Unified Relevance Feedback for Multi-Application User Interest Modeling

Sampath Jayarathna, Atish Patra, and Frank Shipman
Computer Science & Engineering, Texas A&M University
College Station, TX 77843-3112
(Sampath, apatra, shipman)@cse.tamu.edu

ABSTRACT
A user often interacts with multiple applications while working on a task. User models can be developed individually at each of the individual applications, but there is no easy way to come up with a more complete user model based on the distributed activity of the user. To address this issue, this research studies the importance of combining various implicit and explicit relevance feedback indicators in a multi-application environment. It allows different applications used for different purposes by the user to contribute user activity and its context to mutually support users with unified relevance feedback. Using the data collected by the web browser, Microsoft Word and Microsoft PowerPoint, combinations of implicit relevance feedback with semi-explicit relevance feedback were analyzed and compared with explicit user ratings. Our results are two-fold: first we demonstrate the aggregation of implicit and semi-explicit user interest data across multiple everyday applications using our Interest Profile Manager (IPM) framework. Second, our experimental results show that incorporating implicit feedback with semi-explicit feedback for page-level user interest estimation resulted in a significant improvement over the content-based models.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Information filtering, Retrieval models*

Keywords
User interest modeling, implicit and explicit feedback, personalized information delivery

1. INTRODUCTION
Perhaps due to the difficulty in expressing a more precise query, many queries consist of only a few keywords to model the real information need. These short queries often contain only marginally informative content about user's actual intention and therefore may have difficulty returning content relevant to the user's desired topic. Such query term mismatch is compounded by synonymy and polysemy [10], resulting in user confusion.

In order to mitigate the inherent ambiguity of queries, web search engines employ search personalization to customize search results based on the inferred interests of the user. The belief is that detailed knowledge about a user's interests, i.e. the *user interest model*, can improve the support of searching and browsing activities as every user has a particular goal and a distinct combination of context and background knowledge [35].

Even though personalized information delivery has the potential to provide users accurate results relevant to search intentions, personalization is particularly challenging due to two key issues. First, it requires identifying the interests of users in semi-persistent user profiles. Estimating user preferences in a real user interaction with a web search engine is a challenging problem, since the interactions tend to be more noisy than a controlled setting [2]. Second, given the user preferences recorded in a user profile, personalized information delivery requires a way to alter the presentation of search results to reflect those preferences. This paper is focused on the first of these problems. The particular approach being explored here looks to broaden current techniques by including a variety of direct and indirect evidence of interest across multiple applications.

Real-world personalization is often dynamic in nature and information delivered to the user can be automatically personalized and catered to individual user's information needs [25]. However, people interact with different applications, and have extra information about the content they are interacting with. These interactions results in implicit feedback (e.g., click-through data, reading time) and semi-explicit feedbacks (e.g., annotations) data that varies depending on their task and the type of information being explored. For example, a user may examine a list of search results in a web browser; or PDF Reader to examine the contents of individual documents; she may use a note-taking tool to keep track of interesting snippets; and she may use word processing applications or a presentation tool to author her own interpretation of what she has found. Therefore, a user model from a single application is unlikely to be as effective as a user model based on the aggregate activity across applications [4].

1.1 Contributions
We have previously reported [5] on the aggregation of semi-explicit feedback across a web browser and customized organization tool and its use. Here, we present a software framework and server for using both semi-explicit and implicit relevance feedback affects resulting user models in the context of multiple everyday applications. One objective of the research is to collect, measure and evaluate the predictive power of implicit and semi-explicit relevance indicators in a multi-application environment.

The rest of this paper is as follows: Section 2 describes related work in multi-application interest modeling and relevance feedback; Section 3 describes the system architecture; Section 4 explains how the activity data is turned into user models; Section 5 describes the collection of the corpus; Section 6 analyzes the

JCDL'15, June 21–25, 2015, Knoxville, Tennessee, USA.
 ISBN 978-1-4503-3594-2/15/06…$15.00
DOI: http://dx.doi.org/10.1145/2756406.2756914

results from evaluating alternative user modeling approaches with the corpus data; and Section 7 presents discussion, conclusions and some possible future work.

2. RELATED WORK

Our work is informed by related and prior work in the areas of multi-application user modeling and relevance feedback.

2.1 Multi-Application User Modeling

User models can be developed by adapting the content consumed or produced by the user, and their specific task, background, history and information needs [31]. These models can bring users' attention to valuable content via personalized presentations. Recognizing the user interest based on observed user activity is confounded by idiosyncratic work practices. As a result, systems that aggregate evidence of user interest from a wide variety of sources are more likely to build a robust user interest model.

There are two main approaches to user modeling in a component-based architecture. These vary based on the degree of centralization of the user models. Decentralized (or distributed) user modeling had its roots in agent-based architectures; here fragments of user model are kept and maintained by each independent application. In a centralized approach, the integrated user model is stored in a central server and the model is then shared across several user-adaptive applications. These include user modeling servers such as IPM [5], CUMULATE [9], UMS[21] and PersonisAD [3]. Another important distinction among user modeling approaches is whether the model is represented via features or content (see Table 1). Feature-based user models define a set of feature-value pairs representing various aspects of the user, such as interest in a specific category or a level of knowledge in a specific area. Content-based approaches take into account the user's area of interest, as an example, the textual content of documents the user has previously indicated as relevant. These systems generate recommendations by learning user needs with the analysis of available rated content.

Table 1: Related work in multi-application user modeling architectures and software frameworks

	Centralized	Distributed
Feature-based	PersonisAD [3], UMS [21]	Mypes [1], Life-log sharing [15]
Content-based	IPM [5], CUMULATE [9]	G-profile [6]

PersonisAD is a framework for building ubiquitous computing applications. It defines a user model based on data gathered from different sensors and combines their preferences using resolvers to provide a tailored experience. CUMULATE is a generic modeling server developed for a distributed E-Learning architecture to help students select the most relevant self-assessment quizzes by inferring their knowledge of a predefined set of topics based on authored relationships among activities in the educational applications and topics. UMS is a user modeling server based on the LDAP protocol which allows for the representation of user interests using a predefined taxonomy for the application domain. External clients can submit and retrieve information about users using the arbitrary components that perform user modeling tasks on these models.

