Web personalization using Extended Boolean Operations with Latent Semantic Indexing

Preslav Nakov
Bulgaria, Sofia, Studentski grad. bl.8/room 723
(preslav@rila.bg)

Key words: Information Retrieval and Presentation, Search, Data Mining, Natural Language Processing, Data Mining and Knowledge Discovery, Distance Learning and Electronic Commerce, Knowledge Construction from the Web

Abstract. The paper discusses the potential of the usage of Extended Boolean operations for personalized information delivery on the Internet based on semantic vector representation models. The final goal is the design of an e-commerce portal tracking user's clickstream activity and purchases history in order to offer them personalized information. The emphasis is put on the introduction of dynamic composite user profiles constructed by means of extended Boolean operations. The basic binary Boolean operations such as OR, AND and NOT (AND-NOT) and their combinations have been introduced and implemented in variety of ways. An evaluation is presented based on the classic Latent Semantic Indexing method for information retrieval using a text corpus of religious and sacred texts.

Introduction

The pre-Internet era imperative stated that more data means better chance to find the information needed. Internet has imposed new standards and new way of thinking. In 1994 the World Wide Web Worm received an average of about 1500 queries per day, in November 1997 only one of the top four commercial search engines finds itself (returns its own search page in response to its name in the top ten results) and nowadays the AltaVista search engine serves hundreds of millions queries per day. With the enormous growth of the information available on the Web the goal has changed and the main efforts are directed towards the limitation of information presented to the user. ([5])

The first that felt the problem were of course the search engines and they offered the users several possibilities for advanced query refinements. Unfortunately their usage remained highly limited, since as Marchionini argued: "End users want to achieve their goals with a minimum of cognitive load and a maximum of enjoyment. …humans seek the path of least cognitive resistance and prefer recognition tasks to recall tasks; most people will trade time to minimize complexity". [17]

The problem of the relevance of information presented to the users was well understood by the commercial Internet sites. When people find some magazine irrelevant to their information expectations they simply stop to buy it. It is the same with the Web sites: if the information presented does not meet the customers' needs they never return there.

The limited volume of the magazine does not permit to include everything people would find relevant and they tend to specialize in a particular area. People buy only those magazines that are relevant to their specific interests. The Web sites and portals are a different case because there are no so strong limitations of the amount of information to be published, as is the case with the magazines. The biggest Internet portals like those of Yahoo!, MSN or Netscape can offer almost all a customer may need. The problem is how to organize the site in order to help the users find what they are actually looking for.

The Idea of the User Profile

The most valuable decision is the development of a dynamic model of the interests for the specific user. The first attempt in the development of user's profile was asking the user to enter some words that best describe his or her interests. Another possibility is the selection of the relevant ones among a set of articles. Each of these articles can be assigned a list of key words that will be used to limit the information presented to the user. For example, a personalized search engine could return information only from the field of interest to the user. The same way a well-personalized Web site changes dynamically its content in order to present to the user only relevant information, news or advertisements, according to the previously created profile. ([10])

Asking explicitly the users for some kind of relevance feedback may not always be the best way to create their profiles. This is especially the case when using key words. As Furnas, Landauer, Gomez and Dumais have shown in [11], people use the same words to describe the same subject in 10-20% of the time (see also [4]). The relevance feedback when using whole articles may not be correct too, because of the influence of some subjective factors like novelty, informativeness or familiarity to the user. Some sites/portals offer the customers the opportunity to receive a free "passport". The users are asked to fill a form and answer a set of common questions that will be used as a primary source for the construction of their profiles. This may be of great importance and can lead to significant improvements.

The problem is that people tend to get annoyed when are asked to do something in order to help the system. That is why several business sites/portals developed specialized mechanisms for automatic user's profile construction. Some of the recent studies and applications in the field include the automatic tracking and recording of the user's activity when browsing on the site: e.g. page visited, button clicked, hyper link followed, search query entered etc. The information collected this way is called clickstream and is stored in specially designed clickstream data marts and Data Webhouses. Thus, the Web site/portal retains a full history of the user's activity that permits the construction of more effective and objective profile. ([13,14,15,21])

For almost all the cases the user profile has a dynamic character and changes over time since new information becomes available. The general sources of additional information are the raw details of the recorded user activity: the clickstream. Most of the systems use a vector representation of the user profile. This is very convenient and, as we will show later, simplifies its creation, support and usage. Although there are several different techniques for vector generation of the type described, we have chosen the Latent Semantic Indexing for several reasons the primary of which is that it is a well-studied classic method that will allow us to concentrate on the

specific details of profile creation and usage we want to study.

Latent Semantic Indexing

The *Latent Semantic Indexing* (LSI) is a powerful statistical technique for fully automatic indexing and retrieval of information. LSI is generally applied to texts and represents a two-stage process that consists of (see [7], [9], [16] for details):

- *off-line* construction of document index, and
- *on-line* respond to user queries.

The off-line part of the process is the training part when LSI creates its index. First a large word-to-document matrix X is constructed where the cell (i,j) contains the frequencies of occurrence of the i-th word into the j-th document. After that, a *singular value decomposition* (SVD) is performed which gives as a result three matrices D, T (both orthogonal) and S (diagonal), such that $X=DST^t$. Then all three matrices are truncated in such a way that when we multiply the truncated ones D', S' and T' we get a new matrix X' which has the same dimensionalities as X and is the least-squares best fit approximation of X. This results in the compression of the original space in a much smaller one where we have just a few number of significant factors (usually 50-400). Each document is then represented by a meaning vector of low dimensionality (e.g. 100). It is possible to perform a sophisticated SVD which speeds the process by directly finding the truncated matrices D', S' and T' (see [4]).

