Advances in Knowledge-Based and Intelligent Information and Engineering Systems
M. Graña et al. (Eds.)
IOS Press, 2012

© 2012 The authors and IOS Press. All rights reserved.
doi:10.3233/978-1-61499-105-2-555

User Personalization via W-kmeans

Christos Bouras[1,2] and Vassilis Tsogkas[1]

[1] Computer Engineering and Informatics Department, University of Patras
[2] Computer Technology Institute and Press "Diophantus", N. Kazantzaki, Panepistimioupoli
Patras, 26500 Greece
bouras@cti.gr, tsogkas@ceid.upatras.gr

Abstract. With the rapid explosion of online news articles, predicting user-browsing behavior using collaborative filtering techniques has gained much attention in the web personalization area. However, common collaborative filtering techniques suffer from low accuracy and performance. This research proposes a new personalized recommendation approach that integrates user and text clustering based on our developed algorithm, W-kmeans, with other information retrieval techniques, like text categorization and summarization in order to provide users with the articles that match their profiles. Our system can easily adapt over time to divertive user preferences. Furthermore, experimental results show that by aggregating multiple other information retrieval techniques like categorization, summarization and clustering, our recommender generates results that outperform the cases when clustering is not applied.

Keywords: User clustering, recommendation system, personalization, collaborative filtering, k-means, W-kmeans.

1 Introduction

Relying on recommendations from other people is a basic means of filtering through the vast amount of information that the average internet user comes across every day. This natural social process is assisted by recommender systems that have risen over the last years at many large electronic sites and which aim to help people shift through available sources in order to find the most interesting and valuable piece of information for them [1]. Recommendations can roughly be divided into the following approaches: a) content-based, where users are profiled by identifying their characteristic features – something that requires personal data which are difficult to harvest, b) collaborative filtering (CF), where we take advantage of the fact that people who had similar tastes in the past may also agree on their tastes in the future.

Collaborative filtering was initially introduced by the developers of one of the first recommender systems, Tapestry [2], in order to describe this personalized recommendation technique which was based on the similarity of interests. The fundamental assumption of CF is that if users X and Y rate n items similarly, or have similar behaviors, they will rate or act on other items similarly. Several matrix factorization techniques have been applied to CF, like SVD, probabilistic LSA, probabilistic matrix factorization, etc. However, combination of various algorithms typically outperforms single methods. As explained in [1], utilizing user preferences is commonplace for CF techniques allowing them to predict additional items a new user might like. CF algorithms must be able to handle sparse data and scale with the increasing numbers of users and items. Issues like synonymy, shilling attacks, data noise, and privacy protection should also be taken into consideration and dealt with.

One basic problem with CF is that it doesn't always work well due to data scarcity: each person has seen only a small fraction of the data, thus accurate predictions cannot be easily made until the coverage of users / data has increased to a significant value. One way to deal with this situation is to group people into clusters of similar interests. Thus by using symmetry, one might group articles based on whoever sees them and use article groups as opposed to mere users. A vice-versa approach is also possible: consider a group of users that have previously expressed their interest for a particular topic. A newly added article with similarities to some of the articles previously read by the people of this group might also be appealing to the rest of the group. This suggests that instead of depending on choices of single users, the cluster aggregates the needed information. Two techniques have traditionally been applied in this scenario: k-NN and clustering. Another problem with CF is that similarity scores typically do not take into consideration the user interest shifting and they also do not estimate the reliability of the user choices, leading to poor recommendation results. According to [1], in order to improve prediction performance and avoid the problems of memory-based CF algorithms, model-based CF approaches have been explored in the literature. Model-based CF techniques make use of the rating data in order to estimate or learn a model to make predictions [3]. Some examples of Model-based CF techniques are: Bayesian belief nets (BNs) CF models [4], clustering CF models [5], and latent semantic CF models [6].

Content-based filtering is another important class of recommender systems. They make recommendations by analyzing the content of textual information and finding regularities in the content. The major difference between CF and content-based recommenders is that CF only uses the user-item ratings data to make predictions and recommendations, while content-based recommenders rely on the features of users and items for predictions [7]. While CF systems do not explicitly incorporate feature information, content-based systems do not necessarily incorporate the information in preference similarity across individuals [8]. Hybrid CF techniques, such as the content-boosted CF algorithm [9] and Personality Diagnosis (PD) [10], combine CF and content-based techniques, hoping to avoid the limitations of either approach and thereby improve recommendation performance.

