SCENE : A Scalable Two-Stage Personalized News Recommendation System

Lei Li, Dingding Wang, Tao Li
School of Computing and Information Sciences
Florida International University
Miami, FL 33199
{lli003, dwang003, taoli}@cs.fiu.edu

Daniel Knox, Balaji Padmanabhan
DailyMe, Inc.
4000 Hollywood Blvd. Suite 745-S
Hollywood, FL 33021
daniel@dailyme.com, bp@usf.edu

ABSTRACT

Recommending news articles has become a promising research direction as the Internet provides fast access to real-time information from multiple sources around the world. Traditional news recommendation systems strive to adapt their services to individual users by virtue of both user and news content information. However, the latent relationships among different news items, and the special properties of new articles, such as short shelf lives and value of immediacy, render the previous approaches inefficient.

In this paper, we propose a scalable two-stage personalized news recommendation approach with a two-level representation, which considers the exclusive characteristics (e.g., news content, access patterns, named entities, popularity and recency) of news items when performing recommendation. Also, a principled framework for news selection based on the intrinsic property of user interest is presented, with a good balance between the novelty and diversity of the recommended result. Extensive empirical experiments on a collection of news articles obtained from various news websites demonstrate the efficacy and efficiency of our approach.

Categories and Subject Descriptors: H.3.3[Information Search and Retrieval]: Information filtering

General Terms: Algorithms, Design, Experimentation

Keywords: Multi-level News Recommendation, News Entity, Personalization, Submodularity, User Profile

1. INTRODUCTION

Web-based news reading services, like Google News and Yahoo! News, have become increasingly prevalent as the Internet provides fast access to news articles from various information sources around the world. With the gigantic amount of news articles, a key issue of online news services is how to help users find interesting articles that match the users' preference as much as possible, by making use of both news content and user information. This is the problem of personalized news recommendation.

SIGIR'11, July 24–28, 2011, Beijing, China.

Despite a few recent advances, personalized news recommendation remains challenging for at least three reasons. First, the scalability of most news recommendation services needs more research for fast and real-time processing; Second, news articles are not independent in most scenarios, i.e., browsing one news item may affect the subsequent news reading; Third, the popularity and recency of news articles change dramatically over time, which differentiates news items from other web objects, such as products and movies, rendering traditional recommendation methods ineffective. As a result, many critical issues of news recommendation have not been explored in previous studies. These issues include news selection (i.e., how to effectively select news articles for recommendation considering their unique characteristics?), news representation (i.e., how to effectively present the recommended new articles to facilitate user navigation and exploration?), news processing (i.e., how to efficiently handle large scale news collection?), and user profiling (i.e., how to construct high-quality user profiles?).

In our work, to address the issues mentioned above, we propose *SCENE*, a *SC*alable two-stage p*E*rsonalized *N*ews r*E*commendation system with a two-level representation, where the first level contains various topics relevant to users' preference, and the second level includes specific news articles. In our system, we explore the intrinsic relation between users and news articles, along with the special properties (e.g., popularity and recency) of news items when recommending to individual users. Also, the system is capable of efficiently dealing with large scale news corpus.

Specifically, *SCENE* consists of three major components – *Newly-Published News Articles Clustering*, *User Profile Construction* and *Personalized News Items Recommendation.* For news articles clustering, we initially partition newly-published news articles into small groups by making use of *Locality Sensitive Hashing* [10], and then hierarchically separate these groups into intermediate clusters, each of which is summarized using probabilistic language models (e.g., *Probabilistic Latent Semantic Indexing* (PLSI) [12] and *Latent Dirichlet Allocation* (LDA) [3]). For personalization, the user's profile is constructed in three different yet related dimensions – news topic distribution, similar access patterns and news entity preference. Based on the generated topic distribution, we sequentially select news clusters similar to the profile of a given user as the first level of the result representation. In each news cluster, the submodularity hidden in different dimensions of news articles motivates us to incorporate this property into our solution to the second level of the representation. Extensive empirical experiments on a collection of news articles obtained from various news websites demonstrate the efficacy of our approach, in terms of the accuracy of selected top ranking news items and the diversity of the recommended news list.

In summary, the contribution of this paper is three-fold:

- **A novel two-level representation** (see Section 4 and 6): Unlike prior approaches simply providing a list of news items, our system generates a two-level representation, where the first level contains general topics *summarized* from news clusters similar to the user's profile, and the second level includes *representative* items within each cluster. Such a representation can help users easily navigate their preferred articles.

- **A principled framework for news selection** (see Section 6): We observe that the interestingness of news articles with respect to a user could be *regressive*, and based on this "submodularity" property, we model the news selection problems as a budgeted maximum coverage problem [16], which is more realistic than independently selecting news items. The proposed framework achieves a good balance between the *novelty* and *diversity* of the recommendation result.

- **Multi-factor high-quality user profile construction** (see Section 5): We explore the feasibility of incorporating various properties of news articles – *news content*, *access patterns* and *named entities* – into the construction of user profiles. Our work provides the first systematic study on constructing high-quality user profiles using these dimensions. It is with great benefit of using such enriched profile to capture the exact reading interest of users.

In addition, we integrate locality sensitive hashing and hierarchical clustering to address the scalability issue in news processing (See Section 4). The rest of this paper is organized as follows. Section 2 presents a brief summary of prior work relevant to personalized news recommendation. In Section 3, the system framework will be introduced, and the detailed algorithmic descriptions for major components in our framework are presented in Section 4, Section 5,and Section 6, respectively. Extensive experimental results are reported in Section 7. Finally Section 8 concludes the paper.

2. RELATED WORK

Recently, recommending news articles or other document-format web objects has attracted more research attention. Several adaptive news recommending systems, such as Google News and Yahoo! News provide personalized news recommendation services for a substantial amount of online users. Existing news recommender systems can be roughly categorized into two different groups: content-based and collaborative filtering.

Content-based: The systems try to sequentially find newly-published articles similar to the user's reading history in terms of content. Generally speaking, news content is often represented using vector space model (e.g., TF-IDF) [15], or topic distributions obtained by language models (e.g., PLSI and LDA), and specific similarity measurements are adopted to evaluate the relatedness between news articles. For example, News Dude [2], is a personal news recommender agent that utilizes TF-IDF combined with the K-Nearest Neighbor algorithm to recommend news items to individual users. Another content-based method is Newsjunkie [8], which filters news stories by formal measures of information novelty, and shows how the techniques can be used to custom-tailor newsfeeds based on a user's reading history. Content-based recommender systems are easy to implement; however, in some scenario, simply representing the user's profile information by a bag of words is insufficient to capture the exact reading interest of the user.