The on-line part of LSI receives the query (pseudo-document) user typed and finds its corresponding vector into the document space constructed by the off-line part using a standard LSI mechanism. Now we can measure the degree of similarity between the query and the indexed documents by simply calculating the cosine between their corresponding vectors. Other possibilities include the usage of the angle, Euclidean distances between the normalized document vectors, Manhattan, Chebishov's and other measures.

Extended Boolean Operations

We return now to the automatic creation of vector representation of the user's profiles. Consider an e-commerce portal tracking users' clickstream activity as have been discussed above. The information collected can be used in variety of ways including analysis of the quality of the Web site structure and organization, etc. ([13]) There are several things we are interested in when constructing the users' profiles among which the most important are:

- Which sections/pages on the site the customer visits most frequently? What do they content?
- Which pages are "session killers" for our customer?
- How long time does the customer spend on the site?
- Who is our customer? How often he or she visits the site?
- Has the customer purchased something and what, if any? What kind of products?
- Is it a complaining customer that often returns back our products?

Having collected information like this will allow us to create a sophisticated high quality user profile that will permit offering him or her personalized news, advertisements, banners etc. We would like to create a profile vector that is closely aligned to the vectors of the pages the user is interested in and is far from those which seem to be uninteresting. A page of interest for the user may be a page where he goes often or spends a long time. The longer the user stays on the page, the more relevant it may be to his or her interests. On the other hand we must beware not taking too seriously the extremely long times (the user has just left the browser open) giving at the same time higher weights for the pages related to the user's purchases, if any. So, we would like to combine the vectors of the relevant pages, weighted according to the frequency of the visits and the duration of the time spent there, in order to obtain the profile vector. This implies the need for a weighted OR similarity measure. We would like also to exclude the pages that seem to be strongly uninteresting for the user: e.g. those where he or she (often) cancels the session or those, we know he or she is not interested in, according to a relevance feedback, possibly taken from the user "passport" registration information supplied. This implies the need for excluding NOT (MINUS) Boolean operation. These examples show that the extended Boolean operations play a major role in the process of user's profiles creation.

Another possibility is to design a composite profile by keeping several different vectors whose weighted combination gives the profile vector. This results in improved performance since we can manage the different vectors the profile is built of separately and combine (some of) them only when needed. This allows the creation of a *dynamic* profile that may be recalculated when needed and with changed weights. For example, we may like to drop some elements of

the user profile that are no longer relevant (because are old), or at least reduce their weights.

Consider we have collected a complete history of the customers' purchases and clickstream activity, and want to send the users several advertisements by e-mail or show them on the Web when browsing the site/portal. We have already developed LSI index based on the text description of each product. We can think of the purchases/clicks as query components and of the advertisement as a new document in the same space. We need some kind of similarity function that will give us a measure of the similarity between our advertisements and the user's profile. Let us define d_1, d_2, …, d_n as distances (in LSI sense) between the ad and the n components of the query. The classic LSI algorithm calculates the cosines between the vectors in order to find the degree of their similarity. Most of the similarity measures for the Boolean operations we propose below are based on Euclidean distances, although we can use some other distances (angle, Manhattan distance, Chebishov's distance, power distance, etc.). It is important to note that we must first normalize the vectors before calculating Euclidean distances. All Boolean operations proposed return a value between 0 and 1. Almost the same results can be obtained when using the classic cosines but for some functions it is difficult to fit the values returned in the interval [0,1]. We have experimented a lot trying a large quantity of measures and found that the usage of Euclidean distance seems to be the most appropriate.

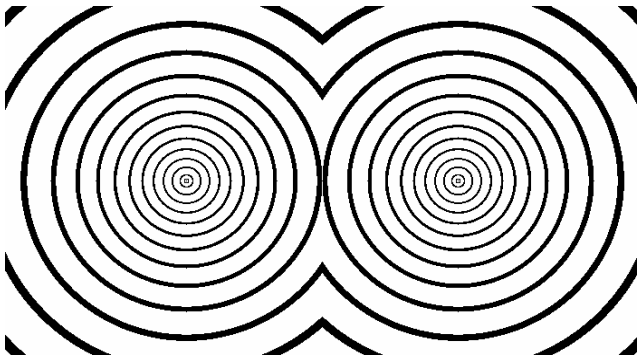

Figure 1. OR similarity for two-component query

There are several similarity measures we have experimented with:

- **OR-similarity measure.** This measure depends only on the minimal distance between the document and the query components and has the following general representation:
  $S_{or} = f(\min(g(d_1),g(d_2),\ldots,g(d_n)))$, where $f(x)$ and $g(x)$ are some one-argument functions.
  In case we have more information for the query we can add weights to the query components and modify $g(x)$ to $g(x, w)$. So the formula is:
  $S_{or} = f(\min(g(d_1,w_1),g(d_2,w_2),\ldots,g(d_n,w_n)))$
  OR similarity measure has well separated picks at the query components vectors.
  Example: The similarity measure for two- and three-component query,
  $f(x) = 1/(1+x)$
  $g(x) = x$
  are shown on figures 1 and 2.

