Clustering user preferences using W-kmeans

Christos Bouras, Professor

Computer Engineering and Informatics Department,
University of Patras and Computer Technology Institute
and Press "Diophantus", N. Kazantzaki,
Panepistimioupoli, 26500
Patras, Greece
bouras@cti.gr

Vassilis Tsogkas, M.Sc.

Computer Engineering and Informatics Department,
University of Patras, Greece, Panepistimioupoli, 26500
Patras, Greece
tsogkas@ceid.upatras.gr

Abstract— **Although commonly only document clustering is suggested by Web mining techniques for recommendation systems, one of the various tasks of personalized recommendation is categorization of Web users. In this paper, a method for clustering navigation patterns of Web users is proposed. We adapt the WordNet-enabled W-kmeans algorithm, an enhancement of standard k-means algorithm which uses the external knowledge from WordNet hypernyms and that has been previously used for document clustering, to user profile clustering by analyzing the users' historical data. We also investigate the effects this approach has on the recommendation engine by evaluating the overall performance it has in terms of precision – recall on our online recommendation system.**

User clustering, session identification, recommendation system, personalization, k-means, W-kmeans

I. INTRODUCTION

Object clustering refers to the process of partitioning a collection of objects into several sub-collections based on their similarity of contents. For the case of user clustering, each sub-collection is called a user cluster and includes users that have revealed similar appeals in their selections of text articles while browsing through a document collection. Clustering has been proven to be a useful technique for information retrieval by discovering interesting information kernels and distributions in the underlying data. In general, it helps constructing meaningful partitions of large sets of objects based on various methodologies and heuristics. It plays a crucial role in organizing large collections. For example, it can be used a) to structure query results, b) form the basis for further processing of the organized topical groups using other information retrieval techniques such as summarization, or c) within the scope of recommendation systems by affecting their performance as far as suggestions made towards the end users are concerned.

Web mining focuses on finding natural groupings of Web resources or Web users. We could roughly divide Web Mining into three basic categories [6]. Firstly, Web content mining, where information is extracted from the content of pages and links (i.e. not from the users themselves). Secondly, Web Structure Mining, where structural information about hyperlinks and organization plays a predominant role. And thirdly, Web Usage Mining which focuses on extracting useful usage patterns from the users' behavior. Clustering of Web users is a particular research topic of Web Usage Mining that aims towards describing generic trends in users' behaviors within some particular time (e.g. a specific time-window).

Cooley et al. in [5] introduced the term Web Usage Mining and explained it as the "automatic discovery of user access patterns from Web Servers". Since then, the field has been explored within the scope of Web personalization by various works, e.g. [7] and [8]. In [10], the authors take into account basically two types of usage patterns and cluster them in order to build generic navigational profiles, without minding the order of accesses. A method that uses attribute-oriented induction where user sessions are represented as vectors in an n-dimensional Euclidian term space is described in [8]. A visualization approach of the user choices has also been explored in [4] for navigation patterns. In [9], the authors introduce a Sequence Alignment Methodology that clusters users based on their navigation patterns. This work focuses on the order in which navigation events take place by users.

Web usage mining results to Collaborative filtering (CF) when it uses the known preferences of a group of users to make recommendations or predictions about the unknown preferences for other users. CF techniques use a database of preferences for items by users to predict additional topics or products a new user might like. They come in roughly three categories: a) memory based, like neighbor-based and item-based top-N, b) model-based, like Bayesian belief nets, latent semantic, dimensionality reduction (SVD) and c) hybrid, which combine the advantages of both categories and improve the prediction performance. Early generation collaborative filtering systems, such as GroupLens [11], use the user rating data to calculate the similarity or weight between users or items and make predictions or recommendations according to those calculated similarity values. Memory-based CF methods are notably deployed into commercial systems such as http://www.amazon.com/ and Barnes and Noble, because they are easy-to-implement

75

and highly effective. Customization of CF systems for each user decreases the search effort for users.

In [13] the authors focus on the personalized recommendation of Web pages that are adapted according to the access patterns constructed by analyzing user navigation information. They prove that the methodology of integrating user clustering within the scope of a recommendation system, while mining interesting user navigation patterns can be beneficial. Beyond the above, there is little work with regards to clustering user preferences within the scope of a recommendation system and how the above can be exploited with a significant effect to the efficiency of such a system.

