Journal of Universal Computer Science, vol. 11, no. 2 (2005), 229-249
submitted: 31/7/04, accepted: 15/12/04, appeared: 28/2/05 © J.UCS

RankFeed - Recommendation as Searching without Queries: New Hybrid Method of Recommendation

Maciej Kiewra
(Fujitsu Services, Spain
mkiewra@mail.fujitsu.es)

Abstract: The paper describes RankFeed a new adaptive method of recommendation that benefits from similarities between searching and recommendation. Concepts such as: the initial ranking, the positive and negative feedback widely used in searching are applied to recommendation in order to enhance its coverage, maintaining high accuracy. There are four principal factors that determine the method's behaviour: the quality document ranking, navigation patterns, textual similarity and the list of recommended pages that have been ignored during the navigation. In the evaluation part, the local site's behaviour of the RankFeed ranking is contrasted with PageRank. Additionally, recommendation behaviour of RankFeed versus other classical approaches is evaluated.

Keywords: Recommendation, Information Retrieval, Web Mining, Personalization, Adaptive Systems, ROSA
Categories: H.3.1, H.3.3,

1 Introduction

There are many similarities between searching and recommendation. Firstly, both of them have the same purpose: to present relevant information to the users basing on their necessities. In case of searching, these necessities are explicitly expressed by queries, whereas recommendation tries to predict users' preferences using various techniques such as collaboration filtering, content analysis etc. Secondly, both of them order the relevant information into a ranking and present their top to the users. Finally, searching and recommendation change the presented information due to the user interactions (requesting of a new page, query modification etc.). For all these reasons, in some sense recommendation can be regarded as searching without query specification.

Normally, searching is an iterative process. The user formulates and sends a query to the server and then it is refined depending on the search result in order to improve accuracy and coverage of the pages from the top of the ranking. Refinement can be performed either manually (the user changes the query) or automatically. The relevance feedback is one of the most typical methods of query refinement. The previous query and documents that the user has marked as relevant or irrelevant are used to obtain a new query. Similarly, the list of recommended documents may adapt as a result of the users' behaviour.

The purpose of this document is to present a hybrid recommendation method called *RankFeed* that operates in a single web site and tries to enhance the quality and coverage of recommendation using well-known searching issues: the document ranking and the user feedback. In this case, the feedback is acquired in the transparent and no invasive way and it depends on the visited pages and the list of recommended documents that have been ignored during the navigation. The rest of this paper is organized as follows: Section 2 describes typical recommendation methods, underlying their principal inconveniences, Section 3 presents the general idea of the new method, Section 4 is dedicated to initial ranking calculation, Section 5 shows how the initial ranking evolves due to the positive and negative feedback. Finally, Sections 6 and 7 present respectively the method's evaluation and final conclusions.

2 Related Works

2.1 Traditional Approaches to Recommendation

Dynamic development of the World Wide Web system and information overload have entailed interest in anticipation of users' necessities. There are many approaches to recommendation. The comprehensive guide through existing techniques can be found at [Montaner, 03]. One of the earliest methods: collaborative filtering [Shardanand, 95], [Buono, 02] and demographic filtering [Krulwich, 97] expect explicit users' ratings of the view content (for example music albums or artists) in order to recommend the content positively scored by the users with similar features. This approach is not very appropriate for web page recommendation because many users' opinions are needed to cause the method work smoothly. First of all, the visitors are not willing to rate the pages. Moreover, since there are too many pages that appear and disappear every month, it is quite impossible to acquire enough opinions on time.

The second group of methods is based on textual content analysis [Lieberman, 95] [Pazzani, 99], [Mooney, 00]. In this case recommended items for, example documents, are represented as vectors whose coordinates correspond to textual features (descriptors). The items similar to those that have been seen are recommended basing on vector similarities calculated, for example, with the cosine measure:

$$cos(\mathbf{a},\mathbf{b}) = \frac{\sum_{i=1}^{N} a_i * b_i}{\sqrt{\sum_{i=1}^{N} (a_i)^2 * \sum_{j=1}^{N} (b_i)^2}}$$

Where N is the dimension of the vectors **a** and **b**.

Content-based recommendation has several drawbacks. The most important is the suspicious quality of recommended content due to the fact that there is no user feedback. For example let's assume that there are two pages about java programming: the first one d_g is very good and frequently visited, and the second d_b is a test page that contains only one word: *java*. If the user entered the page $d_{g,}$ it would be quite probable that the hyperlink to the document d_b would appear as a recommendation item. Moreover, content based recommendation always promotes very similar items that can influence negatively on the recommendation coverage (the user will obtain only the same or almost the same items and not necessarily the complete content related to the searched information).

Another interesting issue is based on applying Data Mining techniques [Madria, 99], [Booley, 99], [Srivastava, 00], [Kazienko, 03a]. The most popular is the usage of association rules and clustering. In the first case, items visited together are recommended to the user. Applying association rules to the web recommendation has three principal shortcomings. First of all, it is quite impossible to recommend anything to new or separately visited documents. Additionally, recommendation is quite static especially in the basic format when the most often co-visited documents are displayed (limited coverage). Finally, many pages are visited together due to the web site structure (for example many visitors pass through the home page in order to access to desired information).

The second solution benefits from clustering. Normally, user sessions or documents are clustered. The former permits the typical usage pattern to be obtained and the latter discovers thematic groups. Typically, clustering is performed within the vector space model. In case of usage pattern discovery, one vector is created for every user historical session where non-zero coordinates correspond to visited documents. It is possible to discover typical navigational patterns, clustering the vectors mentioned above. Then a current session vector is built for the on-line user. Similarly to the previous vectors, non-zero coordinates are related to the documents visited in the current session.

The closest usage pattern can be obtained by calculating similarities between the current session vector and the clusters represented by their mean vectors denominated centroids. Once the closest centroid is known, the documents with the highest coordinates are recommended to the user. In case of thematic groups, the process is practically the same except substitution of historical sessions with descriptor vectors.

