Available online at www.sciencedirect.com
ScienceDirect
Journal of Network and Computer Applications ▌(▌▌▌▌) ▌▌▌–▌▌▌
ELSEVIER
Journal of NETWORK and COMPUTER APPLICATIONS
www.elsevier.com/locate/jnca

NectaRSS, an intelligent RSS feed reader

Juan J. Samper, Pedro A. Castillo, Lourdes Araujo, J.J. Merelo*, Óscar Cordón, Fernando Tricas

Technologia de Computadores ETS Ingenieria Informatica y Telecomunicaciones, Depto. Arquitectura, Daniel Saucedo Aranda, 18071 Granada, s/n, Spain Q1

Received 7 February 2007; received in revised form 31 August 2007; accepted 5 September 2007

Abstract

In this paper a novel article ranking method called *NectaRSS* is introduced. The system recommends incoming articles, which we will designate as newsitems, to users based on their past choices. User preferences are automatically acquired, avoiding explicit feedback, and ranking is based on those preferences distilled to a user profile. NectaRSS uses the well-known vector space model for user profiles and new documents, and compares them using information retrieval techniques, but introduces a novel method for user profile creation and adaptation from users' past choices. The efficiency of the proposed method has been tested by embedding it into an intelligent aggregator (RSS feed reader) which has been used by different and heterogeneous users. Besides, this paper proves that the ranking of newsitems yielded by NectaRSS improves its quality with user's choices, and its superiority over other algorithms that use a different information representation method.

Keywords: RSS; Weblogs; Information retrieval; User profiling

*Corresponding author. Tel.: +34 958 243162; fax: +34 958 248993.

E-mail addresses: nectarss@gmail.com (J.J. Samper), pedro@atc.ugr.es (P.A. Castillo), lurdes@lsi.uned.es (L. Araujo), jmerelo@geneura.ugr.es (J.J. Merelo), oscar.cordon@softcomputing.es (Ó Cordón), ftricas@unizar.es (F. Tricas).

1084-8045/$ - see front matter © 2007 Published by Elsevier Ltd.
doi:10.1016/j.jnca.2007.09.001

1. Introduction

A blog or weblog is a website with entries (usually called *posts*) made in journal style and displayed in a reverse chronological order. Weblogs often provide commentaries or opinions on a particular subject, such as gadgets, politics, or local news; some of them work as more personal online diaries. A typical weblog combines text, images, and links to other weblogs, web pages, and other media related to its topic.

One of the advantages of weblogs, and possibly a factor in their success, is that any new post is automatically published in several formats. HTML (Hypertext Markup Language) is the default, but most if not all weblog publishing systems generate other formats too. These formats strip all non-essential information (such as navigation, ads or simply format marks) from the posts, leaving just the newsitem (title and content) and related metadata (such as author and date of publication). One of these formats, based on XML (eXtended Markup Language) is RSS[1]. RSS is read through programs called *feed readers* or *aggregators*, thus the user subscribes to a feed by supplying to their reader a link to the feed; the reader can then check the user's subscribed feeds to see if any of those feeds have new contents since the last time it checked, and if so, retrieves that content and presents it to the user.

The blogosphere offers millions of weblogs on different topics and in different languages; besides, RSS and other similar formats, such as Atom, are increasingly popular, and most web-based publications (such as mainstream media sites, and even website updates from sites such as arXiv[2]) offer it. Daily browsing of even a small percentage of these weblogs can be very tedious and unapproachable in practice. RSS feed aggregators, which read RSS feeds chosen by the user to a desktop program or to a website, avoid website-to-website browsing, but even so, the task of selecting what to read from a few dozen feeds usually exceeds practical limits. Users often get tired of checking information before reaching whatever they are interested in.

In this paper, we propose the *NectaRSS* system (Samper, 2005), for filtering information gathered from the web by scoring it according to the user's implicit preferences, that is, preferences obtained with the only effort of clicking in whatever newsitem he/she is going to actually read. The system incrementally builds user profiles based on the content (heading or extended content) of these choices.

These techniques will be applied in a novel way to an aggregator of contents to endow it with a certain degree of "intelligence", by ordering the information recovered according to the user profile. Experiments have shown that the results of NectaRSS largely improve those obtained offering the information sorted at random and also using a simple binary algorithm which selects as relevant documents those containing the query terms.

The rest of the paper is organized as follows: in Section 2, we review the state of the art on personalized information access systems. In Section 3, we propose novel approaches to providing relevant information that satisfies each user's information need by capturing changes in the user's preferences without the user's effort. In Section 4, we present the experimental results for evaluating our proposed approaches. Finally, we conclude the paper with a summary and directions for future work in Section 5.

[1] RSS is acronym of "Really Simple Syndication".

[2] http://arXiv.org. The site for Physics (and other disciplines too) preprints.

Please cite this article as: Samper JJ, et al. NectaRSS, an intelligent RSS feed reader. J Network Comput Appl (2007), doi:10.1016/j.jnca.2007.09.001

2. State of the art

Recommendation systems have quickly evolved within interactive web environments. Along this line, Schafer et al. (2001) establish a taxonomy of recommendation systems attending to three categories of features: income and exit functionalities, recommendation methods and design dependent aspects. Middleton et al. (2001) present the recommendation system *Quickstep* to find scientific and research papers. The user's preferences are acquired by monitoring his/her behavior when navigating on the web, applying automatic learning techniques associated with an ontological representation. Mizzaro and Tasso (2002) consider personalization techniques in systems used to access electronic publications; they distinguish between persistent and ephemeral personalization, and apply them to filtering and retrieval of information through a specialized web portal.