In Mypes [1], the authors introduce a cross-system user modeling on the social web based on interoperable distributed model where a single vector-based user model is built using hand crafted alignment rules to map between different social web applications (e.g. Flickr, Twitter, and Delicious). In [15] authors present a distributed, decentralized architecture for sharing and re-using logged data from different systems using standalone agents with the help of broker for a successful exchange. G-profile [6] provides a general-purpose, flexible user model system based on abstract protocol to interact with and concept mapping between user data among applications. In [29], the authors present a vision of a P2P architecture to generate and maintain a distributed user model based on pre-defined information exchange templates. Each peer acts as a stand-alone user model agent which only handles information from a single source. In [11] , the authors present a model for achieving user model interoperability by means of semantic dialogues in a P2P manner.

A number of the related approaches for multi-application interest modeling require a predefined set of potential interests/taxonomy or require pair-wise alignment rules to be developed that map interests between applications. In our approach the set of user interests and the distinctions between them are constructed based on the content encountered rather than pre-agreed upon by the contributing applications. In comparison, our system extends prior work on IPM [4, 5, 18] and enables the comparison of the effectiveness of user models via unified relevance feedback

2.2 Relevance Feedback

User modeling can be viewed as a form of relevance feedback. Relevance feedback has a history in information retrieval systems that dates back well over thirty years and has been used for query expansion during short-term modeling of a users' immediate information need [20]

Implicit interest indicators are based on user actions rather than on explicit value assessments. During a search task, readers indicate their interest in documents by how they interact with them: by how much of the document they examine (e.g. how far into a document they scroll); and through other behaviors and events that are specific to the tools they are using. For example, the Curious Browser [13] records various types of implicit feedback include aspects of mouse usage, keyboard usage and the time spent viewing documents.

Explicit feedback requires users to assess the relevance of documents or to indicate their interest in certain aspects of the content. Explicit feedback has the advantages that it can be easily understood, is fairly precise and requires no further interpretation [13]. Explicit feedback can be recorded in the form of user ratings of documents' "relevance score", "readability score" and "topic familiar before" ratings [37]. WebMate [12], InfoFinder [22], and contextual relevance feedback [14, 23] learn and keep track of user interests incrementally as users provide explicit assessments of pages they examine. Some user actions, particularly annotations, and bookmarking, can be interpreted as semi-explicit feedback in that the user's action is clear evidence of their desire

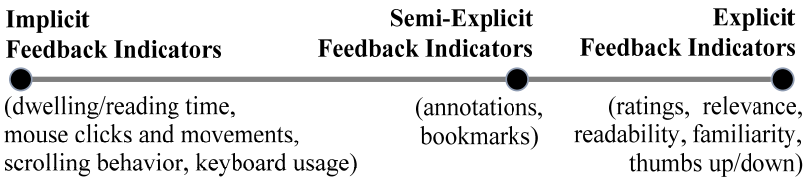

Figure 1: Types of relevance feedback indicators

to re-access this content. A user can mark-up a portion of a document by highlighting a paragraph or attaching an electronic sticky note. Not all reading results in annotations. Annotations are most likely when people read materials crucial to a particular task at hand and are infrequent when reading for fun [34].

Figure 1 shows how user actions form a continuum from implicit to explicit feedback. There is a clear tradeoff between the quantity and quality when comparing implicit feedback with explicit feedback. Explicit feedback indicators are higher in quality but lower in quantity because it is rather burdensome to enter a rating for every item a user liked or disliked [24]. On the other hand, implicit feedback indicators are abundant in quantity but lower in quality because they must be interpreted by heuristic algorithms that make assumptions about the relationships between the observable low-level actions and the high level goals of users. In [28], authors evaluated the costs and benefits of using implicit feedback indicators over explicit feedback indicators. The results suggested that the implicit ratings can be combined with existing explicit ratings to form a hybrid system to predict user satisfaction. In [16], authors showed that implicit and explicit positive feedback complement each other with similar performances despite their different characteristics. This implies that systems can be designed to use the correlation between implicit and explicit feedback to tune the interest modeling algorithms based on implicit feedback.

In this research, we combine semi-explicit and implicit feedback together in a multi-application environment to infer users' information preferences.

3. System Architecture

The Interest Profile Manager (IPM) is a multi-application environment based personal profile server (see Figure 2) to support search personalization. The IPM collects user activity across many applications and infers user interests using this collected implicit and semi-explicit interest information. It also shares the inferred user interests with registered applications that ask for it. We also presents a generic client stub to show that any application that can be modified to include the interest profile client software and communicate with the IPM enabling user interest modeling capability.

We have used Mozilla-Firefox as the application to present search results and also to visualize recommendations and three other applications: PDFPad which is an acrobat add-on; IPCWord which is a Microsoft Word add-on; IPCPowerPoint which is a Microsoft PowerPoint add-on. Records of user activity in PDFPad, Mozilla, MS Word and MS PowerPoint are stored in the IPM and drive the visualizations that the IPM generates for each of the application registered for relevant notification request. An interest profile is made up of the aggregated heterogeneous interest evidence collected from these different IPM clients.

The IPM defines the XML communication interface so that other application clients can interact with IPM over TCP/IP. The IPM framework includes two modules involved in estimating the user interest, the Estimation Manager and the Estimation module which is again decomposed to 3 sub-modules: Multi-Application Weighting module, implicit feedback module and explicit feedback module. The Estimation Manger provides a generic high level interface to the other modules within the IPM and also enables multiple modules to estimate the user's interests using different algorithms. In the Multi-Application Weighting module (see section 4.2 for further discussion), each application is assigned a weight based on the particular user's activities in the various applications. These learned weights are used to merge the estimated interests from the different applications when modeling the overall user interest. The implicit and explicit relevance modules handle the implicit and explicit feedback indicators respectively. The combined outputs from these two modules are used to estimate the final unified user interests for a search task.

