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 which is important for User Modeling systems. To address this issue, we have developed topic modeling based UIMaP: User Interest Management & Personalization, which helps in building a multi-application environment by inferring user profiles based on users’ interactions with multiple applications, coupled with an analysis of the characteristics and content of the documents they are interacting with. It allows different applications used for different purposes by the user to contribute and share their models to mutually support the activity with different forms of feedback. These interest profiles are used to estimate the user’s interests and broadcast it to the participating applications. The novelty of our approach lies in the use of topic models to generate fine-grained models of user interest and visualizations that direct user’s attention to documents or parts of documents that match user’s inferred interests. User annotations in multi-applications are used to help generate personalized visualizations for user’s search tasks. Based on 1267 user annotations from 17 users, we show the performance comparisons of Non-negative Matrix Factorization (NMF) model with Latent Dirichlet Allocation (LDA) topic models: LDA+TopN=1, and LDA+TopN=2.

Keywords – Latent Dirichlet Allocation, Non-Negative Matrix Factorization, Multi-Application, Search Personalization, User Interest Modeling.

I. INTRODUCTION

With the explosive growth of the web and other information exchange tools, vast amount of information has become accessible to everyone. But, a user extracts or operates on very limited amount of information while working on any particular task that indirectly defines her interest model. Users also need much time and efforts to search and select the information before they can find out what they really want. All forms of user modeling systems make use of these user interactions to model, predict and to present information items that are likely to be of interest to the user. But normally, people interact with different applications depending on the type of information or the kind of operation they intend to do on it, resulting in the interests being scattered across multiple applications. For example, a user may consult a web browser that performs a list of search results; she may use MS Word or PDF reader to examine the content of individual documents; she may use a note-taking tool to keep track or summarize the tasks. The user model extracted from a single application will not be effective unless a combined user model aggregated from the individual interest models at each application is developed [1].

A typical recommender system can support its users by recommending documents that best match users’ interests during an open-ended information gathering task, thereby ensuring that their time is spent efficiently on the most relevant documents. In other words, personalized web search recommendations relies on a user’s interest model to filter away irrelevant information and cater only information relevant to a certain task a user particularly interested in. A recommender system rely on user, item and ratings, which can be mapped to an interaction matrix on users and items [2]. The resultant matrix will be sparse as most of the users will tackle only a subset of item ratings as a result of the interaction between user and the items. In the proposed research context, this interaction matrix can be represented as a mapping between documents, terms and the term weights.

This research study examines how inferred models of user interest from different applications may be used in building multi-application environments which can share these interest models. In particular, we are interested in how applications that are normally used for reading, writing and presenting documents can contribute to and share a user model to mutually support the activity with different forms of feedback. Through our present system, we have examined how inferred user interest model from different applications, may be used to generate visualizations to aid the user in performing a web search tasks.

We use a four step approach:

- Users’ interests are inferred from their interactions with a collection of relevant documents (for example, if the user annotates document content representing two documents A and B, these interactions are recorded and are taken as indications of the user’s interest in A and B).
- Cluster of user interest are identified and represented based on relationships among the document content of interest (in this step, the similarity between the annotated content of documents A and B is characterized as a cluster of user interest).
• Documents similar to one or more cluster of interest are selected (if document C has content that is similar to documents A and B, it is assumed to be in the same interest cluster as documents A and B)
• Visualizations are generated to reflect how these documents are related to the inferred interests (the visualization applied to documents A and B partially determines the visualization applied to document C; in this case, the relevant content representing document C would be highlighted).

This paper describes the overall multi-application architecture and the topic modeling algorithms we developed for user interest modeling. Section II surveys related work and section III and IV presents our system components and topic modeling algorithms respectively. In section V we discuss the results of initial user evaluations, and section VI presents conclusions and points out some future work.

