10

PNS: A Personalized News Aggregator on the Web

Georgios Paliouras[1], Alexandros Mouzakidis[1], Vassileios Moustakas[1,2], and Christos Skourlas[2]

[1] Institute of Informatics and Telecommunications, National Center for Scientific Research "Demokritos", Athens, Greece, paliourg@iit.demokritos.gr, alexm@iit.demokritos.gr, bmoustakas@iit.demokritos.gr

[2] Department of Informatics, Technological Institute of Athens, Athens, Greece, cskourlas@teiath.gr

Summary. This paper presents a system that aggregates news from various electronic news publishers and distributors. The system collects news from HTML and RSS Web documents by using source-specific information extraction programs (wrappers) and parsers, organizes them according to pre-defined news categories and constructs personalized views via a Web-based interface. Adaptive personalization is performed, based on the individual user interaction, user similarities and statistical analysis of aggregate usage data by machine learning algorithms. In addition to the presentation of the basic system, we present here the results of a user study, indicating the merits of the system, as well as ways to improve it further.

10.1 Introduction

In recent years, the World Wide Web has experienced a self-feeding increase in the number of users and the quantity of content, data and services. More content makes the Web more interesting for more users, who in turn create more content. This spiral effect seems now to be accelerated by Web 2.0 technologies and the ever-increasing possibilities for user-generated media. A typical example of this is the news industry, which seems to be turning fully online and trying to follow the developments in Web publishing. Most of the news publishers have introduced electronic versions of their content, which in many cases are much richer in structure than the traditional paper versions. Additionally, a number of intermediate services have appeared, such as thematic news portals, which aggregate and re-distribute information from various sources.

In this manner, the end user has gained access to an enormous volume of information, which apart from its clear positive side brings along the problem of information overload. The task of finding interesting information in all that is within reach is as daunting and frustrating for the non-expert user as

G. Paliouras et al.: *PNS: A Personalized News Aggregator on the Web*, Studies in Computational Intelligence (SCI) **104**, 175–197 (2008)
www.springerlink.com

looking for a needle in a hay-stack. Thus, if we are to support this exciting development, we need to devise better and simpler methods of access to interesting information. Personalization is one way of achieving this through the modelling of user interests. Personalization systems typically acquire models of individual users or groups of users and then use these models to filter content, to recommend interesting content or to facilitate search. The acquisition of the user models is either done "manually", i.e. by asking the users or experts to define them, or in a less obtrusive automated manner, by statistical analysis of usage data.

PNS (Personalized News Service),[1] the system that we present here, attempts to personalize the experience of news reading at the level of an intermediate aggregating news service. PNS is a portal that aggregates news from various multi-language electronic news sources and provides a user with a personalized view of recent and past news items. Aggregation is done both through RSS feeds, as well as through information extraction with the help of simple HTML parsing programs (wrappers). Highlights of new items (title, source etc.) are retrieved periodically from various Web portals and e-newspapers. Once retrieved, they are organized according to predefined news categories and a Web-based interface provides personalized views to the users. Personalization in PNS is powered by a general-purpose personalization server called PServer. PServer uses statistical analysis and machine learning methods [14] to support four types of adaptive personalization: (a) personal user models, (b) user stereotypes, (c) user communities, (d) associated items. PNS uses all four types of personalization to rank news items according to the user's individual preferences or the interests of similar users. Thus, PNS is to our knowledge the only news aggregator supporting such powerful and flexible personalized news reading.

Compared to the previous version of PNS that was presented in [11], the version presented here is improved in several ways: (a) it provides a more complete personalization solution, integrating in a better way into the system various of the services available by PServer, (b) it includes many more news sources, which was noted as a major requirement of the users in the evaluation of the previous system, (c) the interface has improved significantly, and (d) a brand new user study has been performed.

The rest of this paper is structured as follows. Section 2 describes the design and implementation of PNS. Section 3 presents the results of an initial user study. Section 4 reviews the state-of-the-art systems for news personalization and in the last section conclusions and future directions are presented.

[1] http://pns.iit.demokritos.gr/

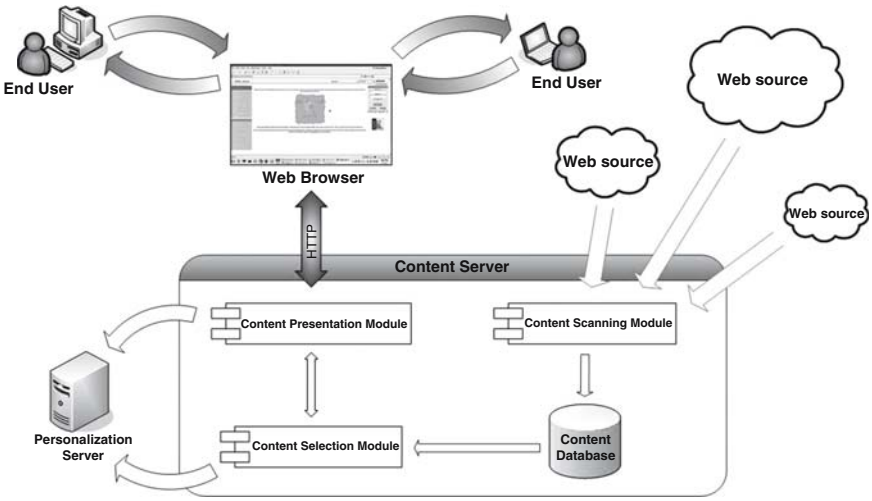

Fig. 10.1. The architecture of the personalized news service (PNS)

10.2 Personalized News Service

The Personalized News Service (PNS) provides its users with personalized access to news items harvested from multiple Web sources. It takes input from both the content sources (news agencies, news portals, electronic newspapers etc.) and the users themselves. This information is processed and a personal newspaper is constructed with recent news items that match the user's preferences. Figure 10.1 illustrates the system's overall architecture with emphasis on the Content Server which consists of the following basic modules: (a) the Content Scanner, (b) the Content Selector, (c) the Content Presenter, and (d) the Content Database, where information about about the news sources, the news items[2] and the wrappers are stored.

The system collects information about users in two ways:

- During the registration, the user specifies a user name and password and may also provide personal information, such as age, gender, occupation. Personal information is fed to the Personalization server (PServer) for improved personalization.
- During the use of the system, the users' browsing activity updates the corresponding user models maintained by PServer.