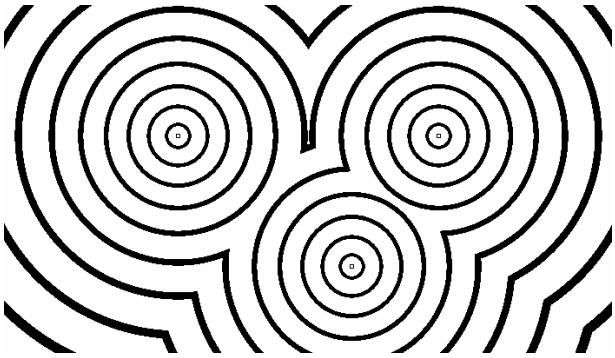

Figure 2. OR similarity for three-component query

- **AND-similarity measure.** This measure depends only on the sum of distances between the document and query components. It has the following general representation:
  $S_{or} = f(g(d_1)+g(d_2)+\ldots+g(d_n)))$, where $f(x)$ and $g(x)$ are some one-argument functions.
  And again if we have more information for the query and we can add weights to the query components and modify $g(x)$ to $g(x, w)$. In this case the formula is:
  $S_{or} = f((g(d_1,w_1) + g(d_2,w_2) + \ldots + g(d_n,w_n)))$
  Usually this measure can be thought of as a superposition of distinct similarity measures of the query components.
  Example: The similarity measure for two- and three-component query,
  $f(x) = 1/(1+x)$
  $g(x) = x$
  are shown on figures 3 and 4.

- **Combination of the previous two (AND-OR)**. This similarity measure is a combination between the previous two.
  $S_{and\text{-}or} = f(S_{and}, S_{or})$
  Example: We can use linear combination between S_{or} and S_{and} measures.
  $S = k.S_{or} + (1–k).S_{and}$, where k is constant and $0 \le k \le 1$.

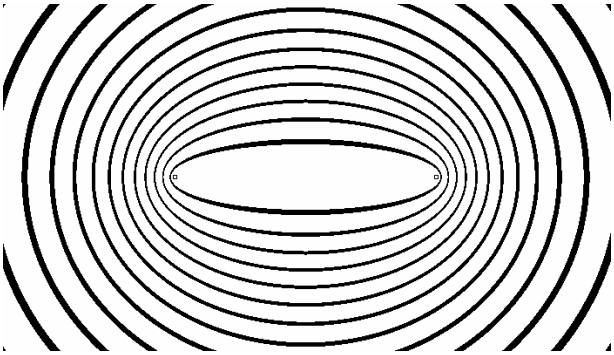

Figure 3. AND similarity for two-component query

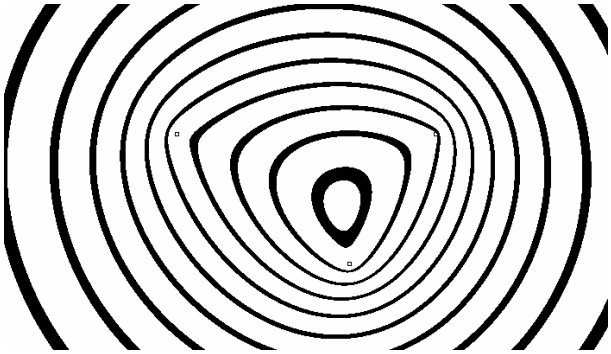

Figure 4. AND similarity for three-component query

Figure 3 shows the two component query results for $k = 0.5$. We still have two distinct parts like the OR-similarity function but higher values in the middle region between them just like the AND-similarity function.

Figures 5 and 6 show the results for a two- and three component queries with $k = 0.5$ Figure 7 is an example of weighed combined similarity measure.

- **MINUS and Binary NOT (AND-NOT)-similarity measure.** In case we want to exclude a vector we can apply two different similarity measures: MINUS and NOT. For the MINUS similarity measure, if the vector considered is more similar to the exclude vector it will receive a similarity measure of 0. (see the second clause below) Otherwise we return a the similarity measure that takes in account the distance to the include vector only.

  Example:

  We can use the following MINUS-measure:

  $S_{not} = d_1$, when $d_1 < d_2$, and

  $S_{not} = 0$, else.

  The result is shown on figure 8.

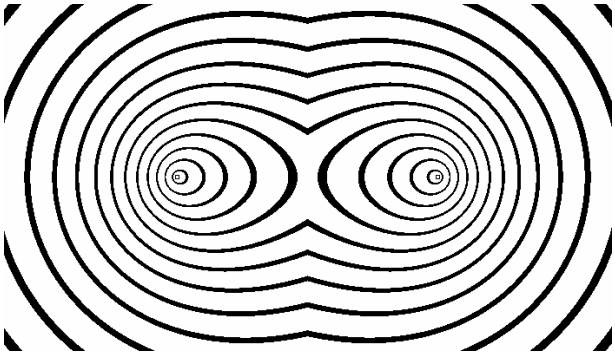

Figure 5. Combined similarity for 2 comp. query, k=0.5

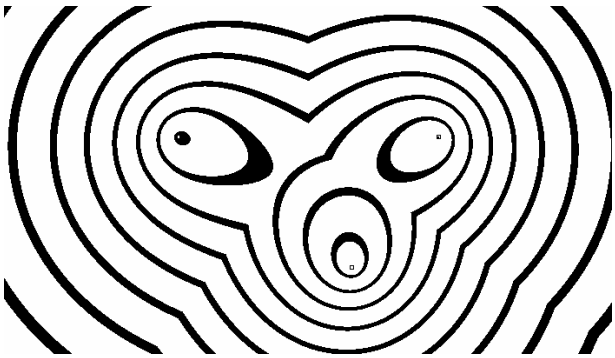

Figure 6. Combined similarity for 3 comp. query, k=0.5

The problem with this measure is that it takes in account d_2 only when deciding whether to cut the value. A more sophisticated implementation may be used: the NOT (AND-NOT) similarity measure. If the document is more similar to the exclude document text it will receive a similarity measure of 0, but otherwise we return a similarity measure between 0 and 1 that takes in account the distances to *both* documents.