II. RELATED WORKS

Nowadays, it’s a norm in many application domains such as service personalization to collect and model information about users. The main consideration is to enable different applications to understand the users, their interests and preferences’ to provide services catered specifically to their needs. Different applications organize user properties, preferences and assumptions based on the user interaction in user profiles [3]. Though the interest models can be developed separately for each individual application with relative ease, this will not guarantee a complete picture of a particular user’s interest that lies in multiple applications. A system supporting an efficient task oriented process can compute an aggregated interest model accumulated from partial models across multiple task-related applications. Our past research shows that combining evidence from a multi-application environment improves the interest model [4].

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 [5]. The relevance feedback can be (i) implicit: information can be derived by studying users behavior while using services and/or (ii) explicit: information can be gathered by a direct intervention of the users themselves by filling some kind of predefined forms [3]. Explicit feedback requires users to assess the relevance of documents or portions of documents or to indicate their interest in certain aspects of the content (e.g. identifying nouns or phrases within search results). 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 that in part rely on the tools they are using. This interest may be recorded as users interact with documents and may be characterized via feature extraction. Explicit feedback has the advantages that it can be easily understood, is fairly precise and requires no further interpretation [6]. Annotations can be interpreted as one form of explicit feedback.

Reading documents happens for many reasons: we read for fun, for general knowledge, or for some specific activity. When reading as part of an activity, we have a particular task in mind. 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 [7]. Explicit interest indicators such as annotations are based on users directly identifying which documents or portions of it are interesting.

As users read through a particular document, they begin to identify the content relevant to the task in hand. If the document text content is large, users will frequently skim or stop reading when they feel what they have is good enough. Consequently, potentially better document contents are left having never been reviewed [8]. A potential solution for this particular problem is to provide visualizations to draw user's attention to similar documents or document parts relevant to their search task [4]. User’s attention to passages of potential interest can be drawn by using colors and icons to highlight them in a document overview [9]. Spatial hypertext systems such as VIKI [10] and VKB [11], use a similar visualization techniques to provide system-identified “interesting document contents” to provide visualization aided navigation.

The task of document content filtering based on user interest can be done by representing the document collection in a vector space and then applying learning algorithms to the vector space [12]. These learning algorithms usually divided into supervised learning, unsupervised learning and semi-supervised learning. The process of supervised document filtering is usually called document classification. The document filtering based on unsupervised learning is usually called document clustering.

Topic models learn bag of words from a collection of documents without any supervision [13]. Topic models assume generative model which can be used to model a collection of documents by topics. With the assumption that a single document covers a mixture of concise topics, these generative topic models can mine topic level relations based on the words used within a document. Three major distinct approaches for topic modeling are the Latent Dirichlet Allocation (LDA) [14], Latent Semantic Analysis (LSA) [15] and Non-negative Matrix Factorization [16]. As shown by [17, 18], NMF outperforms traditional vector space approaches for document clustering such as LSA and learn concise topics with similar performance with LDA. However, NMF learns more incoherent topics compared to LDA [13].

A particular document can be encoded in a n-dimensional vector where n is the total number of terms in the corpus, hence each vector defines the relative importance of corresponding terms with respective to the semantics of the given document [19]. In this vector space model, a collection of documents can effectively represented as a document-by-term matrix with a positive weight per corresponding term presented in the document or zero value otherwise. Given a term-by-document matrix with inherent non-negativity, the
NMF [16] can learn the underlying semantics or patterns in a text collection based on non-negative lower rank factors. The documents can be reconstructed combining these learned semantic features and set of documents with common features can be represented by a cluster. In this work, we explore recommendations of web documents based on these clusters of semantically similar documents created via NMF model.

There have been research studies that perform document clustering using Non-negative Matrix Factorization (NMF) [17, 18]. In the latent semantic space derived by the NMF, each axis can capture a basic topic relevant to a particular document cluster, and each document can be represented as an additive mixture of these latent topics. The cluster membership is determined by the topic (the axis of the latent space) with which the document has the largest projection value.