The clustering approach seems to solve partially the coverage problem due to the fact that not only the closest or the most often co-visited pages can be displayed to the user, but also those that are close to the cluster's centroid. Nevertheless, it is laden with other shortcomings. The gravest is connected with the promotion of the strongest cluster members although they can be relatively far from the user interest. Let's assume that the user wants to read something about java programming in mobile devices. Nevertheless, there are two clusters related to his or her interests: java programming and mobile programming. Therefore the area of the searched information is situated on the border of those two clusters (see the figure 1)

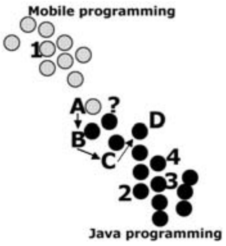
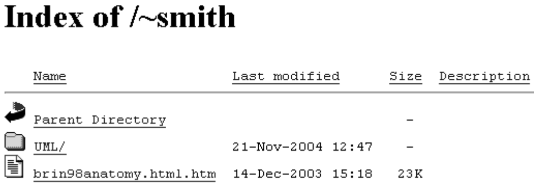


Figure 1: An example of data mining recommendation when the user interests are situated on the border of two clusters

Figure 1 presents an imaginary distribution of the documents reduced to the two dimensions. Black dots are the documents classified to java programming cluster and the grey dots belong to the "mobile" programming cluster. Documents visited by the user are labelled with letters, whereas the documents recommended in each step are labelled with numbers (when the document *A* has been visited, the document *1* has been recommended etc.) It can be easily noticed that the pages from the centres of two clusters are proposed and although the document marked with interrogation seems to enter into user's interest it will not be recommended at all.

Moreover, especially in content clustering, low quality documents tend to form weak clusters. Let's imagine a typical university department's web site in which professors possess their personal web pages. Every personal page is linked with the folder pages that reflect the structure of the directories that correspond to the professor's lectures. An example of a "folder page" that belongs to the professor Smith is presented in the following picture:



Index of /~smith

Name	Last modified	Size	Description
Parent Directory		-	
UML/	21-Nov-2004 12:47	-	
brin98anatomy.html.htm	14-Dec-2003 15:18	23K	

Figure 2: an example of a folder page

All folder pages are quite similar one another (due to the common words such as: "last", "modified", "Size", "Description") and it is quite probable that they will form a cluster or even various clusters (one cluster for professor, for instance).

Last but no least, it is important to mention hybrid approaches that combines the methods presented above [Balabanovic, 97], [Schafer, 01], [Cunningham, 01] .Although integration of various recommendation techniques can improve the final result, to the best author's believes there is no method that deals with the quality of recommended content and at the same time permits almost all relevant content to be presented to the user.

2.2 Document Ranking

Document ranking concept is widely used in Information Retrieval. The most popular ranking algorithms are: HITS [Kleinberg, 98], PageRank [Brin, 98], DirectHit [DirectHit, 02]. The first two assume that the more hyperlinks from "good documents" point to the document, the better the document is. This assumption is generally true in case of global search engines, but it can be false in case of local search engines and recommendation systems, because local hyperlinks reflect rather site navigation model than documents' relevance. For example, many web masters include links to documents that contain site's privacy policy or the site map. As a consequence, those documents would possess high position in the ranking despite their small relevance and interests.

There are some works that try to enhance the PageRank in the local search engines (e. g. [Xue, 03] in which the PageRank algorithm is enriched with information about users' navigation). Similarly, the HITS algorithm is also adapted to the local necessities [Miller, 01].

On the other hand, the DirectHit ranking is based on two usage factors: *click popularity* and *stickiness.* The former measures how many times a particular document has been clicked from the search engine result page. The latter determines the amount of time that users spend on reading a particular document (when the user clicks another document from the result page, it is assumed that the previous page has been abandoned). *Click popularity* and *stickiness* may be laden with errors, because not all users enter the page by means of search engines.

The improvement of the ranking order by means of relevance: [Rocchio, 71], [Salton, 90], [Alla, 01] and pseudo-relevance feedback [Sing, 99], [Xu, 00] have sparked big interest of the IR community.

3 Method Overview

RankFeed is a hybrid method of recommendation that operates on a single web site with HTML documents. Similarly to other recommendation systems all its activities can be divided into off-line and on-line part. In the off-line part the following complex and time consuming tasks are performed:

- Initial RankFeed (IRF) ranking calculation – it is a document ranking that is assigned to every user that enters to the site. During the site navigation this ranking changes due to the implicit feedback provided by the user (visited and ignored document). The initial ranking reflects the quality of the site documents. The detailed algorithm of ranking calculation is presented in the next section
- Usage pattern discovering – all historical users' visits are transformed into the vector space model and grouped in order to discover the typical navigational patterns
- Calculation of the most similar pages for every document belonging to the web site

The on-line part has an iterative character and consists of:

- Aggregation of the current page to the current session vector
- Adding recommended but ignored pages to the ignored document vector
- The closest usage pattern calculation – the closest usage pattern is the usage cluster whose centroid is the most similar to the current session vector
- Calculation of a new ranking as a linear combination of the closest usage centroid, the most similar documents' vector and the ignored document vector
- Presentation of the top – n documents to the user.

4 Initial Ranking

In the majority of ranking algorithms, the position of a document not only depends on its relevance to the query, but also it is determined by its qualities. All documents ordered according to their “contextless goodness” can be considered as the initial ranking. The method of quality calculation varies with the rankings. Since the RankFeed method is a local recommendation (all recommended content comes from the same web site), the initial ranking is also calculated using local features. All of them can be classified into two groups:

- *Usage factors* – session opening rate, traffic popularity, stickiness
- *Quality factors* – availability, dead link rate, freshness rate

The goal of all of these features is to measure the importance of the document in the web site. Every feature will be shortly discussed below. The author strongly believes that the local features that very often are only available in the particular web site (for example traffic popularity) better characterize valuable documents hidden among irrelevant and low quality content. The initial ranking will be also denominated the *IRF* ranking.

4.1 Basic Definitions

This section contains basic definitions that will be used in this paper.

Definition 4.1.1

Timestamp t is the number of seconds that have passed since the midnight of 01/01/1970. Additionally, the following functions are defined:

- $h(t)$ returns the number of hours that has passed since the midnight of 01/01/1970: $h(t)=[t/3600]$
- $day(t)$ returns the number of days that have passed since 01/01/1970: $day(t)=[t/86400]$
- t_c will be the current timestamp (the timestamp that corresponds to the current moment)

where $[x]$ denominates the integer part of the x.