Two generic categories of the various clustering methods exist: hierarchical and partitional. Typical hierarchical techniques generate a series of partitions over the data, which may run from a single cluster containing all objects to n clusters each containing a single object, and are widely visualized through a tree-like structure. On the other hand, partitional algorithms typically determine all clusters at once. For partitional techniques, a global criterion is most commonly used, the optimization of which drives the entire process producing thus a single-level division of the data. Given the number of desired clusters, let k, partitional algorithms find all k clusters of the data at once, such that the sum of distances over the items to their cluster centers is minimal. Moreover, for a clustering result to be accurate, besides the low intra-cluster distance, high inter-cluster distances, i.e. well separated clusters, is desired. A typical partitional algorithm is k-means which is based on the notion of the cluster center, a point in the data space, usually not existent in the data themselves, which represents a cluster. The family of k-means partitional clustering algorithms [15] usually tries to minimize the average squared distance between points in the same cluster, i.e. if d_1, d_2,…, d_n are the n documents and c_1, c_2,…, c_k are the k clusters centroids, k-means tries to minimize the global criterion function:

$$\sum_{t=1}^{k}\sum_{j=1}^{n} sim(d_j, c_i) \qquad (1)$$

Several improvements have been proposed over this simple scheme, like bisecting k-means [14], k-means++ [1] and many more.

WordNet is one of the most widely used thesauri for English. It attempts to model the lexical knowledge of a native English speaker. Containing over 150,000 terms, it groups nouns, verbs, adjectives and adverbs into sets of synonyms called synsets. The synsets are organized into senses, giving thus the synonyms of each word, and also into hyponym / hypernym (i.e., Is-A), and meronym / holonym (i.e., Part-Of) relationships, providing a hierarchical tree-like structure for each term. The applications of WordNet to various IR techniques have been widely researched concerning finding the semantic similarity of retrieved terms [12], or their association with clustering techniques. The use of a WordNet-based clustering approach for users has not been investigated so far.

In this paper we extend our implementation of the WordNet enhanced W-kmeans algorithm to the domain of clustering Web Users generating, thus, offline user clusters which can be used at a later stage by the other information retrieval techniques. In essence, we are able to decode the navigation patterns of users and aggregate their profiles using the W-kmeans clustering algorithm. This allows our recommendation system to suggest content that, with high probability, will be interesting to the users. Our goal is to improve the results of our information retrieval system in terms of precision / recall, and thus serve better filtered and adequate results to their users, helping in essence the decision making process. Our recommendation system, as explained in the next section, could be described as a Hybrid between content-based filtering and CF.

The rest of the paper is structured as follows: section II describes the information flow within our system and the modified components needed for user clustering. Section III presents the algorithms used, while section IV describes the evaluation process and the results. Some concluding remarks and pointers for future work are given in Section V.

II. FLOW OF INFORMATION

Fig. 1 depicts the flow of information within our recommendation system [2]. Initially, at its input stage, news articles are crawled and fetched from news portals from around the Web. This is an offline procedure and once articles as well as metadata information are fetched, they are stored in the centralized database from where they are picked up by the procedures that follow.

A key process of the system as a whole, probably as important as the clustering algorithm that follows it, is text preprocessing on the fetched article's content, that results to the extraction of the keywords each article consists of. Analyzed in [2], keyword extraction handles the cleaning of articles, the extraction of the nouns [3], the stemming as well as the stopword removal process. Following, it applies several heuristics to come up with a weighting scheme that appropriately weights the keywords of each article based on information about the rest of the documents in our database. Pruning of words, appearing with low frequency throughout the corpus, which are unlikely to appear in more than a small number of articles, comes next. Keyword extraction, utilizing the vector space model, generates the term-frequency vector, describing each article as a 'bag of words' (words – frequencies) to the key information retrieval techniques that follow: article categorization, summarization and clustering. Our aim towards increasing the efficiency of the used clustering algorithm is to enhance this 'bag of words' with the use of an external database, WordNet. The above characteristics of our system give its content-based nature. This enhanced feature list, feeds the k-means clustering procedure that follows. In this work, clustering is achieved via regular k-means using the cosine similarity distance measure:

