

**Hubi Helsinki -suositukset**

ETSITKO JOTAIN TÄLLÄISTÄ:

- » 3.9. Yksiköitä, yksiloita ja yhteisoja Työpaja Lume-mediakeskuksessa
- » 3.9. Svenska Talande Klubben Artisteja ja bändejä ilman musiikki- tai kielirajoja
- » 3.9. Peking-oopperan gaala Kinalaisen teatterin vaiheita esittelevä gaala-ohjelma
- » 3.9. Teatteriteltan taidenäyttely Katuteatteria, musiikkia ja performanssia teltan ympärillä
- » 3.9. Apinakapina Vauhdikas saunanukkenäytelmä sopii myös lapsille

[» TUTUSTU HUBI-SUOSITTIEMME!](#)

# Cultural Event Recommendations

## A case study

Teemu Mutanen, Joni Niemi, Sami Nousiainen,  
Lauri Seitsonen & Teppo Veijonen

ISBN 978-951-38-7166-6 (URL: <http://www.vtt.fi/publications/index.jsp>)  
ISSN 1459-7683 (URL: <http://www.vtt.fi/publications/index.jsp>)

Copyright © VTT 2008

JULKAISIJA – UTGIVARE – PUBLISHER

VTT, Vuorimiehentie 3, PL 1000, 02044 VTT  
puh. vaihde 020 722 111, faksi 020 722 4374

VTT, Bergsmansvägen 3, PB 1000, 02044 VTT  
tel. växel 020 722 111, fax 020 722 4374

VTT Technical Research Centre of Finland, Vuorimiehentie 5, P.O. Box 1000, FI-02044 VTT, Finland  
phone internat. +358 20 722 111, fax +358 20 722 4374

VTT, Vuorimiehentie 3, PL 1000, 02044 VTT  
puh. vaihde 020 722 111, faksi 020 722 6027

VTT, Bergsmansvägen 3, PB 1000, 02044 VTT  
tel. växel 020 722 111, fax 020 722 6027

VTT Technical Research Centre of Finland, Vuorimiehentie 5, P.O. Box 1000, FI-02044 VTT, Finland  
phone internat. +358 20 722 111, fax +358 20 722 6027



Series title, number and  
report code of publication

VTT Working Papers 105  
VTT-WORK-105

Author(s) Mutanen, Teemu, Niemi, Joni, Nousiainen, Sami, Seitsonen, Lauri & Veijonen, Teppo		
Title <b>Cultural Event Recommendations</b> <b>A case study</b>		
Abstract This paper presents recommender system for use in cultural event context. A pilot study was conducted by implementing a festival menu on cultural event web pages. The menu implementation is able to deliver five different types of recommendations. Real content and clickstream data are used to evaluate the recommendation methods. Findings show that the festival menu attracts steady interest and the event pages receive new users.		
ISBN 978-951-38-7166-6 (URL: <a href="http://www.vtt.fi/publications/index.jsp">http://www.vtt.fi/publications/index.jsp</a> )		
Series title and ISSN VTT Working Papers 1459-7683 (URL: <a href="http://www.vtt.fi/publications/index.jsp">http://www.vtt.fi/publications/index.jsp</a> )		Project number 19814
Date September 2008	Language English	Pages 17 p.
Name of project Ubiikki Helsinki	Commissioned by	
Keywords cultural events, recommendations, recommendation engine, web page	Publisher VTT Technical Research Centre of Finland P.O. Box 1000, FI-02044 VTT, Finland Phone internat. +358 20 722 4520 Fax +358 20 722 4374	

# Contents

1. Introduction.....	5
2. Recommenders.....	6
3. Festival Menu.....	8
3.1 Trial .....	8
3.2 Menu Implementation .....	8
3.3 Recommendation Content .....	9
3.4 Recommendation Type Rotation.....	9
3.5 Click Tracking.....	10
3.6 Performance Testing.....	10
3.6.1 Test Execution.....	10
3.6.2 Test Results .....	11
3.6.3 Analysis of Bottlenecks.....	11
4. Results.....	12
4.1 Popularity and Users.....	12
4.2 User's Page Visit.....	13
4.3 Rank.....	14
5. Conclusions.....	16
References .....	17

# 1. Introduction

In this paper we present an implementation of recommendation techniques in cultural event context. The literature of recommendation systems and techniques is numerous and varied [1]. However, to our knowledge, studies related to recommending cultural events have not been publicly reported. Cultural events have temporal attributes (start and end time) and are not necessarily repeated. Furthermore, location information can be expected to influence the interest of an event from the point of view of a visitor. Cultural events are versatile in topic and the expected number of visitors can vary considerably between events ranging e.g. from a hundred to several tens of thousands.