The component modules of the system and the basic functionality of PServer are described in more detail in the following sections.

[2] Respecting the copyright of the sources, the server does not store the content of the articles, but simply indexes it, according to its own categorization.

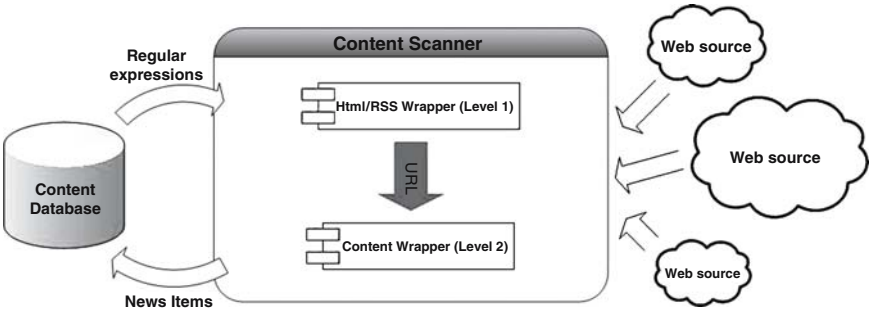

Fig. 10.2. The architecture of the content scanner

10.2.1 Content Aggregation

The content scanning module is responsible for locating and retrieving new items from a list of pre-specified sources and then storing in the database basic indexing information that will allow personalization and retrieval of the item. The aggregation process is done offline with a Web spider that is called periodically.

The spider works on the list of sources and associated URL addresses, which are stored in the database. For each source it follows a two-stage procedure: (a) identifying the addresses of new items in the source, (b) retrieving the items and extracting the information required for indexing them. For each of these two subprocesses an HTML wrapper is invoked, i.e., a small parsing module that identifies the required information within each Web page. Figure 10.2 illustrates this two-level identification and extraction process.

The first level of wrapping (HTML/RSS wrapper) involves the identification of URL addresses of new items in the source and per category of news. Thus, for each source-category pair an address is stored in the database and associated with a corresponding wrapper, which takes the form of a set of regular expressions. If the page that is retrieved is an RSS document, the wrapper has to parse the corresponding XML file and identify the URL addresses of new items and associated information. Typically, RSS documents, annotate the address of each time, each title and other information with XML tags. The information that we retrieve at the moment is the address and the title. For example an RSS document of a news source might have the following text:

```
<item>
    <title> Title of news item </title>
    <link> Address of news item </link>
    <pubDate> Publication date and time </pubDate>
    ...
</item>
```

If on the other hand, the source is not RSS, an HTML page is parsed by the wrapper and the addresses and titles of new items are extracted. The following is a sample extract of an HTML page containing links to articles:

```
<tr>
    <td align="justify" valign="top">
    <a class="title" href=" Address of news item" >
    <b> Title of news item </b></a>
    <br><div class="cat"> Publication date and time </div>
    ...
    </td>
</tr>
```

In order to build a wrapper in this case, one needs to identify the expressions that delimit the information of interest. For instance, the strings `<a class= "title" href="` and `" >` delimit the address of the articles in the above example. In both RSS and HTML sources, new items are distinguished from old ones, based on their URL addresses.

The second level of wrapping (content wrapper) extracts useful information from each new item. This information is used for indexing and retrieving the article, as well as presenting a highlight of it to the user. So far, we are extracting only the first sentence of each article, parsing the corresponding HTML page. In future versions of the system, additional information, such as keywords from the content of the article will be extracted. The wrapper is constructed in a similar fashion as for the level 1 wrapper, i.e. by identifying sufficiently delimiting regular expressions. In some RSS feeds a short description of the article is provided. The description information is becoming increasingly common in RSS news sources. In that case, the second level wrapper is not needed for the information that we are currently extracting.

One problem with both types of wrapper for HTML pages, i.e. non-RSS sources, is that they are source-specific. Each source uses a different format for the presentation of articles and therefore we need different regular expressions for each one. However, almost all of these HTML pages are generated dynamically from a content database and therefore the same wrapper works for all news items. Furthermore, the format does not change very frequently (on average every few months) and therefore the wrappers require only occasional updating. Even that can be difficult sometimes though. For this reason, we are studying wrapper verification and wrapper induction methods [15] that will allow us to recognize when a wrapper has changed and automatically produce the correct wrapper for the new HTML format. Additionally, the increasing use of RSS feeds will eventually remove the need for source-specific wrappers.

In summary, the HTML/RSS wrapper reads from the database, a list of URL addresses corresponding to the source-category pairs and the associated regular expressions and produces as output a list of URL addresses for new items and their titles. This information is received by the content wrapper, together with the corresponding regular expressions and a full record for the

new item, containing its address, title and first sentence, is stored in the content database.

10.2.2 Personalized Content Selection

News aggregation provides one-stop access to many sources, but at the same time reveals in a very immediate manner the problem of information overload. In other words, by combining information from many sources, the user becomes aware of the quantity of information out there and the difficulty of getting to the items of interest. As a result, dealing with the problem of overload is essential in the PNS.

More specifically, the system provides four different personalized views to the news items:

Personal news provides a content-based ranking of the interestingness of news items based on the personal model of the individual. For example, a user may prefer to read financial and sports news, while another might be interested specifically in the world news of yahoo.

Stereotype news provides collaborative ranking of the items based on the model of all individuals with similar characteristics, i.e., age, gender, etc. Such personal information is optionally provided by the users when they register or can be added later and it is used for assigning an individual to a stereotype.

Community news provides collaborative ranking without taking into account the personal characteristics of the user, combining instead the models of the communities in which the user has been assigned automatically, e.g. users that prefer financial and sports news.

Related news provides collaborative navigation, using cluster models of items, in order to associate news and recommend further reading.

The four views are complementary in several dimensions. For instance, new users are likely to find the "stereotype news" more useful, as their personal model will be very poor. On the other hand, "related news" provides navigational help, in contrast to the ranking approach of the other three views. "Community news" is useful for expanding one's interests in a focussed way, especially when no personal information for the user is available, in order to assign the user to a stereotype.

Personalization is achieved with the use of a separate personalization server, called PServer,[3] that provides a variety of services. Pserver is a general-purpose personalization server that can be adapted to any kind of application requiring personalization services. Pserver works like a Web service, taking a request through the http protocol and re-turning XML documents with the

[3] PServer has been developed in the Institute of Informatics and Telecommunications of NCSR "Demokritos" and will soon be made available under a BSD-like license.