$$d(a,b) = \cos(\theta) = \frac{a \cdot b}{|a||b|} \qquad (2)$$

where |α|, |b| are the lengths of the vectors α, b respectively and the similarity between the two data points is viewed by means of their angle in the n-dimensional space. It is important to note, however, that the clustering process is

independent of the rest of the steps, meaning that it can easily be replaced by any other clustering approach operating on a word-level of the input documents.
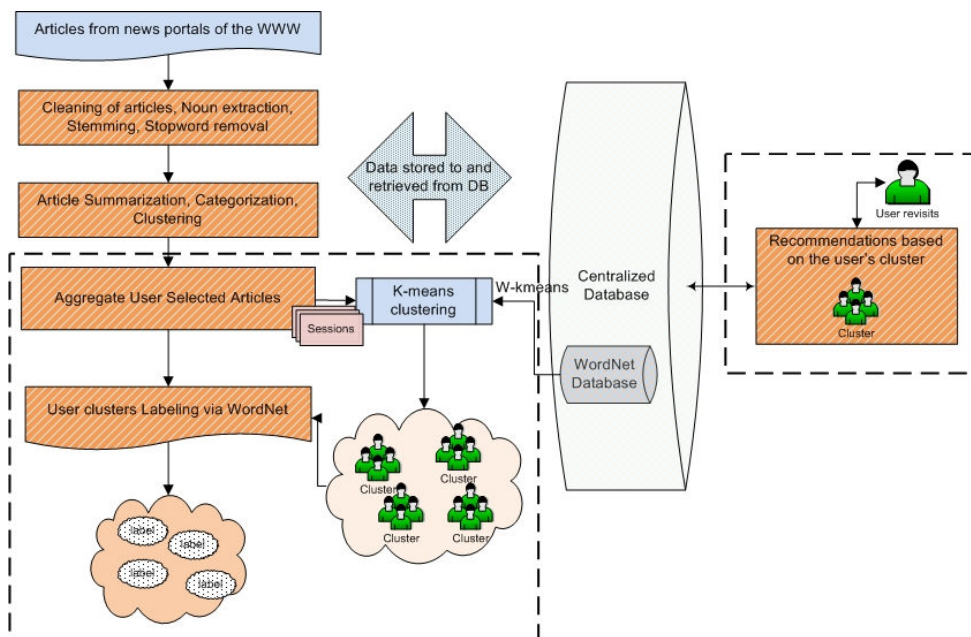

Figure 1. Flow of Information.

For each user viewing news articles, we keep track of the selected actions which characterize a user session. A session is defined as the list of selected articles that a user has decided to view for a minimum duration and within a limited time frame, both of which are fine-tuned at the experimentation stage. The selected articles contained in those sessions are then aggregated at a keyword level generating a time-limited user profile. User profiles from multiple users and timeframes are then clustered using the W-kmeans algorithm forming profile clusters.

W-kmeans is a novel approach that extends the standard k-means algorithm using the external knowledge from WordNet hypernyms for enriching the "bag of words" used prior to the clustering. The W-kmeans algorithm enhances the user profiles with hypernyms deducted from the WordNet database, using a heyristic manner. Those profile clusters, being essentially user clusters, are used at the recommendation stage to enhance the system's usage experience by providing better adapted results to users revisiting the site. Following the session clustering procedure, the resulting clusters are labeled using our WordNet cluster labeling mechanism, which however is beyond the scope of this paper. When a user comes back, his clustered profile is recalled and articles belonging to the clustered sessions of his profile are extracted and sent as viewing recommendations back to the user. Suggested articles do not belong to the ones the user has already visited and also are not closely related to articles that the user has marked negatively in the past.

The approach previously described is essentially the collaborative filtering nature of our recommendation system, which practically involves related users to the decision making process. We expect that combining this method with our keyword extraction (content-based) mechanism, the recommendations towards users will ameliorate.

III. ALGORITHM APROACH

The proposed approach consists of three major algorithmic components that are used for: a) the offline process of identifying the sessions of users navigating through the recommendation system, b) the offline process of clustering of the detected sessions, and c) the online process of recommending news articles to the users based on the clustered profiles. Those components are: session identification, clustering of user sessions and recommendation stage.