Several studies address the issue of recommending web pages. For example, [3] and [2] propose and evaluate hybrid recommendation systems utilizing both content and click data for making recommendations. In both cases, the web page recommender is implemented in university web pages and the clickstream data is collected from web server log files. In terms of application domain, the closest example we have found is Zync [6]. The system relies on explicitly collected information about user interests and ratings.

The rest of the paper is organized as follows: section 2 presents five different types of recommendations; in section 3 the implementation issues are presented. The results of the pilot are presented in section 4, while conclusions and future research directions are discussed in section 5.

## 2. Recommenders

For recommendations to be effective the recommendation system should produce recommendations specific to the user's desires and wishes. The system should be able to adapt to this need by delivering expected type of recommendations to the user. Or, at minimum, the system should indicate what type of recommendations are delivered to the user. In the following we present five different recommendation types in focus on this paper. The types are following:

**Similar Events** Recommendations are based on the event description data. Similar events are determined based on event description text analysis by TF-IDF method. Although the TF-IDF weighting method is relatively old idea, we apply TF-IDF because it is simple and effective [7]. The TF-IDF vectors are generated by counting binary occurrence of the keywords and pruning the keyword space to values between 3 and 15.

**Popular Events** Recommendations are based on the clickstream data and individual user counts. Recommendations inform user about the popular events, for example on a given day.

**Coming Soon** Recommendations are based on the event information: date and time. User will find value from these recommendations when she seeks an event occurring soon, e.g. tonight or this weekend.

**Surprising Events** Recommendations are based on the clickstream data and individual user counts. Surprising recommendations point out rare events overall and previously unknown events to the user.

**Similar Users Go** In our context, we have little information about users; to avoid cold start problem we use combined information from the users' page visits and page contents. The recommender uses a data model that has a feature vector for each event and each user. The event vectors are initialized by content based analysis, user vectors are initialized to null.

When a user visits a cultural event page user's feature vector is updated as follows:

$$w'_u = \frac{N_u w_u + w_e}{N_u + 1} \quad (1)$$

where  $w'_u$  is the new feature vector of the user,  $w_u$  user's current feature vector,  $w_e$  corresponding event feature vector and  $N_u$  number of previous event page loads by the user.

In similar way, the event's feature vector is updated.

$$w'_e = \frac{N_e w_e + w_u}{N_e + 1} \quad (2)$$

where  $w'_e$  is the new feature vector of the event,  $w_e$  event's current feature vector,  $w_u$  corresponding user feature vector and  $N_e$  number of previous event page loads of the events.

In the festival menu implementation the maximum number of individual page loads is restricted to 25. This restriction has a short-term memory effect. The method updates the feature vectors by giving higher importance to recent clicks. Thus, user receives recommendations similar to her current status and interests.

In addition, we also implemented a random recommendation type: five randomly chosen events delivered to the user as recommendations.

## 3. Festival Menu

### 3.1 Trial

The festival menu pilot was arranged in web environment. Seven temporally and thematically overlapping cultural festivals participated in the pilot and had the festival menu included on their web pages. The number of events from each festival is shown in Table 1.

*Table 1. Number of events in each festival organizer.*

	<i>Pages with menu</i>	<i>Recomm. events</i>
Festival 1	18	12
Festival 2	14	6
Festival 3	1	1
Festival 4	26	23
Festival 5	21	16
Festival 6	17	17
Festival 7	1	1
Total	98	76

### 3.2 Menu Implementation

The festival pages included the festival menu by embedding a few lines of code to the HTML or PHP code of the page. The festival menu code lines include a `<div>` element and a JavaScript snippet. Each time when a festival page is loaded on the user's browser, the JavaScript snippet calls a dedicated Servlet on the recommendation server. The Servlet response is the menu content as HTML and JavaScript formatted code. The content is injected into the `<div>` element using `innerHTML` property.

Each menu request as well as the provided recommendations are registered and stored by the server.