Collaborative filtering: The systems make use of news ratings by users to provide recommendation services, and in general, they are content-free. Note that news ratings are typically binary; a click on a piece of news corresponds to a 1 rating, whereas a non-click is represented as a 0 rating [7]. In practice, most collaborative filtering systems are constructed based on users' past rating behaviors, either using a group of users "similar" to the given user to predict news ratings [22, 23], or modeling users' behaviors in a probabilistic way [13, 21]. Collaborative filtering systems can efficiently capture users' behaviors in case where overlap in historical consumption across users is relatively high and the content universe is almost static [24]; however, in many web-based scenarios, the content universe undergoes frequent changes, with content popularity changing over time as well [18]. Moreover, many online users do not have enough historical consumption record, which is known as a *cold-start* problem [25]. These issues render collaborative filtering ineffective.

As discussed above, content-based and collaborative filtering systems can provide meaningful recommendation and in case also have some disadvantages. To get more reasonable results, many researchers investigate the feasibility of combining these two types of methods, and propose hybrid solutions to news recommendation. Representative examples include [4, 5], in which the inability of collaborative filtering to recommend news items is commonly alleviated by combining it with content-based filtering.

Our work is essentially a hybrid recommendation approach. What differentiates our work from prior methods is that we model personalized news recommendation as a budgeted maximum coverage problem, i.e., the selection of one news item will influence the selection of the following news items. From this perspective, our work is similar to [18], in which personalized recommendation of news articles is modeled as a contextual bandit problem, where a learning algorithm sequentially selects articles to serve users based on contextual information about users and articles, while simultaneously adapting its article-selection strategy based on user-click feedback to maximize total user clicks. Our work is orthogonal to theirs in terms of news articles selection, since they focus on the long-term effect of recommendation, whereas our concern is located on single-session recommendation.

Our SCENE system is also closely related to EMM News Explorer [1] and Newsjunike [8] in the use of news content and named entities for news recommendation. However, EMM Newsexplorer does not provide personalized services and Newsjunike does not address the news selection, news presentation and scalability issues as we do.

3. RECOMMENDATION FRAMEWORK

Figure 1 depicts a brief framework of our proposed system, *SCENE*. The recommendation is performed by following a two-stage procedure, where the first stage serves to divide news collection into groups, and the second stage aims to

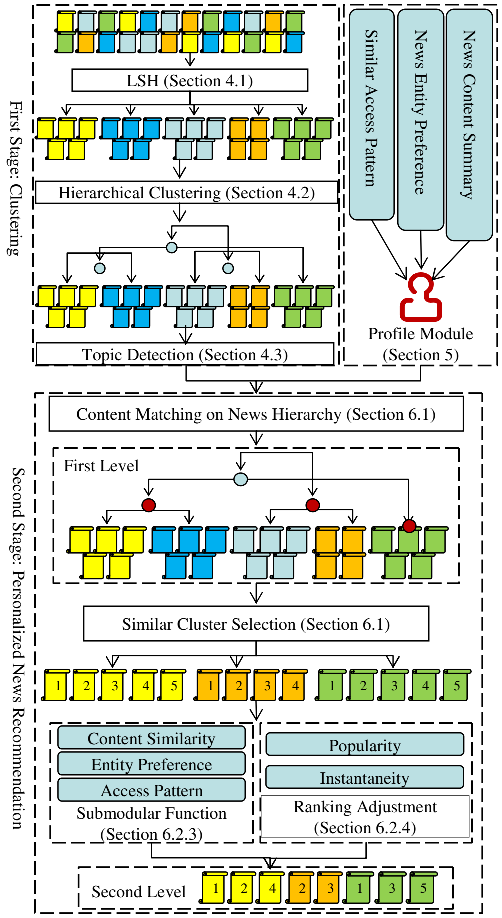

Figure 1: System Framework of *SCENE*.

recommend news items. The three major components in our framework are described as follows.

I. **News articles clustering**: Newly-published news collection is initially divided into small groups using LSH purely based on news content. In order to quickly navigate to specific groups, we employ hierarchical clustering with average-link on these small groups. Then, probabilistic language models are applied to summarizing news articles in each intermediate cluster and small news groups within the cluster. By doing this, a two-layer news hierarchy can be obtained, where leaf nodes denote small groups accompanied by their topic distributions, and internal nodes contain a couple of news groups, representing more general news topics.

II. **User profile construction**: A user's profile is represented as an integrated information capsule, in which three different yet related aspects – accessed news content, similar access patterns and preferred name entities – are combined. All these three factors are extracted from the user's reading history. Specifically, news content is denoted as a topic distribution of the reading history; similar access patterns are generated by analyzing click behaviors of different users; and preferred named entities are extracted from the history using open source NLP tools, e.g., GATE [6].

III. **Personalized News recommendation**: We compare the topic distributions of each intermediate cluster and news content in the user's profile, and then sequentially select the intermediate clusters based on the similarity score, as the first level of the recommendation result. Within each cluster, we continue to compare the similarities between each small news group and the user's accessed news content, and select the most similar group as the base of the second recommendation level. In the selected group, we model personalized news recommendation as a budgeted maximum coverage problem [16], and solve it by selecting news items in a greedy way. Note that when recommending specific news items within each group, the entire user profile is utilized. Moreover, the exclusive properties of news articles, such as popularity and recency, are synthesized into the final news ranking as adjustment factors.

4. NEWS ARTICLES CLUSTERING

Formally, given a set of news articles $\mathcal{N} = \{n_1, n_2, \cdots, n_M\}$, where $|\mathcal{N}| = M$, our goal is to generate a hard clustering result $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_K\}$ on $\mathcal{N}$, where K is a predefined cluster number. Each cluster $\mathcal{C}_i$ is composed of a list of news groups $\mathcal{G}_i = \{G_1, G_2, \cdots\}$, and each news group G_j ($|G_j| = t_j$) contains t_j news articles. Each cluster and its news groups are associated with a topic distribution $\mathcal{T}$ respectively, to describe the latent topics hidden in news articles. The intuition of this representation is simple: news recommendation requires fast response to immediately show the result to users; representing news articles in such way can help quickly navigate to news articles that the user prefers. In *SCENE*, the clustering component lies in the first stage.

4.1 Clustering over News Articles

In reality, large volume of newly-published news corpus requires substantial computational power. For instance, a news corpus with 100,000 news items requires ten billion pairwise comparisons. To tackle this issue, we propose to employ LSH to eliminate unnecessary similarity computations between unrelated articles, and get a rough separation on the original news corpus. In our system, we use a standard Jaccard-based hashing method to find similar news articles. The procedure can be divided into three modules.