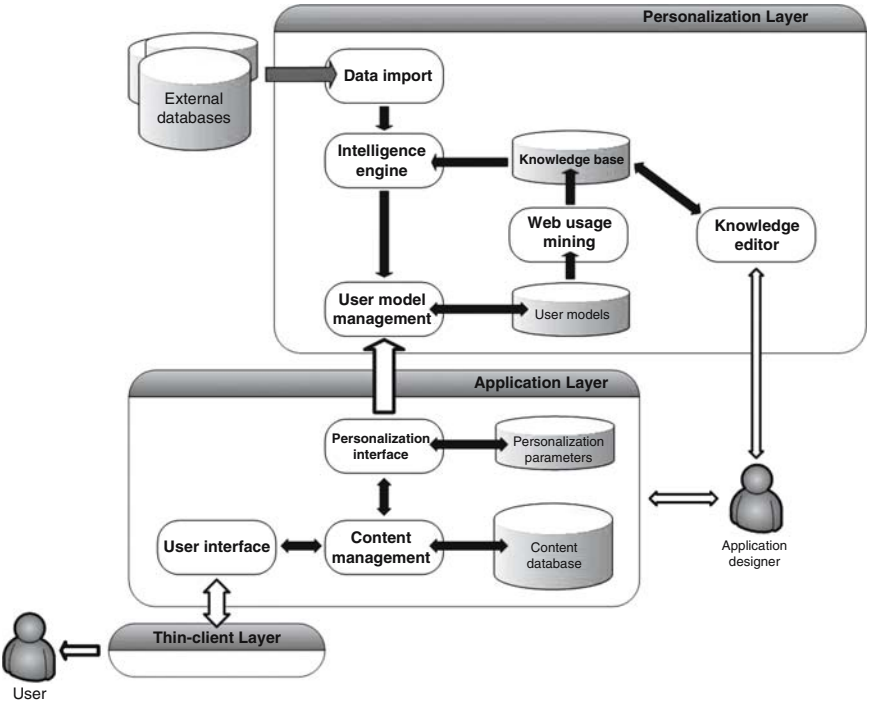

Fig. 10.3. The architecture of the personalization server

results, and can be used by many different applications concurrently. The developers who want to personalize their application do not need to make great modifications to their applications, but just the code required for making their application a client of Pserver. Thus, PServer makes the personalization of existing applications very easy. Figure 10.3 provides a high-level view of the PServer architecture.

The application layer in Fig. 10.3 illustrates a typical personalized application, such as PNS, while the Personalization server presents the main elements of the PServer. The user model management module is the one controlling the changes that happen to all types of user model. The Web usage mining process discovers new knowledge from the existing user models, which is then processed by the intelligence engine and provided to the management module, in order to update the user models. The optional introduction of external data or knowledge to the system, by data import and knowledge editing is not used in PNS. At the application layer, a typical personalized application, such as PNS, maintains a content database, selecting the content appropriate for each user. This content selection module communicates with the Personalization server, retrieving user models and updating them according to the users' actions.

The communication of PServer with the application is based on a common vocabulary of personalization parameters, which are defined by the application and communicated to PServer at the set-up stage of the system. There are two basic types of parameter, called *attributes* and *features*. Attributes capture information that is rather static, i.e., it is not processed statistically, but provided explicitly by the application. An example of such information is the personal information of each user, i.e., age, gender, etc. Features on the other hand capture usually the preferences of the users. In the case of PNS, the features correspond currently to news sources, news categories and their combination. As a result, personalization is only done at the coarse level of news sources and categories. Current work is expanding this to the contents of individual news items and clusters thereof. All users modeled by PServer share the same parameters, but not the same parameter values. PServer provides functions to insert of remove features, attributes and users, to get or set the values of the features for a specific user and functions to increase or to decrease the values of the features.

As explained above, PNS makes use of four types of adaptive personalization, using the corresponding services of PServer: (a) personal user models, (b) stereotype models, (c) community models, and (d) item clusters. Each of the four types requires the acquisition and maintenance of a different user model, which is achieved with the use of statistical analysis and machine learning methods. The corresponding services of Pserver are described below.

Personal user models: Each personal user model stores the attribute and feature values of an individual user. Features are updated according to the actions of each user, either as frequency counts and/or as a histories of actions. In this manner, we can at any point in time infer the level of interestingness of each user in a certain feature, such as the sports news in yahoo. In PNS, we are currently using only frequency counts.

Stereotype models: User stereotypes are sets of users with common attributes. For example all the users within a specific age range and a particular job type may constitute a stereotype. Like personal user models, stereotypes also have features that are updated according to the preferences of the users in the stereotype. However, in contrast to personal models, each stereotype may have a different feature set from all other stereotypes and thus each stereotype can be handled separately by the application. More advanced methods for learning stereotypes from personal models, e.g. [10], are not used currently in PNS.

Community models: The main problem with stereotypes is that users may provide inaccurate personal information, due to privacy or other concerns. For this reason, providing personal information is only an option in PNS. For those users who do not provide personal information, we need a different way to support collaborative filtering or ranking. This is achieved by the clustering of users into user communities. Communities in PServer are constructed on the basis of similarities between the users, using the cluster mining unsupervised learning algorithm. Any clustering method could be used for this purpose [12],

but we are using the cluster mining method described in [13], due to the fact that it allows communities to overlap, i.e., each user may belong to more than one communities. The algorithm is a graph-based clustering method, associating communities with cliques in the graph of users. PServer is easily extensible with new algorithms for community discovery. This is done through a simple SDK and does not require the recompilation of the code of PServer.

Item clusters: In order to discover associations between features, e.g. news categories, one can simply cluster features together, according to their statistics in individual user models. This can be considered as the inverse task of community discovery. Hence, instead of clustering users, based on features, we cluster features based on users. In PServer, this is also achieved with the cluster mining algorithm, although other association discovery methods can easily be added.

10.2.3 Reading Personalized News

The content presenter module of PNS is basically the personalized graphical user interface of the system. The module is responsible for identifying the user, providing the various personalized views of PNS, informing PServer about the actions of the user and providing more traditional retrieval facilities, such as search by date or by keyword. In the following we illustrate the basic functionality of the module with the help of corresponding screenshots.