A. Session Identification

The identification of sessions within a user's browsing history is achieved using the following algorithm.

```
Algorithm find_sessions
Input: history //time  window for sessions to be extracted
Output: Sessions[] //discovered sessions array
    viewing_threshold = 30 // at least 30 seconds
    session_threshold = 10 * 60 // at most 10 minutes
    previous_user = NULL
    current_session = NULL
```

```
    while fetch from DB (user, viewed article, timestamp,
viewing_time) {
if (viewing_time < viewing_threshold || timestamp < history)
        continue
if (current_session.username != user) {
// Since this is sorted by username, when a new user is found this
means a new session begins
    if (current_session.username!="" && current_session.articles
!empty)
            Sessions[]+=current_session
    current_session.username = user;
    current_session.user_id = user_id;
    current_session.start = timestamp;
    current_session.articles.add(article_id);
}
else {
// If the user is the same as before but the access time for this
article exceeds the time limit, a new session begins
    if (timestamp – current_session.start) > session_threshold ) {
        if (current_session.username!=""&&current_session.articles
!empty)
            Sessions[]+=current_session
    current_session.username = user;
    current_session.user_id = user_id;
    current_session.start = timestamp;
        current_session.end = timestamp;
    current_session.articles.add(article_id);
    }
    else {
        // The access time for this article does not exceed the time
limit
    current_session.articles.add(article_id);
    current_session.end = timestamp;
    }
return Sessions[]
```

Algorithm 1. Discovering Sessions in user's access paths.

B. Clustering User Sessions

Once user sessions have been extracted, we proceed to the core procedure described in this paper: session clustering. As described in Algorithm 2, for each user session, we aggregate the news articles that make up this session. At the next step we enrich the keywords that belong to the session using related hypernyms from the WordNet database. Initially, for each given keyword of the session, we generate its graphs of hypernyms leading to the root hypernym (commonly being 'entity' for nouns). Following, we combine each individual hypernym graph to an aggregated one. An example of the hypernym generation and aggregation process is depicted in Fig. 2. There are practically two parameters that need to be taken into consideration for each hypernym of the aggregate tree-like structure in order to determine its importance: the depth and the frequency of appearance. It is observed that the higher (i.e. less deep, walking from the root node downwards) the hypernym is in the graph, the more generic it is. However, the lower the hypernym is in the graph, the less chances does it have to occur in many graph paths, i.e. its frequency of appearance is low. In our approach, those two contradicting parameters are weighted using (3).