I. **Decompose news articles to shingles**: To overcome the disadvantages of the traditional bag-of-words model, we use "shingles" to represent news articles. A k-shingle for an article is a sequence of k continuous words that appear in the article. Before decomposing news articles into shingles, a sequence of preprocessing steps, such as removing stop words, tokenizing and stemming, are executed on the original articles. Since typical news articles are in moderate length, we empirically choose $k = 10$ to guarantee that the probability of any given shingle appearing in any article is low. By shingling, original news corpus can be quantified as a shingle-article matrix $\mathcal{M}$, where rows represent shingles and columns represent news articles.

II. **Minhashing**: In general, $\mathcal{M}$ may still not fit into memory, since the number of news articles tends to be substantial. To handle this issue, we use Minhashing technique to generate a succinct signature for each column in $\mathcal{M}$, such that the probability that two articles have the same signature is equal to the Jaccard similarity between these two. In our system, we construct a length-100 Minhash signature using the technique detailed in [14]. Note that the randomized nature of the Minhash generation method requires further checks to increase the probability of uncovering all pairs of related articles in terms of the signature. Thus, we utilize LSH to increase such probability.

III. **Locality Sensitive Hashing**: LSH [10] is employed to reduce the comparisons, where the generated Minhash signatures are initially decomposed into multiple bands, and for each band, we adopt a standard hash function to hash it into a big hash table. In our experiment, we choose empirically the band length as 5. We treat columns (or news articles) that are hashed into the same bucket[1] at least once as the similar ones. Finally, the original news corpus can be separated into multiple small news groups.

4.2 Clustering over News Groups

When dealing with large news corpus, the number of generated news groups by LSH are still substantial, ranging from hundreds to even thousands. To further speed up the selection of news articles, we propose to hierarchically cluster news groups based on the average-link cross-cluster measurement to provide an elegant global representation of the latent structure of news corpus. Formally, given a number of news groups $\mathcal{G} = \{G_1, G_2, \cdots\}$, hierarchical clustering produces a dendrogram $\mathcal{H}$ of $\mathcal{G}$. With a predefined cluster number K, our system automatically cut $\mathcal{H}$ to get K clusters. The generated clusters serve to be the base of the first level of news recommendation result.

Discussion: As introduced above, we use "LSH + hierarchical" clustering to to separate the original news set into multiple clusters. One may argue that such strategy might result in poor performance in terms of accuracy. Alternative ways of grouping news articles involve the standard K-means and hierarchical clustering directly on the original news corpus. However, these two clustering techniques are not efficient when dealing with large text corpus, and their performance on accuracy cannot be guaranteed when the dataset becomes larger. Also, when recommending news items, we cannot quickly navigate to a small news group that the user might be interested in by using these two clustering techniques. In the experiment, we will verify our claim.

4.3 News Topic Detection

A natural way to explore the correlations between intermediate clusters (or news groups) and the given user's profile is to compare the similarity of latent topics underlying their news articles. In general, detecting latent topics of a text corpus is typically done using probabilistic language models, such as PLSI and LDA, by extracting a list of representative words from the original corpus along with the corresponding weights. In our system, we employ LDA as the language model to detect latent topics, and represent the topic distribution of each news collection located at intermediate clusters (or news groups) as a topic vector, each entry of which denotes the weight of the corresponding word.

Discussion: The PLSI model and the LDA model are similar, except that in LDA the topic distribution is assumed to have a *Dirichlet prior*. Note that the PLSI model is equivalent to the LDA model under a uniform *Dirichlet prior* information, whereas the LDA model is essentially the *Bayesian* version of the PLSI model [11]. *Bayesian* formulation tends to perform better on small datasets because *Bayesian* methods can avoid overfitting. In a very large dataset, the results are probably the same. One difference is that PLSI uses a variable d to represent a document in the training set. As discussed in [3], when a model representing a document has not seen before, PLSI fixes $Pr(w|z)$ – the probability of words under topics – to be that learned from the training set and infers $Pr(z|d)$ – the topic distribution under d. Blei argues that this step is cheating because the model is essentially refitted to the new data [3]. To get a more robust representation of topic distribution, we choose LDA as our language model. A comparison on recommendation results of using LDA and PLSI is provided in Section 7.

[1]In the experiment, we set the size of the hash table to be 1000, which is large enough to distinguish news articles.

5. USER PROFILE CONSTRUCTION

In order to capture a user's reading interests on news articles, news recommendation systems start with constructing the user's profile. Traditionally, a user's profile can be defined by keeping track of what articles the user has read so far (or called consumption history), mainly based on news content. A comprehensive survey of various user profile construction techniques is provided in [9]. However, simply representing a user's profile as a weighted topic distribution cannot effectively capture the user's exact reading preference, as the user's interest might be affected by other users. Another reason of the insufficiency of topic representation is that many news readers tend to glance at news articles, interested in some named entities, such as what happened, who involved, when it happened and so on.

Based on the above analysis, we propose to build users' profiles by the exploration on three different yet related dimensions – news content, similar access patterns and preferred news entities. Each user's profile can be parameterized with a three-attribute tuple $\mathcal{U} = <\mathcal{T}, \mathcal{P}, \mathcal{E}>$, where

- $\mathcal{T}$ represents the topic distribution of news articles the user accessed in the past, in the format of a topic vector $\{<t_1, w_1>, <t_2, w_2>, \cdots\}$, where each entry consists of a representative word and the corresponding weight;
- $\mathcal{P}$ denotes a list of users $<u_1, u_2, \cdots>$ that have similar access patterns with the given user;
- $\mathcal{E}$ is a list of named entities $<e_1, e_2, \cdots>$ extracted from the user's reading history, associated with the corresponding entity type.

Clearly, these three dimensions have mutual interaction with each other. The topic distribution learned from the reading history is likely to be related to the list of entities in the profile, whereas these two dimensions might contribute to the similar access patterns that the two users have. In the following, we will introduce the detailed techniques adopted to construct different aspects of users' profiles.

5.1 News Content Summarization

In our recommendation system, news articles in users' consumption history are summarized as a topic distribution, by making use of the same strategy employed in Section 4.3. The way that we represent users' reading history as the same as the representations for intermediate clusters (or news groups), guarantees that news content matching is applicable to the first level recommendation.

5.2 Access Pattern Analysis

In reality, many online news readers may exhibit similar reading preference. A user's profile information can be enriched in a way of analyzing other users' reading preference similar to the given user, which is essentially collaborative filtering. Under such intuition, we propose to integrate similar access patterns from other users into the given user's profile. Specifically, each user's reading history is maintained. Suppose the news collection read by user A and B are $\mathcal{N}_A$ and $\mathcal{N}_B$, their pairwise similarity of access patterns is defined as the Jaccard similarity of $\mathcal{N}_A$ and $\mathcal{N}_B$. By calculating pairwise user similarities, a similarity matrix can be obtained, each entry of which is the Jaccard similarity between two users' access patterns. The calculation is done offline. Given a specific user u, any other user can be said to be similar with u if their pairwise similarity score is above a predefined threshold τ_u[2]. By doing this, all the users similar to u are stored in u's profile.