Clustering has proven to be a useful technique for information retrieval by discovering interesting information kernels and distributions in the underlying data. It plays a crucial role in organizing large collections. 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.

Personalized search is an important research area that aims to resolve the ambiguity of query terms. To increase the relevance of search results, personalized search engines create user profiles to capture the user's personal preferences and, as such, identify the actual goal of the input query. In reality, positive preferences are not enough to capture the fine-grained user interests. User profiling strategies can be broadly classified into two main approaches: the document-based approach and the concept-based approach. Document-based user profiling methods aim at capturing user's clicking and browsing behaviors. User's document preferences are first extracted from the clickthrough data and then used to learn the user behavior model which is usually represented as a set of weighted features. Concept-based user profiling methods aim at capturing user's conceptual needs. In [11] a method employing preference mining and machine learning to model user's clicking and browsing behavior is considered: when a user would scan the search result list from top to bottom and she skips a document, the next selected item is considered as more interesting for her.

In our previous work [12], we proposed a new clustering method, W-kmeans, which improves the traditional k-means algorithm by enriching its input with WordNet hypernyms. The WordNet lexical reference system, organizes different linguistic relations into hierarchies/hypernyms (Is-a relation) and W-kmeans uses them as a preprocessing stage before the regular k-means algorithm. We extended this algorithm in [13] to the domain of user clustering, where we investigated how user clustering alone can affect the recommender's performance.

The contribution of the current work is the evaluation of W-kmeans within a more generic framework: since we are dealing with the effective and adequate retrieval of personalized news articles that derive from the web, we present the personalization algorithm that takes into account a variety of techniques and heuristics. Our recommendation approach can be classified as 'hybrid' since it is mainly content-based with some collaborative filtering features that enhance the algorithm with the ability to automatically adapt over time to the continuously changing user choices. In contrast to CF techniques, we derive the user groups by arranging the information that is extracted from several IR techniques, like categorization, clustering and also inferred by previous user behavior. Another contribution of the current work is tackling the problem of user shifting interests by rather small but continuous user profile adjustments. Our recommender incorporates several heuristics such as viewed articles by the user, the time a user spends on reading an article, the categorization of articles by the system and the clustering of articles.

The rest of this paper is structured as follows. In Section 2 we present the information flow of our system. In Section 3 we outline the algorithmic approach of our mechanism while Section 4 presents the experimentation. Finally, in Section 5 we conclude this paper and give some pointers regarding future work.

2 Information Flow

Fig. 1 depicts the flow of information within our recommendation system [14]. Since a detailed description of the various components is beyond the scope of this paper, we will outline them briefly. Initially, at its input stage, news articles are crawled and fetched the web. This is an offline procedure, storing the articles as well as their metadata in the centralized database from where they are picked up by the procedures that follow. Html pages are stripped of unnecessary page elements (ads, css, javascript, etc). During this analysis level, our system isolates the "useful text", containing only the article's main body. By storing only the useful text, as well as some other page meta-data, the database is populated with news articles that are ready for the text preprocessing step during which the sentence separation and punctuation removal are applied. Afterwards, the noun identification step takes place and some common text extraction techniques follow: stopwords removal and stemming. The results of the procedures described in this layer are stemmed keywords either marked as nouns or not, their location in the text and their frequency of appearance in it. 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.

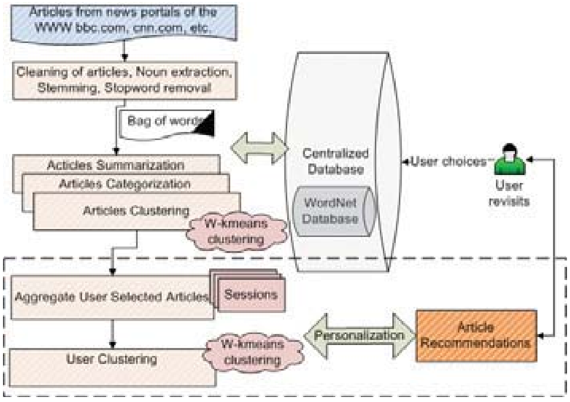

Fig. 1. Flow of Information.