$$W(d,f) = 2 \cdot \frac{1}{1 + e^{-0.125(d^3 \frac{f}{TW})}} - 0.5 \qquad (3)$$

where d stands for the node's depth in the graph (starting from root and moving downwards), f is the frequency of appearance of the node to the multiple graph paths and TW is the number of total words that were used for generating the graph (i.e. total keywords of the session).
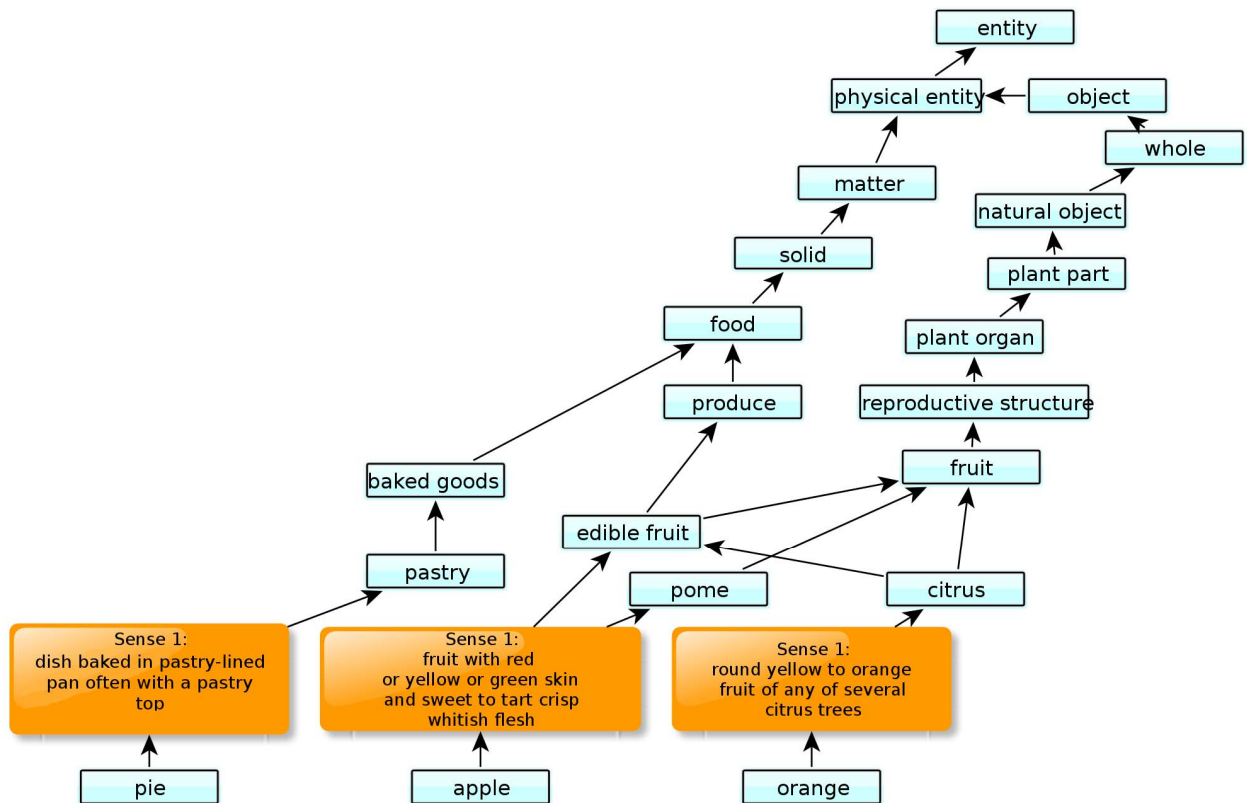
Figure 2. Aggregate hypernym graph for three words: 'pie', 'apple', 'orange'

Algorithm **clustering_user_sessions**
Input: sessions, number of clusters
Output: session to cluster assignments
for each session s {
　　for each article a belonging to s
　　　session.kws += fetch 20% most frequent k/ws for a
　wordnet_enrich(s) // See Algorithm 3
　}
　clusters = kmeans(sessions)
return clusters

Algorithm 2. Clustering User Sessions using WordNet.

Algorithm **wordnet_enrich**
Input: session s
Output: session with enriched list of keywords
total_hypen_tree = NULL
kws = fetch 20% most frequent k/ws for s
　for each keyword kw in kws {
　htree = wordnet_hypen_tree(kw) //extract the hypernym tree
from WordNet
　for each hypen h in htree {
　　if (h not in total_hypen_tree)
　　　h.frequency=1
　　　total_hypen_tree ->append(h)
　　else

　　total_hypen_tree ->at(h)->freq++
　　　}
　}
　for each h in total_hypen_tree {
　　calculate_depth(h)
　　weight = 2 ((1/(1+ exp(-0.0125 * (h->depth ^3 * h->freq/
kws_in_wn->size)))) - 0.5))
　}
sort_weights(total_hypen_tree)
important_hypens = (kws ->size/4)*top(total_hypen_tree)
return kws += important_hypens

Algorithm 3. Enriching user sessions using WordNet hypernyms.

C. Recommendation Stage

When a user returns to the system, his cluster has already been determined, based on the recorded past sessions. It is now safe to assume that selections made by other users belonging to the same user cluster are more likely to be of interest to him/her rather than random articles. Based on this simple assumption, we adjust our recommendation stage to suggest news articles to the user as explained in Algorithm 4. In general, we only keep 10 of the most frequently occurring articles in the user's cluster in order to avoid overloading the user with information.

```
Algorithm cluster_based_recommendation
Input: user u, cluster c
Output: suggestions
suggestions [] = NULL
    num_sug = 10 // number of suggestions
sessions = recover_user_clustering_info(u, c)
for each s in sessions // for users that belong to the same cluster
    suggestions = top_suggestions(s, num_sug, suggestions)
return suggestions
top_suggestions // finds the articles with the highest frequency
Input: session s, total suggestions num_sug, suggestions
Output: suggestions [] // top suggestions for the parsed sessions
for each article a in s
    if freq(a) > min(freq(suggestions))
        suggestions [] += article
return suggestions
```

Algorithm 4. Recommending news articles based on user clusters.