5.3 Named Entities Extraction

Typically, in news articles, named entities include when, where, what happened, who are involved, and so on. News readers might have special preference on some particular named entities contained in news articles. Therefore, named entities are important when recommending news items to individual users. To extract named entities, we employ an open source NLP tool – GATE[6], which is capable to automatically identify named entities in texts by predefining a couple of entity rules. By default, we use the rules provided in GATE. After entity detection, each news article is associated with a list of named entities along with their corresponding entity types.

6. PERSONALIZED RECOMMENDATION

Personalized news recommendation is oriented from exploring the relations between newly-published news articles and the user's profile. In this paper, what we proposed is essentially a hybrid recommendation method; however, different from prior approaches, we provide a two-level recommendation hierarchy, where the first level shows a brief summary for each topic category the user might prefer, and the second level gives a specific list of news articles similar to the user's reading interest. In our system, this component lies in the second recommendation stage.

6.1 Interest Matching for Representation Lv.1

Once we generate the news hierarchy and the user's profile, the first representation level can be obtained by sequentially matching the user's profile onto the news hierarchy, and selecting appropriate intermediate clusters. Note that each cluster corresponds to a topic category. For simplicity, we only consider the similarity between topic distributions of each intermediate cluster and the user's reading history.

Formally, the topic distribution is represented as a topic vector $\mathcal{T} = \{<t_1, w_1>, <t_2, w_2>, \cdots\}$. To ensure that all the topic vectors are within the same dimensionality, we create a topic vocabulary $\mathcal{V}$ based on the existing topics, where $|\mathcal{V}|$ is the total number of representative words. When comparing the similarity between the topic distribution of each cluster, $\mathcal{T}_C$, and the one of the user's profile, $\mathcal{T}_U$, we adopt the cosine similarity:

$$Sim(\mathcal{T}_C, \mathcal{T}_U) = \frac{\mathcal{T}_C \cdot \mathcal{T}_U}{||\mathcal{T}_C||\,||\mathcal{T}_U||}, \quad (1)$$

where $|\mathcal{T}_C| = |\mathcal{T}_U| = |\mathcal{V}|$, and $||\mathcal{T}_C||$, $||\mathcal{T}_U||$ are the l_2 norms.

[2]In the experiment, τ_u is tuned to be an optimal value.

The ranking of intermediate clusters is based on the similarity score calculated by Eq.(1). In general, people tend to have their preference on topic categories, i.e., not interested in all the topics. Therefore, we choose the clusters with the similarity greater than a dynamic threshold[3]. After selecting the appropriate clusters, we dig into each cluster and choose the news group most similar to the user's interest using the same strategy for selecting clusters. By doing this, a list of news groups will be selected, as the recommendation base of the second level.

6.2 News Selection for Representation Lv.2

After obtaining news groups that the user might be interested in, the subsequent step is to select specific news articles to present to the user. We initially maintain a news profile for each news article, and then model personalized recommendation as a budgeted maximum coverage problem, and solve it by a greedy algorithm.

6.2.1 *News Profile Construction*

A news profile includes static descriptors (e.g., topic distribution, named entities) and dynamic characteristics (e.g., accessed users, popularity, recency). Continuous refinement on the news profile will help to optimize the use of the news corpus. For the popularity, it is computed as the ratio of the number of users accessing to the article and the size of the users' pool. For the recency, the score is represented as $(CurrentTime - PublishedTime)/(24 * 60)$.

In *SCENE*, news profiles are helpful to compare two news articles, and to evaluate how the news item can satisfy the user's reading preference. These two types of comparisons are calculated under the same setup. Given a news profile $\mathcal{F}_n = <\mathcal{T}_n, \mathcal{P}_n, \mathcal{E}_n>$ and a user's profile $\mathcal{F}_u = <\mathcal{T}_u, \mathcal{P}_u, \mathcal{E}_u>$, the similarity between $\mathcal{F}_n$ and $\mathcal{F}_u$ is computed as

$$Sim(\mathcal{F}_n, \mathcal{F}_u) = \frac{\alpha Sim(\mathcal{T}_n, \mathcal{T}_u) + \beta Sim(\mathcal{P}_n, \mathcal{P}_u) + \gamma Sim(\mathcal{E}_n, \mathcal{E}_u)}{\sqrt{\alpha^2 + \beta^2 + \gamma^2}}, \quad (2)$$

where α, β and γ are parameters to control how we trust the corresponding components. $Sim(\mathcal{T}_n, \mathcal{T}_u)$ is computed by Eq.(1), whereas $Sim(\mathcal{P}_n, \mathcal{P}_u)$ and $Sim(\mathcal{E}_n, \mathcal{E}_u)$ are calculated by the Jaccard similarity.

Discussion: As noted, three components in Eq.(2) evaluate the similarities of news content, access patterns and preferred named entities, respectively. In prior approaches, either news content, or access pattern, or their combination is considered. Comparatively, we assign equal importance to these three factors in our recommender system, since they represent different yet related aspects of news articles or users' profiles. In Section 7, we present our evaluation on the effects of different combinations of these three factors.

6.2.2 *Introduction to Submodularity*

Let E be a finite set and f be a real valued nondecreasing function defined on the subsets of E that satisfies

$$f(T \cup \{\varsigma\}) - f(T) \leqslant f(S \cup \{\varsigma\}) - f(S), \quad (3)$$

where $S \subseteq T$, S and T are two subsets of E, and $\varsigma \in E \setminus T$. Such a function f is called a **submodular** function [20]. Intuitively, by adding one element to a larger set T, the value increment of f can never be larger than that by adding

[3]The dynamic threshold is set to be the median of all similarity scores with respect to a specific user's profile.

one element to a smaller set S. This intuitive diminishing property exists in different areas, e.g., in social network, adding one new friend cannot increase more social influence for a more social group than for a less social group.

The budgeted maximum coverage problem is then described as: given a set of elements E where each element is associated with an influence and a cost defined over a domain of these elements and a budget B, the goal is to find out a subset of E which has the largest possible influence while the total cost does not exceed B. This problem is NP-hard [16]. However, [16] proposed a greedy algorithm which sequentially picks up the element that increases the largest possible influence within the cost limit and it guarantees the influence of the result subset is $(1-1/e)$-approximation. Submodularity resides in each "pick up" step. A key observation is that submodular functions are closed under non-negative linear combinations [17].

