyth, B., Cotter, P., 2004. MP3—mobile portals, profiles and personalization. In: Leveneand, M., Poulovassilis, A. (Eds.), Web Dynamics Adapting to Change in Content, Size, Topology and Use. Springer SMB, London, pp. 411–434.

Srivastava, J., Cooley, R., Deshpande, M., Tan, P.N., 2000. Web usage mining: discovery and applications of usage patterns from web data. SIGKDD Explorations 1 (2), 12–23.

Widenius, M., Axmark, D., DuBois, P., 2002. Mysql Reference Manual. O'Reilly & Associates, Inc., Sebastopol, CA, USA.