Example:

We can use the following NOT-measure:

$S_{not} = 1 - d_1 / (1 + d_2)$, when $d_1 < d_2$, and

$S_{not} = 0$, else.

The result is shown on figure 9.

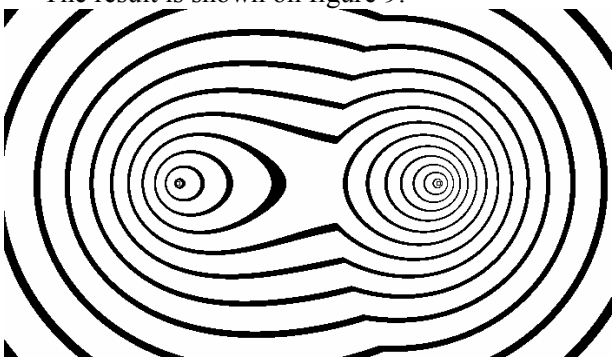

Figure 7. Weighted combination for 2 comp. query, k=0.5

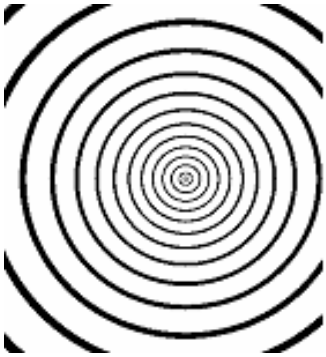

Figure 8. MINUS similarity measure

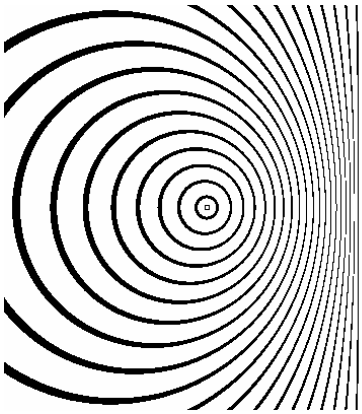

Figure 9. NOT (AND-NOT) similarity measure

Application to Religious and Sacred Texts

The first step toward the construction of the dynamic user profile is development of the appropriate extended Boolean operations. We have experimented the performance of the Extended Boolean Operations presented above on a large number of different corpuses containing thousands of documents by thousands of words and hundreds of megabytes.

We will demonstrate how the functions we introduced above work on a small corpus of religious and sacred texts we found at: http://davidwiley.com/religion.html. We selected 196 different religious and sacred texts from 14 categories: apocrypha (acts, apocalypses, gospels, writings), Buddhism, Confucianism, Dead Sea scripts, The Egyptian Book of Dead, Sun Tzu: The Art of War, Zoroastrianism, The Bible (Old and New Testaments), The Quran and The Book of Mormons. The experiments were made in a 30 dimensional space (see Fig.1) with a preliminary to SVD replacement of the frequencies in X (196 documents × 11451 words) with their logarithms. Fig. 10 illustrates the inter-document similarities given by the correlation matrix (196×196), shown in 5 different colors for the five correlation intervals:

- 87,5-100%, black color;
- 75-87,5%, dark gray;
- 62,5-75%, gray;
- 50-62,5%, light gray;
- 0-50%, white.

The dark rectangles in the main diagonal show the high correlation between texts belonging to the same religion. For example: the black rectangle from the bottom right corner contains texts from the Book of Mormons. To the left and up on the main diagonal can be found the Quran, then the Old Testament (The Bible), then come the Zoroastrian texts, The New Testament (The Bible), the Sun Tzu's Art of War, the Egyptian Book of the Dead and so forth. And the smooth rectangle in the upper left corner shows the relatively high similarity between all kinds of apocrypha present. We see for example that The Book of Mormons is more correlated to the New Testament than to The Old Testament.

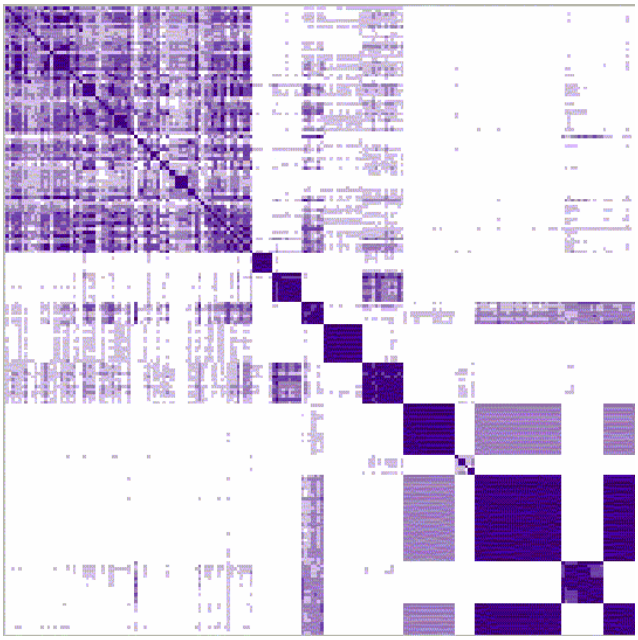

Figure 10. Correlation between religious texts (196 × 196)

Some of the religious texts are too big (like the Bible or the Quran) and we have selected just a part of them. The corpus of texts selected for our experiments on the behavior of the different Boolean operations we will work on is about 5,5 MB and is well structured: there are well-separated clusters. The clear rectangles we can see on Fig. 10 correspond to texts from different religions, which made the analysis of the correctness of the results we obtained easier.

