Tailored News in the Palm of Your HAND: A Multi-Perspective Transparent Approach to News Recommendation

Mozhgan Tavakolifard[1], Jon Atle Gulla[1], Kevin C. Almeroth[2], Jon Espen Ingvaldsen[3],
Gaute Nygreen[4], Erik Berg[4]

[1]Norwegian University of Science and Technology, {mozhgan, jag}@idi.ntnu.no
[2]University of California, Santa Barbara, almeroth@cs.ucsb.edu
[3]Kantega AS., Ingvaldsen@kantega.no
[4]Telenor Research Center, {gaute, erik.berg}@telenor.com *

ABSTRACT

Mobile news recommender systems help users retrieve news that is relevant in their particular context and and can be presented in ways that require minimal user interaction. In spite of the availability of contextual information about mobile users, though, current mobile news applications employ rather simple strategies for news recommendation. Our multi-perspective approach unifies temporal, locational, and preferential information to provide a more fine-grained recommendation strategy. This demo paper presents the implementation of our solution to efficiently recommend specific news articles from a large corpus of newly-published press releases in a way that closely matches a reader's reading preferences.

Categories and Subject Descriptors

H.4.7 [**Information Systems Applications**]: Communications Applications—*Information browsers*

General Terms

Design, Experimentation, Human Factors

Keywords

Recommender system, news, mobile, user modeling

1. INTRODUCTION

Because keyword-based searches and browsing on mobile devices inconvenient, there is a strong need for personalization in order to recommend "relevant" and "well-timed" news content to users. More specifically, given a set of users and a large and growing set of articles, what are the ways that a ranked set of articles can be recommended to a user.

News article recommendation differs in several way from other well-known types of recommender systems such as for music and movies. The main differences are: (i) freshness is sometimes more important than relevancy, (ii) similarity between news articles does not necessarily mean they are related–unrelated news articles might share many words, (iii) the unstructured format of a news story is more difficult to analyze than other objects with structured properties such as a friends network, (iv) news readers might have special preference on some particular events contained in news articles, which are also called named entities, (v) serendipities (*i.e.,* variety in recommended news articles) are of particularly importance, and (vi) breaking and trendy news articles might be of interest even if they are not related to the user's general interests.

Despite the significant progress of recommender systems in general, there are still challenges that limit the effectiveness of currently available solutions in terms of news recommendation for mobile users. The following challenges shape our work: (a) mobile devices have *limited user interfaces*, (b) news articles have *short life cycles*, (c) the challenge of *cold-start users* when they first request a recommendation and their interests are initially unknown, (d) the challenge of *cold-start news articles*, which refers to the difficulty of recommending new articles that have not been tied to many users' preferences, (e) explicit signals about which news articles a user truly wishes to see are typically weak and the *user's desirability* for a news article can not simply be detected, and (f) users' preference for particular articles depends not only on the topic and on propositional content, but also on users' current *context*. A user's current context can cover various types of information like the user's current location, access time, social environment, and external events.

This demo paper presents our solution and implementation for a personalized mobile news recommender system. Our solution combines various methods of recommendation particularly designed for the domain of news articles and it considers a variety of factors including freshness and users' preferences as the main recommendation factors. Cold-start problems are alleviated by considering the user's contextual information such as location. Relevancy among news articles is detected based on several measures in addition to their

similarity. Trendy and popular articles are recommended as well as articles related to the user's general interests. Recommended news is presented and visualized for the user through an intuitive and easy-to-use interface. Moreover, the user's desirability is detected implicitly based on several facets of the user's behavior. In summary, we offer entity-based content analysis, multi-perspective recommendation strategy, and transparent recommendations.

2. BACKGROUND

Different existing techniques for recommendation are briefly explored to serve as background for our solution. These techniques include:

Content-based filtering: a new article is recommended to a user if it is "related" to previous articles desired by the user in the past. Articles are represented as sets of keywords and the article-article relatedness function typically depends on three aspects [4]: (i) *relevance and novelty*, which can be captured in one score based on similarity, (ii) *connection size*, which considers topic overlap as opposed to simple word overlap, and (iii) *transition smoothness*, which indicates if the combination of the articles is coherent. The main disadvantage of this technique is that, in the case of a domain such as news, a user's interest in an article cannot always be characterized by its terms/topics.