### 3.3 Recommendation Content



*Figure 1. Illustration of the festival menu.*

The recommendations are links to the selected events of other festival sites. The information displayed for each link includes date, event title and a brief event description, an example is shown in Figure 1.

As the festival menu is profiled to include only cross-festival recommendations, events of the festival site itself are not included in the recommendations. As an exception to this are the Popular recommendations. Another constraint on the recommendation content is a balancer filter for limiting the number of recommendations of events by each festival so that no single festival would dominate more than 3 recommendations on any menu.

### 3.4 Recommendation Type Rotation

The five-item menu content will be dynamically generated for each request. Six recommendation types are rotated in the menu content so that one recommendation type will be applied for each menu request. The user cannot indicate which type of recommendations she likes to receive. The user was provided a subtitle of the recommendation type. The purpose of the title was to make it more evident to the user what kind of recommendations the menu contains. The titles for the different recommendations were formulated as self-explanatory as possible. However, random recommendations were titled as *Selected samples* to conceal the fact that those "recommendations" are in fact just random shots.

## 3.5 Click Tracking

The tracking data contains information on which festival event pages a user has visited, what menu content has been provided for those pages, and which recommendation link the user has clicked. For gathering this data from distinct festival sites and for providing cross-festival recommendations, user's browser will be provided with a tracking cookie containing a unique user ID string. The cookie is written when the user loads a festival page containing the festival menu for the first time. Consequent menu loads and recommendation clicks on any of the festival pages are then registered to the same user.

User's browser will be provided a third party cookie. Using third party cookies is a drawback in user tracking, because some browsers will not accept third party cookies by default [5]. Although first party cookies would be accepted more often, the only suitable solution in cultural cross festival context is to use third party cookies. Only with a third party cookie can the user be tracked across multiple websites.

## 3.6 Performance Testing

The festival menu implementation was subjected to performance testing in order to determine the overall feasibility of the approach taken, to survey effectiveness of current implementation, and to find possible performance bottlenecks.

### 3.6.1 Test Execution

The first goal of testing was to identify upper limits for normal operational capacity of the system.<sup>1</sup> After this limit was discovered, performance bottlenecks were identified and a baseline for future regression testing was established. The tests were executed by having virtual users send a series of HTTP requests to the recommender server in consecutive batch runs. The usage profile for a virtual test user was to alternate between recommendation menu retrieval (web page hit) and recommendation usage (clicking on a recommendation link).

---

<sup>1</sup> test set up: the recommendation implementations and the MySQL 5.0.45 database server were located on a Ubuntu Linux 7.10 server (2 GB of RAM). The tests were run with Microsoft Web Application Stress tool 1.1

### 3.6.2 Test Results

The capacity of the non-optimized recommendation implementation turned out to be around 40 HTTP page requests per second, with average turn-around time being about 2,5 seconds. The performance bottleneck proved to be the number of database transactions. With continuous load of 40 recommendator HTTP page requests per second, the application server reported around 80 database transactions per second. Given that the MySQL database server was only rudely tuned, the transaction throughput in itself was not very bad [4].

*Table 2. Database roundtrips.*

<i>Recommendation type</i>	<i>Database queries</i>	<i>Database updates</i>
Similar	8	-
Popular	6	-
Surprising	8	-
Similar users go	11	2
Coming Soon	2	-
Random	7	-

### 3.6.3 Analysis of Bottlenecks

The tracking data was recorded to database synchronously, which proved to be the biggest bottleneck in the system. Every user page hit caused at minimum one database transaction, as in the case of user clicking on a recommendation link. The database resources required for construction of recommendation menus varied much more. Table 2 depicts the overall number of database queries and updates consumed by different recommendation menus. Most of the queries were from static tables and thus read from database cache. In addition to the largest number of database queries, the similar users go -recommendation implementation was the only recommendation type requiring updates in the database, which made its performance considerably worse than that of others.

## 4. Results

The festival menu implementation was running the whole month of May, in this paper we analyze data gathered from time period: April 26-May 26. One drawback for using the whole time period is that the number of events decreased as events were removed from the possible recommended events list upon ending.