Entering the system, the user views a welcome screen, as shown in Fig. 10.4. From this point on the user can either browse all the news using the category and source menus on the left hand side of the screen or log into the system as

Fig. 10.4. The introductory page

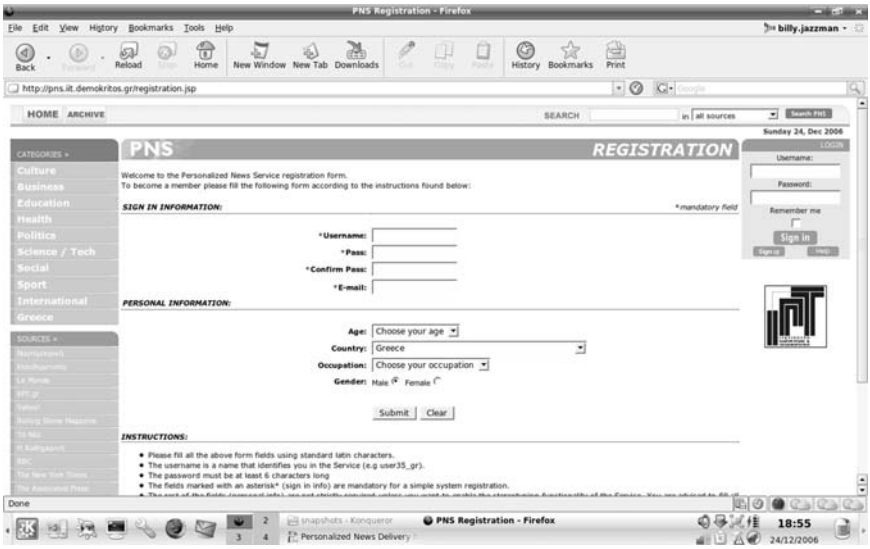

Fig. 10.5. The registration form

a known user (top right corner of the screen). If the user is not logged in, the news are presented in a non-personalized manner and no information is kept about the user's actions. By logging in, the system gets into a personalized mode and the news are presented according to the user's model. In particular the news are ranked according to their expected interestingness for the user, based on the personalization parameters that are used.

If the user is not already registered and wants to do so, they select to sign up and the registration form shown in Fig. 10.5 appears. The form is separated into a set of compulsory fields required for user identification, i.e., username, password and email address, and a set of optional fields that are to be used for assigning the user to a stereotype. If the user chooses not to provide these, stereotype-based personalization will not be available. A broader consequence of many users choosing not to provide personal information is that the stereotype models become statistically weak, as they are based only on a small fraction of the users who actually belong to the stereotype.

When logging into the system, the user moves to the "personal news" view described in Sect. 10.2.2 above. An example is shown in Fig. 10.6. In this view, the ranking of the articles is based on the information recorded in the user's profile, assumed to represent the user's preferences. Some basic information is shown for each article, avoiding the reproduction of the full article, due to copyright issues. In particular, only the title and the first sentence of the article are shown and a link to the original source is provided if the user wishes to read the whole article.

When a new user enters the system, this personalized view is actually the default one, as no information is available yet about the preferences of the

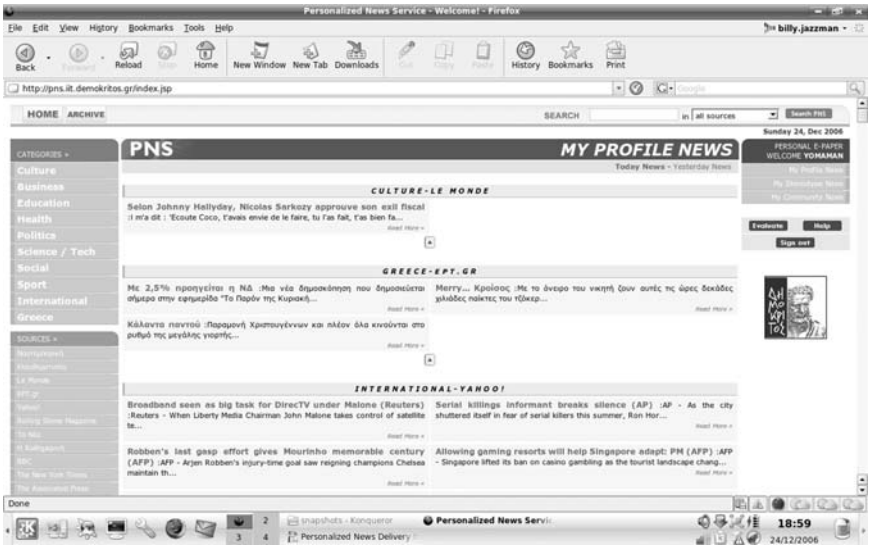

Fig. 10.6. The personal news page

user. In that case, the user can switch to the "stereotype news" view, using the menu on the top right corner of the screen. As explained above, this is only possible if the user has provided personal information. The effect on the ranking of the news items is similar to that of the "personal news" view, but a different model is used, i.e. the stereotype. The "community news" is also not meaningful for new users, as no information is available for assigning the user to communities of "similar" users. However, once the user has used the system a few times, assignment to communities becomes possible and the user can choose to rank the articles according to the models of the corresponding communities. The presentation of the news is again similar to that shown in Fig. 10.6.

The fourth personalization view discussed in Sect. 10.2.2, i.e. "related news" becomes available when the user selects to view an article. By selecting the article, a separate screen appears showing the article as shown in its original source, and adding an interesting link "Users who read this article also read ...", see Fig. 10.7. The PNS header is also added in order to show to the user that the system is still in a personalized mode.

When choosing the "Users who read this article also read ..." link, the user gets a list of articles ordered according to the clusters in which the first article belongs. The appearance of the ranked items is shown in Fig. 10.8 and is similar to that in Fig. 10.6. The user can then choose one of the recommended articles, causing its display, follow the "related news" link and so on. In this manner the navigation of the user in the content database becomes personalized in a collaborative way.