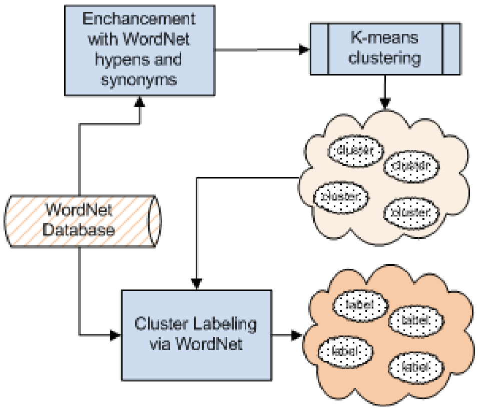

Fig. 2. Article / User Clustering.

The IR tasks of our mechanism are located in the next analysis level, where the summarization, categorization and clustering algorithms are applied. The main scope of the categorization module is to assist the summarization procedure by pre-labeling the article with a category and has proven in [14] to be providing better results. Summarization then proceeds with extracting a short but useful piece of textual information that can convey the article's meaning. As far as our clustering approach is concerned, one of our aims is to enhance this 'bag of words' mentioned earlier with the use of external databases, and in particular WordNet. This enhanced feature list, feeds the k-means clustering procedure that comes next and is depicted in Fig. 2, resulting to item clusters.

Following the core IR tasks of our mechanism, the personalization algorithm takes place. The personalization module (dashed in Fig. 1) that is described in this paper, can easily adapt to subtle user preference changes. Those changes, as expressed by the user's browsing behavior, are detected and continuously adjust her profile. The algorithm uses a variety of user-related information in order to filter the results presented to the user. Furthermore, it takes into account in a weighted manner the information originating from the previous levels regarding the summarization/categorization and news/user clustering steps. For each user viewing news articles, we keep track of the selected actions which characterize a user session. For connecting the user clustering component with our personalization algorithm, we define the notion of a session 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 [12] algorithm forming profile clusters. W-kmeans is a novel approach that extends the standard k-means algorithm by using the external knowledge from WordNet hypernyms for enriching the "bag of words" used prior to the clustering process. The W-kmeans algorithm enhances the user profiles with hypernyms deducted from the WordNet database, using a heuristic manner. Those profile clusters are used at the recommendation stage in order to enhance the system's usage experience by providing more adapted results to users revisiting the site. Following the session clustering procedure, the resulting clusters are labeled using our WordNet cluster labeling mechanism.

When a user comes back her clustered profile is recalled. Articles matching her profile are extracted and are considered for user recommendations. 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.

3 Algorithmic Approach

In this section we are presenting a novel personalization algorithm that is utilized by our recommendation system. We are explaining how the user's profile is generated, the way that it is dynamically updated and the weighting scheme that takes into account the various parameters for producing the user recommendations. Note that before the personalization procedure kicks in, the user sessions are extracted as explained in [13].

The steps that are followed by the personalization procedure are presented in Algorithm 1. When a new user is registering to the system, she states the keywords of her preference as well as the scores that describe this preference initializing, thus, her profile. This procedure is trivial and can be avoided altogether since the personalization subsystem keeps track of the user's choices and browsing history, and so the user's preferences are updated on each visit. The user's profile consists of two keyword lists: a positive one, where the user-preferred keywords are placed, and a negative one where uninteresting keywords for the user are kept. By using these lists, we can personalize the news articles and summaries with exceptional results.