We performed several different experiments that can be separated in two general classes: "practical" and "theoretical". The "practical" experiments include the composition of two or more different queries and their combination with Boolean operations. The second class of "theoretical" experiments includes the choice of two or more texts from the same space and performing queries using Boolean operations (OR, AND and NOT). We think that the second class illustrates better the general idea and the results we obtained.

We developed several specialized software command line tools supporting both the on-line and off-line LSI stages using the standard SVDPACKC routines for the singular value decomposition (see [4]). A LSI based natural language query search engine has been developed based on these tools and exposed on the Web at (http://**self reference withdrawn**).

Below are presented eight different tables that contain experimental results obtained for two example texts from the corpus belonging to different well-separated clusters (religions): the first chapter of the Sun Tzu's Art of War (suntzu1.txt) and the first chapter of the Confucianism religious texts (conf1.txt). The first table contains the ranked top list of the documents similar to the first chapter of Sun Tzu's Art of War with the corresponding degree of similarity. Then follow seven tables containing the results from the application of different Boolean operations. This corresponds to the case when the system tries to judge whether a particular document is relevant to the user. Consider the user's clickstream activity shows he is interested in information common to both the documents. The system needs to perform a Boolean AND operation on the LSI vectors of those documents and to produce a ranked document sort list in order to choose the relevant documents. The results are shown in the second table. The following two tables contain the results of the application of two different excluding operations: NOT and MINUS, whose behavior has been discussed above. Then follow four tables showing the results of the application of four different types of OR operations for different values of k (see above).

The experiments with the combined OR and AND similarity measure search using different values for the parameter k were among the most interesting ones. Fig. 11 gives another view perspective on the results, showing the distribution of the correlation coefficients for all the 196 documents using a text from the Sun Tzu's Art of War and another one from the Egyptian Book of the Dead. We can see again that the results vary which suggests that we can obtain quite different results by tuning the parameter k. In fact the sorted order of the best matches returned by the query is almost the same for all the cases.

Discussion

The results presented above show that the Boolean operations proposed perform well and can be used successfully in the meaning vectors construction by using any kind of Boolean expressions. As have been mentioned above, the correct application of the Boolean operations is a key point in the development of the dynamic user profile.

The operations can be useful also in the construction of a natural language query system giving the users the opportunity to combine any kind of natural language queries. After the ranked list has been

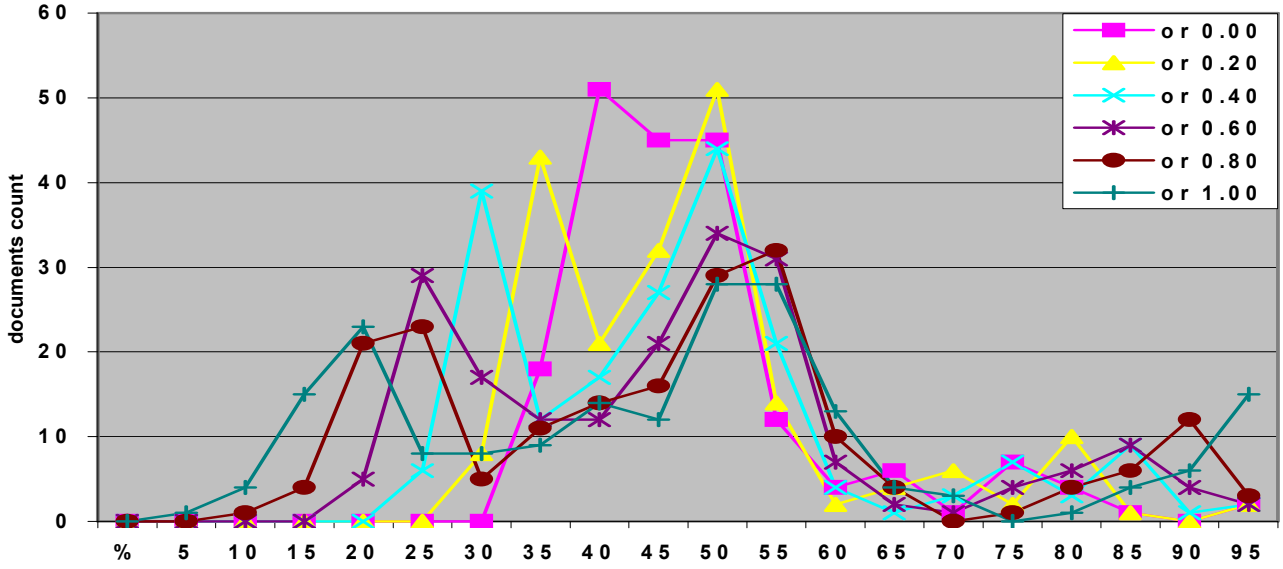

Figure 11. OR similarity. Correlation coefficients distribution

returned the user can provide the system a relevance feedback by pointing out some of the documents as relevant or non-relevant to his or her query. The system will then provide a second ranked list of documents by combining the vector of the user query with the vectors of those documents using the appropriate extended Boolean operations, like this is done at http://lsi.research.telcordia.com/lsi-bin/lsiQuery.