Fig. 10.7. Reading an article

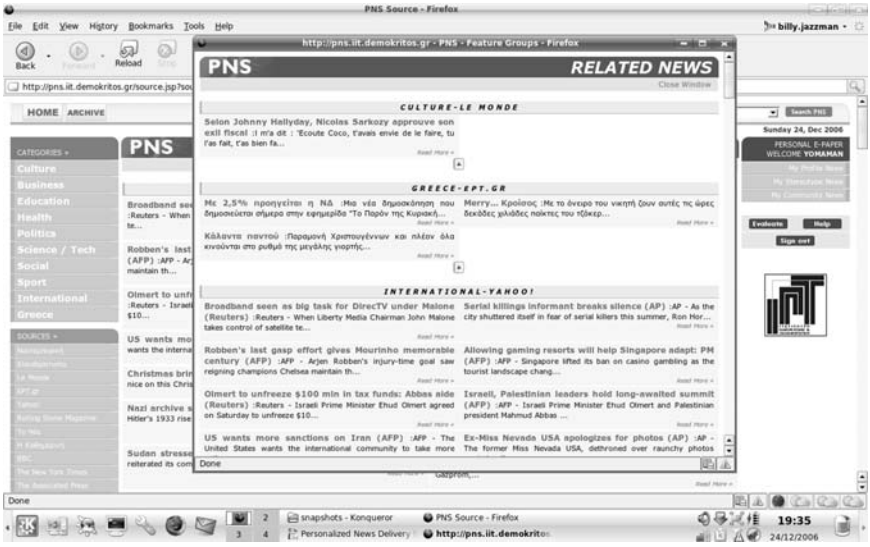

Fig. 10.8. Viewing related articles

Finally, in addition to the various personalized views, which refer to the most recent articles published in the corresponding sources, the user is able to search the content database by date, category and source (Fig. 10.9). By doing this, the user will retrieve all news published in the specified date range, and

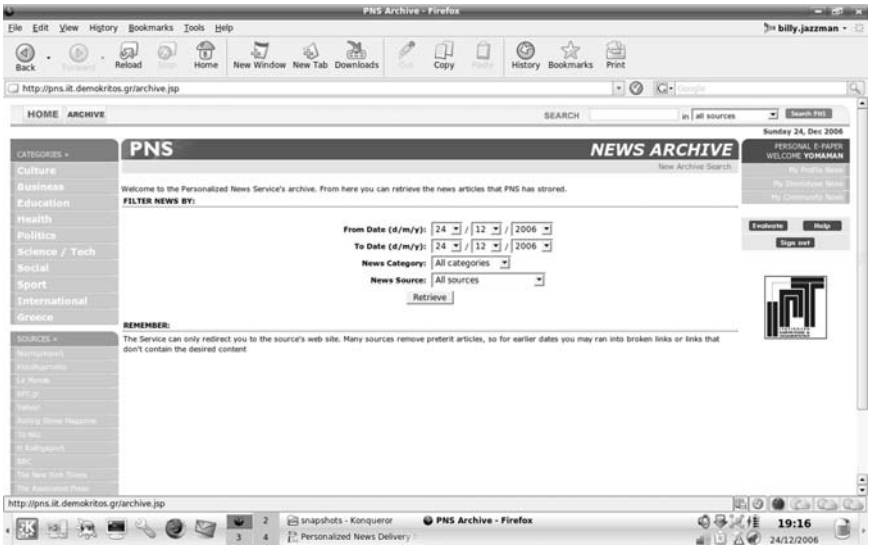

Fig. 10.9. The search-by-date page

in a particular source-category pair. In a similar manner, the user can search by a keyphrase within a particular source or in the whole database.

10.3 User Evaluation

10.3.1 Set-Up of the Study

In order to assess the usefulness of the system, users were asked to test the system for a short period of time. On a daily basis, the system collected the most recent news, which were then presented to the users. At the end of the test period, the users were asked to fill an electronic questionnaire with their observations and comments. The role of the user study was to gather feedback on several different aspects of PNS:

- Validating the personalization services
- Evaluating the functionality of the system
- Providing input to the design of the system

Thus, the questionnaire was separated into four sections. The first section asked for some basic characteristics of the evaluator, focusing mainly on computer literacy and use. The second section assessed the functionality of the system, focusing on usability issues. The third section, which is more interesting for this paper, assessed the value of different personalization views. Finally, the fourth section asked the users to provide suggestions for improvement.

In order to make the completion of the questionnaire easier and allow useful conclusions to be drawn even with a small number of users, most of the questions had a three-choice answer, like "Satisfied–Partially satisfied–Not satisfied" or "Very useful–Not so useful–Not useful".

At the end of the study we collected 34 answers and most of the users were highly literate in computer usage. Actually, the majority were either computer science students or academics. Therefore, the results that are presented below cannot be considered representative of the average user, but more biased on the technical issues of the system. This is particularly helpful for improving the system technically, but it is clear that a wider-audience evaluation is still needed.

10.3.2 Evaluating the Functionality of the System

In order to evaluate the usability of the system, the users were asked to respond to a number of questions concerning the user interface and important design parameters of the system, such as the news sources and news categories that are used. Most of these questions, though not all, had been asked also in the user study that was performed for the previous version of the system, presented in [11]. Table 10.1 presents the response of the users in the relevant questions and where available the results of the previous study are presented for comparison purposes.

The results of the study, regarding the functionality of the system are particularly encouraging. Comparing to the previous version of the system,

Table 10.1. Results on the functionality of the system

Do you find the web interface usable and comprehensible?			
Answer	Yes (%)	Partially (%)	No (%)
New study	73.5	23.5	2.9
Old study	70	30	0
How much time did it take you to get familiarized with the system?			
Answer	<30 min	About 60 min	>60 min
New study	76.5%	23.5%	0.0%
Did you find the news you are interested in quickly and easily?			
Answer	Yes (%)	Partially (%)	No (%)
New study	79.4	20.6	0.0
Old study	70	20	10
Are you satisfied with the news categories used by the system?			
Answer	Yes (%)	Partially (%)	No (%)
New study	55.9	44.1	0.0
Old study	20	45	35
Are you satisfied with the news sources included in the Service?			
Answer	Yes (%)	Partially (%)	No (%)
New study	41.2	58.8	0.0
Old study	10	50	40

usability has improved in all aspects. In the first three questions, there is a great majority, between 70 and 80% of people who are satisfied with various aspects of the user interface. Thus, there seems little room for improvement in this direction. Even more encouraging though are the results in the last two questions, regarding the news sources and categories that we are using. This aspect of the system has been criticized in the previous survey and the system seems to have improved significantly, since the answers have shifted towards the "satisfied" side of the spectrum, leaving the non-satisfied quadrant completely empty. Having said that, there is still room for improvement.