The profile update procedure described in Algorithm 2, running constantly at every user's visit, takes note of the following aspects: i) the browsed articles, i.e. the ones that the user selected to view, ii) the amount of time a user spends viewing the summary or the full text of a specific article, iii) the articles that the user avoids viewing (either their summary or their full text); the above derives from the simple logical assumptions that follow. A user will most likely spend an amount of time above a certain threshold, R_{ar_thr1} or R_{sum_thr1}, reading an article's full text or its summary respectively, that is of interest for her (factor a). However, an upper bound, R_{ar_thr2} and R_{sum_thr2}, should be used for these metrics since we don't want the mechanism to mistake forgotten browsed articles for the really interesting ones.

```
Update_profile(factor a, factor b, factor c, factor d){
 Get_articles(a,b,d) //for factors a,b,d
 for each article{
  if (full article)
  if (time_viewed > Rar_thr1 && time_viewed < Rar_thr2){
     Keywords_positive = top 5 frequent keywords
     Update_list(Positive, Keywords_positive)}
 else
    if(time_viewed> Rsum_thr1 && time_viewed< Rsum_thr2){
    Keywords_positive = top 5 frequent keywords
    Update_list(Positive, Keywords_positive)}
 Get_articles(c) //for factor c
 for each article{
    Keywords_negative = top 5 frequent keywords
    Update_list(Negative, Keywords_negative) }
Get_article(lists ...){
//Recovers: i) articles and the time spent reading the article or its sum-
mary(a,b), ii) articles with negative preference(c), iii)most frequently viewed
articles by the user's cluster (d)}
Update_list(list, keywords){
    for each (keyword in keywords)
    if (keyword not in list[])
        list.add(keywords[keyword])
    else
        list.update_freq(keywords[keyword]) }
```

Algorithm 1 Personalization steps for utilizing user feedback

We found that the best thresholds for R_{ar_thr1} and R_{ar_thr2} are 30 seconds and 3 minutes respectively defining thus which article's keywords should be added (or have their weight increased) in the user's positive keywords list. The summary viewing thresholds are calculated in an analogous way: $R_{sum\_thr1} = R_{ar\_thr1} * S_{ratio}$ and $R_{sum\_thr2} = R_{ar\_thr2} * S_{ratio}$ where: Sratio = #words(summary) / #words (fulltext). Moreover, most of the times a user will select to browse articles of a topic that he finds interesting (factor b) as advertised by the article's title and/or summary. Lastly, a user will probably avoid visiting articles that he finds uninteresting and thus the keywords that represent those articles should be receiving a lessened or negated weight (factor c). In addition to the above factors (a-c), having deduced the user's cluster by following the steps described in Section 2 via W-kmeans, we can also take into account the user clustering information. More specifically, from the cluster the user belongs to we can enrich the user's positive keywords list using articles that have been frequently viewed by the cluster members. From those articles we keep the top 5 keywords which have also been previously enriched by WordNet. We call this user clustering heuristic: factor d. Using factors a-d, the personalization algorithm keeps track of the keywords that the user has expressed preference to, combined with similar preferences of people from the same cluster, and thus, the articles (containing these keywords) that she will likely be willing to read in the future. The parameter that depicts the user's preference for a keyword according to the aforementioned factors (a-d) is U_{wi} and is based on the relative frequency that the keyword has on the (positive or negative) list, a frequency that is constantly modified by the user's choices. U_{wi} derives from the following equation:

$$Uwi = rel(fr(kwi)) * (1 + Tkwi) \tag{1}$$

where T_{kwi} is the normalized total time spent on the specific keyword if it belongs to the positive list, however, if the keyword is in the negative list, T_{kwi} is set to 0 since no time is actually spent on these keywords by the user. In case the keyword originates from the user clustering process and thus has not been explicitly preferred by the user, we average on the total amount of time the users of the cluster spend on the article this keyword comes from. Furthermore, we expect that the mean times of the keywords preferences will be correct when the user profile reaches its steady state, hence depicting the overall user preferences. The overall personalization factor for each keyword i, named Upi, is:

$$Upi = B * Uwi \tag{2}$$

where for the parameter B: if the keyword belongs to the positive keyword list, then B>1; whereas if the keyword belongs to the negative keyword list, then B < 1. The norm of the B parameter can take any value that we desire, thus increasing or decreasing at will the effect that personalization and dynamic profile generation have on the sentence weighting procedure. In our analysis, we found that for |B|=1.5 we get the best results. From the previous, U_{pi} can be positive, negative or zero if there is no information about the user's preference of the specific keyword.