Information filtering: a new article is recommended to a user if it matches to the user's profile. The similarity measure can be as simple as the similarity of term frequencies between the vectors of the user profile and the news article or it can be based on more sophisticated methods like those based on semantic reasoning and domain ontologies [1]. The main weakness of this technique is that it does not consider the short-term changes in the user's preferences, for example, based on the user's current active context.

Collaborative filtering: A new article is recommended to a user if it has been preferred by another similar user [2]. This technique can be further categorized into two types: *memory-based* and *model-based*. The former predict a user's ratings for an article as a weighted average of the ratings by other users where the weight is proportional to the similarity between users. The model-based techniques use users' past ratings to predict the ratings of unseen articles. The main advantage of these techniques for news recommendation is that serendipitous recommendations can be made. On the other hand, most of systems using collaborative filtering do not use context or content. For personalized news recommendation, news articles are regarded as different items since there is usually no explicit ratings on news articles. Furthermore, these techniques have disadvantages like sparsity and cold-start problems, *i.e.,* new relevant articles with no historical ratings cannot receive high predicted scores.

Social filtering: A new article is recommended to a user if it has been preferred by the user's contacts in online social networks [6, 5, 3]. The main advantage of this technique is that the cold-start problem is solved since it does not require any previous interaction with the news website. The main weakness of this technique is the sparsity of the social networks and the lack of context consideration. Since the context is defined by your group of friends or contacts, the recommendations may be of a more general nature.

None of these techniques can by themselves provide a comprehensive solution for the aforementioned news recommendation challenges for mobile users. Hybrid solutions that combine different types of methods also have been developed recently. We propose a multi-facet model for news recommendation that overcome these challenges through a specially designed architecture for mobile users.

3. ARCHITECTURE

Our mobile news recommender system is designed to overcome the identified challenges in Section 1 through a two-layered client-server architecture as depicted in Figure 1. It uses two sources of information: *real-time news streams* and *real-time web services* such as microblogs. We chose "PolarisMedia", which generates regional news in Norway, as the source of the real-time news stream. We chose Twitter as the real-time web service. The following subsections describe how the design of each part of the system addresses the different challenges.

3.1 Server-side Development

The server-side goal is twofold. On the one hand, the server must regularly fetch the newest news stories. On the other hand, it has to listen to requests for recommendations from the client-side and provide proper recommendations considering the variety of important factors previously described.

The two tasks are achieved by two separated components, namely the "News data preparation and aggregation" and the "Recommendation engine". Each news article is represented as a vector consisting of title, term-frequency of title words in the article content, names and places, keywords, and category. Each user is represented by a profile, representing the user's general preferences, as a vector of keywords. The user's profile can be either manually specified or can be automatically built using other sources of information such as the user's online activities.

3.1.1 News Data Aggregation and Preparation

The process of transforming the initial data sources from the PolarisMedia real-time news streams to indexed documents consists of four steps: "Fetching", "Syntactic Processing", "Semantic Processing", and "News Indexing" respectively, as depicted in Figure 1. A list of URLs to RSS feeds of news articles is manually provided in the beginning. The list is traversed continuously at a fixed time interval.

In the first step, Fetching, news articles' entries are extracted from the RSS feeds. These entries typically contain information such as title, summary, images, publication timestamp, publisher and URLs pointing to the full text sources. Then, the URLs are used to extract the complete document's text in HTML format. These HTML documents contain some noisy information such as advertisements, HTML tags, scripts, and header/footer texts. Next, the Apache Tika library[1] is used to identify the body part of the HTML document and scrape off the unnecessary information.

---

[1]http://tika.apache.org/

In the second step, Syntactic Processing, the Apache OpenNLP library[2] is used to apply a trained Named Entity Recognition (NER) model to identify names of persons, organizations, and locations. The location names are used as input for the third step and are also used later by the recommendation engine together with information about the recent preferred articles by the user to model the user's current active context.

In the third step, Semantic Processing, the Google Map API is used to find relevant geocodes of the location names. At this point in the pre-processing pipeline, we have news documents containing information about the title, summary, full text, images, publication timestamp, publisher, involved persons, organizations and locations and relevant geocodes. In the last step, News Indexing, this information is indexed via SolR search indexing. SolR allows client applications to search for the contents and form queries and relevance scoring functions that include and combine content keywords, timing information, and geospatial properties.