III. USER INTEREST MODELING

Fig. 1 shows the overall architecture of UIMaP. The reading application (web browser) communicates with the UIMaP via the browser plug-in annotation tool. The interest profile stores inferred user interests; records of user activity in reading application. UIMaP then drives the visualizations (system generated underlines of text content) of documents based on the inferred interests the topic models generated.

A. Interest Profile

The interest profile manager (IPM)[8] plays the central role in the UIMaP. It collects and stores information about interest-related activity from document reading application and this information are processed to create a user interest profile based on UIMaP interest modeling algorithms. The IPM then estimates the user interest based on the inferred user interest profile and broadcast it to the document reading application to generate visualizations. Any application that can be modified to include the interest profile client software API can communicate with the interest profile manager enabling multi-application user interest modeling capability. Currently, WebAnnotate, Microsoft Word and PowerPoint include this interface. While the 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 it. WebAnnotate support two way communications while MS Word and PowerPoint support one way communication.

B. Explicit User Annotations

During information gathering search task, useful documents may be long, and cover multiple subtopics; users may read some segments and ignore others. In order to record which portions of the document pique the user’s interests most, an explicit interest expressions capturing tool can be used. The WebAnnotate [4] provides basic annotation capabilities, collect data on user’s interactions with web documents, and uses interest data returned from UIMaP to create visualizations that enhance document skimming and reading. With annotation tool, users can provide explicit feedback via annotations and convey it to UIMaP with terms associated with the annotation.

![Fig. 1. UIMaP: Overview and System Components](image)

![Fig. 2. The user annotations (highlights) and system generated underlines indicating the similarity with the user generated annotations](image)

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, a thick underline of the appropriate color for the class is used to signal the similarity. For example, it can be seen in the **Error! Reference source not found.** user has opened Wikipedia page for Human Genome Project and highlighted some text related to its history. It can be seen that other paragraphs are underlined with the same color indicating that it is similar to the passages which user has shown interest by annotation.

C. MS Word and Powerpoint

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. The text content of a document is parsed into paragraphs, and each paragraph is assigned a different weight depending on how close they are to
the highlighted phrase. Highest weight is assigned to the paragraph containing the highlighted phrase, and the weight of the paragraphs on either sides (above and below) decrease, as we move farther from it. This is because; generally there is a higher probability for the user to be interested in the paragraphs closer to the phrase she has highlighted.

IV. TOPIC MODELING

A. Document Representation

We define two interest classes, *source interest class* and *target interest class*. Each of user-generated annotations in the web browser and user content in other applications (MS Word, PowerPoint) are attributed in the source interest class. Whenever user open a web page in the browser, each paragraph in the web document consider as a target interest class. The granularity in this scenario is the paragraph. We consider each paragraph of content from both source and target interest class as an individual document and create weighted document-term matrix for clustering.

We use weighted term-frequency inverse-document-frequency (tf-idf) vector to represent each individual document. Each individual annotation is weighted from predefined weight value vector and the learning of weights from individual annotation is beyond the scope of this paper. For this research, each application weight is learned heuristically. For both LDA and NMF, we multiply each document term-vector from the weight of the learned application weight vector.

Let \( W = \{w_1, w_2, \ldots, w_m\} \) be the vocabulary of the document corpus after stop word removal and stemming. The term-frequency inverse-document-frequency vector \( X_j \) of document \( d_j \) is defined by,

\[
X_j = \left[ x_{1j}, x_{2j}, \ldots, x_{mj} \right]^T, \quad x_{ij} = t_{ij} \cdot \log \left( \frac{n}{\text{idf}_{j}} \right)
\]

The \( t_{ij}, \text{idf}_{j}, n \) define the term frequency of word \( w_j \) in document \( d_j \), the number of documents containing word \( w_j \) and the total number of documents in the corpus, respectively. We create \( m \times n \) term-document matrix \( X \) and apply weighting and then use NMF to obtain document clusters from the factorization result.