4 Experimental Procedure

For our experimentation we analyzed the logs of the browsing patterns as well as the recommendations offered to 50 of the registered system users. The users had been using the system for two months after their registration and for this period of time, the recommendations with and without the application of user clustering via W-kmeans were recorded. The total amount of articles recommended or browsed, i.e. the used corpus was over 8000 articles which belonged to various fields of interest: politics, technology, sports, entertainment, economy, science and education. One of the most widely used evaluation metrics for predicting performance of CF and recommender systems is Mean Absolute Error (MAE). MAE, expresses the average absolute deviation between predicted and true ratings and can be computed using formula (3).

$$MAE = \frac{\sum r'(u,i) \in R' \,|\, r(u,i) - r'(u,i) \,|}{|\, R'|} \tag{3}$$

where r(u, i) is the preference of user u for article i and r'(u,i) the predicted / recommended preference for user u of articles belonging to R'.

Fig. 3 depicts the MAE results that we obtained during this experimental procedure. We observe that the application of article and user clustering via the W-kmeans algorithm has significantly reduced the MAE of the recommendations provided to the users. More specifically, we observed that as users were viewing more and more articles and their profiles got shaped, the MAE of the recommendations were reduced. This was true both when user clustering was applied and when it was not applied. What this means from the practical point of view is that the recommendations given to the users were with increasing tendency accurate, since users opted on viewing them. This result was expected and has also been previously observed [14]. However, by taking into account the clustering information, the MAE of article suggestions compared to the actual user choices dropped by an average of 15% over the case when user clustering wasn't applied.

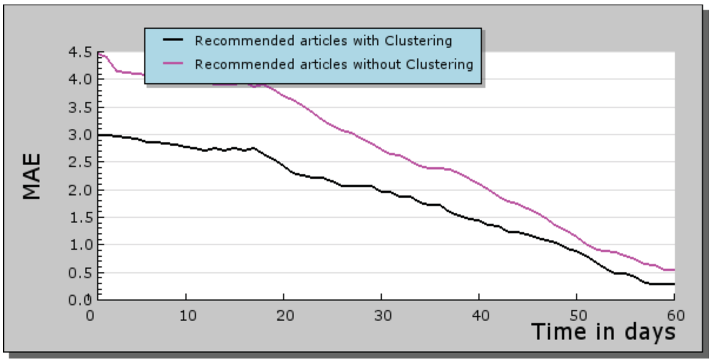

Fig. 3. MAE of recommendations with and without the use of W-kmeans.

This was more obvious at the early days of system usage during the experiment, when the user profiles had not yet been determined to a good extent by our system. Nevertheless, even when the user profiles had on average reached a 'steady state', around day 45 (as observed from the average numbers of profile updates), the MAE was still less when clustering was taken into consideration by the recommendation process, proving in effect the significance of our approach.

For our next experiment we tried to evaluate the overall performance and efficiency improvement of our personalization procedure when article and user clustering is applied. As an evaluation metric we used the F-measure, defined in (7). The F-measure is a weighted combination of the precision and recall metrics. 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.

$$F(c'_i, c_j) = 2 \cdot \frac{r(c'_i, c_j) p(c'_i, c_j)}{r(c'_i, c_j) + p(c'_i, c_j)} \tag{4}$$

Where: $r(c'_i, c_j) = \frac{doc(c'_i, c_j)}{doc(c'_i)}$ and $p(c'_i, c_j) = \frac{doc(c'_i, c_j)}{doc(c_i)}$. Using the same user logs as in the previous experiment and for the same time window, we extracted the F-measure results for the produced recommendations depicted in Fig. 4. We observe that the recommendations which utilize the generated article and user clusters produce on average 0.1 better scores in terms of F-measure. As before, the improvement gets even bigger after some days of system usage. The above has two explanations: a) the system has more data regarding the user's choices/preferences, and b) the system has more time to generate more coherent and generally better user clusters. Initially the F-measure scores are too low due to the fact that the recommender hasn't yet determined the user profiles to an acceptable extend. It is also observed that around day 45, the recommendations have reached almost their performance peak revealing that on average, the steady state for the user's profiles has been achieved.

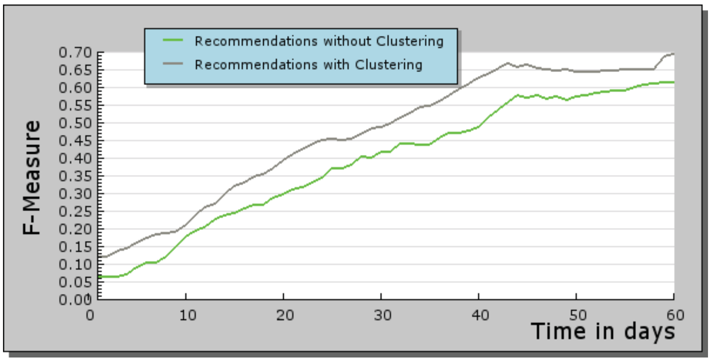

Fig. 4. F-Measure of recommendations with and without the use of W-kmeans.