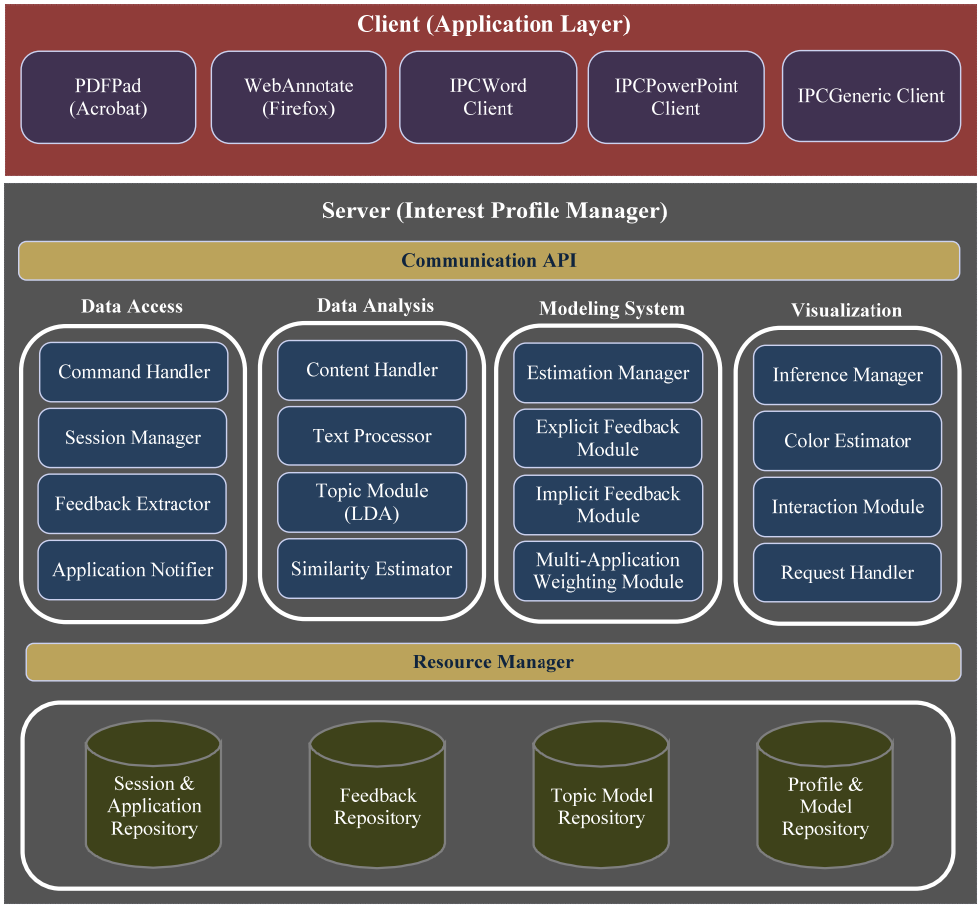

Figure 2: Interest Profile Manager architecture and software components

The Resource Manager communicates with data repository to update the user interests according to the user activity data sent from application clients. The Data Repository also saves session data both in terms of contextual and temporal features so that the user activity can be defined as a group of search tasks related to each other in order to make inferences about evolving information needs. This is

particularly important because if we are able to accurately identify changes to the users' information seeking intent, then we will be in a better position to limit the application of particular inferences about user interests [19]. The Data Repository also saves both feedback data and application data received from application clients for further processing at the estimation modules.

3.1 Interest Representation

Although each application has unique information that may be used to gauge human interest, this interest assessment needs to be sharable among the different applications to be useful in building the complete interest model of a user. The IPM depends on an abstract XML representation for receiving interest-related information from applications and for broadcasting inferred interest to client applications. Because we realize that we cannot foresee all of the ways different applications will allow users to interact with documents, the representation is extremely general and extensible. Thus an interest profile consists of a document identifier, an application identifier, and a list of application-specific attribute/value pairs. In this way, new applications only have to inform the IPM of the attributes and how they demonstrate user interest when registering.

While some of these applications support two-way communication, this is not required; an application could merely provide information to the IPM or only receive interest information from the IPM. PDFPad and WebAnnotate support two-way communications while Microsoft Word and PowerPoint support one-way communication. Applications also can be categorized into (i) *Consumption Applications*, for examining existing content; and (ii) *Production Applications*, for creating content.

3.2 Interest Extraction

Whenever a document is opened in Microsoft Word or PowerPoint, event handlers are registered for user events. Event handlers save each interaction and their values locally and send them in XML format to IPM. Additionally, the content of the document and document characteristics are sent to the IPM at the time of closing the document. Similarly, WebAnnotate parses raw text to identify every paragraph when a new web page is opened. It also appends mouse and keyboard events in a buffer and saves the color and relevance score assigned to each annotation until the browser is moved to the background. All the raw information is sent to IPM in an XML format at focus out event or at the web page close event. The buffer is reset once the focus is brought back to the web page.

3.3 Explicit Feedback

During an information gathering activity, useful documents may be long and cover multiple subtopics; users may read some segments and ignore others. The browser plug-in WebAnnotate [5] enables basic annotation capabilities so that users can make persistent annotations on web pages and passages and get suggestions within these documents based on estimated user interests. The interest classes can be defined based on annotations' color, type and content in WebAnnotate. To identify segments of new or unread documents to bring to the user's attention, these classes are then compared against the segments of the document currently displayed in WebAnnotate generated by the text-tiling algorithm. When a match is identified, an underline (based on the intensity of the inferred interest value) of the appropriate color for the class is used to signal the similarity. In Figure 3 the user has opened the Wikipedia page for the Human Genome Project and highlighted text related to the history of the project. It can be seen that other paragraphs are underlined with the same color indicating that they are similar to the passage highlighted.

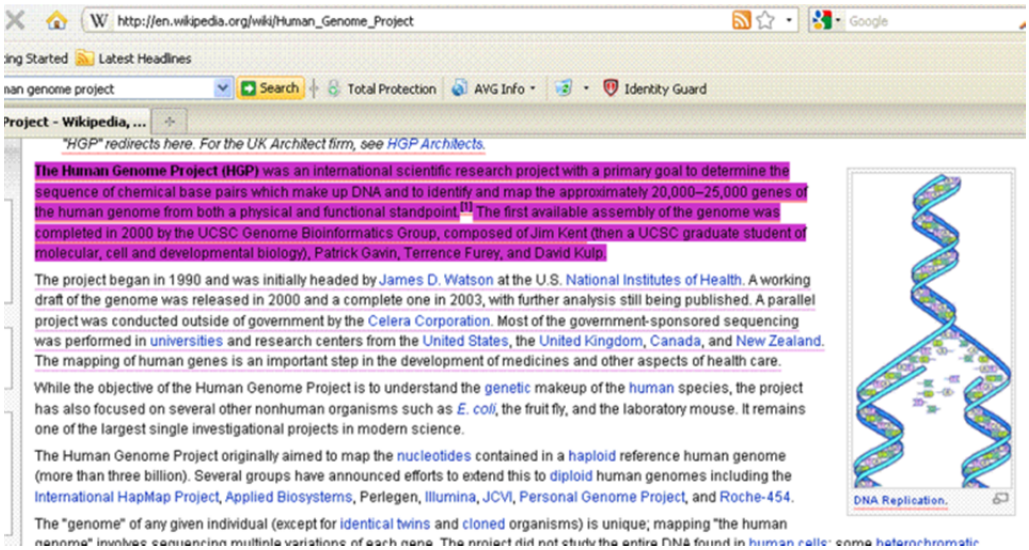

Figure 3: User highlights and system generated recommendations underlined

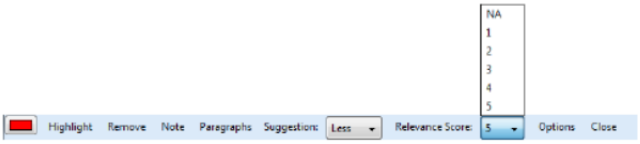

Figure 4: WebAnnotate toolbar for rating paragraphs

In the current study, WebAnnotate was extended to include three types of explicit ratings for content: "page relevance", "page familiarity", and "paragraph relevance" on a 5-point scale after each paragraph annotation, WebAnnotate allows the user to mark individual paragraphs as relevant to their task (see Figure 4).