Merelo et al. (2006) propose a system that recommends weblogs to a reader based on its current reading habits; the system uses the results of a pool and applies association rules (Agrawal and Srikant, 1994). The goal is to find *attribute-value* conditions which appear frequently in a data set.

Products based on this technology are also offered on the web. For instance, Criteo[3] is a recent recommendation engine, based on collaborative filtering technology that includes a predictive tool for e-commerce recommendations and portal ads.

These recommendation systems have not been so far applied to content aggregators, which are a relatively recent product. There is a large list of aggregator programs,[4] most of which offer similar functionalities. In this paper we present what, up to our knowledge, is the first recommendation system that works as an aggregator and ranks news items according to a user profile that is automatically computed from past user choices.

3. NectaRSS

The system that we propose, called *NectaRSS,*[5] is designed to rank newly arrived information according to an automatically elaborated user profile. We will restrict our system to information that appears periodically and whose structure is similar to a news story. Thus, the pieces of information the system retrieves will be generically referred to as *newsitems*, each of which will be composed by a *headline*, a *hyperlink* to its content and optionally a summary. Information aggregators usually show the headline hyperlinked to the content, and sometimes the summary; besides, the hyperlink is a unique ID for the newsitem. We will assume that if a newsitem is shown in the aggregator, and the user clicks on it, it corresponds to a topic the user is interested on, and thus it will be used to build his/her profile.

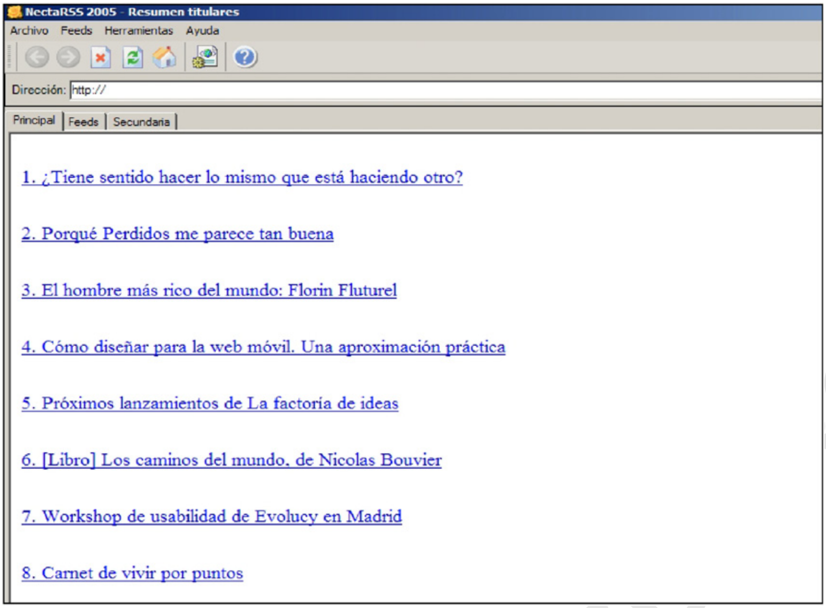
Additionally, NectaRSS uses *sessions*; each *session* is a complete execution of the system, understood as the retrieval and scoring of the information available on the web in that particular moment (according to the preferred sources), the monitoring of user choices and the calculation of the user profile at the end of the execution of the system. This ensemble of techniques used in each session is original and exclusive to the NectaRSS system,

[3] The Criteo recommendation engine is available at http://www.criteo.com.

[4] RSSfeeds. The RSS, Atom and XML directory and resource, March 2006. On line: http://www.rssfeeds.com/readers.php

[5] Nectar means essence or nectar in Spanish.

Please cite this article as: Samper JJ, et al. NectaRSS, an intelligent RSS feed reader. J Network Comput Appl (2007), doi:10.1016/j.jnca.2007.09.001



1. ¿Tiene sentido hacer lo mismo que está haciendo otro?

2. Porqué Perdidos me parece tan buena

3. El hombre más rico del mundo: Florin Fluturel

4. Cómo diseñar para la web móvil. Una aproximación práctica

5. Próximos lanzamientos de La factoría de ideas

6. [Libro] Los caminos del mundo, de Nicolas Bouvier

7. Workshop de usabilidad de Evolucy en Madrid

8. Carnet de vivir por puntos

Fig. 1. Typical aspect of the NectaRSS experimental system.

configuring a kind of *intelligent content aggregator*. In this context, *intelligence* is the capability of automatically adapting to a context or service based on implicit behavior and learning instead of explicit solicitation from users (Pasi, 2003). In Fig. 1 the typical aspect of the experimental system NectaRSS is shown.

In principle, NectaRSS might work on the client computer or on a server. However, making it work on a server includes an additional infrastructure, which makes it experimentally more difficult; that is why, right now, it has been implemented on a client using Microsoft tools. A typical screenshot is shown in Fig. 1. This also avoids privacy concerns, since all profiles are stored on the client, and are as secure as the client computer itself. It also avoids scalability problems, since all profile calculations are done on the client, not on a server. Besides, as we will see in the following, the user profile finally takes the form of the usual “bag of words”, i.e., a list of weighted terms defining the user’s interests, and those weights are computed “on the fly”; so the storage requirements are very small. A copy of the experimental system NectaRSS can be found in the following link: http://www.neoyet.com/NectaRSSvs.zip.

3.1. User profile construction based on browsing history

In our approach, the user profile is built in an implicit way: the user will not have to take any additional action such as explicit feedbacks or evaluations to build his profile, which will be constructed automatically according to his navigation history by the newsitems headlines which are presented to him (Hanani et al., 2001).

