AI^{2}R: An Adaptive Clustering Technique for information Retrieval

Sándor Dominich

Department of Computer Science, University of Veszprém, Hungary, Egyetem u. 10, e-mail: dominich@dcs.vein.hu;[1] *Department of Computing and Information Technology, Buckinghamshire Chilterns University College, High Wycombe, Alexandra Road, HP11 2JZ, United Kingdom, e-mail: sdomin01@bcuc.ac.uk*

The paper suggests a novel adaptive clustering technique for information retrieval based on the Interaction Information Retrieval method. Theoretical, experimental and practical results are presented and discussed. Results based on both test collections and a real application show that this technique proves useful when retrieving from homogeneous documents, and when emphasis is on high precision.

1 Introduction

Clustering is a well-known technique applied in Information Retrieval (IR). It is typically used to group documents (to be searched), in which case the result is a – typically disjoint – set of document groups called clusters; each cluster contains – in some sense – similar documents. Several clustering methods and techniques have been proposed, such as, for example, based on similarity measures (e.g., van Rijsbergen, 1979; Salton and McGill, 1983), neighborhoods (e.g., Voorhees, 1985), hierarchies (e.g., Lebowitz, 1987; Willett, 1988; Fisher and McKusick, 1989; Crawford et al., 1991; Tanaka et al., 1999), on matrix theory (diagonalisation, singular value decomposition, e.g., Deerwester et al., 1990).

Retrieval is performed based on a cluster representative, which may but need not be one of the cluster members. If the particular retrieval method used associates a cluster representative to a query, then all members of that cluster are returned in response to the query. This view of retrieval is based on the well-known cluster hypothesis according to which closely associated documents tend to be relevant to the same request (van Rijsbergen, 1979). The cluster hypothesis is applied *a priori*, i.e., the document clusters are established before any retrieval is performed and independently of any queries. Clustering algorithms should be stable under growth (i.e., the cluster structure is unlikely to change significantly when a new document is added), under description (i.e., the cluster structure is hardly influenced by small changes in the representation or description of documents), and under ordering (i.e., the cluster structure is independent of any ordering of documents, if any). Recent research reveals a sound mathematical background for clustering, as well as for its evaluation (e.g., Mather, 2000).

This way of clustering documents is hence fixed, and good retrieval results were obtained with it. As somewhat opposed to fixed clustering, adaptive clustering (i.e., a clustering in which the cluster structure is being developed under or is being influenced by an interaction with the user, e.g., by taking into account queries, rule bases, learning patterns, etc.) has proved to be a viable approach to IR, as shown in, e.g., Cheing et al., 1985; Belew, 1989; Johnson et al., 1994, 1996; Shaw et al., 1997; Mobasher et al., 1998;). Retrieval is then viewed similar to that in fixed clustering: those documents are said to be retrieved which form the same cluster.

One way of conceiving adaptive clustering is to adopt a network- or connectionist-based (semantic networks, neural networks) view (e.g., Cohen and Kjeldsen, 1987; Belew, 1989; Doszkocs et al., 1990; Chen, 1994, 1995), but, as yet, there have not been any precise — both theoretical and practical — evaluations as to the computational complexity as well as standard and practical effectiveness of such retrieval systems. Following the connectionist line, the present paper proposes a retrieval model using adaptive clustering based on the Interaction IR (I^{2}R) paradigm. Both theoretical and practical results (based on standard test collections and a real application) are presented and discussed.

2 Associative Interaction Information Retrieval (AI^{2}R)

2.1 Interaction IR

The Interaction-based paradigm of *IR* (*I^{2}R*) as well as implementations were first suggested in [Dominich, 1994; van Rijsbergen, 1996] based on the concept of

[1] This research was partly supported by research grant no T 030194 of the National Science Foundation (OTKA) Hungary.

interaction according to the Copenhagen Interpretation in Quantum Mechanics.

The documents are represented as a flexibly interconnected network of objects (or artificial neurons). The interconnections are adjusted each time a new object (e.g., a document) is fed into the network. The query interacts with the other objects, i.e. it is treated like any other object: it is interconnected with the already interconnected other objects. Thus, on the one hand, new connections will develop (between the object-query and the other objects), and on the other hand, some of the existing connections can change. Retrieval is defined as recalled memories, i.e. those documents are said to be retrieved which belong to reverberative circles triggered by a spreading of activation started at the object-query. The reverberative circles correspond to clusters, which are not fixed as they develop under the presence and influence of the query (see below).

Note
Just like in, e.g., the Hopfield network the AI^{2}I network is a single-layered interconnected network, but unlike in Hopfield net, where nodes are activated in parallel and then relaxed until a stable state is reached, in the AI^{2}R network nodes are sequentially activated according to a winner takes all strategy. In principle, convergence is represented by the stable state in the Hopfield net, and by the reverberatice circle in the AI^{2}R network.

The idea of flexible, multiple and mutual interconnections from *I^{2}R* also appear and are investigated in [Salton, Allan and Singhal, 1996; Salton, Singhal, Mitra and Buckley, 1997; Pearce and Nicholas, 1996; Carrick and Watters, 1997; Liu, 1997; Mock and Vemuri, 1997; Dominich, 1999, 2000].

In what follows, an implementable model for I^{2}R is described briefly.