A user might also use Microsoft Word or PowerPoint applications to open, read or modify some documents. The user's actions while working on these applications can also be used to infer some type of user's interests. MS Word and PowerPoint consider all the data in one document to belong to a single interest class. The default color of the application is used to define the interest class.

3.4 Implicit Feedback

We utilize a set of the implicit feedback indicators during a document reading activity to characterize the interactions between the user and documents. These document reading activities include user actions during a passive reading in a consumption application (web browser or PDF reader). This consists of time spent in a document, number of mouse clicks, number of text

selections, number of document accesses and characteristics of user scrolling behaviors such as number of scrolls, scrolling direction changes, time spent scrolling, scroll offset, total number of scroll groups. Furthermore, we collect time spent on a production application (MS Word or PowerPoint), focus in/out and other formatting activities. Table 2 summarizes the user events and document attributes collected from both production and consumption applications during this research study.

Table 2: Interest indicators from applications

Interest Category/ Application	Microsoft Word/PowerPoint	Browser (Firefox)
User characteristics	Click, double click, right click, focus in/out, total Time, edit time, idle time, away time	Click, double click, right click, focus out, total Time, reading time, away time, number of scrolls, number of scrolling direction changes
Document characteristics	Size, number of characters, images, links, last access time, number of slides, text boxes	Images, links, document relevance and familiarity score (explicit)
Textual characteristics	Text edited (semi-explicit)	Text annotated (semi-explicit)

The interest profile broadly contains three types of interest indicators, characteristics of the user, the document as a whole, and the textual content of the document. The user features are derived from implicit feedback data. All these features vary from one user to another as they heavily depend on the individual practices. Document features are high level features of the documents that are the same across users. Finally, document text features are generated from the user's annotations in consumption applications and from the user's produced content from production applications. Document text content provides evidence of more focused interest than the general document features. Such evidence is important when identifying the specific parts of documents that are expected to be relevant.

Another type of feature important in this work is content similarity. Content similarity metrics are used to measure the overlap between the textual content of the user's previous interactions and any future text content. These similarities are computed between text considered valuable to the user (user authored or annotated text) and all other paragraphs displayed in the browser. The similarity score represents the user's interest expressed through the textual content. In this work, Latent Dirichlet Allocation (LDA) is used to compute the content similarity using Hellinger Distance measure (see section 4.1 and 6.1 for further discussion) and are then normalized to be between [0-1] using max-min normalization.

4. MODELS OF USER INTEREST

The IPM uses the document attributes (e.g. metadata, term vectors, user-assigned color) to determine classes of user interest. Attributes of the document as a whole and textual characteristic of document segments are selected based on evidence of interest in individual documents. To aid in the creation of descriptions of document classes, the IPM includes term vector and metadata analysis capabilities as well as text tiling capabilities to allow clients and the IPM to analyze text at the sub-document level. Currently, user-assigned annotation color is used to identify the known members of an interest class while the identification of documents and document components similar to that class is based on the other document attributes and user characteristics.

The next subsections describe the use of topic modeling for similarity assessments of textual content in the user model or of potential value to the user, the weighting of features across the different applications, and the development of semi-explicit and unified feedback models.

4.1 Topic Modeling of Textual Content

Before introducing our topic modeling approach for inferring user interests, we first give a brief review of the statistical model LDA and its parameters used in this research study. LDA [8] is a hierarchical Bayesian model that assumes each document is a finite mixture of a set of topics K and each topic is an infinite mixture over a set of topic probabilities. Unlike clustering methods, LDA does not assume that each document can only be assigned to one topic. Given a document collection, we use LDA to find a set of topics discussed in the document collection. Each topic is represented as a set of words that have a higher probability than others to appear in the text unit related to the topic. Based on the probability distribution of words in each topic, we can calculate the probability that each document may contain a topic and obtain a document-topic assignment.

We set LDA parameters; a number of topics $K = 5$ to match the number of topic clusters anticipated (see section 6.1 and Figure 5 for a detailed discussion on topic selection), two smoothing parameters $\alpha = 0.01$ and $\beta = 0.01$ [27]. As words are the only observable variables in an LDA model, conditional independence holds true for the outputs of LDA model which are document-topic and topic-words distributions Φ and Θ.

For a corpus containing D documents, the parameters, the $D \times K$ matrix of document-topic probability distribution per each document and the $K \times W$ matrix of topic-words probability distribution per each topic must be learned from the data. Parameter fitting is performed using collapsed Gibbs sampling [30] with sampling and burn-in iterations set to 1 and 5 respectively. We look at the difference in the content from two text units by first computing the LDA document-topic distributions Φ_i and Φ_j $(i, j = 1..K, i \neq j)$ and then by calculating the divergence between these two document-topic distributions. The smaller the divergence is, the stronger the associated similarity is.

We performed an evaluation to determine the feasibility of topic modeling divergence methods in our context and to select among alternative topic modeling approaches (this is described in detail in section 6.1). Based on those results, we use Hellinger distance [7] to compare the similarity between document-topic distributions.

$$D_{LDA+H}(\Phi_i || \Phi_j) = \sqrt{\frac{1}{2}\sum_{i,j=1}^{K}\left(\sqrt{\Phi_i} - \sqrt{\Phi_j}\right)^2} \qquad (1)$$

4.2 Multi-Application Weighting

Once we have user, document, and textual characteristics as well as textual similarity measures, we need to weight the various features to predict the likelihood of interest in the target. Rather

than using one set of weights for all users, we train the interest model using weighted K nearest neighbor (WKNN). This enables weights to adapt to the user-specific patterns present in the feature space. The weights for the features result in a classifier algorithm that tries to predict relevance score for each paragraph on a 5-point scale. From here onwards, we denote *C* as the relevance label.