B. Non-Negative Matrix Factorization

Matrix factorization is the task of approximating the matrix \( X \in \mathbb{R}^{m \times n} \) by the product of two reduced-dimensional matrices \( W \in \mathbb{R}^{m \times k} \) and \( H \in \mathbb{R}^{k \times n} \) so that \( X \approx WH^T \). Dimensions of \( W \) and \( H \) are \( m \times k \) and \( k \times n \) respectively, where \( k \) is the select number of topics for \( 0 < k \ll \min(m, n) \)[20]. Then, the minimization problem can be stated as,

\[
\min_{s.t.\ W \geq 0, H \geq 0} f(W, H) := \| X - W \cdot H \|_F^2 \tag{1}
\]

Where \( \| \cdot \|_F \) is the Frobenius norm. We note that other objective functions can be used to measure the error of the approximation instead of the Frobenius norm, but it is the most appropriate when errors are normally distributed[21].

The \( H \) is initialized to zero and \( W \) to some randomly generated matrix where each \( W_{ij} > 0 \) and these initial estimates are updated with alternating iterations of NMF multiplicative update rules [16]. The NMF algorithm successively updates \( H \) and \( W \) which fixing the other, by taking a step in weighted negative gradient direction for the \( f(W, H) \) as defined in (1).

\[
W_{ij} \leftarrow W_{ij} - \zeta_{ij} \left[ \frac{\partial f}{\partial W_{ij}} \right] \\
\equiv W_{ij} + \zeta_{ij}(XH^T - WH^T)_{ij} \tag{2}
\]

\[
H_{ij} \leftarrow H_{ij} - \eta_{ij} \left[ \frac{\partial f}{\partial H_{ij}} \right] \\
\equiv H_{ij} + \eta_{ij}(W^TX - W^TH)_{ij} \tag{3}
\]

where \( \zeta_{ij} \) and \( \eta_{ij} \) are individual weights for the corresponding gradient elements with following weight values,

\[
\zeta_{ij} = \frac{(W)_{ij}}{(WHH^T)_{ij}} \quad \eta_{ij} = \frac{(H)_{ij}}{(W^TH)_{ij}}
\]

Now, we can define the updating formulas:

\[
W_{ij} \leftarrow W_{ij} \frac{(XH^T)_{ij}}{(WHH^T)_{ij}} \tag{4}
\]

\[
H_{ij} \leftarrow H_{ij} \frac{(W^TX)_{ij}}{(WH)_{ij}} \tag{5}
\]

C. Latent Dirichlet Allocation

Before introducing our topic model algorithms for inferring user interests, we first give a brief review of the statistical model Latent Dirichlet Allocation (LDA) and its parameters used in this research paper. LDA is a hierarchical probabilistic generative model which can be used to model a collection of documents by topics [14]. Given LDA parameters, a number of topics \( K \), a document corpus of \( W \) distinct words, two smoothing parameters \( \alpha \) and \( \beta \), and prior distribution over document corpus, LDA can create random documents whose contents are a mixture of topics. As words are the only observable variables in an LDA model, conditional independence holds true for the outputs of LDA model which are document and topic distributions \( \theta \) and \( \phi \). For a corpus containing \( D \) documents, the parameters \( \theta \), the \( D \times K \) matrix of topic probability distribution per each document and \( \phi \), the \( K \times W \) matrix of topics must be learned from the data. The remaining parameters \( \alpha \) and \( \beta \), and \( K \) are specified by UIMaP. For the LDA models used in this paper, parameter fitting is performed using collapsed Gibbs sampling [22] to estimate \( \theta \) and \( \phi \). We use \( \alpha = 0.01 \) and \( \beta = 0.01 \) [23]. Two additional parameters for the Gibbs sampling are the number of sampling and burn-in iterations, which we set to 1 and 5 respectively.

In our experiments with LDA models, we will create similarity matrices to compare the source interest class to target interest class; hence we define proposed measures as similarities. The following LDA+TopN measures have been evaluated in our experiments.
LDA+TopN, N=1 and N=2: The simplest way to support Top-N topic probabilities is to sort the resultant document topic probability distribution in the desired order and then discard all but the first N topic tuples. Then compare the Top-N topics between source interest class to find the similarity of target interest class. The main motivation behind this method is to find document-based results, such as finding main topics of a document or finding the top topics that are most related to a specific document content or user annotation.