Definition 4.1.2

Let the tuple $d=(url_d,t_0,t_u,H)$ be a document (web page); url_d corresponds to the unique document’s URL address that identifies the tuple unambiguously, t_0 is the timestamp in which the given document has appeared in the web site; t_u is the total number of seconds in which the document has been unavailable. Finally, $H=\{url_1$, *url2, url3, urlm*$\}$ is the set of URLs addresses of resources to which the document d points. It is important to emphasize that elements of the set H do not have to come from the same web site nor be a document (cascade style sheet or multimedia files are also regarded as internet resources).

Definition 4.1.3

Let $D=\{d_1, d_2, d_n\}$ be the set of all documents (web pages) available in the web site.

Definition 4.1.4

Let the tuple v=(url_v, t_v) be a single document visit (denominated also a document hit) where t_v is a visit's timestamp:

$$\underset{d\in D}{\exists}(d=(url_d, t_0, t_u, H)\wedge url_d = url_v).$$

Definition 4.1.5

Let $s=(v_1, v_{2,\ldots,} v_p)$ be a *user session* - a sequence of single document hits performed by the user during one site visit. A user session that fulfil the condition t_c-t_p<600 is called *current user session.* The rest of sessions are denominated *historical user sessions*.

Definition 4.1.6

Let $S=\{s_1, s_2,\ldots,s_m\}$ be the set of all user sessions.

Definition 4.1.7

Let the S_d ($S_d \subseteq S$) be the usage of the document *d*, $d\in D$ that in other words can be defined as the set of *user sessions* in which the document *d* has been visited:

$$S_d = \{s : s = (v_1, v_2,\ldots, v_p)\wedge \underset{0<i\le p}{\exists}(v_i = (url_{vi}, t_{vi})\wedge(url_{vi} = url_d))\}\,.$$

Definition 4.1.8

Let the $S^m_{d,}$ ($S^m_d \subseteq S_d$) of document *d*, $d\in D$ be the set of *user sessions* from the last month in which the document *d* has been visited:

$$S^m_d = \{s : s = (v_1,\ \ldots,\ v_p)\wedge$$
$$\wedge \underset{0<i\le p}{\exists}(v_i = (url_{vi}, t_{vi})\wedge(url_{vi} = url_d))\wedge month(t_c) - month(t_{vi})\le 1\}\,.$$

Definition 4.1.9

Let the S_{fd} ($S_{fd} \subseteq S_d$) be the set of user sessions in which the document *d* has been visited as the first from the session:

$$S_{fd} = \{s : s = (v_1, v_2,\ldots, v_p)\wedge v_1 = (url_{v1}, t_{v1})\wedge(url_{v1} = url_d)\}\,.$$

Definition 4.1.10

Let *W* be the set of all words (terms) extracted from the textual content of the documents belonging to the set *D.*

Definition 4.1.12

Let $\boldsymbol{dw_i}$ be a content vector of the i^{th} document from the set D whose coordinates corresponds to the terms coming from the set W

$$\boldsymbol{dw}=(dw_1, dw_2, dw_3,..dw_{card(W)}\}$$

Definition 4.1.13

Let DW be the set of all content vectors:

$$W=\{\boldsymbol{dw}_1, \boldsymbol{dw}_2, \boldsymbol{dw}_3,..\boldsymbol{dw}_{card(D)}\}$$

4.2 Session Opening Rate

Statistics that visualize from which document the users begin their navigation within the site is widely used in applications that analyse web traffic (for example WebTrend, ROSA). These documents can be regarded as gateways that link the site with the external world. Analysing three principal ways, in which the user can enter the given site, it is possible to understand why session opening rate ought to influence positively on the document's ranking position:

- The user has typed the URL address in the browser or he has chosen it from the favourite's list. (the user considers the document as important)
- The user has followed a link from an external web site (at least the author of the external page considers the document as important)
- The user has found the document in a global search engine (the user must consider the document as relevant to the query)

The principal goal of session opening rate is to promote the documents by means of which users enter the site.

Opening rate of the document d can be defined as follows:

$$or(d) = \left\{ \begin{array}{lll} \dfrac{card(S_{fd})}{card(S_d)} & for & card(S_d) > 0 \\ & & \\ 0 & for & card(S_d) = 0 \end{array} \right\}$$

4.3 Traffic Popularity

Traffic popularity is used to increase the ranking position of the documents that are visited frequently. Unlike global search engines (that are able to gather only click popularity – the number of hits that a given page receives from a search result page), a local system can also benefit from traffic popularity that can be regarded as the total number of hits for each document. The main problem concerning traffic popularity is cyclic reinforcing of frequently visited documents (since they appear at the top of the ranking, the users visit them, and as a consequence their position is getting higher). One of the methods that weaken a bit this negative trend is the usage of temporal traffic popularity that reflects the mean number of document visits per a time unit (for example per day). The temporal traffic popularity of the document *d* can be obtained from the following formula:

$$tp(d) = \begin{cases} \dfrac{card(S_d)}{day(t_c) - day(t_0)} & for \quad day(t_c) - day(t_0) > 0 \\ 0 & for \quad day(t_c) - day(t_0) = 0 \end{cases}$$

It is important to underline that temporal traffic popularity reflects the global documents' popularity and does not determine if the document's visits are higher now or two years ago. For example, let's assume that a document *A* and a document *B* have the mean traffic popularity equal to 200 hits per day. The document *A* has been in the service for two months and during first five days of its existence received 2400 hits per day, but then it deprecated and nobody has visited it. Whereas the document *B* has been presented in the site for two weeks and it obtained 200 hits every day.

The last month traffic popularity has been introduced in order to enhance the position of documents that have obtained more visits during the last month:

$$mtp(d) = \begin{cases} \dfrac{card(S_d^m)}{day(t_c) - day(t_m)} & for \quad day(t_c) - day(t_m) > 0 \\ 0 & for \quad day(t_c) - day(t_m) = 0 \end{cases}$$

Either temporal traffic popularity or month traffic popularity must be normalized to 1 by dividing all non-zero values by the maximum value of temporal traffic popularity or month traffic popularity, respectively.