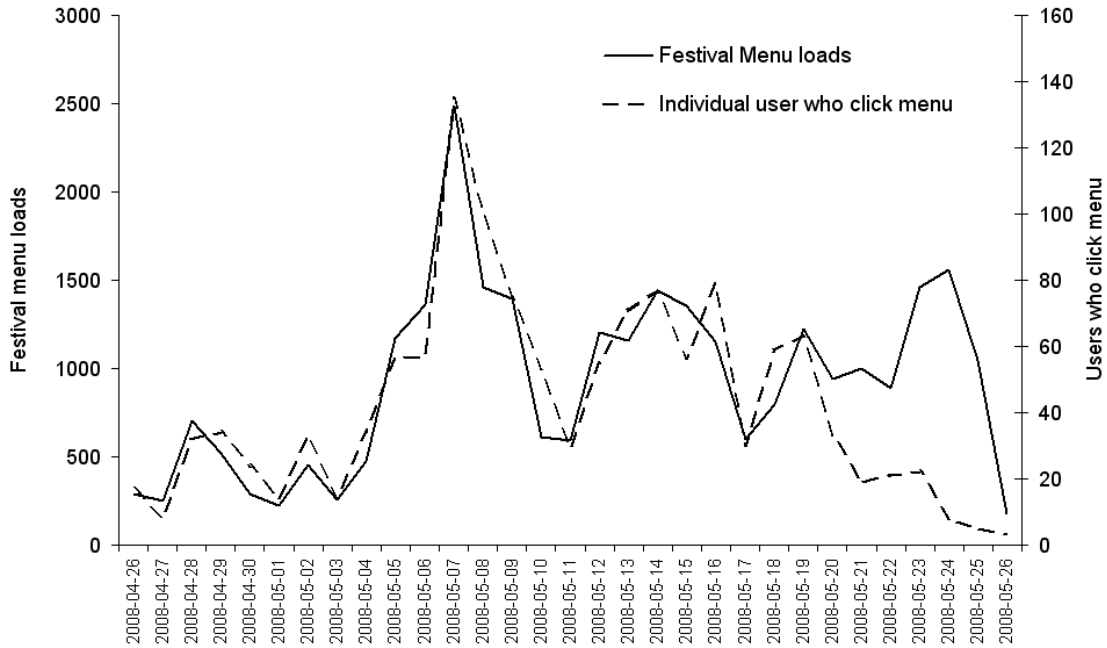


Figure 2. Individual users who click menu and festival menu loadings on each day.

### 4.1 Popularity and Users

During the research period of 31 days, a total of 59 803 menus were displayed, each containing 5 recommendations. Recommendation links were clicked 1 406 times. The number of individual users who clicked the menu during a given day is presented in Figure 2, along with the number of festival menu page loadings. Figure 2 also shows that number of recommendation clicks goes down in the end of May. There is no sign of word of mouth -effect because the proportion of users using the festival menu do not grow. Of course the overall interest grows, but the proportional amount of users, menu loadings and users who click the links on the menu remains the same.

Table 3. The number of clicks on different recommendation types on the given time period 26.4.–26.5.

<i>Type</i>	<i>Clicks</i>	<i>Impressions</i>	<i>% share</i>
Similar	213	9898	2,15
Popular	324	10210	3,17
Surprising	217	9950	2,18
Similar users go	244	9915	2,46
Coming soon	218	9904	2,20
Random	190	9926	1,91

The number of impressions and recommendation clicks for each recommendation type is presented in Table 3. The click rates show that all recommendation types were close to each other in popularity. The number of different recommendation menu impressions were almost even because the different types were evenly distributed to the users.

Table 4 shows the statistical significance between recommendation types. The click counts of each recommendation type are alone compared to the click counts of the random recommendation type. The p-values are presented with a null hypothesis: same underlying distribution. The p-values show that the Similar, Surprising and Coming Soon recommendations do not have statistical difference from random recommendations. The two other recommendations: Popular, and Similar Users Go have different click count behaviour from random recommendations.

Table 4. The p-values from t-test when different type of recommendation click rates are compared.

<i>Similar</i>	<i>Popular</i>	<i>Surprising</i>	<i>Similar users go</i>	<i>Coming soon</i>
0,076	0,001	0,135	0,004	0,053

## 4.2 User's Page Visit

In Figure 3 the distributions of reading times are presented. Users are divided in to two groups according to whether or not they use festival menu. The Figure 3 shows that those users who arrive to the event page from the festival menu spend more time in the event page. In Figure 4 the distributions of page loadings by individual users are

presented. The Figure 4 shows users load more event web pages, and visit number of organizers, when they find and use the festival menu recommendations than they would without festival menu.