The user profile P will be generated and updated at the end of each session from two different session profiles P_s (built from the index terms included in the newsitems

Please cite this article as: Samper JJ, et al. NectaRSS, an intelligent RSS feed reader. J Network Comput Appl (2007), doi:10.1016/j.jnca.2007.09.001

headlines) and P_r (which considers the index terms in the summaries) whose relative importance can be adjusted; they will be defined in the next paragraphs.

Both the newsitems and the user profile will be represented using the vector space model proposed by (Salton, 1971; Salton and McGill, 1983). Thus, we define S_j ($j = 1,2,\ldots,N$) as the number of news headlines that the user has chosen in session j. In each session, P_s will be built as follows: first of all, we will denote the characteristic vector w^h of the headline h ($h = 1,2,\ldots,S_j$) as follows:

$$w^h = (w^h_{t_1}, w^h_{t_2}, \ldots, w^h_{t_m}), \tag{1}$$

where m is the number of different terms in the headline h and t_k denotes each specific term. In our study the stop words are left out, but stemming is not performed on the text.

Using the *tf* scheme, or term frequency, each index term weight $w^h_{t_k}$ in w^h is defined as

$$w^h_{t_k} = \frac{tf_{h,k}}{\sum_{s=1}^{m} tf_{h,s}}, \tag{2}$$

where $tf_{h,k}$ is the frequency of the term t_k in the headline h.

Finally, P_s is defined as

$$P_s = (ps_{t_1}, ps_{t_2}, \ldots, ps_{t_u}), \tag{3}$$

where u is the number of different terms in all the headlines chosen in session j and t_k denotes each specific term; ps_{t_k} is defined using the formulation (2) as follows:

$$ps_{t_k} = \frac{1}{S_j}\sum_{h=1}^{S_j} w^h_{t_k}. \tag{4}$$

On the other hand, we consider a characteristic vector w^{hr} composed of the terms that appear in the summary r associated with a headline h. Sr_j ($j = 1,2,\ldots,R$) is the number of headlines with associated summary which have been chosen by the user in the session j. Similarly to the session profile, P_r will be the summary elaborated with the terms of the summaries following a process very similar to what was done with the headlines. The characteristic vector w^{hr} of the summary associated with a headline h ($h = 1,2,\ldots,Sr_j$) is defined as

$$w^{hr} = (w^{hr}_{t_1}, w^{hr}_{t_2}, \ldots, w^{hr}_{t_v}), \tag{5}$$

where v is the number of different terms in the summary r associated with the headline h and t_k denotes each specific term. Using the *tf* scheme of the frequency of the term, each element $w^{hr}_{t_k}$ in w^{hr} is defined as

$$w^{hr}_{t_k} = \frac{tf_{hr,k}}{\sum_{s=1}^{v} tf_{hr,s}}, \tag{6}$$

where $tf_{hr,k}$ is the frequency of the term t_k in the summary r associated with the headline h.

Then, P_r is defined as

$$P = (pr_{t_1}, pr_{t_2}, \ldots, pr_{t_z}), \tag{7}$$

where z is the number of different terms in all the summaries chosen in session j and t_k denotes each term.

Please cite this article as: Samper JJ, et al. NectaRSS, an intelligent RSS feed reader. J Network Comput Appl (2007), doi:10.1016/j.jnca.2007.09.001

We define each element pr_{t_k} using formula (6) as follows:

$$pr_{t_k} = \frac{1}{Sr_j}\sum_{h=1}^{Sr_j} w^{hr}_{t_k}. \tag{8}$$

The elaboration of the user profile P at the end of each session proceeds as follows: let P_j be the user profile stored after the session j, and let $P_{s,j+1}$ be the profile of the session j+1. Then, for all $t_k \in (t_1,t_2,\ldots,t_u)$, where u is the number of different terms found in the session j+1 and t_k denotes each term, the profile P_{j+1} built at the end of session j+1 is given by the following expressions:

$$P_{j+1} = (0.5 \cdot P_j + 0.5 \cdot P_{s,j+1}) + P_{r,j+1} \quad \text{if} \;\; p_{t_k} \in P_j, \tag{9}$$

$$P_{j+1} = P_{s,j+1} + P_{r,j+1} \quad \text{if} \;\; p_{t_k} \notin P_j, \tag{10}$$

where $P_{r,j+1}$ is the profile P_r in the session j+1. When $j = 1$, the profile is built using the initial session and summary profiles $P_{s,1}$ and $P_{r,1}$.

We should note that this way of updating the user profile is similar to the classic relevance feedback techniques for the vector model such as *Ide dec-hi* (Salton and Buckley, 1990). Our algorithm is summarized in the text Box 1.

3.2. Retrieving newsitems from the user profile

In order to compute the retrieval status value (RSV) associated with a newsitem headline h, we will compare its corresponding characteristic vector $w^h = (w^h_{t_1}, w^h_{t_2}, \ldots, w^h_{t_m})$ with the user profile $P = (p_{t_1}, p_{t_2}, \ldots, p_{t_n})$.

The similarity, $sim(P,w^h)$, between the user profile P and the characteristic vector of the headline h, w^h, is calculated applying the cosine measure (Salton, 1989):

$$sim(P, w^h) = \frac{P \cdot w^h}{|P| \cdot |w^h|} = \frac{\sum_{k=1}^{m} p_{t_k} \cdot w^h_{t_k}}{\sqrt{\sum_{k=1}^{m} (pt_k)^2 \cdot \sum_{k=1}^{m} (w^h_{t_k})^2}}. \tag{11}$$

The similarity value given by Eq. (11) is the RSV for headline h according to the user profile P. Then the newsitems headlines are ordered for each user according to his/her profile, presenting in the first positions those headlines with greater RSV.