4.4 Stickiness

Stickiness is the amount of time that users spend on reading or viewing a particular document. The stickiness factor has been introduced in order to enhance the importance of the documents that pay the visitors' attention. Having access to the users' activity, it is possible to calculate it better than in case of global search engines. The *stickiness* of the document *d*, $d \in D$ in the session s=(v_1, $v_{2,\ldots,}$ v_p), $s \in S_d$ is equal to:

$$stick_s(d) = min(sec(t_{vi+1}) - sec(t_{vi}), timeout)$$

where $vi=(url_{vi},t_{vi}) \wedge url_{vi}= url_d \wedge i<p$. *Timeout* has been introduced because of experiments which have revealed that there are sessions in which the stickiness is very high. Single sessions with abnormal high stickiness influence negatively on the final results.

Due to the fact that *stickiness* of the document *d* is not defined for those sessions in which the document *d* is the last one, the subset of the set S_d has been defined:

$$S_d' = \{s : s = (v_1, v_2, \ldots, v_p) \wedge \underset{0<i<p}{\exists} (v_i = (url_{vi}, t_{vi}) \wedge (url_{vi} = url_d))\}$$

The mean stickiness of the document $d=(url_d,\ t_{0d},\ t_{ud},\ m_d, H_d)$, $d \in D$ is equal to:

$$stick(d) = \frac{\sum_{s \in U_d^{'}} stick_s(d)}{card(S_d^{'})}$$

The stickiness function should also be normalized to 1, dividing it by the maximum value of stickiness.

This method of stickiness calculation (although more precise than in DirectHit) is also laden with errors. First of all, it is not possible to get the stickiness of the last document in the session. Moreover, we cannot be sure that the user was really reading the document (and not taking a coffee-break). For all these reasons stickiness should not influence so firmly on a ranking position like, e. g., traffic popularity.

4.5 Availability

Even the most relevant document is useless if it is unavailable. The *availability score* of the document *d* is the percent part of the document life in which document has been available:

$$av(d) = 1 - \frac{t_u}{t_c - t_0}$$

The purpose of this feature is to weaken the ranking position of the documents that are unavailable very often or suffer from internal server errors.

4.6 Dead Link Rate

One of the most important tasks related to web site administration is the validation of hyperlinks correctness. Hyperlinks to the pages that do not exist or are temporally unavailable (denominated also *dead links*) may cause users' confusion and question site's credibility. Analysing hyperlinks' target availability, it is important to take into account all types of resources (not only HTML documents, but also images, cascade style sheet files etc.) Dead link rate of the document *d* can be obtained from:

$$dl(d) = \left\{ \begin{array}{ll} \frac{card(H_u)}{card(H)} & for \quad card(H) > 0 \\ & \\ 0 & for \quad card(H\ ) = 0 \end{array} \right\}$$

$H_u \subseteq H_d$ is a set of those hyperlinks that have their source in the document *d* and are unavailable.

The principal goal of the dead link rate is to decrease the importance of the documents that possess many hyperlinks to unavailable resources.

4.7 Freshness Rate

New documents do not possess high position in search engines' rankings due to two main reasons. Firstly, there are few pages that point to them (PageRank, HITS). Secondly, they have been visited less times that old ones (DirectHit). Underestimation of new documents' relevance is not a grave problem in case of global search engines because many documents appear and disappear every day. In contrast, promotion of new documents is a very important task in local systems. For all these reasons the *freshness rate* is used in order to promote new documents.

The freshness rate of document d is calculated from the formula:

$$fr(d) = \begin{cases} q^{day(t_c) - day(t_0)} & for \quad day(t_c) - day(t_0) > \varepsilon \\ 0 & for \quad day(t_c) - day(t_0) \leq \varepsilon \end{cases}$$

where $q \in (0, 1)$. The goal of introduction of the constant $\varepsilon \approx 0.01$ is to omit the time-consuming calculation when the value of the *freshness rate* is close to 0.

4.8 Initial Ranking Calculation

Due to practical performance issues, the initial ranking score is calculated as a linear combination of all factors presented above:

$$IRF(d) = \lambda \cdot or(d) + \mu \cdot max(\ tp(d) + mtp(d), fr(d)) + \chi \cdot stick(d) + \phi \cdot av(d) + \psi(1 - dl(d))$$

where the *max* function returns greater parameter. The constants: $\lambda, \mu, \chi, \phi, \psi$ regulate the influence of the related factors. In the practical usage, their values should reflect the importance of each factor for the system administrator. For example if the web site does not possess enough usage data λ, μ, χ ought to be close to 0. Generally, χ value should be much smaller than the rest of parameters due to the reasons described in the section 4.4. For the experiments presented below the following values have been assumed $\lambda=0.75$, $\mu=0.5$, $\chi=0.25$, $\phi=0.5$, $\psi=0.5$.

5 User's Feedback and Ranking Evolution

The initial ranking in the *RankFeed* method is treated as a starting point of recommendation. Every time the user visits a new page, initial ranking is modified using the indirect feedback that is transparent to the user. There are three main factors of the user feedback: the most similar documents, the recommended documents that the user has ignored and the visited documents that permit the user's profile to be discovered. All of them are described in this chapter.

5.1 Historical Sessions' Clustering

Classification of the user to one of predefined usage patterns is a very important task. Similarly to the initial ranking calculation, usage patterns are constructed off-line basing on clustering of historical user sessions $s=(v_1, v_{2,...,} v_p)$.

Let's assume that sequences of historical usage sessions are available (a detail method of calculation of historical usage sessions can be found at [Mobasher, 01]). Since there is no point in grouping empty or almost empty sessions, the first operation that must be performed is the restriction of the set S to the set S' that contains only these sessions in which at least n^s documents from the set D were visited:

$$S' = \{s \in S : s = (v_1, v_2, ..., v_p) \wedge p \geq n^s\}$$

In the implementation n^s=*5* has been assumed. Too small value of n^s will worsen clustering quality, whereas too high value will eliminate the majority of the sessions and therefore the results will not be representative.