In this work, we have combined two variants of KNN, i.e., attribute-weighted and distance-weighted KNN to a build our weighted KNN classifier. By introducing a feature weight component in the distance metric (Equation 2), the quality of the feature is also considered in addition to the difference in value of the feature. Thus, more useful features are given more weight while the less useful features have less weight in the ultimate distance measurement. As a result, useful features have greater impact on the distance function compared to irrelevant features.

$$d(x,y)_w = \sqrt{\sum_{j=1}^{d} w_{cj}^2 (y^j - x^j)^2} \quad \textbf{(2)}$$

Where c = class(x), x ∈ F, w_{cj} = weight of feature j belonging to class c

Since we intend to learn the individual importance of each feature corresponding to each class, we have implemented a normalized version of the class dependent RELIEF algorithm, NCW-R [26]. All the feature weight vector values are initialized to zero and updated iteratively by processing each data point *x* in *X* as per Equation 3.

$$w_c = \sum_{x \in X_c} \left\{ \sum_{z \in WKNN(x,c)} -|x-z| + \sum_{\substack{z \in WKNN(x,\acute{c}) \\ \acute{c} \notin c}} |x-z| \right\} \Big/ N_c \quad \textbf{(3)}$$

4.3 Semi-Explicit Feedback Model

In this section, we first focus on the user interest model based on semi-explicit and implicit relevance feedback. For the semi-explicit model, we use baseline-LDA to infer content similarity and use it in the user interest estimation to determine how likely a page or a segment is of interests to a user.

Suppose at time t, the user has annotated a segment from document d_{ti} whose previous annotations (from same user) are $a_1, \ldots, a_n$. We update our baseline-LDA model by modified Rocchio algorithm [32, 33] computing the centroid vector of all annotations created by the user for the given task and interpolates it with the previous source document vector to obtain an updated term vector (Equation 4). In this context we define the set of annotations as the combination of the relevant user annotations from the browser and the produced text from content producer applications (MS Word or PowerPoint).

$$\vec{Q}_t = \lambda \vec{Q}_{t-1} + (1-\lambda)\frac{1}{n}\sum_{i=1}^{n} \vec{a}_i \quad \textbf{(4)}$$

Where $\vec{Q}_{t-1}$ is the previous source vector, n is the number of annotations the user created immediately following the current annotation, and λ is the parameter that controls the influence of the annotations on the inferred user model. In our experiments, λ is set to 0.5.

4.4 Unified Relevance Feedback Model

Previous work shows that implicit relevance feedback alone is not adequate to estimate the interest of a user during document interactions in some situations [24, 36]. The results suggested that the implicit ratings can be combined with existing explicit relevance data to form a hybrid system to predict user interest.

For a target document d_{ti}, we define a scalar valued interest prediction from the observations of user behavior as

$$r_i = \mu R_E(i) + (1-\mu) R_I(i), \quad 0 \le R_E(i) \le 1, \quad 0 \le R_I(i) \le 1 \quad \textbf{(5)}$$

Where $R_E(i)$ is the similarity score estimated from semi-explicit feedback model, $R_I(i)$ is an implicit feedback estimated from the following equation, and $\mu = 0.8$ is a heuristically tuned scaling factor representing the relative importance of the implicit feedback. We calculate $R_I(i)$ from,

$$R_I(i) = \sum_{j \in F} w_j f_j(i) \quad \textbf{(6)}$$

Where w_j is the weight for each feature *j* of the implicit feedback generated from WKNN. All the features were normalized to zero mean and unit variance.

5. MULTI-APPLICATION ACTIVITY AND CONTENT RELEVANCE COLLECTION

31 undergraduate and graduate students (ages 21 to 40) were recruited to perform a set of four tasks requiring the use of the Firefox web browser with the WebAnnotate extension, Microsoft Word and Microsoft PowerPoint. All participants reported spending at least 1-3 hours daily browsing the Internet. None of the participants had any prior experience with WebAnnotate.

Participants were given the task of writing summaries and generating short slide presentations on topics in four different domains (technology, science, finance, and sports; shown in Table 3) based on a set of eight web resources per domain. The instructions suggested that each task would take about 30 minutes, but that they could continue working as long as they needed to.

The resources provided were selected from the top documents returned from a Google query on the topic and were chosen to include pages with varying degrees of relevance to each task. Table 3 includes the average and variance of post-task relevance scores assigned by participants for the documents per task. It shows that each task contained both relevant and non-relevant web pages in similar proportions.

Table 3: Task topics with mean and variance of post-task document relevance assessments

Task No	Task Name	Relevance Score Mean and Variance
1	How does Google Glass work?	3.55 ± 0.96
2	What is mars one project?	3.23 ± 1.11
3	How to improve your credit score?	3.53 ± 0.98
4	What are the rules of American football?	3.52 ± 1.01

User activity data in the three applications and post-task relevance assessments of each document were collected. Activity data collected during the tasks included all the features originally

described (in Table 2). Due to experimental setup, this data required preprocessing. For example, as it is expected due to the data collection process, document features such as last access time, creation time, and last write time features are not informative because each individual task lasted approximately 30 minutes. Thus, these features are not considered during the evaluation process. In total, the data captured includes 34 potentially useful features out of 48 features.

In addition to the post-task page level assessments of relevance, each participant was requested to annotate and rate individual segments of documents, so that each segment in a page could be considered as a unique piece of content with the goal of the interest model learning to identify relevant segments in web pages. Pre-processing of the data assumes any segment that was not explicitly annotated and rated by a participant was irrelevant ($C = 1$). At the end of the tasks we conducted a survey about participant's prior knowledge of the applications involved, understanding of tasks and other details. The average score for the question "How comfortable were you doing the tasks" is 4.35 on a scale from 1 to 5 (1 being Lowest & 5 being Highest). This indicates that participants did not have many issues comprehending the topics.

Small segments were also removed from consideration; any segments with less than 10 words are ignored from the data set to avoid noise. We ignored data collected for tasks when participants did not generate the requested document or slides and for participants that did not annotate at least fifty paragraphs across the four tasks. Finally, since the web pages shown to the participants are real web pages and there may be some unwanted segments (comments, page headers) in the content. We removed 6247 such data instances during data filtering stage. Final dataset includes 33212 data instances across 108 tasks available for model evaluation

5.1 Evaluation Metrics

We evaluate our models by examining their performance in interest prediction in both page-level and paragraph-level interest modeling. We use Root Mean Square Error (RMSE) to measure the rating prediction quality where a smaller RMSE value indicates better performance.

Given that our primary goal is to learn the user's preference from her relevance feedback and use these to identify relevant document content, we consider the standard information retrieval domain evaluation metrics such as precision, recall, harmonic mean (F1), and mean average precision (MAP) to compare the performance of alternative user modeling techniques. MAP gives us an overall sense of how well we identify relevant estimations to recommend from sent of annotation content.