IV.   EXPERIMENTAL PROCEDURE

For the evaluation process of the W-kmeans algorithm within the scope of user clustering, we used a set of 10,000 news articles obtained from major news portals like bbc.com, cnn.com, reuters.com, etc. over a period of 6 months. These articles were evenly shared among the 7 base categories that our system features: business, politics, health, education, science, sports and entertainment. After the preprocessing procedure and most notably stemming and noun identification, we kept for each article its list of stemmed nouns. Notice that duplicate articles originating from different sources have been removed from the dataset based on their title and main body. We also used the navigational patterns that we recorded for the 50 registered system users at the same period. Those are the selected articles as well as the time spent on them as explained in Algorithm 1.

For our evaluation metrics we used Clustering Index (CI) and F-measure. In order to determine the efficiency of each clustering pass, together with the right number of clusters for your dataset, we used the evaluative criterion of Clustering Index (CI), defined as:

$$CI = \bar{\sigma}^2 / (\bar{\sigma} + \bar{\delta}) \tag{4}$$

where $\bar{\sigma}$ is the average intra-cluster similarity

and $\bar{\delta}$ is the average inter-cluster similarity. Intuitively, since the most efficient clusters are the ones containing articles close to each other (within the cluster), while sharing a low similarity with articles belonging to different clusters, CI focuses on increasing the first measure (intra-cluster similarity) while decreasing the second (inter-cluster similarity). The F-measure, as defined in (5) is a weighted combination of the precision and recall metrics and is employed to evaluate the accuracy and efficiency of our recommendation system when using user profile clustering. We define a set of target articles, denote C, that the system suggests and another set of articles, denote C', that are visited by the user after the recommendation process. Moreover, $doc(c'_i, c_j)$ is used to denote the number of documents both in the suggested and in the visited lists.

$$F(c'_i, c_j) = 2 \cdot \frac{r(c'_i, c_j) p(c'_i, c_j)}{r(c'_i, c_j) + p(c'_i, c_j)} \tag{5}$$