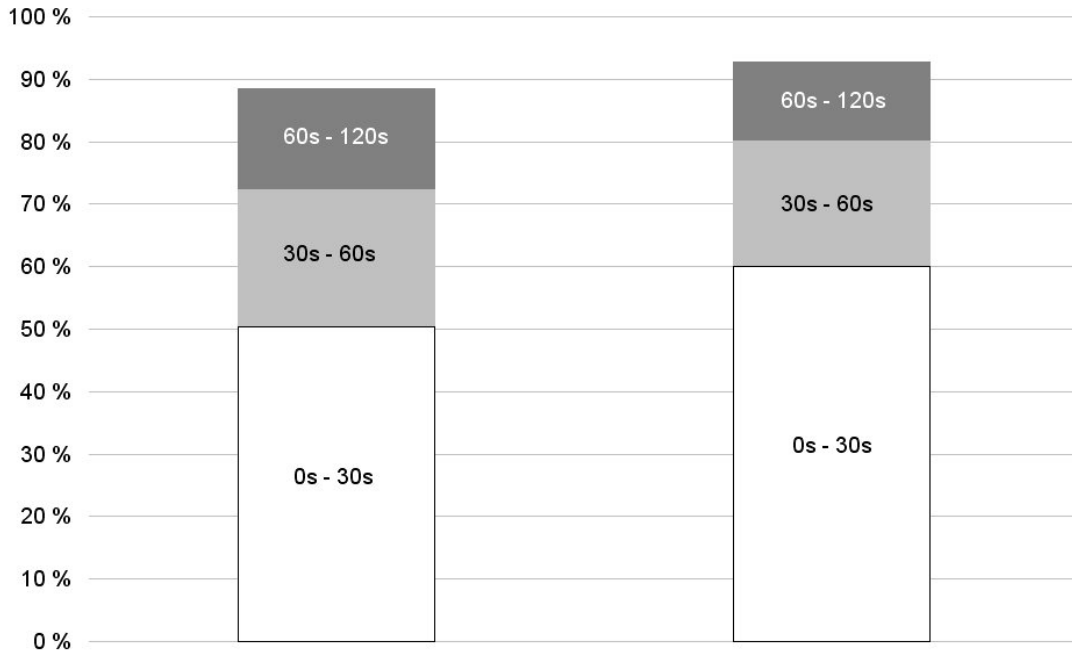


Figure 3. The distribution of the user's page reading time, with (left) and without (right) using Festival menu.

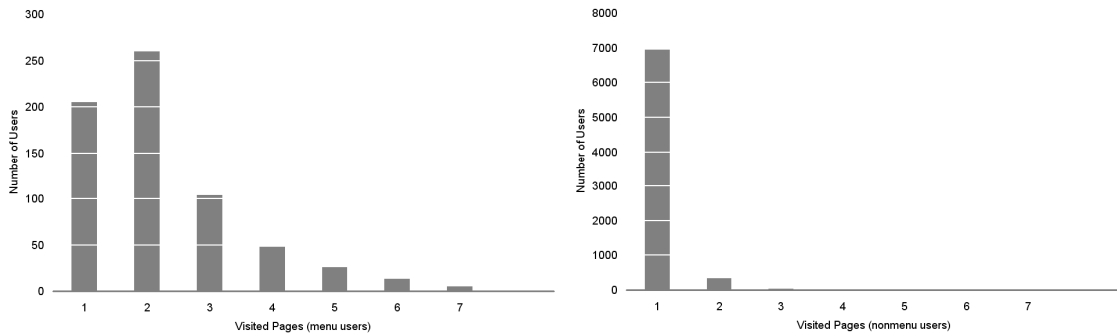


Figure 4. The distribution of how many festival pages user visited, with (left) and without (right) using Festival menu.

### 4.3 Rank

In the festival menu, five recommendations were presented in ranked order in each menu impression. The relevance of the rank is illustrated in Figure 5, which shows the click count on each rank position for four recommendation types. The three types with significant relevance are compared with the random recommendation type. Figure 5

shows that the click distributions are skewed towards to the top-ranked positions, except for the random recommendation type.