Each historical session from the set *S'* must be transformed in a document binary vector in which each coordinate corresponds to one particular document visited in this session. This type of transformation can be defined as function *tsv*, that fulfil the following condition:

$$\underset{s\in S'}{\forall}(tsv(s) = \boldsymbol{s}^v \wedge \boldsymbol{s}^v = (s_1^v, s_2^v, ..., s_{card(D)}^v) \wedge \underset{url\in Us}{\forall}(s_{cord(url)}^v = 1) \wedge \underset{url\notin Us}{\forall}(s_{cord(url)}^v = 0))$$

where *cord(url)* is a function that for each url_d returns an integer number of corresponding coordinate and U_s is a set of URL addresses visited during the session *s*.

$$U_s = \{url : \underset{v\in s'}{\exists}(v = (url_v, t_v) \wedge url_v = url)\}$$

Once the binary vectors are created, it is possible to cluster them. For the purposes of our experiments, the hierarchical agglomerative clustering method (HACM) has been used. The distance between vectors has been calculated using the Jaccard formula. As a result, the set of clusters is created. Every created cluster reflects one usage pattern and it is represented by the centroid - the mean vector c_i of all vectors that belong to the cluster. In other words, the centroids' set $C=\{\boldsymbol{c_1},\boldsymbol{c_2},\boldsymbol{c_{3...}},\boldsymbol{c_{nc}}\}$ is created as the result of clustering. The centroid of i^{th} cluster can be calculated from the following formula:

$$\mathbf{c}^i = \frac{1}{m_i}\sum_{j=1}^{m_i}\boldsymbol{s}_{\boldsymbol{ij}}^{\boldsymbol{v}}$$

Centroid coordinates can be intuitively described as the measure in which a given document belongs to the centroid's cluster. For example, in a centroid $\boldsymbol{c^p}$=(0.1,0.0,0.8,0.9,0.2,0.0) the forth and the fifth coordinates correspond to documents that are "good cluster's members" because they have been visited in respectively 80% and 90 % of all historical sessions classified to this cluster.

5.2 Similar documents

Similarity calculation in the vector space model is a well-known task. Once the set of the terms that occur in each document is created the document vectors are calculated. Every coordinate corresponds to the particular term. The wide-known Salton's formula *term frequency* (*tf*) - *inverse document frequency* (*idf*) has been used. Terms, that occur frequently in one document (*tf*), but rarely in the rest of the set (*idf*), seem to be more relevant to the subject of the document. Therefore, *tf-idf* measure for the i^{th} coordinates is based on the weight w_{ij} of the term t_i in the document d_j, as follows:

$$dw_{ij} = tf_{ij} \times idf_i = tf_{ij} \times \log\left(\frac{card(D)}{n^{t_i}}\right)$$

where: tf_{ij} - term frequency (the number of times term t_i appears in d_j), *card*(*D*) - the number of all documents, n^{t_i} - the number of documents in which term t_i occurs. Terms that appear in many documents are not useful in distinguishing between relevant and irrelevant documents. The inverted document frequency idf_i reduces the influence of these terms.

The similarity between all pairs of **dw** vectors is calculated using the Jaccard formula and as a consequence for each document $\mathbf{sim}^d$ vector is created. The coordinate corresponding to the j^{th} document is obtained from the following equation:

$$sim_j^d = \begin{cases} 0 & for \quad jacc(\mathrm{d}, \mathrm{d}_j) < \tau \\ jacc(\mathrm{d}, \mathrm{d}_j) & for \quad jacc(\mathrm{d}, \mathrm{d}_j) \geq \tau \end{cases}$$

The main goal of τ constant's introduction is to eliminate the documents that are not very similar ($\tau \approx 0.2$)

The purpose of the $\mathbf{sim}^d$ vector of the document *d* is to represent all documents similar to the document *d* in the form of vector.

5.3 Ranking Evolution

When the user enters the site (viewing the first document), the following data structures are assigned to his or her session: the current session vector, the closest usage pattern, the ignored document vector and the personalized document ranking. All of them are explained in the following subsections.

5.3.1 Current Session Vector

The current session vector $\boldsymbol{s}^c=(s^c_1, s^c_1, s^c_{1,\ldots,} s^c_{card(D)})$ reflects the set of pages that the user has seen in the current site's visit and it can be regarded as a profile of the on-line user. At the beginning, every coordinate is equal to 0. After a new document *d* is visited, the vector coordinates are modified according to the following formula:

$$s_i^c = \begin{cases} \tau s_i^c \quad for \quad i \neq ord(url_d) \\ 1 \qquad for \quad i = ord(url_d) \\ 0 \qquad for \quad s_i^c < 0.01 \end{cases}$$

where $\tau<1$, ($\tau=0.8$ in the performed experiments). The higher the constant τ is, the longer the method "remembers" that the particular document has been visited.

The formula presented above enforces the importance of the documents that have been visited recently (every time a new document is visited, the weights of all visited documents decrease geometrically)

5.3.2 The Closest Usage Pattern

The closest usage pattern can be regarded as a typical user session that is the most similar to the current user's profile. The documents whose coordinates possess high values are frequently visited in the sessions represented by this usage pattern. Every time the current session vector changes, it is assigned to the closest usage pattern represented by the centroid $\mathbf{c_c}$ that fulfils the following condition:

$$\underset{c \in C}{\forall}\, dist(\boldsymbol{c}, \boldsymbol{s}^c) > dist(\boldsymbol{c}_c, \boldsymbol{s}^c) \vee \boldsymbol{c} = \boldsymbol{c}_c$$

where the *dist* is the distance between two vectors. Similarly to the clustering, the Jaccard distance formula has been used.

The intuition of the closest centroid is to classify on-line user's behaviour to one of predefined patterns in order to recommend documents typically visited in the pattern.

5.3.3 Ignored Document Vector

The ignored document vector $\mathbf{g}^{\mathbf{c}}=(g^{c}_{1}, g^{c}_{2}, g^{c}_{3,\ldots,} g^{c}_{card(D)})$ reflects the set of pages that have been recommended to the user and have been ignored. At the beginning every coordinate is equal to 0. After a new list of n^r recommended pages is presented to the user, their URL addresses are saved in a temporary set R={url_1,url_2,…, url_{nr}}. Every time a new document d is visited, the vector coordinates are modified according to the following formula:

$$g_i = \left\{ \begin{array}{lll} \tau g^c & for & \underset{url \in R}{\forall} \; i \neq ord(url) \\ 1 & for & \underset{url \in R}{\exists} \; i = ord(url) \wedge url \neq url_d \\ 0 & for & (i = ord(url_d) \wedge url_d \in R) \vee g_i < 0.01 \end{array} \right\}$$

where $\tau<1$, in the performed experiments τ has been equal to 0.8. The higher the constant τ is, the longer the method "remembers" that the particular document has been ignored. For example if the τ=0.8, the fact that the document has been ignored, method will stop influencing negatively on the document after 20 page requests.