where:
$$r(c'_i, c_j) = \frac{doc(c'_i, c_j)}{doc(c'_i)}$$

and:
$$p(c'_i, c_j) = \frac{doc(c'_i, c_j)}{doc(c_i)}$$

For our first experiment we compared W-kmeans to standard k-means when applied to user clustering. The results, depicted in Fig. 3, show that W-kmeans clearly outperforms standard k-means by at least a factor of 10, thus providing clusters of users more tightly bound. Consequently, the generated clusters can capture with a better accuracy users with similar interests while successfully separating user's with contradicting interests. From Fig. 3 it can also be deducted that both of the CI graphs peak at around 30 clusters. This is a good indication about the best suited amount of clusters applicable for our dataset, a finding that shows that the W-kmeans algorithm generates fine-grained clustering results.
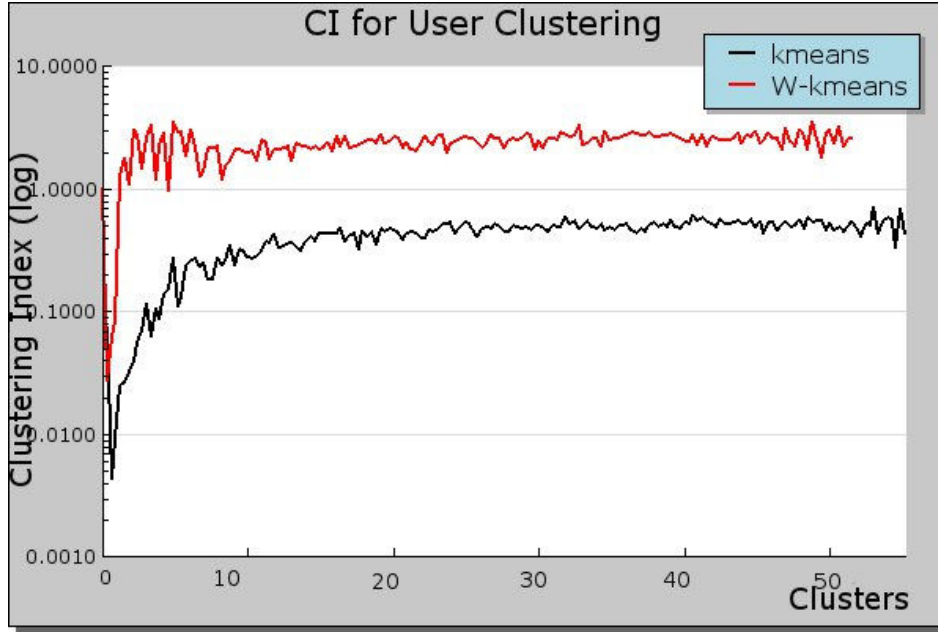
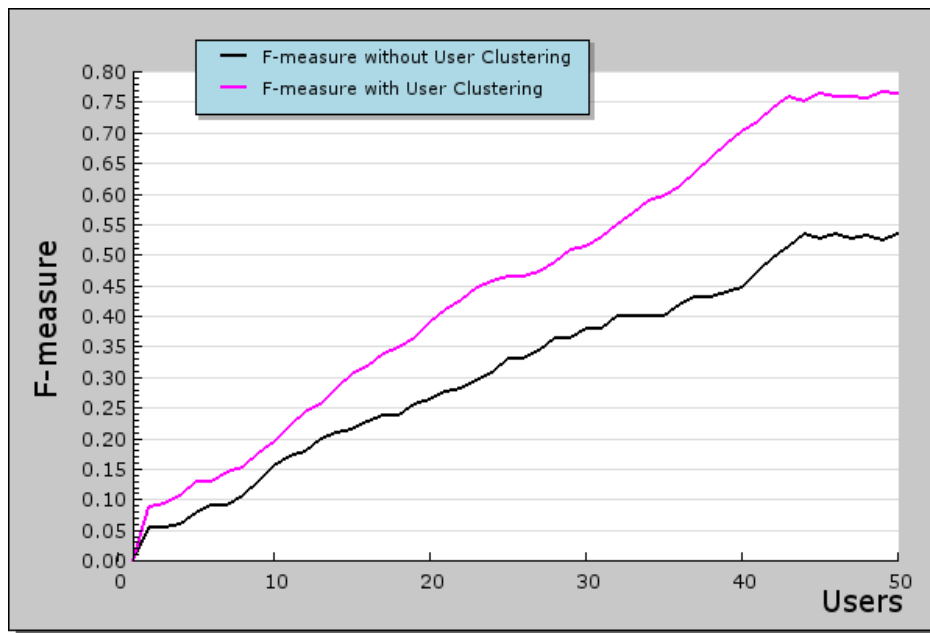Figure 3. Comparison of W-kmeans and k-means for user clustering.

Figure 4. Comparison of the recommendation engine performance.

For our second experiment we tried to determine the overall improvement of our recommendation engine when taking into consideration existing user clusters. As explained in Algorithm 4, for returning users we modified our recommendation stage to suggest 10 of the top viewed articles belonging to the user's cluster. Following, we recorded which of the suggested articles were viewed by the user within a time frame of 30 minutes. The process was repeated without the user clustering enhancement of the recommendation engine but with other heuristics, such as text categorization and personalization still enabled [2]. The results, presented in Fig. 4 show the average F-measure for each case as users increase.

From Fig. 4, we observe an average improvement of 10% with regards to the F-measure when user clustering is deployed. The efficiency also rapidly increases as more users

are taken into consideration by the system, something that is expected, given the personalization features of our recommendation engine. From a natural point of view, our experiments showed that the resulting suggestions matched the user's choices in average 7 out of 10 times. In our opinion this proves that our approach has greatly benefited the recommendation stage.

For our last experimentation procedure, we tried to determine the efficiency of the proposed methodology compared to some state of the art CF methods, like latent semantic CF, neighbor-based CF and dimensionality reduction techniques like SVD. The results on the same dataset, presented in Table 1, revealed that W-kmeans outperformed or was at least as equal as those methods in terms of F-measure average over various users.