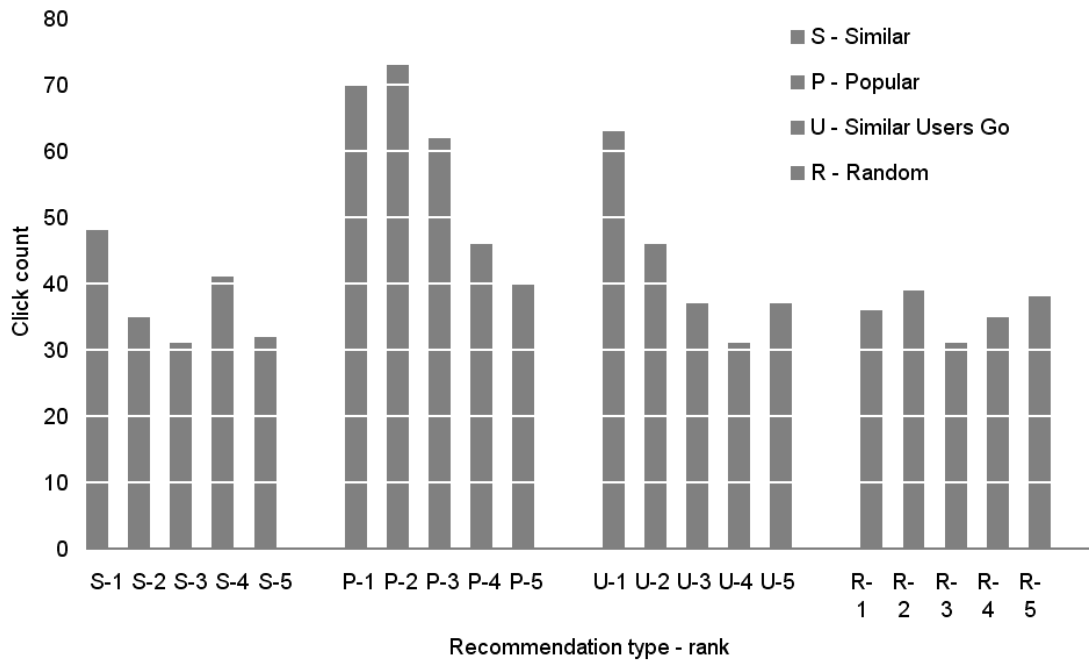


Figure 5. Click counts on different recommendation types and the rank in the menu.

## 5. Conclusions

In this paper we presented a festival menu implementation in a cultural event context. The recommendations produced are not exactly long-tail recommendations because of the small number of events. However, the idea of producing recommendations for targeted audience is illustrated with these cultural events. The Festival Menu implementation produced more relevant traffic to the event organizer's websites, as can be seen from Figure 3 and Figure 4.

In case of Similar Events recommendations, it was observed that such irrelevant words as *today* and *take* got chosen to the term vectors describing the events. Expanding the stop-word list might resolve the issue, but another way is to request a keyword list for each event from the festival organizer so that only more meaningful words in the event context (such as *dance*, name of the performer, *Finnish*) would appear in the term vector.

One future research direction would be to develop location and time based recommendations. In cultural event context this would mean map based interface and recommendations with location restrictions. Time based recommendations could be more meaningful to the users in mobile context than in web.

Another future research direction is the development of more relevant recommendation types. A future implementation could give higher importance for page loads with only few clicks. Rare events could be made more meaningful than blockbusters when recommendation system tries to analyse user behaviour and produce relevant recommendations based on it.



## References

1. Burke, R. 2002. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12, pp. 331–370.
2. Haruechaiyasak, C., Shyu, M.-L. and Chen, S.-C. 2004. A data mining framework for building a web-page recommender system. In: 2004 IEEE International Conference on Information Reuse and Integration. Pp. 357–362.
3. Li, J. and Zaïane, O. R. 2004. Combining usage, content, and structure data to improve web site recommendation. 5th International Conference on Electronic Commerce and Web Technologies. Pp. 305–315.
4. MySQL. Mysql performance benchmarks – measuring mysql’s scalability and throughput. Technical report, MySQL AB, Retrieved on 2008-04-15.
5. [http://en.wikipedia.org/wiki/http\\_cookie](http://en.wikipedia.org/wiki/http_cookie). www-address, 2008.
6. <http://www.zync.com>. www-address, 2008.
7. Salton, G. and Buckley, C. 1988. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24, pp. 513–523.