6.2.3 Submodularity Model for Recommendation

In a particular news group, most of news articles concentrate on similar or even the same topic, with minor difference on major aspects of the corresponding topic. For example, given a news group talking about a popular movie "*Inception*", one piece of news may focus on the actor cast of this movie, while another may describe the high-end techniques used in this movie. Typically, a news reader is interested in some specific aspects of the given topic, but not all of them. Based on this intuition, our news selection strategy can be described as follows (note that $\mathcal{N}$ denotes the original news group, $\mathcal{S}$ represents the selected news set, and ς is the news item being selected). After selecting ς,

- $\mathcal{S}$ should be similar to the general topic in $\mathcal{N} \setminus \mathcal{S}$;
- The topic diversity should not deviate much in $\mathcal{S}$;
- $\mathcal{S}$ should provide more satisfaction to the given user's reading preference.

Per the above strategy, we define a quality function f to evaluate the current selected news set $\mathcal{S}$ over the entire news group $\mathcal{N}$ as

$$f(\mathcal{S}) = \frac{1}{|\mathcal{N} \setminus \mathcal{S}| \cdot |\mathcal{S}|} \sum_{n_1 \in \mathcal{N} \setminus \mathcal{S}} \sum_{n_2 \in \mathcal{S}} sim(n_1, n_2) + \frac{1}{\binom{|\mathcal{S}|}{2}} \sum_{\substack{n_1, n_2 \in \mathcal{S} \\ n_1 \neq n_2}} -sim(n_1, n_2) + \frac{1}{|\mathcal{S}|} \sum_{n_1 \in \mathcal{S}} sim(u, n_1), \tag{4}$$

where n_1 and n_2 denote news items, u represents the given user, and $sim(\cdot, \cdot)$ represents the similarity between two profiles, either the user profile or the news profile.

In Eq.(4), three components are involved, corresponding to the news selection strategy we list above. The first one aims to evaluate the quality of how representative that the selected news set $\mathcal{S}$ is over the original news set; the second one provides a perspective on how diverse that the topics underlying the selected news articles are; and the last component gives us the evidence that how much the user's preference is satisfied by the selected news set $\mathcal{S}$. $f(\mathcal{S})$ balances the contributions of different components. Note that all these three components are naturally submodular functions. Based on the non-negative linear invariability of the submodular function, $f(\mathcal{S})$ is also a submodular function. Suppose ς is the candidate news article, the quality increase is therefore represented as follows:

$$I(\varsigma) = f(\mathcal{S} \cup \{\varsigma\}) - f(\mathcal{S}). \tag{5}$$

The goal is to select a list of news articles which provide the largest possible quality increase within the budget[4]. Hence, personalized news recommendation is transformed to the budgeted maximum coverage problem.

In each news group, a greedy algorithm is employed to solve the budgeted maximum coverage problem, by sequentially selecting the news article providing the largest quality increase based on the selected news set until the budget is reached. To integrate recommended news items from different news groups into the final recommendation list, we select top ranking items within each group, where the number of items selected in one group is proportional to the interest weight of the user on the corresponding topic category.

Discussion: The submodularity-based news selection strategy provides us a diverse news recommendation list within each topic category. In addition, multiple topic categories are recommended to individual users, by which the diversity of the final recommendation result is explicitly enriched to a great extent. In Section 7, we will present detailed evaluation to verify our claim.

6.2.4 Ranking Adjustment

By adopting the greedy algorithm introduced above, we can obtain a list of news articles for each topic category. Taking into account the exclusive characteristics of news articles, e.g., the popularity and the recency, the ranking of the selected news articles needs to be adjusted in order to make the recommendation result more reasonable.

In our system, the popularity and the recency of news articles are maintained in news profiles, as described in Section 6.2.1. When doing adjustment on the selected news list, we combine the normalized values of these two types of properties. Formally, given a news article n, the popularity $n_{\mathcal{P}}$ and the recency $n_{\mathcal{I}}$ can be combined as

$$n_\phi = \frac{n_{\mathcal{P}} - n_{\mathcal{P}_{min}}}{n_{\mathcal{P}_{max}} - n_{\mathcal{P}_{min}}} - \frac{n_{\mathcal{I}} - n_{\mathcal{I}_{min}}}{n_{\mathcal{I}_{max}} - n_{\mathcal{I}_{min}}}. \tag{6}$$

Note that the recency is restricted by time; the smaller it is, the higher ranking the article is. Given a ranking list of news articles, we sequentially select two adjacent news items n_i and n_j from top to bottom, and compare their dynamic score n_ϕ. If $n_{\phi_j} - n_{\phi_i}$ is larger than 0, we swap the order of n_i and n_j; otherwise, we skip them and continue to compare the next article pair. By such minor adjustment, the generated ranking can emphasize more popular and fresh news items, as well as concentrating on news articles that satisfy the user's reading preference.

7. EXPERIMENTAL EVALUATION

In this section, we provide a comprehensive experimental evaluation to show the efficacy and efficiency of our proposed news recommendation system, *SCENE*. We start with an introduction to a real-world news collection obtained from multiple news service websites, and then describe the experimental design based on the recommendation framework introduced in Section 3.

[4]Here the budget can be regarded as the maximum number of recommended items in each news group.

7.1 Real World Dataset

For experiments, we gather news articles along with users' access history from popular news websites which contain multiple news topic categories, such as sports, movies and politics. We gather the news data for 9 categories on purpose, where the data collection ranges from Aug 15th, 2010 to Nov 16th, 2010. In order to embody the role of similar access patterns in users' profiles, we preprocess the data by removing news articles that are rarely accessed (i.e., the accessed frequency is less than 10 times per day) and by storing users with frequent online reading behaviors (i.e., users who read news articles every day and read more than 10 pieces of news each day). After preprocessing, 112,380 news items are stored, with 4,630 users, each day in average with 1,221 news articles.

7.2 Experiments

SCENE contains three major components: (a) An offline component responsible for periodically clustering news articles published within a time range; (b) An online component of dynamically constructing and updating user profiles; and (c) An online component capable to recommend news articles to individual users based on the generated cluster hierarchy and user profiles. From the experimental perspective, we verify our system components separately, where Component a is tested in an offline manner, and Component b and c are tested under a unified online environment. In the following we will describe the detailed experiments, along with a user study to evaluate the real-world users' satisfaction.

7.2.1 Clustering Evaluation

To evaluate the performance of the clustering component, we provide three different comparisons: (i) "LSH + Hierarchical" clustering against direct clustering on news articles; (ii) hierarchical clustering against K-means on news groups; and (iii) LDA against PLSI on topic detection.

I : *"LSH + Hierarchical" Clustering* V.S. *Direct Clustering*

We select multiple time ranges with different intervals and run the clustering procedure on news articles from these time ranges. We also implement the general K-means and hierarchical clustering with average-link, to compare the F-measure and the time cost. All the clustering algorithms are implemented using Java and tested under the same experimental environment. For K-means and hierarchical clustering, we represent each news article using vector space model, each entry of which is denoted by TF-IDF value of the corresponding term. For each time range, we execute different algorithms 10 times respectively, except hierarchical clustering, and compare micro-averaging $F1$ and macro-averaging $F1$ to evaluate the average performance across multiple clusters with different cardinality. The number of clusters is set to be the number of categories introduced in Section 7.1.