Conclusion

We think the application of dynamic vector-based user profiles by means of the extended Boolean operations presented above is very promising.
We continue our work by experimenting with different kinds of extended Boolean similarity functions and their behavior on different kind of corpuses. A research has been started whose goal is the application of methods for meaning vector creation, different from LSI, because the latter cannot be easily scaled to extremely large quantity of documents.
The next stage is the design of a clickstream activity capture and a sophisticated analyzer of the user behavior in order to move further towards the creation of the personalized Web site.

```
Z:\>new_doc suntzu1.txt


    SUNTZU1.TXT: 1.00000000
   SUNTZU10.TXT: 0.96812259
    SUNTZU8.TXT: 0.96652910
   SUNTZU11.TXT: 0.93972055
    SUNTZU3.TXT: 0.93858290
    SUNTZU9.TXT: 0.93604917
    SUNTZU5.TXT: 0.93365826
    SUNTZU2.TXT: 0.93192063
    SUNTZU6.TXT: 0.93054489
    SUNTZU4.TXT: 0.92828905
    SUNTZU7.TXT: 0.92593509
      CONF2.TXT: 0.91226262
   SUNTZU13.TXT: 0.91114359
   SUNTZU12.TXT: 0.85816521
      CONF1.TXT: 0.82958573
      CONF5.TXT: 0.78001097
      CONF3.TXT: 0.75975322
      CONF8.TXT: 0.75835731
      CONF9.TXT: 0.74499495
   PLNSENCA.HTM: 0.73306848
      CONF7.TXT: 0.71974991
      CONF4.TXT: 0.71070083
    COMRULE.HTM: 0.68268537
      CONF6.TXT: 0.68213084
    APCTHOM.HTM: 0.65737315
   TOMCNTND.HTM: 0.65597126
   REPORTPL.HTM: 0.64527600
   ACTPTNPL.HTM: 0.64414208
   REPTPILT.HTM: 0.63537118
        BKS.HTM: 0.63271447
   CONSTITU.HTM: 0.60308042
.............................
```

```
Z:\>new_doc_bool
suntzu1.txt conf1.txt AND

      CONF2.TXT: 0.93506011
    SUNTZU1.TXT: 0.91479286
      CONF1.TXT: 0.91479286
   SUNTZU13.TXT: 0.90802675
    SUNTZU8.TXT: 0.90745685
   SUNTZU10.TXT: 0.89604427
    SUNTZU2.TXT: 0.88122315
    SUNTZU3.TXT: 0.87397853
   SUNTZU11.TXT: 0.86994911
      CONF5.TXT: 0.85956386
    SUNTZU6.TXT: 0.85553152
      CONF3.TXT: 0.85421266
      CONF8.TXT: 0.84673427
    SUNTZU5.TXT: 0.84201628
      CONF9.TXT: 0.83800404
    SUNTZU4.TXT: 0.83730812
    SUNTZU9.TXT: 0.83378707
    SUNTZU7.TXT: 0.82986823
      CONF7.TXT: 0.82701671
      CONF4.TXT: 0.80956085
      CONF6.TXT: 0.78754286
   SUNTZU12.TXT: 0.77238908
   PLNSENCA.HTM: 0.71858016
    COMRULE.HTM: 0.65251025
   TOMCNTND.HTM: 0.64263567
   SENTANCE.HTM: 0.60424740
        BKS.HTM: 0.58734007
    APCTHOM.HTM: 0.57461640
    FGAPCPT.HTM: 0.57040597
    GOSMARY.HTM: 0.56554976
    MYSTERY.HTM: 0.56087279
.............................
```

```
Z:\>new_doc_bool
suntzu1.txt conf1.txt NOT

    SUNTZU1.TXT: 1.00000000
   SUNTZU10.TXT: 0.98252302
    SUNTZU8.TXT: 0.98189181
   SUNTZU11.TXT: 0.96651472
    SUNTZU3.TXT: 0.96605616
    SUNTZU9.TXT: 0.96306676
    SUNTZU2.TXT: 0.96280884
    SUNTZU5.TXT: 0.96209854
    SUNTZU6.TXT: 0.96099163
    SUNTZU4.TXT: 0.95893613
    SUNTZU7.TXT: 0.95728178
   SUNTZU13.TXT: 0.95335401
   SUNTZU12.TXT: 0.91590555
   PLNSENCA.HTM: 0.84335849
    COMRULE.HTM: 0.80440870
   TOMCNTND.HTM: 0.78884876
    APCTHOM.HTM: 0.77033574
        BKS.HTM: 0.76180693
   REPORTPL.HTM: 0.75937276
   ACTPTNPL.HTM: 0.75767365
   REPTPILT.HTM: 0.75380184
    FGAPCPT.HTM: 0.73594581
   MARTBART.HTM: 0.73030361
   CONSTITU.HTM: 0.72582505
    MYSTERY.HTM: 0.72447034
   ACTJNTHE.HTM: 0.71790911
    APCJMS1.HTM: 0.71604588
     ACTMAT.HTM: 0.71421530
    REVSTEV.HTM: 0.71192762
    NAGHAM6.HTM: 0.70218790
   DEATHPLT.HTM: 0.70177880
.............................
```

```
Z:\>new_doc_bool
suntzu1.txt conf1.txt MINUS

    SUNTZU1.TXT: 0.99999999
   SUNTZU10.TXT: 0.84153095
    SUNTZU8.TXT: 0.83314816
    SUNTZU9.TXT: 0.79360580
   SUNTZU11.TXT: 0.78726789
    SUNTZU5.TXT: 0.78655673
    SUNTZU3.TXT: 0.78331581
    SUNTZU4.TXT: 0.77882682
    SUNTZU7.TXT: 0.77748381
    SUNTZU6.TXT: 0.77582141
    SUNTZU2.TXT: 0.76678133
   SUNTZU13.TXT: 0.70645429
   SUNTZU12.TXT: 0.70273529
    APCTHOM.HTM: 0.58777081
   PLNSENCA.HTM: 0.58703387
   ACTPTNPL.HTM: 0.58462459
   REPORTPL.HTM: 0.58416049
   REPTPILT.HTM: 0.57699034
    COMRULE.HTM: 0.57378553
   CONSTITU.HTM: 0.56559168