6. RESULTS

A number of subcomponents of our approach to unified relevance feedback for multi-application user interest modeling were evaluated. We used data from an earlier data collection activity that included annotations and post-task relevance assessments to test the feasibility of alternative topic modeling techniques.

6.1 Topic Modeling Approach Selection

We evaluated alternative topic modeling approaches within our context to determine how well they would work with the type of data available (a small collection of small and large segments of annotated or authored text). We applied LDA to compute the probability distributions of topics for two or more selections of textual content. We then used three distance measures of the divergence between these probability distributions and compared those assessments to the user-provided assessments. The three distance measures are: the Hellinger Distance (H), the Kullaback-Leibler divergence (KL), and the Jensen-Shannon divergence (JSD). The algorithmic details of these similarity measures are beyond the scope of this paper. Additional information about these similarity measures are available in our previous work [18]. In addition, we also evaluated the performance of a Non-negative Matrix Factorization (NMF) model to the three LDA-based techniques. Additional information about the NMF and its parameters used in this research study are available at [17].

To compare these approaches and before the data collection effort described in Section 5, we collected a set of text selections from web documents that indicated relevance to given search tasks. The data was based on 17 participants selecting the relevant paragraphs (text segments) from a set of 20 pre-selected web documents for each of five different information gathering tasks. This resulted in a total of 1267 text segments being selected across the 100 documents.

To assess the quality of the topic modeling alternatives, we used each of the user-selected text segments to predict the remainder of that user's selections based on the similarity metrics. When the user-selected paragraph reached a similarity value of 0.5 (experimentally chosen to have reasonable performance) it was assumed to be recommended by the system. When a system-generated recommended by the system was indeed one of that user's other selections, it was counted as a true positive. When a paragraph in the text did not reach that threshold it was counted as a true negative. Table 4 presents the resulting average precision, recall, F-measure and accuracy across the 5 search tasks.

Table 4: Performance comparison of 4 similarity measures

	Precision	Recall	F1	Accuracy
LDA+H	0.944	0.367	0.499	**0.722**
LDA+KL	**0.954**	0.350	0.485	0.719
LDA+JSD	0.736	**0.548**	**0.576**	0.713
NMF	0.814	0.418	0.500	0.692

We also examined the effect of varying the number of latent topics in the LDA model on performance. Figure 5 shows the overall accuracy of the different distance measure for 5, 10, 15, 20, 25 topics across the 5 information selection tasks. From these results, we first observe that the effect on the final performance is consistent for all three LDA models.

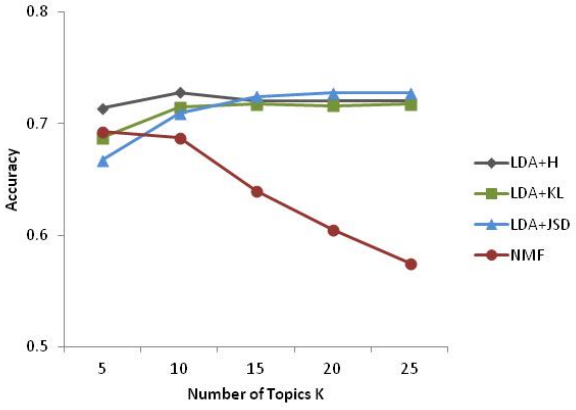

Figure 5: Impact of varying the number of latent topics

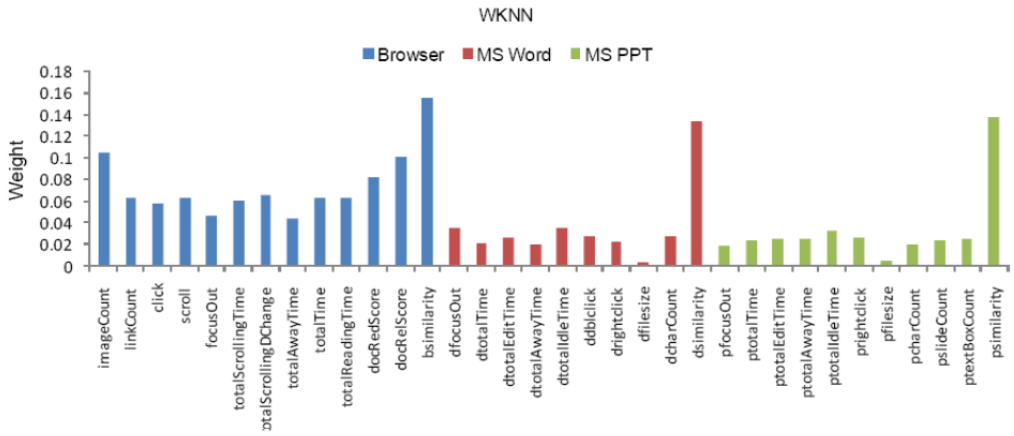

Figure 6: Comparison of feature weights computed from WKNN

The Hellinger distance measure offers the best overall accuracy among the similarity measures. Therefore, the Hellinger distance was used for the remainder of our experiments involving further analysis of our unified relevance model.

6.2 Multi-Application Feature Weights

We explored the use of WKNN to assign weights to the various features in our unified model to predict the likelihood of interest.

The feature weight values are obtained after averaging 200 iterations of the WKNN classifier. The training data set is generated by randomly selecting 70% data points from the entire data set and the remaining 30% is treated as test data for each iteration. The optimal parameter K=5 for the WKNN is selected based on performance after a 5-fold cross validation.

In WKNN, features computed (see Figure 6) from the content-consumer application have higher weights than the features from the content-producer applications except for content similarity. One interpretation of this is that similarity to content being produced by the user is such a strong signal that other features from content-production applications are not needed to help interpret that assessment.

The same cannot be said of content consumption applications. While content similarity is also the strongest feature for the browser, many other features also (including measures of clicks, scrolling, and reading) have strong weights. As opposed to the results from the content production applications, this shows that when assessing activity in the browser, it is important to gauge just how much interest the user has in the content, not just that the content was visited. Each of the three applications contributed one of the three highest strength features. This reinforces the potential for multi-application interest models to improve personalized information delivery via visualizations or recommendations. Feature weight comparison results indicated that WKNN performed well in understanding the importance of individual features of user activity. It also indicated that while content similarity is important across all applications, content consumption applications benefit considerably from additional features in order to interpret the perceived value of that content.