Box 1
NectaRSS algorithm outline.

```
For each session
  1. Obtain RSV for each heading matching its *characteristic vector* to user profile vector **P**.
  2. Rank headings according to RSV.
  3. Repeat.
     1. Store info about newsitems chosen by the user: $w^h$ and $w^{hr}$.
  4. Build *session and abstracts profiles* $P_{s,j+1}$, $P_{r,,j+1}$.
  5. Update user profile P.
```

Please cite this article as: Samper JJ, et al. NectaRSS, an intelligent RSS feed reader. J Network Comput Appl (2007), doi:10.1016/j.jnca.2007.09.001

Another way of computing the RSV associated with a headline h involves applying a simple criterion of binary relevance: a document is relevant if it contains the requested word, without discriminating between different degrees of relevance, which is equivalent to a Boolean algorithm. This simple criterion of binary relevancy is introduced to be compared with our NectaRSS algorithm afterwards.

Thus, given the corresponding characteristic vector of the headline h, $w^h = (w^h_{t_1}, w^h_{t_2}, \ldots, w^h_{t_m})$, and the user profile $P = (p_{t_1}, p_{t_2}, \ldots, p_{t_n})$, the similarity, $sim(P, w^h)$, between the user profile P and the characteristic vector of the headline h, w^h, will be given by the following expression:

$$\begin{aligned} &sim(P, w^h) = 0 \quad \text{if} \;\; p_{t_k} \notin P \quad \forall_{t_k} \in (t_1, t_2, \ldots, t_m), \\ &sim(P, w^h) = 1 \quad \text{otherwise.} \end{aligned} \tag{12}$$

The similarity value given by the former expression is the RSV of the headline h according to the user profile P. It is enough that any term of the headline is found in the user profile to consider that there exists similarity between both of them and assign an RSV value of 1 to this similarity. Then, only those newsitems headlines with RSV of value 1 are shown to the user.

4. Experiments and results

In order to obtain reliable results and determine the validity of our proposal, several sessions were carried out with different real users. Each user was offered a headline list, ordered by RSV, from which he selected whatever headlines he found interesting. The number of headlines offered, in this case 14, allowed the user to see all of them at the same time, without the need of vertical page displacements (which would introduce a bias in the selection). Fifteen users with heterogeneous thematic interests tested the system; which is considered enough to reach some statistical significance, taking into account the difficulty of obtaining experimental subjects for this kind of experiment. At the beginning of each experiment, the user profile was empty and during the sessions it was elaborated and completed.

4.1. Measures for the experimental evaluation of the system

The well-known average RSV and R-Precision measures, which are defined later, have been used to check the validity of the proposed method. Values obtained during experiments and conclusions drawn will be explained below.

4.1.1. Average RSV of a set of newsitems headlines and maximum mean RSV

In each session the user is offered a certain number T of headlines and he/she must choose those of his/her interest, named the *chosen headlines* or $E(T)$. Then, the average RSV or $\overline{v(E(T))}$ is computed for the set of headlines selected by the user in that session. On the other hand, it can be computed a maximum average RSV value or $\overline{v_{\max}(T)}$ for that set of headlines. It is obtained when the news (N) chosen are the same to the first N headlines (in RSV order) offered by the system in a given session. This $\overline{v_{\max}(T)}$ value is computed automatically by the system. To quantify the relationship between the value $\overline{v(E(T))}$ of the headlines chosen by the user and the value $\overline{v_{\max}(T)}$, the *rate* C_D is defined:

$$C_D = \frac{\overline{v(E(T))}}{\overline{v_{\max}(T)}}, \tag{13}$$

where $\overline{v_{\max}(T)}$ is the average of the first N RSV values associated with the N headlines with greater RSV among those offered to the user, with N being the number of headlines chosen by the user.

4.1.2. The R-Precision

According to Baeza-Yates and Ribeiro-Neto (1999), a simple summary value for a set of headlines ranked according to their RSV can be generated by computing the *precision* in the Rth position of the ordered list, with R being the total number of relevant headlines of the session. In the case of NectaRSS, the latter value corresponds to the number of headlines chosen by the user among those offered by the system.

Hence, the R-Precision measure is then defined as

$$RP(i) = \frac{posR(E(T_i))}{card(E(T_i))}, \tag{14}$$

where $posR(E(T_i))$ is the number of headlines chosen among the first R headlines orderly offered to the user in session i, and the value of $card(E(T_i))$ is the total number of headlines chosen in such session.

4.2. Test of the algorithm NectaRSS with different users

NectaRSS has been tested with different users. In each session the user is shown a selection of 14 headlines ordered by RSV; this quantity has been chosen so that all headlines are shown at once in a single page, without forcing the user to scroll down to get them. The total amount of headlines varies with the sessions. The selection of headlines is carried out using a fixed group of sources of information with certain thematic variety and frequent upgrades. These sources of information are shown in Table 1.