        BKS.HTM: 0.56207412
    REVJON2.HTM: 0.56055113
     ACTMAT.HTM: 0.55897387
    ACTPHIL.HTM: 0.55627470
    APCPETE.HTM: 0.55580713
   MARTBART.HTM: 0.55501092
   TOMCNTND.HTM: 0.55428700
     YASNAE.TXT: 0.55068365
    REVSTEV.HTM: 0.55059070
   AVENGSAV.HTM: 0.54946890
     ACTANM.HTM: 0.54908653
.............................
```

```
Z:\>new_doc_bool
suntzu1.txt conf1.txt OR 0

    SUNTZU1.TXT: 0.99999997
      CONF1.TXT: 0.99999993
      CONF2.TXT: 0.82317039
    SUNTZU8.TXT: 0.72128584
   SUNTZU10.TXT: 0.69019913
   SUNTZU13.TXT: 0.68069817
    SUNTZU2.TXT: 0.61376306
    SUNTZU3.TXT: 0.60314637
   SUNTZU11.TXT: 0.59609037
      CONF5.TXT: 0.57673504
      CONF3.TXT: 0.57598572
    SUNTZU6.TXT: 0.56392683
      CONF8.TXT: 0.55314244
    SUNTZU5.TXT: 0.54518708
      CONF9.TXT: 0.53777290
    SUNTZU9.TXT: 0.53540358
    SUNTZU4.TXT: 0.53511372
      CONF7.TXT: 0.52538150
    SUNTZU7.TXT: 0.52381202
      CONF4.TXT: 0.49115954
      CONF6.TXT: 0.46332613
   SUNTZU12.TXT: 0.44084160
   PLNSENCA.HTM: 0.38921508
    COMRULE.HTM: 0.35047941
   TOMCNTND.HTM: 0.34533693
   SENTANCE.HTM: 0.32821505
        BKS.HTM: 0.32180010
    APCTHOM.HTM: 0.31799264
    FGAPCPT.HTM: 0.31500315
    GOSMARY.HTM: 0.31317246
   REPORTPL.HTM: 0.31256336
.............................
```

```
Z:\>new_doc_bool   suntzu1.txt
conf1.txt OR 0.5

    SUNTZU1.TXT: 0.99999999
      CONF1.TXT: 0.99999996
      CONF2.TXT: 0.89051399
    SUNTZU8.TXT: 0.84390747
   SUNTZU10.TXT: 0.82916086
   SUNTZU13.TXT: 0.79592088
    SUNTZU2.TXT: 0.77284184
    SUNTZU3.TXT: 0.77086463
   SUNTZU11.TXT: 0.76790546
      CONF3.TXT: 0.76232891
      CONF5.TXT: 0.75792590
    SUNTZU6.TXT: 0.74723586
      CONF8.TXT: 0.74412684
    SUNTZU5.TXT: 0.73942267
    SUNTZU9.TXT: 0.73572637
      CONF9.TXT: 0.73439302
    SUNTZU4.TXT: 0.73170138
      CONF7.TXT: 0.72983250
    SUNTZU7.TXT: 0.72487356
      CONF4.TXT: 0.69979020
      CONF6.TXT: 0.67814051
   SUNTZU12.TXT: 0.64950340
   PLNSENCA.HTM: 0.56114178
    COMRULE.HTM: 0.51658239
   TOMCNTND.HTM: 0.50065410
    APCTHOM.HTM: 0.48768289
   REPORTPL.HTM: 0.47891968
   ACTPTNPL.HTM: 0.47778194
        BKS.HTM: 0.47725729
   SENTANCE.HTM: 0.47653668
   REPTPILT.HTM: 0.47355391
.............................
```

```
Z:\>new_doc_bool   suntzu1.txt
conf1.txt OR 0.75

    SUNTZU1.TXT: 0.99999999
      CONF1.TXT: 0.99999998
      CONF2.TXT: 0.92418579
    SUNTZU8.TXT: 0.90521829
   SUNTZU10.TXT: 0.89864172
      CONF3.TXT: 0.85550050
    SUNTZU3.TXT: 0.85472377
   SUNTZU11.TXT: 0.85381300
   SUNTZU13.TXT: 0.85353224
    SUNTZU2.TXT: 0.85238124
      CONF5.TXT: 0.84852132
      CONF8.TXT: 0.83961903
    SUNTZU6.TXT: 0.83889037
    SUNTZU5.TXT: 0.83654047
    SUNTZU9.TXT: 0.83588777
      CONF9.TXT: 0.83270307
      CONF7.TXT: 0.83205800
    SUNTZU4.TXT: 0.82999522
    SUNTZU7.TXT: 0.82540432
      CONF4.TXT: 0.80410553
      CONF6.TXT: 0.78554770
   SUNTZU12.TXT: 0.75383431
   PLNSENCA.HTM: 0.64710513
    COMRULE.HTM: 0.59963388
   TOMCNTND.HTM: 0.57831268
    APCTHOM.HTM: 0.57252802
   REPORTPL.HTM: 0.56209784
   ACTPTNPL.HTM: 0.56096201
        BKS.HTM: 0.55498588
   REPTPILT.HTM: 0.55446255
   SENTANCE.HTM: 0.55069749
.............................
```

```
Z:\>new_doc_bool   suntzu1.txt
conf1.txt OR 1

    SUNTZU1.TXT: 1.00000000
      CONF1.TXT: 1.00000000
   SUNTZU10.TXT: 0.96812259
    SUNTZU8.TXT: 0.96652910
      CONF2.TXT: 0.95785760
      CONF3.TXT: 0.94867210
   SUNTZU11.TXT: 0.93972055
      CONF5.TXT: 0.93911675
    SUNTZU3.TXT: 0.93858290
    SUNTZU9.TXT: 0.93604917
      CONF8.TXT: 0.93511123
      CONF7.TXT: 0.93428351
    SUNTZU5.TXT: 0.93365826
    SUNTZU2.TXT: 0.93192063
      CONF9.TXT: 0.93101313
    SUNTZU6.TXT: 0.93054489
    SUNTZU4.TXT: 0.92828905
    SUNTZU7.TXT: 0.92593509
   SUNTZU13.TXT: 0.91114359
      CONF4.TXT: 0.90842086
      CONF6.TXT: 0.89295488
   SUNTZU12.TXT: 0.85816521
   PLNSENCA.HTM: 0.73306848
    COMRULE.HTM: 0.68268537
    APCTHOM.HTM: 0.65737315
   TOMCNTND.HTM: 0.65597126
   REPORTPL.HTM: 0.64527600
   ACTPTNPL.HTM: 0.64414208
   REPTPILT.HTM: 0.63537118
        BKS.HTM: 0.63271447
   SENTANCE.HTM: 0.62485831
.............................
```