5.3.4 Personalized Document Ranking and Hyperlink Recommendation

Personalized document ranking is ordered taking into account the user feedback. When the user enters the site the initial document ranking *IRF* is assigned to his or her session: $\mathbf{p}^0=(p^0_1, p^0_2, p^0_3,\ldots,p^0_{card(D)})$ where:

$$\underset{d \in D}{\forall} \; p^0_{rank(url_d)} = IRF(d)$$

After the calculation of changes in the current session vector, the ignored document vector and the assignment of usage class, the document ranking is updated according the following formula:

$$\mathrm{p}^{i+1} = (\alpha \mathrm{p}^i + \beta \mathrm{c}_\mathrm{c} + \delta \mathrm{sim}^{d_c} - \gamma \mathrm{g}^c)(1 - \mathrm{s}^c)$$

where d_c corresponds to the current document. The α, β, γ, δ constants regulate the influence of the old ranking, the closest usage pattern, the most similar documents and the negative feedback, respectively.

If the α parameter's value is too big, high quality documents will be recommended but their relevance can decrease drastically. On the other hand, the relatively high value of the β and δ will bring the overcomes related to data mining and content approach respectively (see the section 2.1). Moreover, it is important to emphasize that $\mathbf{p}^i$ and $\mathbf{c}_\mathbf{c}$ elements promote documents that have been relevant to the documents visited before, if α, β parameters are too big comparing to δ, the documents seen before will possess to much impact on the recommended content. For the experimental purpose, the following values have been assigned: α*=0.25*, β*=0.5*, γ*=0.5*, δ *=0.5*.

As it can be seen, the new ranking favours the documents that have high values in the usage pattern assigned to the user. It means that the content that was seen in the majority of the sessions similar to the active one may result relevant to the current user (they are treated as a positive feedback). Additionally, the documents that are similar to the current one are also regarded as a positive feedback.

At the same time, the ranking weakens the influence of the documents that have been recommended but the user has not visited them (they are regarded as a negative feedback). In this case, it suggests that from his point of view they are irrelevant. The expression (1-$\mathbf{s}^c$) drastically decreases the position of the documents that have been seen recently (there is no point in recommending just visited content).

Once the new personalized ranking is calculated, the first n^{th} hyperlinks with the document title and the short summary can be presented to the user. It’s not worth presenting more than two or three documents at the same time, because it is quite probable that the last documents would be ignored without even being read. As a consequence, some relevant documents might enter into the negative feedback.

6 Evaluation and Practical Issues

The evaluation of the method has been focused on two main aspects described in the following subsections: the IRF ranking’s behaviour in a local site contrasted with PageRank and evaluation of the RankFeed method comparing with other standard approaches to recommendation. Data from the departmental web site has been used for the purpose of experiments: 56479 user usage sessions registered between October 2003 and October 2004 have been analysed. Those sessions correspond to 4784 documents that have been indexed. Since the departmental web site was migrated from the windows to the linux platform at the beginning of 2004 and some documents were not copied, a copy of the web site has been created. This copy includes all documents that have been visited in the period mentioned above (even if they are not available any longer).

6.1 IRF versus PageRank in a Local Environment

The PageRank in a local site has been calculated for the site of Information Systems Department of the Wroclaw University of Technology.

PageRank	**Hits**	**Document’s URL address**
26.20	0	/tomcat/catalina/docs/api/overviewsummary.html
21.07	0	/tomcat-docs/catalina/docs/api/index.html
20.79	0	/tomcat-docs/catalina/docs/api/allclasses-noframe.html
20.75	0	/tomcat-docs/catalina/docs/api/index-all.html
20.24	0	/tomcat-docs/catalina/docs/api/deprecated-list.html

Table 1: Documents with the highest PageRank values

The department web site uses the Tomcat server that provides the on-line documentation. “The ranking winners” are, in fact, the main pages of tomcat documentation, but nobody wants to view them (see the second column). Their high position is related to the fact that the tomcat documentation contains more than 800 documents that create a complete structure independent from the real department site. Since according to PageRank the more hyperlinks point to the document, the better document is, PageRank promotes the documents to which there are many hyperlinks. Considering the number of hits that the tomcat documentation receives, it is clear that this ranking is far from users’ expectations.

The table 2 contains 11 documents with the highest PageRank and their corresponding visits ranking (the tomcat documentation has been omitted)

Order Num.	**PageRank**	**Number of visits**	**Visit’s ranking**	**Document’s URL address**
1	4.26	109	44	/rosasite/welcome.jsp (the main page of ROSA’s site)
2	3.39	47	99	/zsi/eng/index.html (the main page of English section)
3	3.27	130	31	/zsi/index.html (the main page of department description)
4	3.09	799	3	/zsi/pracownicy/pracownicy.htm (the professors’ main page)
5	3.06	148	23	/zsi/dzialalnosc/dzialalnosc.htm (the program of department research)
6	3.02	21	171	/zsi/info.htm (information about the department)
7	3.02	152	21	/zsi/aktualnosci/aktualnosci.htm (the last news from department)
8	2.89	293	8	/zsi/dydaktyka/dydaktyka.htm (didactics issues)
9	2.86	102	46	/zsi/zaklad/zaklad.htm (description of the department)
10	2.61	27	148	/zsi/kontakt.htm (contact information)
11	2.57	45	104	/zsi/mapa.htm (the sitemap page)

Table 2: Documents with the highest PageRank (*tomcat documentation was omitted*)

In this case also documents that are referenced from many pages are at the top. There are, at least, three documents that have obtained the high ranking position due to the characteristics of the navigation structure (site map - position 11, contact information - 10, general information 6). Additionally, the visits’ number of all documents from the table is relatively small (the most visited document had 1403 visits).