Every one of the 15 voluntary users carried on 2 training sessions and 30 experimental sessions, choosing the information of their interest from among those 14 headlines offered by the system; experimental sessions are also used to improve the user profile, at the same time that it is showing ranked headlines (ORDER experiment). Furthermore, in order to

Table 1
Group of sources of information used to test the operation of the algorithm NectaRSS with different users

http://abraldes.net/feeds/elmundo.xml
http://backends.barrapunto.com/barrapunto.rss
http://www.kriptopolis.org/rss
http://xataka.com/es/index.xml
http://www.maestrosdelweb.com/blog/index.rdf
http://www.filmica.com/sonia_blanco/index.xml
http://rss.time.com/web/time/rss/top/index.xml
http://www.artnovela.com.ar/backend.php
http://www.stardustcf.com/rdf.asp
http://www.pjorge.com/rss
http://mp.blogalia.com/rdf.xml
http://www.elblogsalmon.com/index.xml

http://bitacoras.com/noticias/index.xml
http://abraldes.net/feeds/marca.xml
http://www.ecuaderno.com/index.xml
http://www.alzado.org/xml/alzado.xml
http://www.tintachina.com/index.xml
http://www.britannica.com/eb/dailycontent/rss
http://reviews.cnet.com/4924-5_7-0.xml
http://www.blogdecine.com/index.xml
http://furtivos.bloxus.com/rdf.xml
http://atalaya.blogalia.com/rdf.xml
http://www.librys.com/feed.rss

compare results, the participants run over 30 new test sessions where every user chooses at random headlines of his/her interest among 14 offered (RANDOM sub-experiment). Obviously, since there is no user profile at the beginning of the experiment, headlines in the first training session are randomly ordered.

Results obtained for C_D in the 30th experimental session for the 15 users are shown in Fig. 2. As it can be seen, all users yield better values in the ORDER case, than in the RANDOM case. This means that the headlines chosen by the user in the ORDER case have greater RSV than those chosen in the RANDOM case, that is, the user finds a larger amount of interesting headlines among these presented to him/her when the user profile computed by NectaRSS is used to rank headlines.

The R-Precision measure has only been applied to the ORDER case, since it needs an ordered set of headlines to compute the precision for the Rth position in the ranked headlines list. To compare the R-Precision throughout 30 experimental sessions the user with the worst (#8) and the best (#11) average for this measure have been chosen, which act as *de facto* upper and lower bounds in performance; the rest of users will yield figures between these two. Fig. 3 shows graphically the values of the R-Precision obtained by these users in 30 experimental sessions along with the trend line from each one, Linear (User # 8) and Linear (User #11).

As it can be seen in Fig. 3, the slope of the trend line is positive in both cases. This indicates that the system improves its headline rankings with the number of sessions, which is a desired behavior, and it shows that the ranking offered by NectaRSS increasingly matches the one that would have been done by the user himself.

This means that the user profile constructed by NectaRSS really characterizes the corresponding user and permits improving the response of the information retrieval systems consulted by this user.

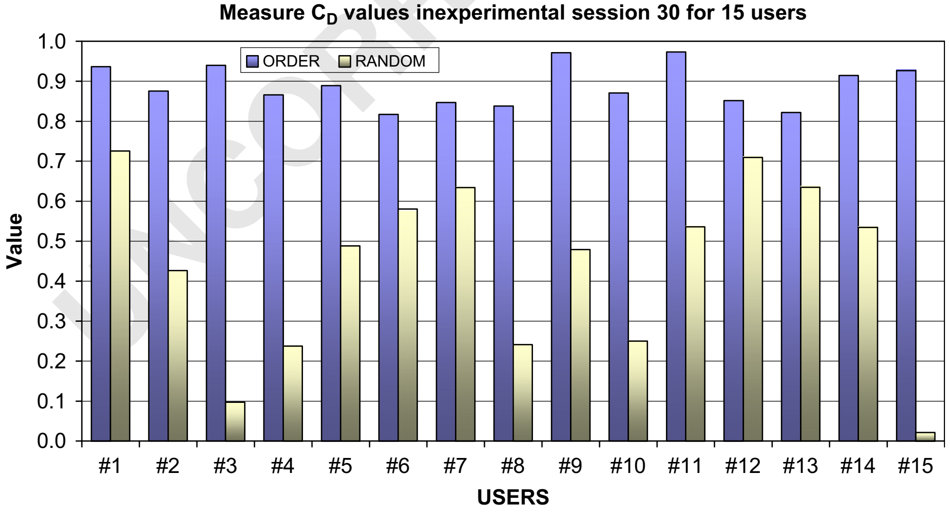


Fig. 2. Results obtained by 15 users for the C_D rate in the experimental session 30, when headlines are offered ranked by the NectaRSS algorithm (ORDER case), and at random (RANDOM case). As can be seen, NectaRSS outperforms the case where headlines are offered at random.

Please cite this article as: Samper JJ, et al. NectaRSS, an intelligent RSS feed reader. J Network Comput Appl (2007), doi:10.1016/j.jnca.2007.09.001

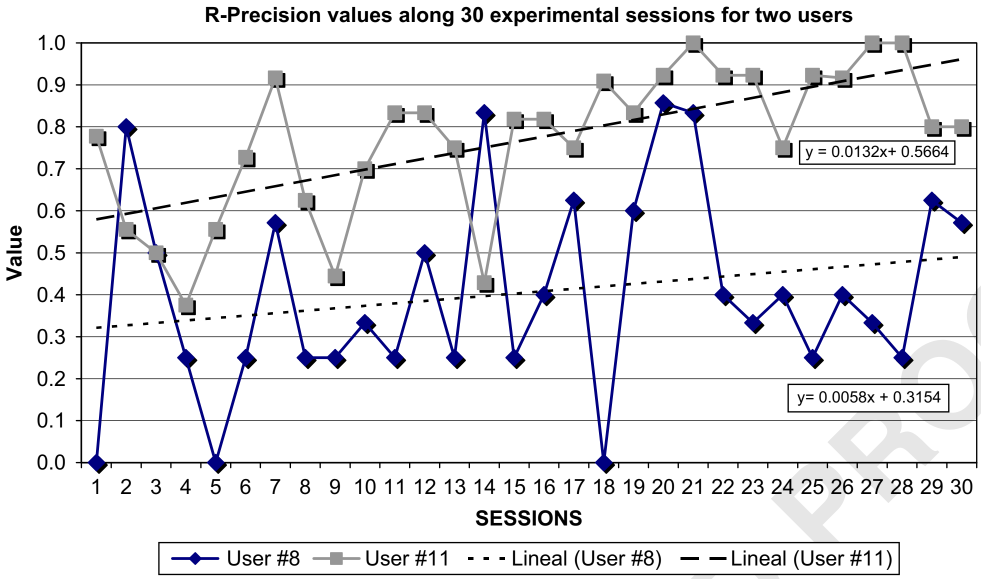


Fig. 3. Results obtained by the user #8 and by the user #11 for the R-Precision throughout 30 experimental sessions, together with the trend lines from the data. It is observed a favorable evolution of the R-Precision.