References

[1] Aldenderfer M., Blashfield R. *Cluster Analysis*. A SAGE University Paper. Sage Publications. 1984.

[2] Anick P. and Vaithyanathan S. *Exploiting clustering and phrases for context-based information retrieval*. Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval pp. 314 - 323. July 27 - 31, 1997, Philadelphia, PA USA

[3] Bates M. *Subject Access in Online Catalogs: A Design Model.*
Journal of the American Society for Information Sciences, Number 37, pp. 357-376, 1986.

[4] Berry M., Do T., O'Brien G., Krishna V., and Sowmini Varadhan, *SVDPACKC (Version 1.0) User's Guide.*
April 1993.

[5] Brin S., Page L., *The Anatomy of Large Scale Search Engine*, Stanford University, June 1998.

[6] Caid W., Carleton J. *Visualization of information using graphical representations of context vector based relationships and attributes.* United States Patent 6,794,178. Aug. 11, 1998.

[7] Deerwester S., Dumais S., Furnas G., Laundauer, T. and Harshman R.. *Indexing by Latent Semantic Analysis.* Journal of the American Society for Information Sciences, Number 41, pp. 391-47, 1990.

[8] Dumais, S. T. *Using LSI for information filtering: TREC-3 experiments*. In: D. Harman (Ed.), The Third Text REtrieval Conference (TREC3) National Institute of Standards and Technology Special Publication , in press 1995.

[9] Dumais, S. T. (1997) Using LSI for Information Retrieval, Information Filtering, and Other Things. Talk at Cognitive Technology Workshop, April 4-5, 1997.

[10] Foltz, P. W. and Dumais, S. T. (1992) Personalized information delivery: An analysis of information filtering methods. Communications of the ACM, 35(12), 51-60. Experiment using LSI for information filtering.

[11] Furnas G., Landauer T., Gomez L. and Dumais T., *Statistical semantics: Analysis of the Potential Performance of Keyword Information Systems*. Bell Syst.Tech.J., 62, Number 6, pp. 1753-1806, 1986.

[12] Harman, D. *An experimental study of the factors important in document ranking*. In Association for Computing Machinery Conference on Research and Development in Information Retrieval . Association for Computing Machinery. 1986.

[13] Kimball R. *Clicking with your customer*. Warehouse Architect, Number 1, Volume 2, January 05, 1999.

[14] Kimball R. *The Data Webhouse Has No Center*. Warehouse Architect, Number 10, Volume 2, July 13, 1999.

[15] Kimball R. *The Special Dimension of the Clickstream*. Data Webhouse, Number 2, Volume 3, January 20, 2000.

[16] Laudauer T., Foltz P., Laham D. *Introduction to Latent Semantic Analysis*. Discourse Processes, 25, pp. 259-284.

[17] Marchionini G. *Interfaces for End-User Interfaces Seeking*. Journal of the American Society for Information Science, 43(2): 156-163, 1992.

[18] Oard, D. *Adaptive Vector Space Text Filtering for Monolingual and Cross-Language Applications*, Department of Electrical Engineering, Univ. of Maryland, August 1996.

[19] Stairmand M. A. *Textual context analysis for information retrieval*. Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval pp. 140 - 147. July 27 - 31, 1997, Philadelphia, PA USA

[20] Vlajic N., Card H. *An adaptive Neural Network Approach to Hypertext Clustering*. University of Manitoba. 1998

[21] Winter R. *More Than You Hoped For*. Scalable Systems. Volume 3, Number 6, April 10, 2000.

[22] LSA 1990-99, http://lsa.colorado.edu
[23] Religions, http://davidwiley.com/religion.html
[24] The Bible,
http://www.bible.org/netbible/download.htm
[25] The Quran, http://www.usc.edu/dept/MSA/quran/