Finally, the document ranking using IRF has been calculated. All top-ten documents seem to possess valuable information:

Document's URL address with short description	**RankFeed**
/ (the home page)	1
/pracownicy.htm (the professors' main page)	0.83
/zsi/dydaktyka/usm.htm (the main page of postgraduate studies)	0.66
/missi2000/referat26.htm a conference paper about interaction between the human	0.65
/zsi/dydaktyka/usm_program_szczegolowy.htm (the detailed program of postgraduate studies)	0.40
/neuman/java/skladnia.htm (description of java)	0.39
/neuman/kierunki/bezpieczenstwo.htm (security issues)	0.32
/missi2000/referat15.htm (a paper from the MISSI conference about cordless transmission using IRDA)	0.24
/zsi/dydaktyka/pmag.htm (the list of master degree projects)	0.25
/stopka/laboratorium/laboratorium.html (laboratory issues)	0.23

Table 3: Documents with the highest RankFeed

Comparing the IRF result with the PageRank it is important to emphasize that all pages from the first ten positions of RankFeed are high-quality and wide recognized documents that could be recommended to all visitors independently from their preferences. The RankFeed ranking seems to be more profitable for the user because it promotes good pages that very often are deeply hidden in the site structure, while PageRank gives more importance to the pages that are pointed from many pages and as a consequence are easily accessible.

6.2 RankFeed and Other Approaches to Recommendation

The evaluation of recommending algorithm is not a trivial task, since almost all criteria that measure the quality of recommendation are very subjective. The evaluation strategy chosen in this paper tries to compare the characteristics of the presented RankFeed method with other typical approaches used in recommendation:

- Content approach – in this case a set of documents, that are the most similar to the current one, are presented to the user
- Usage approach – in this case documents that are the most often visited with the current one are recommended

- Web mining approach – this method is based on the integration of content mining and usage mining proposed in [Mobasher, 2000] and extended in [Kazienko, 03a]

All types of recommendation mentioned above have been implemented in the ROSA project described in [Kazienko, 03b],[ROSA, 2004].

For the purpose of the experiment, 10 imaginary web sessions have been chosen (see the table 4). Those sessions reflect the kind of information that is searched in the departmental web site. Two documents have been recommended for each web page visited during the experiment. Accuracy and coverage have been used as two quality measures of recommendation. The former can be obtained from the formula:

$$accuracy(s)=\frac{rel(R_S)}{card(R_S)}$$

where R_s is the set of pages recommended during the sessions *s*, $rel(R_s)$ is the number of relevant documents recommended to the user during the session *s* and $card(R_s)$ is the number of all documents recommended within the session *s*.

Coverage of recommendation in the session *s* can be obtained from:

$$coverage(s) = \frac{distinct\_rel(R_S)}{card(R_S)}$$

where $distinct\_rel(R_s)$ denominates the number of distinct relevant documents recommended to the user during the session *s* $rel(r) \geq distinct\_rel(R_s)$. If the recommended document has been already visited in the session it does not increase the *distinct_rel* value. The following table presents the experiment's results:

Session name	**visited**	**RankFeed**		**Web Min.**		**Similar**		**Covisited**	
	pages	**acc.**	**cov.**	**acc.**	**cov.**	**acc.**	**cov.**	**acc.**	**cov.**
HTML course	5	0.7	0.5	0.1	0	1	0.5	0	0
Didactic issues	4	0.7	0.69	0	0	1	0.5	0.88	0.38
ROSA tour	6	0.9	0.5	0.67	0.17	0.67	0.58	0.83	0.08
Information about Wroclaw city	6	1	0.92	0	0	0.17	0.08	0.83	0.41
Personal web pages of two professors: N.T. Nguyen and P. Kazienko	5	0.8	0.7	0.7	0.3	0.75	0.55	0.3	0.1
Introduction to java	3	0.5	0.5	0	0	1	0.5	1	0.33
MISSI conference	2	1	1	1	0.	0.25	0.25	1	0.75
MMISTech2004 conference	7	0.9	0.86	1	0.43	0.79	0.64	0.79	0.43
Student club	4	0.5	0.38	0.5	0.13	1	0.63	0.88	0.25
Department research	4	1	0.88	1	0.25	1	0.5	0.5	0.13
Mean values	**4.6**	**0.8**	**0.69**	**0.5**	**0.13**	**0.77**	**0.47**	**0.7**	**0.29**

Table 4: Evaluation of distinct type of recommendation

The comparison of the three traditional approaches with the RankFeed can be concluded as follows:

- Introduction of the feedback has meaningfully increased the coverage of recommendation. Other methods are quite static. They very often recommend the same content
- In spite of high diversity of recommended content RankFeed has obtained the highest accuracy. This result, among others, is related to good behaviour in case of documents that do not have any similar or together visited pages. Every time a "rare" document is requested that does not have any similar nor together visited page, the RankFeed takes into account the documents previously seen
- In the content approach, "the similar pages" very often consist of documents that possess low quality, but they are considered as similar, due to some keywords. In case of RankFeed, the quality ranking eliminates almost all low quality documents
- In the usage approach, "new documents" do not have many possibilities to be recommended due to the small number of hits. Moreover, some documents are visited together due to the site navigation structure although they concern completely different issues
- In the web mining approach, only the documents whose coordinates possess high values in centroids are presented to the user. It is especially inconvenient in case of content mining, because some of the created clusters are concentrated within low quality documents (for example directory lists) Additionally, if the user's navigation does not correspond to any cluster the accuracy is low

7 Conclusions and Future Works

A new recommendation method RankFeed has been presented in this paper. It is based on document quality and user's feedback. According to the experiments, the IRF ranking behaves better than PageRank in a single web site, because the majority of hyperlinks possesses navigational purposes if and only if the local documents are considered.

Taking into account the RankFeed recommendation characteristics, it seems to behave better than traditional approaches. The most important achievement has been related to the increasing of recommendation coverage, maintaining at the same time the high accuracy. Nevertheless it will be indispensable to perform some additional experiments, for example, the number of clicks that receives the recommended content in function of the used method and checking the method's behaviour in more web sites. Another interesting issue is to investigate if the method recommends the documents that are separated from the current one by many other pages (click saving).

The presented method can be regarded as the integration of three separate rankings (the quality ranking, the similarity ranking and the usage profile ranking). The paper presents the simplest and the cheapest way of integration (linear combination), but the usage of consensus model (see [Nguen, 02] for more details) seems to be a challenge worth investigating. In this case, every ranking should be considered as a multi-attribute knowledge of an agent responsible for the ranking creation, and the final ranking may be worked out as a consensus of three distributed agents' opinions

Concluding, it is important to emphasize that although the method depends on the text analysis it can be applied not only to documents but also to other items in which text information is limited to the name (pictures, songs, products). The only modification that has to be performed is setting the ψ and the δ constants to 0.