As opposed to current news applications, our solution's analysis of news does not stop at the level of articles, categories, and sets of keywords. We identify and classify relevant entities in every news article to enable a more fine-grained analysis of content and user preferences. This allows us to distinguish between entities like events, persons, products and places that may themselves be modeled taxonomically in lightweight ontologies. In addition to providing standard news similarity functions, we can go deeper into the semantics of news articles and the relationships between different news stories. In particular, we take advantage of this entity approach for the following:

- **Geographical locations** at different levels of granularity are identified and taxonomically related as part of our short-term recommendation strategy. If several location names appear in a news story, we compare their exact coordinates to determine which locations are relevant. If taxonomically related locations are mentioned, we assume the most specific location to be the most relevant one.

- **News similarity** is interpreted differently for different people in different contexts. Whereas some people may consider a story similar if it relates to the same type of event (*e.g.,* "give me other car accidents"), others focus on people (*e.g.,* "show me other news about Lionel Messi"), locations (*e.g.,* "what else is going on in Paris"), products (*e.g.,* "what are new competitors to the iPhone"), or a combination of different factors. An entity-based approach makes it possible to offer specialized news similarity functions that also takes into account taxonomic relations that are not even present in the articles themselves.

This entity-based content analysis is particularly important in a news setting, as the limited size of news articles and often biased language make standard categorization and clustering techniques less effective.

3.1.2   The Recommendation Engine
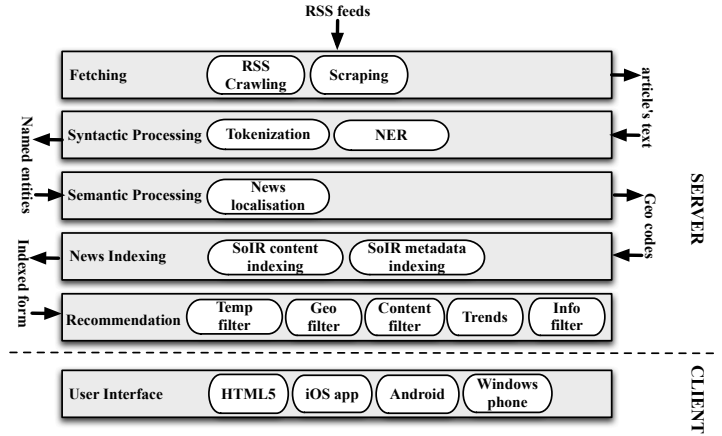
---
[2]http://opennlp.apache.org/

Figure 1: The architecture for our mobile news recommender system.

We next outline the overall design of the recommendation engine. As shown in Figure 2, we have used various factors leading to different recommendation techniques in our mobile news recommender system. The two main factors are *freshness/recency* and *user preferences.*

In reality, the general topics that the user is interested in would be relatively stable over a long period of time, whereas the content accessed by the user might change in the *short-term*. The long-term preferences are determined from the user's profile. As a result, relevant news articles to the user's profile are recommended regardless of their popularity and by using an information filtering technique. On the other hand, short-term preferences are decided based on the user's *current context* and the *trendiness/popularity* of news articles. In this way, the system can adjust more rapidly to the user's changing interests and provide proper recommendations for cold-start users. We chose the user's *location* and *recent preferred news articles* to represent the user's current context. The news articles are ranked higher based on the closeness of their location to the user's current location.

Trendy articles are discovered on Twitter by measuring their popularity and taking into account their aging effects [3]. In our solution, the weights of these factors are either decided by the user or learned automatically through interactions with the user. All in all, the recommendation of fresh articles (cold-start articles) is not challenging for the system because of the use of a combination of filtering techniques, as opposed to any single technique.

3.2   Client Development
The main design challenges were how to present articles to the user given user interface limitations and how to capture the user's desirability and contextual information. Therefore, the client has two main goals: to pull and present the recommended news articles from the sever to the user in an efficient manner and to gather and send information about the user's current active context and the desirability indicators to the server.
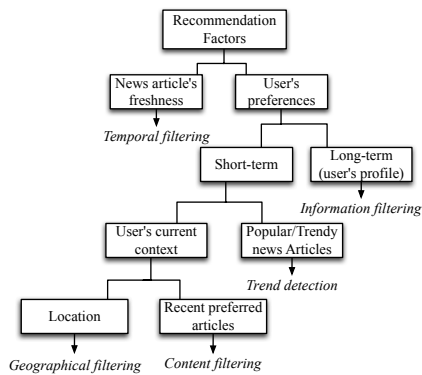
3.2.1   User Interface

Figure 2: Different recommendation factors.