Table 1 lists the clustering evaluation results. Statistical significance test is performed on $F1$ scores and shows the significance of the combined clustering technique; however, due to the space limit, the test is not reported. Based on the comparison, we observe that: (i) Our proposed "LSH + Hierarchical" clustering on news collection significantly outperforms the general K-means and hierarchical clustering techniques in terms of accuracy and efficiency. A straightforward explanation for the accuracy increase is that *SCENE* uses shingles to represent news articles. Shingling aims to separate articles into shingles where the probability of any given shingle appearing in any article is low. In this way, similar articles will have more shingles in common, whereas dissimilar articles share rarely few shingles. (ii) The accuracy of K-means and hierarchical clustering on news corpus decreases when the volume of the dataset increases.

II : *Hierarchical Clustering* V.S. *K-means Clustering*

An alternative approach to divide the news corpus into news groups is K-means clustering. In this section, we provide comparison between K-means and hierarchical clustering on news groups. We select 10 time ranges with 3-days interval, and run LSH on news articles located in these ranges. Then we apply K-means and hierarchical clustering with average-link on small news groups, respectively. For evaluation purpose, we randomly select 100 users and recommend news articles to them. The recommendation result consists of top 30 news articles. We evaluate the averaged *precision*, *recall* and *F-score* based on the 100 users' actual reading history in these time ranges. Table 2 represents the comparison results. We can observe that hierarchical clustering based system outperforms its rival to a great extent, which verifies our claim in Section 4.2. Statistical significance test is performed on F-scores and demonstrates the qualification of hierarchical clustering on news groups.

Time Range	# of Articles	K-means			Hierarchical(average-link)		
		P	R	F	P	R	F
08/20–08/22	3,340	0.1892	0.3115	0.2241	0.2935	0.4073	0.3406
09/06–09/08	2,985	0.2031	0.3230	0.2408	0.3016	0.4252	0.3512
09/23–09/25	3,920	0.1938	0.2988	0.2376	0.2863	0.3908	0.3389
10/18–10/20	2,807	0.2101	0.3296	0.2492	0.2997	0.4186	0.3557
11/03–11/05	3,742	0.1834	0.3053	0.2283	0.2874	0.3922	0.3392
Average	3.305	0.1959	0.3136	0.2360	**0.2937**	**0.4068**	**0.3451**

Table 2: Comparison between K-means based and hierarchical clustering based systems.

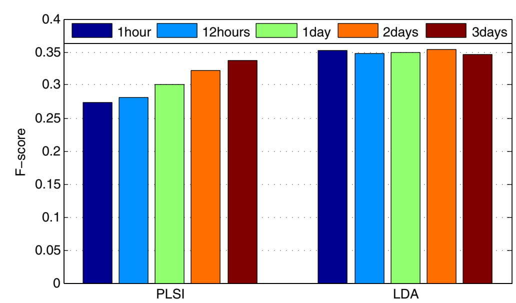

Figure 2: Recommendation F-score of different time periods for PLSI-based and LDA-based systems.

III : *LDA* V.S. *PLSI*

In reality, news article clustering happens per day, or even per hour. The number of news events happened in a small time range may be at a scale of hundreds or thousands, but not more in regular. Based on our analysis in Section 4.3, LDA tends to perform better than PLSI in terms of topic detection when the dataset is relatively small. In order to verify this claim, we design the experiment as follows: (1) use 1 hour, 12 hours, 1 day, 2 days, and 3 days as the time periods; (2) for each time period, randomly select 10 time ranges, and extract news articles in these time ranges; (3) perform PLSI and LDA on these news sets respectively; (4) perform top 30 news recommendation to 100 users randomly

Time Range	# of Articles	K-means			Hierarchical(average-link)			"LSH + Hierarchical"		
		Micro-F1	Macro-F1	Time Cost	Micro-F1	Macro-F1	Time Cost	Micro-F1	Macro-F1	Time Cost
08/15–08/16	2,340	0.4530	0.3744	3min	0.4398	0.3551	5min	0.5016	0.4622	2min
08/19–08/22	5,572	0.4365	0.3653	5min	0.4207	0.3506	9min	0.4897	0.4406	4min
08/23–08/31	10,985	0.4227	0.3421	10min	0.4156	0.3412	15min	0.4902	0.4535	7min
09/01–09/15	18,841	0.3962	0.3302	21min	0.3820	0.3409	25min	0.4830	0.4480	10min
09/01–09/30	35,920	0.3783	0.3153	38min	0.3777	0.3344	50min	0.5128	0.4598	15min
10/01–11/16	63,659	0.3529	0.2976	59min	0.3755	0.3220	78min	0.5239	0.4679	25min
08/15–11/16	112,380	0.3217	0.2711	121min	0.3622	0.3118	152min	0.5093	0.4506	38min
Average	–	0.3945	0.3280	–	0.3962	0.3367	–	**0.5015**	**0.4547**	–

Table 1: "LSH + Hierarchical" Clustering V.S. Direct Clustering.

selected from the users' pool; and (5) compute the average F-score for PLSI-based and LDA-based systems. The result is shown as in Figure 2.

From the result, we have the following observations: (i) LDA-based recommender system has stable recommendation performance in terms of F-score, regardless of different size of news corpus; and (ii) PLSI-based recommender system has comparable results when the news corpus becomes larger. However, when the dataset is relatively small, the performance of PLSI-based system is comparatively lower than LDA-based system. Essentially, it results from overfitting when the dataset is small, and verifies our claim in Section 4.3. Therefore, the probabilistic language model LDA is more applicable to our news recommendation system.

7.2.2 *Profile Evaluation*

In our system, we propose to utilize three different yet relevant types of information to construct users' profiles – *news content*, *similar access pattern* and *named entities preference*. Prior approaches often use one or both of "news content" and "similar access pattern" to fulfill the recommendation requirement. However, based on our analysis in Section 5, news readers might have more interest on named entities appeared in news articles. In order to evaluate the performance of our hybrid approach, we test the effect of each possible combination of the three factors to the recommendation results[5]. Specifically, we select 100 users from the users' pool, randomly pick up 10 time ranges with 3-day interval from our news dataset, and recommend news items (top @10, @20 and @30) to these users based on different aspect combinations. For comparison, we compute the averaged F-score of recommendation results for the 100 selected users over 10 time ranges. Figure 3 depicts the result.