References

[Alla, 01] J. Allan, A. Leuski, R. Swan, D. Byrd, Evaluating combinations of ranked listsand visualizations of inter-document similarity, Information Processing and Management, Vol. 37, No. 3, 2001, 435–458

[Balabanovic, 97] M. Balabanovic, Y Shoham, Combining Content-Based and Collaborative Recommendation, Communications of the ACM, 1997, 66-72

[Booley, 99] D. Boley, M. Gini, R. Gross, E.H. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, J. Moorey, Document Categorization and Query Generation on the World Wide Web Using WebACE, Artificial Intelligence Review, 1999 ,Vol. 13 No. 5-6, 365-391

[Buono, 02] P. Buono, M.F. Costabile, S. Guida, A. Piccinno, Integrating User Data and Collaborative Filtering in a Web Recommendation System, OHS-7, SC-3, and AH-3, LNCS 2266, Springer Verlag, 2002, 315-321

[Brin, 98] S. Brin, L. Page, The anatomy of a large-scale hypertextual web search engine, In Proc. 7th International World Wide Web Conference, Brisbane, Australia 1998, 107-117

[Cunningham, 01] P. Cunningham, R. Bergmann, , S. Schmitt, , R. Traphoner, S. Breen, B. Smyth, WebSell: Intelligent Sales Assistants for the World Wide Web, In E-2001, 2001, 28-32

[DirectHit, 02] DirectHit search engine, 2002, http://www.directhit.com

[Kazienko, 03a] P. Kazienko, M. Kiewra, "Link Recommendation Method Based on Web Content and Usage Mining", Proceedings of the International IIS: IIPWM´03 Conference, Advances in Soft Computing, Springer Verlag, 2003a, 529-534, http://www.zsi.pwr.wroc.pl/~kazienko/pub/IIS03/pkmk.pdf

[Kazienko, 03b] P. Kazienko, M. Kiewra : ROSA - Multi-agent System for Web Services Personalization. In Proc. First Atlantic Web Intelligence Conference Proceedings, LNAI 2663, Springer Verlag, 2003b, 297-306

[Kleinberg, 98] J. Kleinberg, Authoritative Sources in a Hyperlinked Environment, In Proc. ACM-SIAM Symposium on Discrete Algorithms, 1998, 604-632

[Krulwich, 97] B. Krulwich, Lifestyle Finder: Intelligent User Profiling Using Large-Scale

Demographic Data, AI Magazine, Vol. 18, No. 2,1997 ,37-45.

[Lieberman, 95] H. Lieberman, Letizia: An Agent that Assists Web Browsing. In Proc. IJCAI'95, 1995, 924–929

[Madria, 99] S.K. Madria, S.S. Bhowmick, W.-K. Ng, E.P. Lim, Research Is-sues in Web Data Mining" DaWaK '99, Springer Verlag, LNCS 1676, 1999, 303-312.

[Miller, 01] J. Miller,G. Rae, F. Schaefer, Modifications of Kleinberg's HITS algorithms Using

Matrix Exponentiation and Web Log Records, in: Proc. the 24th Annual International ACM SIGIR Conference on Research and Development in IR, 2001, 63-72

[Mobasher, 00] B. Mobasher, H. Dai, T. Luo , Y. Sun, J. Zhu, Integrating Web Usage and Content Mining for More Effective Personalization. LNCS 1875 Springer Verlag (2000), 2000, 156-176

[Mobasher, 01] B. Mobasher, B. Berendt , M. Spiliopoulou KDD for Personalization. PKDD 2001 Tutorial

[Montaner, 03] M. Montaner, B. Lopez, J. L. De La Rosa, A Taxonomy of Recommender Agents on the Internet, Artificial Intelligence Review, Volume 19 , Issue 4, June 2003, 285-330

[Mooney, 00] R. J. Mooney, L. Roy, Content-based book recommending using learning for text categorization, In Proc. 5th ACM Conf. on Digital Libraries, 2000, 195-204.

[Nguyen, 02] N.T. Nguyen, Consensus System for Solving Conflicts in Distributed Systems. Information Sciences – An International Journal Vol. 147, 2002, 91-122

[Pazzani, 99] M. Pazzani, A Framework for Collaborative, Content-Based and Demo-graphic

Filtering, Artificial Intelligence Rev., 1995, 13 (5-6), 393-408

[Rocchio, 71] J. J. Rocchio, Relevance feedback in Information Retrieval, The SMART retrieval system - experiments in automatic document processing, Prentice Hall, Englewood Cliffs, 1971, 313-323

[ROSA, 04] http://www.zsi.pwr.wroc.pl/rosa

[Salton 89] G. Salton, Automatic Text Processing. The Transformation, Analysis, and Retrieval of Information by Computer, Addison-Wesley, 1989, Reading, MA.

[Salton, 90] G. Salton, C. Buckley, Improving retrieval performance by relevance feedback, Journal of the American Society for Information Science, Vol. 41(4), 1990, 288-297

[Shardanand, 95] Shardanand, Upendra & Pattie Maes, Social Information Filtering: Algorithms for Automating Word of Mouth, In Proc.of CHI'95 Conference on Human Factors in Computing Systems, ACM Press, 1995, 210-217

[Sing, 99] A. Singha,J. Choi, D. Hindle, D. Lewis, Pereira F, AT&T at TREC-7, The Seventh Text Retrieval Conference (TREC-7), NIST Special Publication 500-242, 1999, 239-252

[Schafer, 01] J.B. Schafer, J.A. Konstan, J. Riedl, E-Commerce Recommendation Applications, Data Mining and Knowledge Discovery, Vol. 5 No. 1/2, 2001, 115-153

[Srivastava, 00] J. Srivastava,R. Cooley,M. Deshpande, T. Pang-Ning, Web usage mining: Discovery and applications of usage patterns from web data.SIGKDD Explorations 1(2) ,2000 , 12–23

[Xu, 00] J. Xu, W. B. Croft, Improving the Effectiveness of Information Retrieval with Local Context Analysis, ACM Transactions on Information Systems, Vol. 18, No. 1, 2000, 79-112

[Xue, 03] G. Xue, H. Zeng, Z. Chen, W. Ma, H. Zhang, Ch. Lu, Implicit link analysis for small web search, In Proc Annual ACM Conference on Research and Development in Information Retrieval, 2003, 56-63