Considering the limitation of a mobile user interface, we assumed simplicity as the main design requirement for the user interface. In addition, the user interface should be parameterized, interactive, and exploratory. Because of the small and limited user interface, solutions like pull-to-refresh and sliding-side-menu are used to create a better user experience.

The client application is developed for the three major platforms: Android, iOS, and Windows. The main page [3] or the news feed, as shown in the left part of the Figure 3, is the first view that is loaded and is the root navigation point. From here, it is possible to navigate to all the other views such as full view for a single article, map view for a single article, and map view of recommended articles. The set of recommended articles is presented in a scrollable view in the main page, while each article is represented by one square, or tile. Each tile has a title, lead text, and a background color to represent the category of the news article. Every three seconds a random tile among the articles is flipped horizontally and the news article's image is shown for a short period of time. It is also possible to do a pull-to-refresh gesture to refresh the news feed and display the latest news. The user can set the preferred category of news articles in the profile using the Filter menu and can send location information to the server using the Location menu. A map of recommended articles can also be shown, see the right part of Figure 3. A clickable dialog box on each article's pin on the map moves the screen to the full view of the corresponding article.

The main properties of the user interface design are user control, parameterized design, exploratory, and interactivity. The user is able to interactively and directly modify time, location, and other parameters such as news source, topics, and news categories and have the system respond immediately with updated views of relevant news. Furthermore, the system supports source tracking by allowing users to go from any news summaries to the original news articles.

3.2.2 Detection of the User's Desirability

The goal of the recommender system is to choose a subset of articles that maximizes the user's utility; that is the user's *desirability* for the articles. It is important to con-

---

[3]The video demonstration of the personalized mobile news recommender system is available at http://youtu.be/BK-YIFq4OZQ.
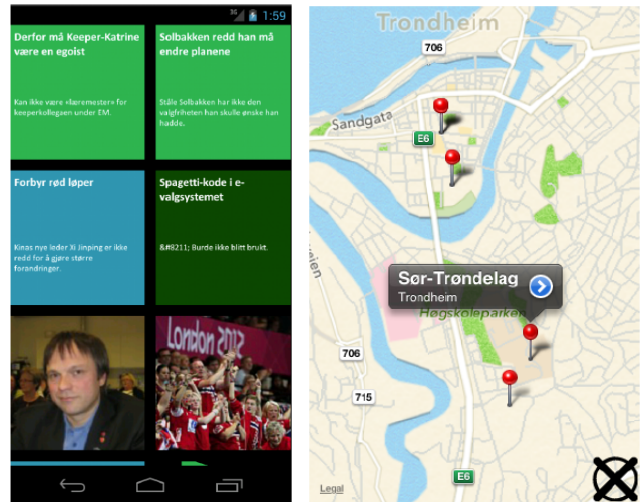

Figure 3: Two parts of the user interface.

sider a broad array of complementary implicit indicators of the users' interests, which can effectively leverage the weak explicit signals about the user's desirability. We considered both "pre-read" user actions such as clicking on the news article and "post-read" engagement signals like discarding, favoriting, sharing, commenting, and e-mailing article links as indicators of the user's desirability. These techniques contrasted with the majority of existing solutions that only use the observed click-through rate as the primary indicator of a user's engagement.

4. CONCLUSION AND FUTURE WORK

Personalized news article recommendation is not only important to solve the information overload and attention scarcity problem, but also is important to improve users engagement on news sites. This demo paper presents the development of a personalized mobile news recommender system.

5. REFERENCES

[1] I. Cantador and P. Castells. Semantic contextualisation in a news recommender system. In *Workshop on Context-Aware Recommender Systems (CARS)*, 2009.

[2] A. S. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization: scalable online collaborative filtering. In *The 16th international conference on World Wide Web*, 2007.

[3] G. De Francisci Morales, A. Gionis, and C. Lucchese. From chatter to headlines: harnessing the real-time web for personalized news recommendation. In *The fifth ACM international conference on Web search and data mining*, 2012.

[4] Y. Lv and et al. Learning to model relatedness for news recommendation. In *The 20th international conference on World wide web*, 2011.

[5] Sh. O'Banion, L. Birnbaum, and K. Hammond. Social media-driven news personalization. In *The 4th ACM RecSys workshop on Recommender systems and the social web*, 2012.

[6] X. Shuai, X. Liu, and J. Bollen. Improving news ranking by community tweets. In *The 21st international conference companion on World Wide Web*, 2012.