From the comparison, we observe that: (i) our hybrid approach that considers all the three aspects always performs the best. (ii) recommendation purely based on one single aspect cannot outperform the system based on the combinations. The reason is straightforward: more intrinsic properties of news articles and users' profiles can be revealed as we take more aspects into account. (iii) recommender systems with preferred named entities involved perform better than the ones without considering named entities. This observation verifies our claim that news readers tend to show more interest on simple but representative named entities contained in news articles.

In order to find similar access patterns for a given user, we use a threshold τ_u to eliminate irrelevant users. To find an optimal value for τ_u, we adopt the same experimental setup with the above procedure, except that when selecting

[5]For simplicity, we set $\alpha = \beta = \gamma = 1$ in Eq.(2).

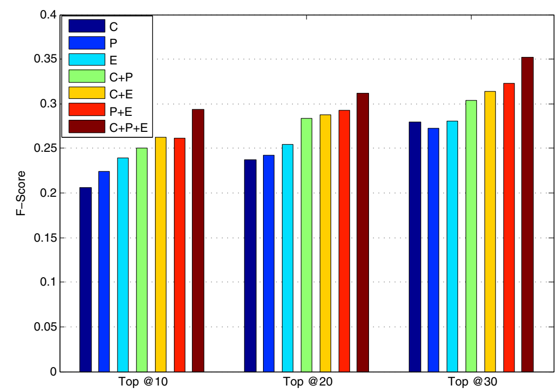

Figure 3: Recommendation F-score of different combinations of profile factors. Remark: C – News Content; P – Similar Access Pattern; E – Preferred Named Entities.

news articles, we only consider the effect of similar access pattern. As we change the value of τ_u, averaged F-scores of the recommendation results are observed. The tuning result is shown in Figure 4. As is depicted, for top @10, @20, @30 recommended news articles, the performance achieves the best when $\tau_u = 0.25$. Therefore, we set τ_u as 0.25 for *SCENE*.

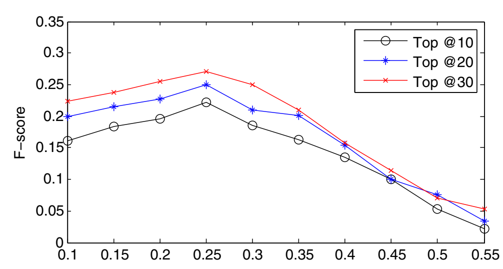

Figure 4: τ_u tuning curve.

7.2.3 *Selection Strategy Evaluation*

In order to verify the effectiveness of our proposed news selection strategy, we provide detailed comparison between ours and the general greedy selection strategy simply based on pairwise similarities. Also, we implement a recommender system that models the recommendation as a contextual bandit problem [18], as the comparison base. For each approach, we randomly select 100 users to provide recommendations for them. We plot the precision and recall pair for each user on top @10, @20, and @30 news items recommended to these users. Figure 5 shows the comparison results. From Figure 5, we observe that besides the higher precision and recall, the performance distribution of *SCENE* is more compact than the other methods, which demonstrates the stability of our proposed news selection strategy.

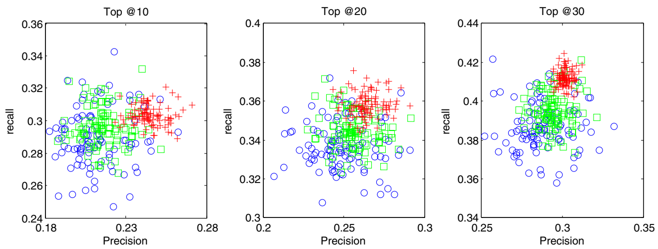

Figure 5: Precision-recall plot for different news selection strategies. Remark: "◯" represents users using the general greedy-based recommender system; "□" denotes users using the bandit-based recommender system; and "+" represents users using *SCENE*.

In the above experiments, all the users are equally treated as the experimental subject. In reality, users with different news access patterns, such as different reading frequency every day, may have distinct patterns of news topic preference, and therefore the dynamic interest on news articles may vary a lot. In addition, many news recommendation systems cannot address the so-called "cold-start" problem. In order to verify the performance of our proposed algorithm on different user groups, we separate the selected users into three groups based on their reading habits. Suppose a user reads N news articles per day, then the three groups are: (i) $N \leq 10$; (ii) $10 < N \leq 50$; (iii) $N > 50$. We apply different algorithms on these three users groups with top @10, top @20 and top @30 recommended news, and record the F1-score respectively. Here, the comparison base includes two existing approaches: [7] (Goo) and [19] (ClickB). The former is a collaborative filtering based method, whereas the latter is a content-based method. Figure 6 shows the comparison results. From the comparison, we observe that our system *SCENE* can achieve a reasonable recommendation result when it is subject to the "cold-start" problem. The reason is that besides considering similar access patterns when recommending news articles to individual users, we also measure the importance of news content and named entities preferred by news readers.

7.2.4 Diversity Evaluation

The recommendation news list provided by *SCENE* shows a great diversity on both topic categories and topic aspects. Such diversity is oriented from two sparkles of our two-stage recommender system: the "LSH + Hierarchical" clustering component and the news selection strategy based on "submodularity". "LSH + Hierarchical" clustering provides a diverse list of topic categories as the first level of recommendation, whereas the "submodularity" model increases the diversity of the result in terms of news content. To evaluate how diverse our recommendation result is, we compare the set diversity described in [26] between the results of *SCENE* and other recommender systems. The news set diversity is

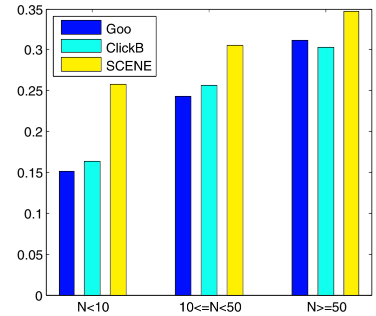

Figure 6: Comparison on F-score of different algorithms for three distinct user groups.

defined as the *average dissimilarity* of all pairs of news items in the recommendation list. Specifically, given a news set $\mathcal{N}$, the *average dissimilarity* of $\mathcal{N}$, $f_d(\mathcal{N})$, is defined as

$$f_d(\mathcal{N}) = \frac{2}{p(p-1)} \sum_{n_i \in \mathcal{N}} \sum_{n_j \in \mathcal{N}, n_j \neq n_i} (1 - Sim(n_i, n_j)) \quad (7)$$

where $|\mathcal{N}| = p$, and the dissimilarity of a news pair is represented as $1 - Sim(n_i, n_j)$, in which $Sim(n_i, n_j)$ denotes the news profile similarity between the news item n_i and n_j.

For diversity evaluation, we choose [7] (Goo), [19] (ClickB), [5] (Bilinear) and [18] (Bandit) as the comparison baselines. We employ the same experiment setup described in Section 7.2.2, to compare the diversities of recommendation lists with different cardinalities. Table 3 shows the averaged diversity result for 10 time ranges.