4.3. Headline scoring using a Boolean algorithm

We have also tried to prove that NectaRSS outperforms more naïve algorithms and, in particular, that the vector space representation is better than a more straightforward purely Boolean (binary) representation, which has been introduced in Section 3.2.

In this experiment the 15 voluntary users were submitted to 30 additional experimental sessions with NectaRSS, configured now to rank incoming newsitems through a simple criterion of binary relevancy.

In the new sessions, the users have been presented the same set of news that was used to rank the information with the cosine measure. This allows us to compare the results obtained by NectaRSS in the ORDER case with those obtained using the Boolean algorithm.

The average C_D for the 15 users can be seen in Fig. 4, which shows the C_D rate by user for the two cases: the cosine measure and the Boolean algorithm.

As it can be seen above, the C_D average rates along the 30 sessions are better for all users in the NectaRSS case than in the binary case. Also, the two-tailed *P*-value of the *t*-test is 0.0008, considered extremely significant, thus the means differ significantly. We can conclude that the headlines presented to the user applying the binary algorithm are much less related to the user's interest than the ones presented by applying NectaRSS.

The R-Precision measure for both kinds of ranking/similarity functions have also been compared, and results are shown in Fig. 5. In this case the two-tailed *P*-value of the *t*-test is 0.0006, considered extremely significant, thus the means also differ significantly. We can conclude in this case that the improvement in rankings shown by NectaRSS is better for the 15 users than the one shown by the binary measure.

Please cite this article as: Samper JJ, et al. NectaRSS, an intelligent RSS feed reader. J Network Comput Appl (2007), doi:10.1016/j.jnca.2007.09.001

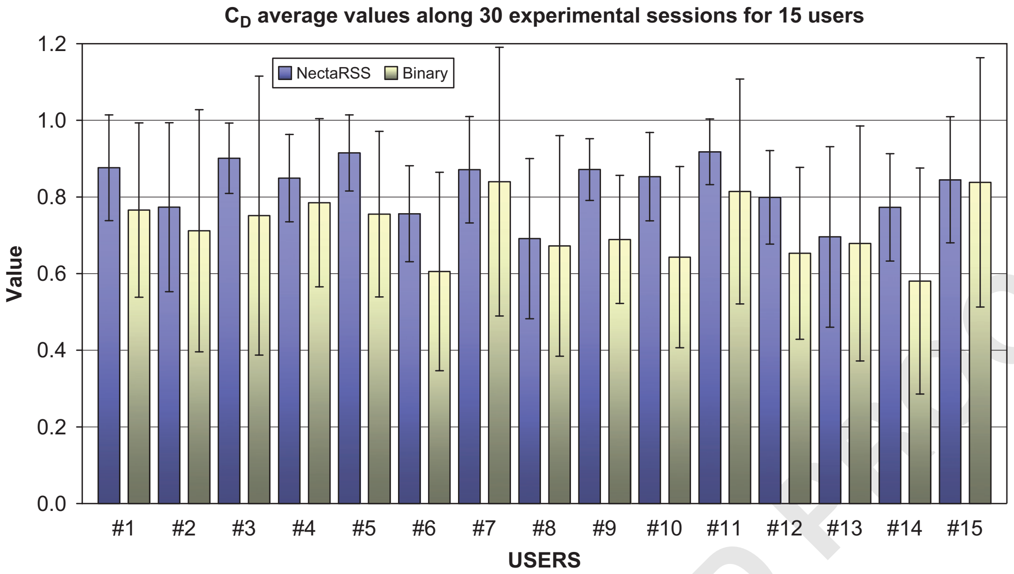


Fig. 4. C_D average measures along 30 experimental sessions, using the cosine measure to score the headlines with an RSV value (NectaRSS) and using a binary algorithm to calculate such RSV value (Binary).