6.3 Unified User Model Performance

Once the particular topic modeling and evidence weighting schemes were determined based on the results in Sections 6.1 and 6.2, the overall user modeling approach could be examined. The central question being how the unified user model would perform relative to simpler models. To compare the performance of semi-explicit and unified feedback we compared the performance of classifiers provided with the different sets of features and report on the resulting classifications.

We performed our evaluation on page-level user interest estimation by running each user data through the three levels of interest models from baseline-LDA (text edited from production applications), semi-explicit (data from previous model + text annotated from consumption application), and unified (data from previous two + implicit relevance feedback through equation (5)). Each evaluator provided RMSE on the relevance of each page. The RMSE results (see Table 5) for the 4 tasks were computed by averaging the values obtained per each task performance. Although baseline-LDA (M=1.31, SD=0.14) and semi-explicit models (M=1.29, SD=0.05) are quite close; t(3)=0.9459, p=0.414, there was a significant difference in the RMSE for baseline and unified (M=1.21, SD=0.12); t(3)= 8.2641, p=0.0037, and semi-explicit and unified; t(3)= 3.9641, p=0.0287. In all cases the unified relevance model improvement over the semi-explicit relevance models is statistically significant. This demonstrates the importance of implicit relevance feedback indicators in interest predictions.

Table 5: Page-level performance of interest models

	Page-Level RMSE			
	Task-1	**Task-2**	**Task-3**	**Task-4**
Baseline-LDA	1.180	1.315	1.239	1.515
Semi-Explicit	1.126	1.326	1.258	1.463
Unified	**1.097**	**1.198**	**1.162**	**1.388**

Clearly the unified approach was of value when locating whole resources of interest. But being able to identify relevant segments within the pages is also important for personalized information delivery. We were thus particularly interested in these models performance in this respect.

To examine this segment-level performance we compared the ordering of the segments' similarity to the user models for each task performed by each user to that user's ordered rating of those segments. We calculate MAP and F1 for each task, judging a segment as relevant when it was annotated by the user. Results are shown in Figure 7 and Table 6. Unfortunately, the implicit data captured is limited to page-level analysis (we do not know what particular content was being presented when users performed each recorded event). Therefore we only compare the baseline model and the model including semi-explicit content. Table 6 points out the benefit of exploiting paragraph-level user interest via user annotations. MAP improvement of semi-explicit model is both substantial and significant over the baseline-LDA.

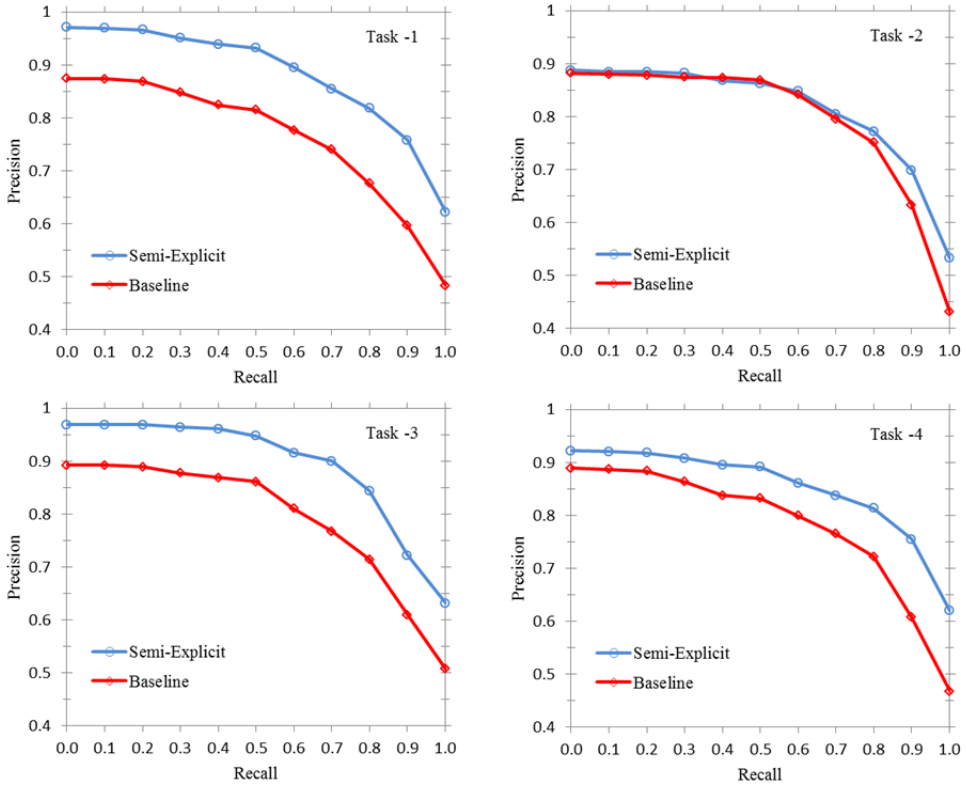

Figure 7: Precision-recall curves. Segment-level performance comparison

Table 6: Segment-level performance of semi-explicit interest models

	Segment-Level							
	Task-1		Task-2		Task-3		Task-4	
	MAP	F1	MAP	F1	MAP	F1	MAP	F1
Baseline-LDA	0.6276	0.5308	0.6371	0.5486	0.6586	0.5739	0.6293	0.5376
Semi-explicit	**0.7827**	**0.6208**	**0.6943**	**0.5568**	**0.7912**	**0.6391**	**0.7488**	**0.5804**

7. DISCUSSION AND CONCLUSION

The work presented in this paper addresses a rarely investigated topic: the potential of aggregating activity across multiple applications for user interest modeling. While there are theoretical or software frameworks for distributed user modeling, assessments of modeling techniques are almost always reported in terms of single applications. In this work, we present and evaluate a multi-application modeling technique that combines implicit and semi-explicit feedback across multiple everyday applications.

Our system and tool set supports a wide range of potential applications communicating with the user interest server. To affect the contents of the user interest model an application must be augmented to capture some information about content and its usage. The features described are occasionally specific to the applications (e.g. MS Word and PowerPoint, Firefox) but similar features would be available in most content producer and consumer applications involving text. Thus, the overall architecture and approach will generalize across a wide range of software applications. To the best of our knowledge, this is the first software framework designed to share explicit and implicit relevance feedback among applications.

The evaluation of the alternative modeling techniques involved collecting activity data and post-task relevance assessments for a common type of activity: rapidly browsing/reading content and writing a report or presentation based on that content. While other types of information tasks exist, this is a frequent and broad enough category of task to warrant investigation.