10.3.3 Evaluating the Personalization Aspects

The most interesting aspect of the study concerned the personalization functionality of PNS. For the first time, this study tried to assess the value of the individual personalization views, in addition to the overall value of personalization in PNS. Figure 10.10 shows the results that we obtained for the latter, which are almost identical with these obtained in the previous survey. This lack of improvement shows that we have not been able to provide the added value that we wanted to the user, through the use of the advanced personalization functionalities.

In an attempt to understand better where the problem lies, we asked more specific questions about the satisfaction of the users with the various personalization views. Figure 10.11 shows the results that we obtained. Based on these results, it seems that the problem focuses mainly on the collaborative views, as the personal news view is assessed rather positively. Of the collaborative views, "community news" is assessed very negatively and "related news" rather positively, while "stereotype news" is in between the two.

As an initial explanation of the situation, one needs to note the difficulty of assessing collaborative personalization. Especially communities require the use of the system for a significant amount of time and by a substantial number of people, in order to start adding value to the user. Stereotypes are a

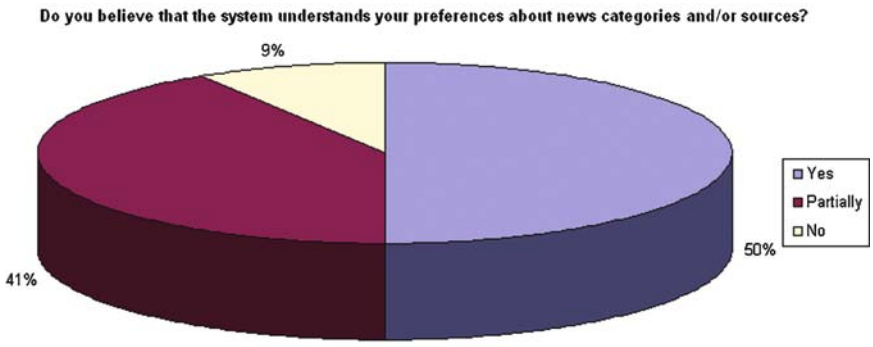

Fig. 10.10. Overall evaluation results for the personalization in PNS

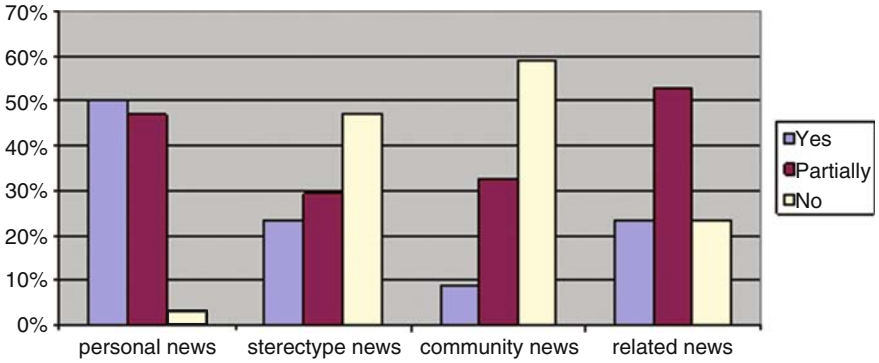

Fig. 10.11. Evaluation results for the various personalization views in PNS

Table 10.2. Correlation of responses for the collaborative views

	Stereotype news (%)	Community news (%)	Related news (%)
Stereotype news	100	63	81
Community news	63	100	58
Related news	81	58	100

bit better due to the fixed specification of the groups of people belonging to each stereotype, i.e., based on their personal characteristics which are entered at registration time. Still, though in order to obtain statistically significant evidence about the interests of each group more usage data are required than what we have been able to collect. Finally, in the "related news" view the situation is better, due to the small number of items, i.e., news sources and categories, that also have rather straightforward associations, e.g. when viewing a news item about politics, one would expect other politics news to be recommended.

Focusing further on the explanation of the problem, we measured the correlation in the users' responses, regarding the collaborative personalization views. Table 10.2 shows the percentage of common responses received for the three views. Each cell in the table shows the percentage of identical response for two views. For example, in 81% of the responses users gave the same assessment for "stereotype news" and "related news".

The above results show a low level of agreement in the assessment of "community news" with the other two views, which when combined with the negative assessment of this view, leads to the conclusion that agreement is mainly towards the negative responses, i.e., users who agree that the two views are equally unsatisfactory. On the other hand, there seems more positive agreement on the "stereotype news" and "related news" views.

10.3.4 Recommendations for Improvement

In addition to the quantitative evaluation of various aspects of the system, the users provided very interesting suggestions on how the system could be improved. Some of these concerned the functionality and the user interface of the system, while others addressed the personalization aspects of it.

Regarding the functionality of PNS, despite the positive quantitative results, we received some interesting suggestions to improve the system further. First, there was a common demand of the users to improve the way that articles are shown on the screen. The highly textual presentation seems tiring and a more visual approach is needed. There were also some suggestions about news sources from different countries and new categories or sub-categories, for example specialized technology news and news about specific kinds of music. These suggestions seem particularly relevant for the biased user group that took part in the evaluation, but a wider audience is expected to make different requests. Finally, we need to add a procedure for password recovery, as users often forget it.

On the personalization aspects of the system, the most common request was for more personalized suggestions. This is due to the fact that personalization is currently done only at the level of news sources and news categories. This is an important problem, which is also responsible to some extent for the negative assessment of collaborative personalization views. As a response to this problem, we are already extending personalization to the level of individual articles and their content, extending it also to the results of free-text queries on the archive. Additionally, users suggested that some of the collaborative personalization views, particularly communities, are an overkill for the system. As mentioned above, we believe that this is due to the short evaluation period and small number of users taking part, which did not allow the discovery of interesting communities. As a response to this comment we are looking into a tighter and more intuitive integration of the various collaborative personalization views, as well as the testing of new clustering methods. Furthermore, the evaluation of the next version of the system should be done over a longer period of time and with more users if possible. The initialization of the system with communities that have been discovered in previous user studies may also help in arriving more quickly at interesting community models.

10.4 Related Work

There exist many Web sites/portals available online that provide similar services to PNS. Many of them are experimental systems that were developed for research purposes while others are real-world commercial services. The category of commercial systems has been growing in the past couple of years

as the potential of the technology has been acknowledged.[4] It is therefore important to see other systems' choices and directions to various problems in order to better place PNS on the map of existing systems.