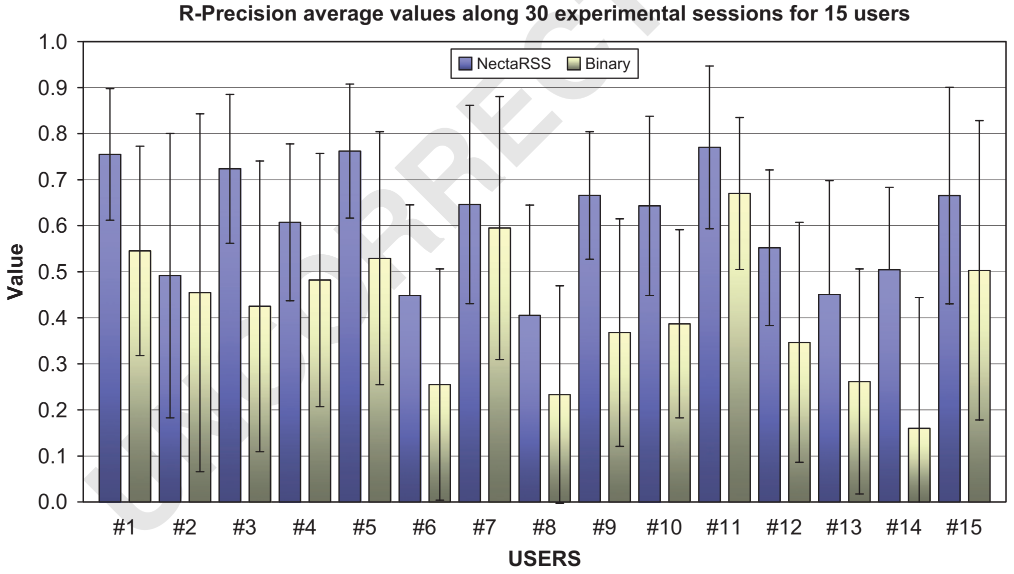


Fig. 5. R-Precision average measure along of 30 experimental sessions, when the cosine measure (NectaRSS) and a binary algorithm (Binary) are used to score the headlines.

In order to have a global idea of the behavior of each considered algorithm, we have analyzed the results of both, the R-Precision measure and the C_D rate, obtained by all the users at each experimental session. At each session, we compute the difference between the

Please cite this article as: Samper JJ, et al. NectaRSS, an intelligent RSS feed reader. J Network Comput Appl (2007), doi:10.1016/j.jnca.2007.09.001

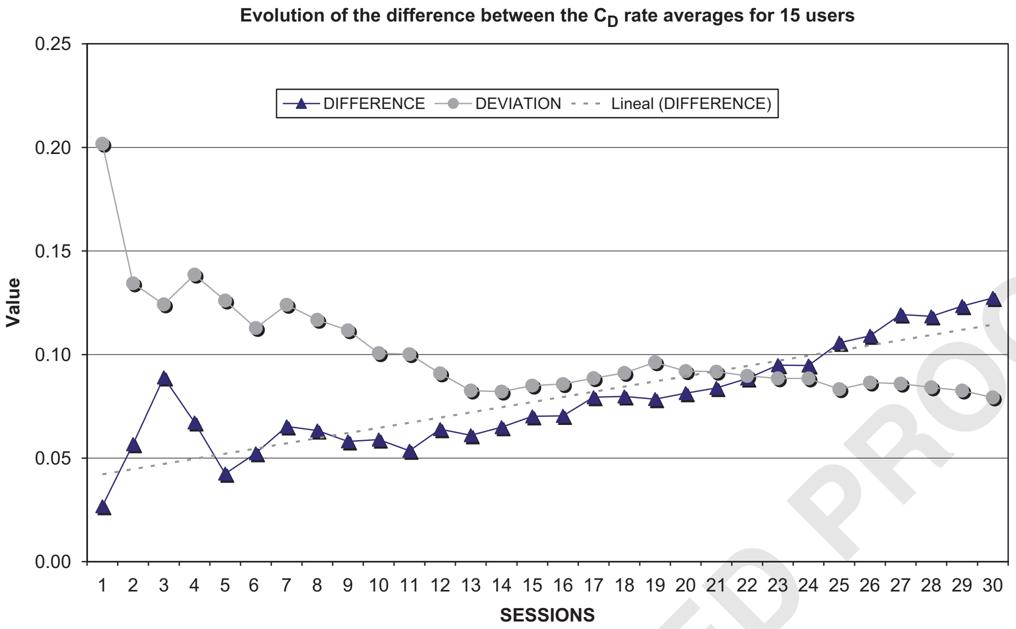


Fig. 6. Evolution of the difference between averages (with standard deviation) with respect to the C_D rate during 30 experimental sessions.

average values, for the 15 users, obtained with the NectaRSS algorithm (average NectaRSS), and the binary algorithm (average binary). Positive values indicate that NectaRSS beats the binary algorithm.

Figs. 6 and 7 show the values of this difference for the C_D rate and the R-Precision measure, respectively, along the 30 experimental sessions.

Fig. 6 shows that the difference between the C_D rate averages increases along the sessions, what means that the advantage of NectaRSS over the binary algorithm gets larger with the system training. We can also observe that the deviation clearly decreases with the sessions, denoting that the values are less scattered. Thus, we can observe a progressive improvement of the NectaRSS algorithm respect to the purely binary algorithm for all the users.

Fig. 7 shows the difference between the R-Precision values. This difference also increases with the sessions, indicating that the advantage of NectaRSS over the binary algorithm improves with the sessions. The deviation also decreases with the sessions.

5. Conclusions and discussion

NectaRSS has demonstrated to be useful in the personalization of intelligent retrieval systems, providing them with flexibility and some kind of *intelligence*. Considering the experimental results obtained in Section 4, we can assert that the newsitems scoring achieved applying the user profile computed via the NectaRSS algorithm is significantly useful. The user is shown more interesting documents or, at least, more documents related to his/her preferences. These advantages of the proposed system have been shown for different and heterogeneous users (in the sense that their technical background and