V. Evaluation

In this section we discuss user evaluation we have done to evaluate our proposed methods and show the results. We first describe our evolution metrics, and then experimental setup. Next we present the results from our user survey that measures the perceived quality of our user interest models.

<table>
<thead>
<tr>
<th>TABLE I. Confusion Matrix for System Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Generated</td>
</tr>
<tr>
<td>System Generated</td>
</tr>
<tr>
<td>Not-Underlined</td>
</tr>
</tbody>
</table>

A. Evaluation Metrics

Given that our primary goal is to learn the user’s preference from her explicit feedback and use these user generated annotation results to visualize relevant document content, we may consider the standard information retrieval domain evaluation metrics such as precision, recall, accuracy, F1 measure. Precision is the ratio of correctly underlined as a class to the target interest class. For example, the precision (P) of the underlined class in table is \( tp/(tp + fp) \). Recall (R) is the ratio of correctly underlined document content as a class to the actual source interest class. The recall of the underlined class in the table is \( tp/(tp + fn) \). Accuracy is the proportion of the total number of underlines that were correct. The accuracy in the table is \( (tp + tn)/(tp + fp + fn + tn) \). F1 is a measure that trades off precision versus recall. F1 measure of the underlined class is \( 2PR/(P + R) \).

B. The Data Set

Since our approaches are based on annotated document contents, we need to collect user’s annotations for a set of search tasks. In the meantime, users are required to supply a set of annotations using the WebAnnotate tool that reflects and relevant to the main idea of the given search tasks. The data is composed of five search tasks and twenty web documents. Documents are preprocessed and removed graphics and tags before experiments. We recruited 17 students to annotate the documents relevant to the given search tasks. Users are told to make annotations freely which reflects the main idea of the given task and relevance to the given documents. We collected total of 1267 annotations.

C. Results

It is important to identify the optimum number of topics in each LDA model and in the NMF model as they determine the quality of the user interest modeling. We calculate the \( P/R/F1 \) values at first 5, 10, 15, 20, 25 topics respectively.

The results are shown in Fig. 3. From these results, we first observe that the effect on the final performance is not quite large in all three models for the given duration of topics. K=5 gives the best average F1 measure for all three models. All three models show constant change in performance after 25 topics, and it is possible that the limitations in the corpus size may be a possible culprit in this particular situation.

The Fig. 4 shows the overall performance of all three algorithms. The improvement on recall and F1 of LDA+TopN=2 is very significant. This is very encouraging since recall is a more important factor in generating user interest models to provide relevant content as suggestions/recommendations. The results demonstrate that the LDA+TopN=2 consistently outperform the other two methods in terms of hit recall and F1 measure. From this comparison, it can be concluded that the proposed approach is capable of making accurate and effective search suggestions.

VI. Conclusion and Future Work

In this paper we have presented topic model based study of the effectiveness of three variant of algorithms for user interest modeling. We extracted user annotations from 17 users and these annotations are used to help generate personalized visualizations for user’s search tasks. Three different topic models are produced: NMF, LDA+TopN=1, and LDA+TopN=2. Performance comparisons between these three topic models are made. This paper also describes the usage of user interest models using topic modeling as a basis for visualizations that draw a user’s attention to similar documents and to portions of documents that match these interests.

We have evaluated the effectiveness of the visualizations in recommending interesting new documents and passages within documents based on what the user has explicitly indicated interests using annotations. The classification of documents and parts of a document into different user interests in the current UIMaP is based on explicit user annotations in a multi-application environment. In the future we plan on
incorporating implicit feedback data as an expressions of interest (e.g. scrolling, click-through records) to create a complete picture of the user’s interest model.

REFERENCES


Fig. 4. Performance comparisons of different models. (a) Precision, (b) Recall, (c) F1 measure, and (d) Accuracy.