In the next few paragraphs there is a small description of the philosophy and the various techniques and technologies incorporated in the most influential systems on the Web. The presentation is based on the two primary technologies adopted by PNS: first the aggregation feature is examined and then we look at how each system (if available) personalizes the provided information.

Content aggregation is mainly achieved nowadays through the use of RSS feeds. Feeds can be imported either statically or dynamically to fit a specific user's preferences. In the first case, this means that a set of news Web sites are imported by the system's administrator into the aggregation module and these constitute a global source pool from where all users draw information. Web services that use this static kind of aggregation include Findory,[5] EMM's News Brief[6] and NewsExplorer,[7] Cebil[8] and Phigita News.[9] Google News[10] takes this approach a step further as it facilitates the addition of new RSS sources through a public procedure were users ask for and evaluate sources to be included or even excluded from the system, as in the National Vanguard magazine controversy. On the other hand, other Web sites, give each individual user the ability to select their own sources and therefore be exempted from content they consider unreliable or of no interest to them. My Yahoo (http://my.yahoo.com) contributed significantly to the wide adoption of RSS, even by newcomers, by letting people customize their own My Yahoo page with feeds from any source. To that helped the acquisition of SearchFox a pioneering company in the area. Additionally a suggestion system using popularity or editorial picks informs new users for the "hottest" feeds available. A similar approach is followed by Netvibes,[11] topix[12] and Feeds2.0.[13]

News portals pioneered the development of personalized systems which adapt to a user's specific needs. The adaptation may concern something so primitive as the medium in which news are delivered to a variety of other things, such as the advertisements displayed.

[4] http://www.jdlasica.com/articles/personalization.html
[5] http://www.findory.com/
[6] http://emm.jrc.it
[7] http://press.jrc.it/NewsExplorer/home/en/latest.html
[8] http://www.cebil.gr/
[9] http://news.phigita.net/
[10] http://news.google.com/
[11] http://www.netvibes.com/
[12] http://www.topix.net/
[13] http://www.feeds2.com/

A basic ingredient to a successful personalization system is the way in which it collects user data and models each user. Findory, Spotback[14] and Feeds2.0 collect only implicit data from the user as the user navigates through the Web pages. This means that no registration process is needed, other than the creation and use of a user name and password. The user modelling is for the same reason adaptive as the model changes based on the analysis of usage data. Google News also tracks the browsing actions of users in order to collect implicit data. Additionally it requires full registration where the user provides data explicitly to the system such as demographic and preference information. All information is stored inside the user's Google Account, thus making the access of the personalized edition of Google News possible from any computer. A totally different approach is used by, the still in private beta, Leaptag.[15] LeapTag allows the user to define the things he is interested in using tagging. For each interest he/she creates a tag for, LeapTag will produce results that include news, blogs, books, etc. Advertisements are also personalized to the user's interest.

Finally, different personalized systems use different approaches to filtering and/or ranking of articles. These approaches fall into two basic categories, that have been briefly mentioned in previous sections: Content-based and collaborative filtering. Content-based filtering is based on the analysis of the article's content, aiming to identify important keywords for the user. The personal user models are an example of this type of filtering. On the other hand, collaborative filtering, such as stereotypes, communities and item clusters in PNS, is based on the assumption that users who regularly view the same articles have similar interests. Based on this assumption, recommendation mechanisms are built in order to help users to implicitly help each other find interesting articles.

There are several systems that provide content-based filtering. Examples are the Personal Wall Street Journal and the electronic edition of the San Francisco Chronicle which uses Fishwrap [6]. Other examples are: WebMate [5], the Mercurio system [7] of personalized access to the electronic variant of the Spanish ABC newspaper, NewsDude [4] and SmartPush [9]. On the other hand, an example of a collaborative filtering system is Findory.

Some web sites combine the two filtering approaches. Google News, Feeds2.0, Krakatoa [3] and its successor Anatagonomy [8] are some of them. Other such systems include SeAN [2] and the automated personalization system studied by Aggarwal and Yu [1]. Also Feeds 2.0 uses a language identification mechanism that allows the service to identify the language in which an item is written and automatically extract the most important keywords as tags for each item. Thus, search of items related to a particular object is made easier, while users are able to provide their own personal tagging of items.

[14] http://spotback.com/
[15] http://leaptag.com/

Table 10.3. Personalization techniques in research prototypes

	Data collection	User modelling	Filtering type
Fishwrap	Explicit	Non-adaptive	Content-based
Krakatoa, Anatagonomy	Explicit, implicit	Adaptive	Content-based, collaborative
SmartPush	Explicit	Non-adaptive	Content-based
SeAN	Explicit, implicit	Adaptive	Content-based, collaborative
Aggarwal and Yu	Explicit, implicit	Adaptive	Content-based, collaborative
WebMate	Explicit, implicit	Adaptive	Content-based
Mercurio (ABC newspaper)	Explicit	Adaptive	Content-based
NewsDude	Explicit	Adaptive	Content-based

Table 10.4. Personalization techniques in commercial systems

	Data collection	User modelling	Filtering type
Google News	Explicit, implicit	Adaptive	Content-based, collaborative
Findory	Implicit	Adaptive	Collaborative
Feeds2.0	Implicit	Adaptive	Content-based, collaborative
Personal Wall Street Journal	Explicit	Non-adaptive	Content-based
San Francisco Chronicle	Explicit	Non-adaptive	Content-based

Tables 10.3 and 10.4 summarize the personalization characteristics of the systems presented in this section.

PNS has two main differences from the systems presented above: the combination of RSS feeds with plain HTML sources and the provision of many complementary personalization views on the same data. The combination of RSS documents with HTML ones remains important for as long as there are interesting sources that do not use RSS feeds. The number of such sources is decreasing, but they still are the majority. It is expected that this situation will change in the medium to long term. The combination of multiple personalization views, powered by the use of PServer is more important. In its current state, PNS is able to personalize the reading of even new users, through stereotype modelling, while the added value increases for long-term users, both in the retrieval of articles, through personalized ranking, as well as in the navigation through the articles, using the "related news" view. The combination of content-based ("personal news") and collaborative ("community news") filtering is also particularly useful for a complete personalization solution.