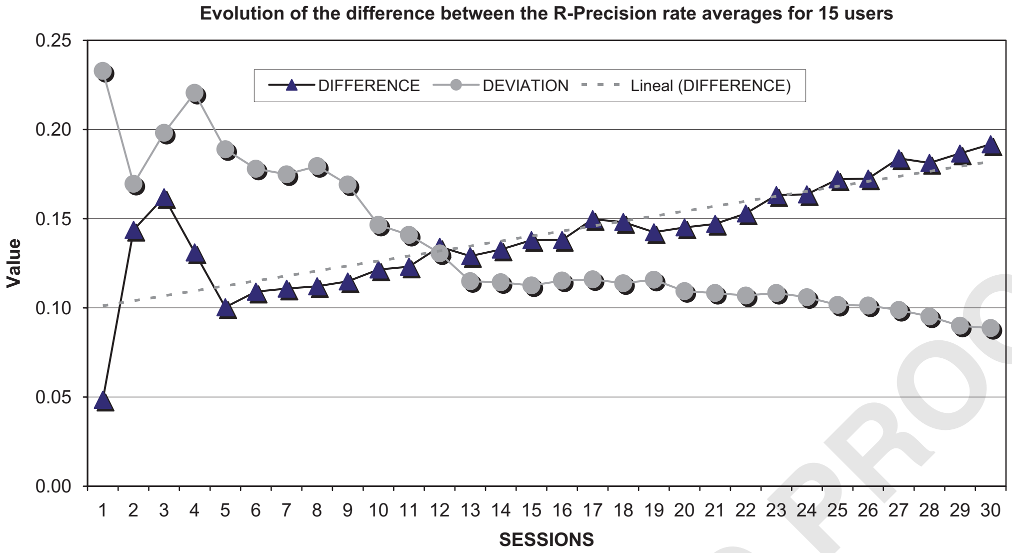


Fig. 7. Evolution of the difference between averages (with standard deviation) with respect to the R-Precision rate during 30 experimental sessions.

preferences are different). Comparing the NectaRSS algorithm with a different way of scoring the information retrieved, namely a simple binary algorithm, it is observed that in the former case the ranking of the information is more appropriate, and moreover, the improvement of the system response is faster.

Furthermore it has been observed a positive evolution of the NectaRSS algorithm throughout the experimental sessions for all users, obtaining better results than a simple binary algorithm and with increasingly stable values.

We can conclude that our NectaRSS system has been able to endow a certain degree of "intelligence" to a typical content aggregator, filtering its RSS contents better than either a random system or a system with a simple binary scoring. This approach is novel in two different senses: first, the profile building algorithm has been designed *ab initio*, although it is based on mainstream information retrieval ideas, and second, it is the first time this kind of algorithms has been used on feed aggregators.

The proposed system can be improved along three different lines of future work:

- Application of linguistic analysis to the retrieved information, which allows the fine tuning of the features used to characterize the documents by selecting particular types of words, such as names, or using stemming over extracted words.
- Use of web text collections[6] specifically defined for the evaluation of information retrieval systems. Since queries and relevance assessments are available for these collections, we plan to use them to evaluate our system.
- More extensive experimentation, using more users and a wider source selection, to confirm these data and analyze if there are differences among users.

[6] Web Research Collections. TREC Web & Terabyte Tracks. http://ir.dcs.gla.ac.uk/test_collections/

Please cite this article as: Samper JJ, et al. NectaRSS, an intelligent RSS feed reader. J Network Comput Appl (2007), doi:10.1016/j.jnca.2007.09.001

Acknowledgments

This paper has been partially funded by projects TIC2003-09481-C04-04 and TIN2007-68083-C02-01 awarded by the Spanish Ministry of Science and Technology; P06-TIC-02025, awarded by the regional Science and Technology Council, and by Resolution 8-7-2004 of General Management of Educative Innovation and Professorship Formation of the Science Department of the Regional Government.

References

Agrawal R, Srikant R. Fast algorithms for mining association rules. In: Proceedings of the 20th international conference on very large data bases, VLDB; 1994.

Baeza-Yates R, Ribeiro-Neto B. Modern information retrieval. New York, Reading, MA: ACM Press, Addison-Wesley; 1999.

Hanani U, Shapira B, Shoval P. Information filtering: overview of issues, research and systems. User Modelling User-Adapted Interact 2001;11:203–59.

Merelo JJ, Carpio J, Tricas F, Ferreres G, Prieto B, Castillo PA. Weblog recommendation using association rules. In: Kommers, Isaías, Goikoetxea, editors. Web-based communities 2006, Proceedings of the international conference. p. 127–32. Q2

Middleton S, De Roure D, Shadbolt N. Capturing knowledge of user preferences: ontologies in recommender systems. In: Proceedings of the 1st international conference on knowledge capture (K-Cap2001), Victoria, BC, Canada; 2001. p. 100–7. Available also from: 〈http://arxiv.org/abs/cs.LG/0203011〉.

Mizzaro S, Tasso C. Ephemeral and persistent personalization in adaptive information access to scholarly publications on the web. In: De Bra P, Brusilovsky P, Conejo R, editors. Adaptive hypermedia and adaptive web-based systems, second international conference AH2002. Springer; 2002. p. 306–16.

Pasi G. Intelligent information retrieval: some research trends. In: Benítez J, Cordón O, Hoffmann F, Roy R, editors. Advances in soft computing engineering design and manufacturing. Berlin: Springer; 2003. p. 157–71.

Salton G. The SMART retrieval system. Englewood Cliffs, NJ: Prentice-Hall; 1971.

Salton G, McGill MJ. Introduction to modern information retrieval. New York: Computer Science Series, McGraw-Hill; 1983.

Salton G. Automatic text processing: the transformation, analysis and retrieval of information by computer. Reading, MA: Addison-Wesley; 1989.

Salton G, Buckley C. Improving retrieval performance by relevance feedback. J Am Soc Inform Sci 1990;41(4):288–97.

Samper JJ. Study and evaluation of an intelligent system for recovery and filtering information of Internet. PhD thesis, University of Granada, 2005.

Schafer JB, Konstan J, Riedl J. Electronic commerce recommendation applications. J Data Min Knowl Discovery 2001;5(1–2):115–52.