5 Conclusions and Future Work

In this paper we presented the application of our WordNet-enabled clustering algorithm, W-kmeans, within a more generic recommendation framework. Trying to deal with the task of effective and adequate retrieval of personalized news articles that derive from the web, we presented the personalization algorithm that is used for presenting the categorized, clustered and summarized articles to the user. Our recommendation approach can be classified as 'hybrid' since it is mainly content-based with some collaborative filtering features that enhance the algorithm with the ability to automatically adapt over time to the continuously changing user choices. Our experimentation showed a significant MAE diminution, on average 15%, when clustering was applied before the recommendations instead of when not using it. We also noticed that the recommendations were scoring on average 0.1 better in terms of F-measure. We proved that adding keywords from user clusters to user keyword lists results in an improvement in the recommendation performance, something not examined before on hybrid recommenders. We believe that the above results have justified that the use of clustering (both article and user based) can be beneficial for a recommendation system.

For the future, we are planning on enriching the various components of our system with various improved techniques, i.e. for keyword extraction / enrichment and categorization. We will also be focusing on creating suitable communication channels for delivering the article recommendations to the user's desktop or handheld device.

Acknowledgments

This research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Heracleitus II. Investing in knowledge society through the European Social Fund.

References

1. Su, X., Khoshgoftaar, T. M.: A survey of collaborative filtering techniques. Advances in Artificial Intelligence (2009)
2. Goldberg, D., Nichols, D., Oki, B. M., Terry, D.: Using collaborative filtering to weave an information tapestry. Communications of ACM, vol. 35, no. 12, pp. 61–70 (1992)
3. Breese, J. Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI '98) (1998)
4. Su X. Khoshgoftaar, T. M.: Collaborative filtering for multi-class data using belief nets algorithms. In Proceedings of the International Conference on Tools with Artificial Intelligence (ICTAI '06), pp. 497–504 (2006)
5. Ungar, L. H. Foster, D. P.: Clustering methods for collaborative filtering. In Proceedings of the Workshopon Recommendation Systems, AAAI Press (1998)
6. Hofmann, T.: Latent semantic models for collaborative filtering. ACM Transactions on Information Systems, vol. 22, no. 1, pp. 89–115 (2004)
7. Si, L. Jin, R..: Flexible mixture model for collaborative filtering. In Proceedings of the 20th International Conference on Machine Learning (ICML '03), Washington, DC, USA, vol. 2, pp. 704–711 (2003)
8. Ansari, A. Essegaier, S. Kohli, R..: Internet recommendation systems. Journal of Marketing Research, vol. 37, no. 3, pp. 363–375 (2000)
9. Melville, P. Mooney, R. J. Nagarajan, R..: Content boosted collaborative filtering for improved recommendations. In Proceedings of the 18th National Conference on Artificial Intelligence (AAAI '02), pp. 187–192, Edmonton, Canada (2002)
10. Pavlov, D. Y. Pennock, D. M.: A maximum entropy approach to collaborative filtering in dynamic, sparse, highdimensional domains. In Advances in Neural Information Processing Systems, pp. 1441–1448, MIT Press, Cambridge, Mass, USA (2002)
11. Joachims, T.: Optimizing search engines using clickthrough data. In Proc. of ACM SIGKDD Conference (2002)
12. Bouras, C., Tsogkas, V.: W-kmeans: Clustering News Articles Using WordNet. In Proceedings of KES (3). pp. 379-388 (2010)
13. Bouras, C., Tsogkas, V.: Clustering user preferences using W-kmeans. In proceedings of SITIS 2011, pp. 75 – 82 (2011)
14. Bouras, C., Poulopoulos, V., Tsogkas, V.: PeRSSonal's core functionality evaluation: Enhancing text labeling through personalized summaries. Data and Knowledge Engineering Journal, Elsevier Science, Vol. 64, Issue 1, pp. 330 – 345 (2008)