The experimental results show that incorporating implicit feedback in page-level user interest estimation resulted in significant improvements over the original models, using both baseline and semi-explicit data. Furthermore, incorporating semi-explicit content (e.g. annotated text) with the authored text is effective in identifying segment-level relevant content. Our results open up many possibilities for using unified feedback in predictive tasks, especially in the context of search personalization. Since we have a model that relates this unified feedback to ratings, we can use methods used for explicit feedbacks on unified data. In the future, we plan to study how semi-explicit feedback can be combined with implicit feedback for segment-level assessment and in additional personalized information delivery contexts.

8. ACKNOWLEDGMENTS

This work was supported in part by National Science Foundation grant DUE–0938074.

REFERENCES

[1] Abel, F., et al., *Cross-system user modeling and personalization on the social web.* User Modeling and User-Adapted Interaction, 2013. 23(2-3): p. 169-209.

[2] Agichtein, E., E. Brill, and S. Dumais. *Improving web search ranking by incorporating user behavior information*. in *ACM SIGIR*. 2006. p. 19-26.

[3] Assad, M., et al., *PersonisAD: Distributed, active, scrutable model framework for context-aware services*, in *Pervasive Computing*. 2007, Springer. p. 55-72.

[4] Badi, R., et al. *Recognizing user interest and document value from reading and organizing activities in document triage*. in *IUI*. 2006. ACM: p. 218-225.

[5] Bae, S., et al. *Supporting document triage via annotation-based multi-application visualizations*. in *JCDL*. 2010. p. 177-186.

[6] Bennani, N., et al. *Multi-application profile updates propagation: a semantic layer to improve mapping between applications*. in *WWW*. 2012. p. 949-958.

[7] Bishop, C.M., *Pattern recognition and machine learning*. Vol. 1. 2006: springer New York.

[8] Blei, D.M., A.Y. Ng, and M.I. Jordan, *Latent dirichlet allocation.* the Journal of machine Learning research, 2003. 3: p. 993-1022.

[9] Brusilovsky, P., S. Sosnovsky, and O. Shcherbinina, *User modeling in a distributed e-learning architecture*, in *User Modeling 2005*. 2005, Springer. p. 387-391.

[10] Carpineto, C. and G. Romano, *A survey of automatic query expansion in information retrieval.* ACM Computing Surveys (CSUR), 2012. 44(1): p. 1.

[11] Cena, F. and R. Furnari, *A model for feature-based user model interoperability on the web*, in *Advances in Ubiquitous User Modelling*. 2009, Springer. p. 37-54.

[12] Chen, L. and K. Sycara. *WebMate: a personal agent for browsing and searching*. in *AGENTS*. 1998. p. 132-139.

[13] Claypool, M., et al. *Implicit interest indicators*. in *IUI*. 2001. p. 33-40.

[14] Harper, D.J. and D. Kelly. *Contextual relevance feedback*. in *Information interaction in context*. 2006. p. 129-137.

[15] Iyilade, J. and J. Vassileva. *A Decentralized Architecture for Sharing and Reusing Lifelogs*. in *UMAP Workshops*. 2013. Citeseer.

[16] Jawaheer, G., M. Szomszor, and P. Kostkova. *Comparison of implicit and explicit feedback from an online music recommendation service*. in *proceedings of the 1st international workshop on information heterogeneity and fusion in recommender systems*. 2010. p. 47-51.

[17] Jayarathna, S., A. Patra, and F. Shipman, *Learning Topic Models for Multi-Application User Interest Modeling*, 2014, Texas A&M University.

[18] Jayarathna, S., A. Patra, and F. Shipman. *Mining user interest from search tasks and annotations*. in *CIKM*. 2013. p. 1849-1852.

[19] Jones, R. and K.L. Klinkner. *Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs*. in *CIKM*. 2008. p. 699-708.

[20] Kelly, D. and J. Teevan. *Implicit feedback for inferring user preference: a bibliography*. in *ACM SIGIR Forum*. 2003. p. 18-28.

[21] Kobsa, A. and J. Fink, *An LDAP-based user modeling server and its evaluation.* User Modeling and User-Adapted Interaction, 2006. 16(2): p. 129-169.

[22] Krulwich, B. and C. Burkey, *The InfoFinder agent: Learning user interests through heuristic phrase extraction.* IEEE Expert, 1997. 12(5): p. 22-27.

[23] Limbu, D.K., et al. *Contextual relevance feedback in web information retrieval*. in *Proceedings of the 1st International Conference on Information interaction in Context*. 2006. p. 138-143.

[24] Liu, N.N., et al. *Unifying explicit and implicit feedback for collaborative filtering*. in *CIKM*. 2010. p. 1445-1448.

[25] Lu, Z., D. Agarwal, and I.S. Dhillon. *A spatio-temporal approach to collaborative filtering*. in *RecSys*. 2009. p. 13-20.

[26] Marchiori, E., *Class dependent feature weighting and k-nearest neighbor classification*, in *Pattern Recognition in Bioinformatics*. 2013, Springer. p. 69-78.

[27] McCallum, A.K., *Mallet: A machine learning for language toolkit.* 2002.

[28] Nichols, D., *Implicit Rating and Filtering*, in *The fifth delos workshop on filtering and collaborative filtering*1997.

[29] Paraskevopoulos, F. and G. Mentzas. *A Peer to Peer Architecture for a Distributed User Model*. in *UMAP*. 2014.

[30] Porteous, I., et al. *Fast collapsed gibbs sampling for latent dirichlet allocation*. in *KDD*. 2008. p. 569-577.

[31] Renda, M.E. and U. Straccia, *A personalized collaborative digital library environment: a model and an application.* Information processing & management, 2005. 41(1): p. 5-21.

[32] Rocchio, J.J., *Relevance feedback in information retrieval.* 1971.

[33] Shen, X., B. Tan, and C. Zhai. *Implicit user modeling for personalized search*. in *CIKM*. 2005. p. 824-831.

[34] Shipman, F., et al., *Identifying useful passages in documents based on annotation patterns*, in *Research and Advanced Technology for Digital Libraries*. 2003, Springer. p. 101-112.

[35] Sieg, A., B. Mobasher, and R. Burke. *Web search personalization with ontological user profiles*. in *CIKM*. 2007. p. 525-534.

[36] Wang, B., et al., *Expectation-Maximization collaborative filtering with explicit and implicit feedback*, in *Advances in Knowledge Discovery and Data Mining*. 2012, Springer. p. 604-616.

[37] Zigoris, P. and Y. Zhang. *Bayesian adaptive user profiling with explicit & implicit feedback*. in *CIKM*. 2006.p. 397-404.