Methods	Top @10	Top @20	Top @30
Goo	0.4101	0.3074	0.1105
ClickB	0.4329	0.3128	0.1562
Bilinear	0.4234	0.2517	0.0933
Bandit	0.5056	0.4126	0.2925
SCENE	**0.6930**	**0.6671**	**0.6059**

Table 3: Diversity evaluation on the result list.

From the result, we observe that: (i) The diversity decreases as the recommendation news list enlarges. It is straightforward that when more news articles are selected, the topic distribution of the news list becomes closer to the user's reading interest, and therefore the selected news items are more similar. (ii) The diversity of the recommendation list provided by the baseline methods drops dramatically as the list size increases, since they did not take the diversity into account. (iii) *SCENE* outperforms the others very significantly, and since we intentionally consider the requirement of news readers, the diversity decreases very smoothly when we recommend more news items to individual users.

7.2.5 A User Study on *SCENE*

In order to verify the efficacy of our proposed recommender system, we implement a prototype system of *SCENE* and conduct a user study on it. Specifically, *SCENE* gathers newly-published news articles from popular news websites per day, as the news articles pool for recommendation. Then 50 volunteers are hired to experience *SCENE* in 15 days. Most of them have online news reading hobbies. For the first 5 days, we collect the users' accessing news articles as the reading history; for the successive 10 days, we recommend news items to the volunteers. For evaluation purpose, we define several indexes to measure the satisfaction of user experience. Each experience index is rated by the volunteers in a range of 1 to 5, where 1 – "Execrable", 2 – "Below Average", 3 – "Average", 4 – "Above Average", and 5 – "Exceptional". The experience indexes include: (i) The response time once a user logs onto the system; (ii) The preference on the recommended news articles; (iii) The diversity of the recommended news list; and (iv) The ordering of the recommended news articles.

We collect all the volunteers' ratings on these four experience indexes at the end of the testing period. To analyze the investigation result, we plot the number of users with different ratings on the four indexes, and get a histogram as shown in Figure 7. From the result, we observe that all the four aspects of *SCENE* can satisfy the requirements of the majority users. Particularly for the diversity of the recommended news list, most volunteers thumb up towards our prototype system. This also verifies our claim in Section 6.2.3.

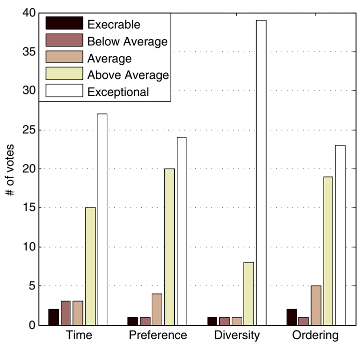

Figure 7: User experience on different indexes.

8. CONCLUDING REMARKS

In this paper, we propose *SCENE* to tackle personalized news recommendation. We explore the intra relations among news articles, along with different characteristics of news items, including news content, similar access patterns and named entities preferred by users. Our system supports efficient clustering on newly-published news articles, as well as high quality of recommendation results. Extensive evaluation has demonstrated the efficacy and efficiency of *SCENE*.

For future work, to handle high volume press releases, we plan to deploy the offline clustering component onto the Map-Reduce framework. It requires careful investigation on the seamless integration of LSH on news articles and hierarchical clustering on news groups. Another remarkable point is the interest evolution of users, which is able to provide insights on the exploration of users' reading behaviors.

ACKNOWLEDGEMENTS

The work is partially supported by US National Science Foundation under grants IIS-0546280 and CCF-0830659.

9. REFERENCES

[1] C. Best, E. van der Goot, M. de Paola, T. Garcia, and D. Horby. Europe media monitor–emm. *JRC Technical Note No.I*, 2, 2002.

[2] D. Billsus and M.J. Pazzani. A personal news agent that talks, learns and explains. In *Proceedings of the 3rd Annual Conference on Autonomous Agents*, pages 268–275, 1999.

[3] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[4] R. Burke. Hybrid systems for personalized recommendations. *Intelligent Techniques for Web Personalization*, pages 133–152, 2005.

[5] W. Chu and S.T. Park. Personalized recommendation on dynamic content using predictive bilinear models. In *WWW 2009*, pages 691–700.

[6] D.H. Cunningham, D.D. Maynard, D.K. Bontcheva, and M.V. Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *ACL*, 2002.

[7] A.S. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization: scalable online collaborative filtering. In *WWW 2007*, pages 271–280.

[8] E. Gabrilovich, S. Dumais, and E. Horvitz. Newsjunkie: providing personalized newsfeeds via analysis of information novelty. In *WWW 2004*, pages 482–490.

[9] S. Gauch, M. Speretta, A. Chandramouli, and A. Micarelli. User profiles for personalized information access. *The adaptive web*, pages 54–89, 2007.

[10] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *VLDB 1999*, pages 518–529.

[11] M. Girolami and A. Kabán. On an equivalence between PLSI and LDA. In *SIGIR 2003*, pages 433–434.

[12] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR 1999*, pages 50–57.

[13] T. Hofmann. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems*, 22(1):89–115, 2004.

[14] P. Indyk. A small approximately min-wise independent family of hash functions. In *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 454–456, 1999.

[15] D. Jurafsky, J.H. Martin, A. Kehler, K. Vander Linden, and N. Ward. *Speech and language processing*. Prentice Hall, 2000.

[16] S. Khuller, A. Moss, and J.S. Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45, 1999.

[17] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *SIGKDD 2007*, pages 420–429.

[18] L. Li, W. Chu, J. Langford, and R.E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *WWW 2010*, pages 661–670.

[19] J. Liu, P. Dolan, and E.R. Pedersen. Personalized news recommendation based on click behavior. In *Proceedings of the 15th International Conference on Intelligent User Interfaces*, pages 31–40. ACM, 2010.

[20] GL Nemhauser, LA Wolsey, and ML Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294, 1978.

[21] D.M. Pennock, E. Horvitz, S. Lawrence, and C.L. Giles. Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach. In *UAI 2000*, pages 473–480.

[22] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. GroupLens: an open architecture for collaborative filtering of netnews. In *Proceedings of ACM conference on Computer Supported Cooperative Work*, pages 175–186. ACM, 1994.

[23] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *WWW 2001*, pages 285–295.

[24] J.B. Schafer, J. Konstan, and J. Riedi. Recommender systems in e-commerce. In *Proceedings of the 1st ACM Conference on Electronic Commerce*, pages 158–166, 1999.

[25] A.I. Schein, A. Popescul, L.H. Ungar, and D.M. Pennock. Methods and metrics for cold-start recommendations. In *SIGIR 2002*, pages 253–260.

[26] M. Zhang and N. Hurley. Avoiding monotony: improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, pages 123–130.