10.5 Conclusions and Future Work

In this paper we have presented the Personalized News Service (PNS), which aggregates news from various Web sources and provides personalized access to it, using a variety of personalized views, namely "personal news", "stereotype news", "community news" and "related news". Personalization is powered by a general-purpose personalization server (PServer), which provides a variety of personalization capabilities to applications that require personalization. The use of this powerful personalization server is one of the distinguishing characteristics of PNS, allowing it to provide a complete personalization solution by integrating complementary types of personalization. Another characteristic of PNS, as compared to the state-of-the-art systems is its ability to process sources that do not provide their content in a structured format, i.e. through RSS feeds.

As part of this work, we have performed a user study, assessing various aspects of the system, and compared the results to those obtained for a previous version of the system. The results of the study have been particularly interesting, showing where more work is needed and also providing suggestions for improvement. Compared to the older version of the system, the satisfaction of the users with the basic functionality of the system and the user interface has increased significantly. Still it can be improved in several ways, such as the presentation of the articles that could become more user-friendly. Additionally, we would like to make the user interface more multilingual, as at the moment only English and Greek are supported.

Other improvements that we are working on are related to the maintenance of the wrappers for non-RSS sources, and the clustering of articles that talk about the same subject. Manual wrapper maintenance, i.e., changing the wrappers when the sources change their format, is becoming a major obstacle to the scalability of the system to many sources. Thus, we are integrating methods that we have been developing independently [15] to learn new wrappers without the intervention of the user. Article clustering is also becoming a major requirement as the number of sources increases, because the same article appears often in many sources. If these alternative versions of the article are treated as separate articles, the list of articles recommended by the system is going to become very large and with a high degree of repeated information.

The results of the user study have been particularly critical of collaborative personalization views, while the "personal news" view has been judged more positively. The users did not see the value of collaborative recommendations for the coarse level of personalization offered by the system, i.e. based on news sources and categories. Another reason for the criticism was the short evaluation period that did not allow for meaningful community models to be learned. Thus, one of our main goals is to move to a finer level of personalization, by using as personalization parameters the contents of individual articles. Furthermore, the evaluation of the next version of the system should

be done over a longer period of time and if possible with a larger and more diverse user group.

In summary, apart from a very useful system for the end user, as a research prototype PNS has raised a number of interesting issues that we are trying to address with related methods that come out of our relevant research.

Acknowledgments

We would like to thank particularly the people who took part in the evaluation study. Their feedback is invaluable to us and we promise to do our best to improve the system in the directions that these comments point at. The presented work is part of a long-term project of the Software and Knowledge Engineering Laboratory at the Institute of Informatics and Telecommunications of NCSR "Demokritos". Part of this work was done in collaboration with the Department of Informatics of the Technological Institute of Athens, in the context of the research project PA_CO_CLIR (Parallel, Content Based Cross Language Information Retrieval) that is co-funded by the European Social Fund and National Resources (EPEAEK-II)-ARXIMHDHS.

References

1. C.C. Aggarwal and P.S. Yu. An automated system for web portal personalization. In *Proceedings of 28th International Conference on Very Large Data Bases (VLDB 2002)*, pages 1031–1040, 2002.
2. L. Ardissono, L. Console, and I. Torre. An adaptive system for the personalized access to news. *AI Communications*, 14(3):129–147, 2001.
3. K. Bharat, T. Kamba, and M. Albers. Personalized, interactive news on the web. *Multimedia Systems*, 6(5):349–358, 1998.
4. D. Billsus and M.J. Pazzani. A hybrid user model for news classification. In *Proceedings of the International Conference on User Modeling (UM), CISM Courses and Lectures, n. 407*, pages 99–108, 1999.
5. L. Chen and K.P. Sycara. Webmate: A personal agent for browsing and searching. In *Proceedings of the Second International Conference on Autonomous Agents*, pages 132–139, 1998.
6. P. Chesnais, M. Mucklo, and J. Sheena. The fishwrap personalized news system. In *Proceedings of the IEEE 2nd International Workshop on Community Networking Integrating Multimedia Services to the Home*, pages 275–282, 1995.
7. A. Diaz Esteban, M.J. Mana Lopez, M. de Buenaga Rodriguez, J.M. Gomez Hidalgo, and P.G. Gomez-Navarro. Using linear classifiers in the integration of user modeling and text content analysis in the personalization of a web-based spanish news service. In *Proceedings of the Workshop on User Modeling, Machine Learning and Information Retrieval, 8th International Conference on User Modeling (UM2001)*, 2001.
8. T. Kamba, H. Sakagami, and Y. Koseki. Anatagonomy: a personalized newspaper on the world wide web. *International Journal of Human-Computer Studies*, 46(6):789–803, 1997.

9. T. Kurki, S. Jokela, R. Sulonen, and M. Turpeinen. Agents in delivering personalized content based on semantic meta-data. In *Proceedings of the AAAI Spring Symposium Workshop on Intelligent Agents in Cyberspace*, pages 84–93, 1999.
10. G. Paliouras, V. Karkaletsis, C. Papatheodorou, and C.D. Spyropoulos. Exploiting learning techniques for the acquisition of user stereotypes and communities. In *Proceedings of the International Conference on User Modeling (UM), CISM Courses and Lectures, n. 407*, pages 169–178, 1999.
11. G. Paliouras, A. Mouzakidis, C. Ntoutsis, A. Alexopoulos, and C. Skourlas. Pns: Personalized multi-source news delivery. In *Proceedings of the 10th International Conference on Knowledge-Based & Intelligent Information & Engineering Systems (KES), Lecture Notes in Artificial Intelligence, n. 4252*, pages 1152–1161, 2006.
12. G. Paliouras, C. Papatheodorou, V. Karkaletsis, and C.D. Spyropoulos. Clustering the users of large web sites into communities. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 719–726, 2000.
13. G. Paliouras, C. Papatheodorou, V. Karkaletsis, and C.D. Spyropoulos. Discovering user communities on the internet using unsupervised machine learning techniques. *Interacting with Computers*, 14(6):761–791, 2003.
14. D. Pierrakos, G. Paliouras, C. Papatheodorou, and C.D. Spyropoulos. Web usage mining as a tool for personalization: a survey. *User Modeling and User-Adapted Interaction*, 13(4):311–372, 2003.
15. G. Sigletos, G. Paliouras, C.D. Spyropoulos, and M. Hatzopoulos. Combining information extraction systems using voting and stacked generalization. *Journal of Machine Learning Research*, 6:1751–1782, 2005.