TABLE 1. CF METHODOLOGIES COMPARISON

CF Methodology	Average F-measure over all users
W-kmeans	0,45
Latent semantic CF	0,4
Neighbor-based CF	0,3
Dimensionality reduction (SVD)	0,45

V. CONCLUSIONS AND FUTURE WORK

In this paper we presented the WordNet-enabled k-means algorithm, which explores the usage of word hypernyms extracted from the WordNet database, to the field of profile clustering as well as its application to our recommendation system. We examined the performance of this approach compared to standard k-means and discovered a 10-fold amelioration in terms of cluster coherence. Furthermore, we found an average improvement of around 10% in terms of F-measure for the resulting suggestions of our recommendation engine when used by real system users. Additionally, some basic experimentation showed that W-kmeans performs usually better compared to other CF techniques when applied to our recommendation system. We believe that the above facts prove the significance of user clustering and in particular W-kmeans to the recommendation process.

As far as future work is concerned, we are planning on incorporating cluster labeling for the generated profile clusters to the system, as well as automate the detection of the best suited number of clusters for W-kmeans that is best for the underlying data.

REFERENCES

[1] D. Arthur, and S. Vassilvitskii, "k-means++: the advantages of careful seeding," In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, 2007, pp.* 1027 – 1035.

[2] C. Bouras, V. Poulopoulos, and V. Tsogkas, "PeRSSonal's core functionality evaluation: Enhancing text labeling through personalized summaries," *Data and Knowledge Engineering Journal, Elsevier Science*, Vol. 64, Issue 1, 2008, pp. 330 – 345.

[3] C. Bouras, and V. Tsogkas, "Improving text summarization using noun retrieval techniques," *Lecture Notes in Computer Science. Knowledge-Based Intelligent Information and Engineering Systems*, Vol. 5178/2008 pp. 593 – 600.

[4] I. Cadez, D. Heckerman, C. Meek, P. Smyth, and S. White, "Visualization of Navigation Patterns on a Web Site Using Model-Based Clustering," In *proceedings of the International Conference of Knowledge Discovery and Data Mining* , 2000, pp. 280 – 284.

[5] R. Cooley, B. Mobasher, and J. Srivastava, "Grouping Web Page References into Transactions for Mining World Wide Web Browsing Patterns," In *Proceedings of the 1997 IEEE Knowledge and Data Engineering Exchange Workshop*, November 04 1997, p.2.

[6] R. Cooley, B. Mobasher, and J. Srivastava, "Data preparation for mining world Wide Web browsing patterns," *Journal of Knowledge and Information Systems*, 1999 (1) 1.

[7] M. Eirinaki, and M. Vazirgiannis, "Web mining for Web personalization," *ACM Transactions on Internet Technology*, 3(1), 2003, pp. 1 – 27.

[8] Y. Fu, K. Sandhu, and MY. Shih, "Clustering of Web Users based on Access Patterns," In *proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, San Diego, USA, 1999.

[9] B. Hay, G. Wets, and K. Vanhoof, "Clustering navigation patterns on a website using a sequence alignment method," *In proceedings of Intelligent Techniques for Web Personalization: 17th Int. Joint Conf. Artificial Intelligence*, vol. s.l, 200, 2001, pp. 1 – 6.

[10] B. Mobasher, R. Cooley, and J. Srivastava, "Automatic personalization based on Web usage mining," *Communications of the ACM*, 43(8), 2000, 142 – 151.

[11] P. Resnick, N. Iacovou, M. Suchak, M. Bergstrom, and J. Riedl, "Grouplens: an open architecture for collaborative filtering of netnews," In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, New York, NY, USA, 1994, pp. 175 – 186.

[12] G. Varelas, E. Voutsakis, P. Raftopoulou, E. Petrakis, and E. Milios, "Semantic similarity methods in WordNet and their application to information retrieval on the web," *In Workshop On Web Information And Data Management, Proceedings of the 7th annual ACM international workshop on Web information and data management*, 2005, pp. 10 – 16.

[13] F.H. Wang, and H.M. Shao, "Effective personalized recommendation based on time-framed navigation clustering and association mining," *Expert Systems with Applications* 27 (3), 2004, pp. 365 – 377.

[14] L. Yanjun, and C. Soon, "Parallel bisecting k-means with prediction clustering algorithm," *The Journal of Supercomputing*, 39, 2007, pp19 – 37.

[15] Y. Zhao, and G. Karypi, "Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering," *Machine Learning*, v.55 n.3, 2004, p.311 – 331.