2.2 Network of Interconnected Documents

Any object-document o_i, $i = 1, 2, ..., M$, is associated a set of identifiers (e.g., keywords, index terms) t_{ik}, $k = 1, 2, ..., n_i$. There are weighted and directed links between any pair (o_i, o_j) of objects. The one is the frequency of a term given a document, i.e. the ratio between the number f_{ijp} of occurrences of term t_{jp} in object o_i, and the length n_i of o_i, i.e. total number of terms in o_i:

$$w_{ijp} = \frac{f_{ijp}}{n_i}, \quad p = 1, ..., n_j$$

Because the value w_{ijp} is analogous to the probability with which object o_i 'offers' t_{jp}, the corresponding link can be viewed as being directed from object o_i towards object o_j. The other is the extent to which a given term reflects the content of a document, i.e., the inverse document frequency. f_{ikj} denotes the number of occurrences of term t_{ik} in o_j, df_{ik} is the number of documents in which t_{ik} occurs, w_{ikj} is given by the inverse document frequency formula, and thus represents the extent to which t_{ik} reflects the content of o_j:

$$w_{ikj} = f_{ikj} \log \frac{2M}{df_{ik}}$$

Because the value w_{ikj} is a measure of how much content of object o_j is 'seen' by term t_{ik}, the corresponding link can be viewed as being directed from o_i towards o_j. The other two connections — in the opposite direction — have the same meaning as above: w_{jik} corresponds to w_{ijp}, while w_{jpi} corresponds to w_{ikj} (Figure 1).

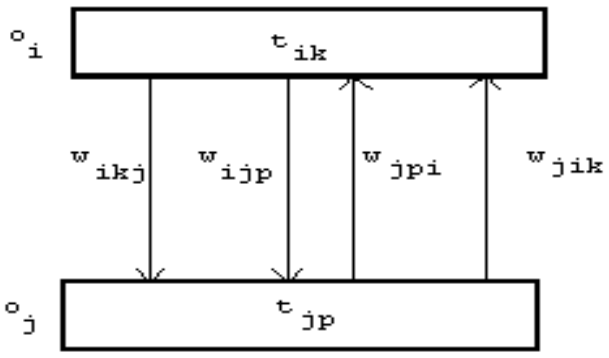

Figure 1. Connections between object pairs.

Let us consider and example. Given two object-documents o_1 and o_2 as follows:

o_1 = *The paper is a good survey of free-text retrieval in the vector model.*

o_2 = *These notions are important in modelling information retrieval. They help to clearly understand the vector model of retrieval.*

The objects are now represented as tuples (collections) of terms (index terms or terms considered to reflect their content), e.g.:

o_1 = (t_{11} = free-text, t_{12} = retrieval, t_{13} = vector model),

o_2 = (t_{21} = information retrieval, t_{22} = vector model, t_{23} = retrieval).

The associated connections and weights are shown in Figure 2.

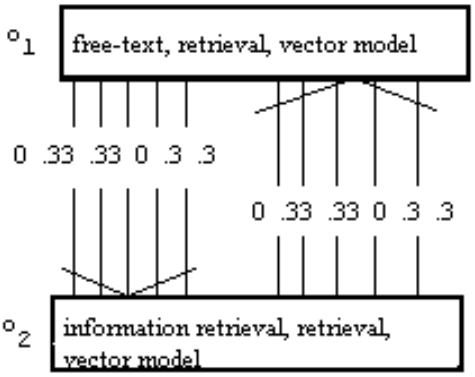

Figure 2. Weighted connections.

2.3 Interaction Between Query and Documents

Let us consider now a query *Q* as follows:

Q = Usage of free-text technique and user model in the vector model of retrieval.

Q is incorporated first into the network of interconnected objects, i.e., *Q* becomes a member of this structure as if it were just another object, say o_3:

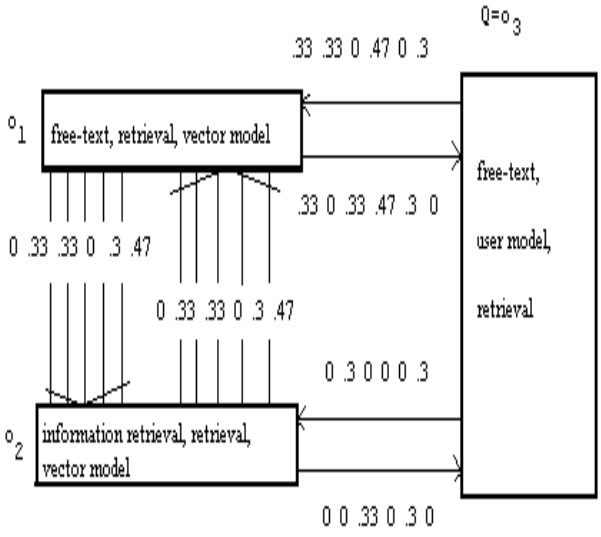

Figure 3. Interaction between query and documents.

$Q = o_3 = (t_{31}$ = free-text, t_{32} = user model, t_{33} = retrieval)

Figure 3 shows the new structure: on the one hand, there are the new connections (between o_3 and o_1, and o_3 and o_2), on the other hand, some of the old weights change: in our example 0.47 instead of 0.3 between o_1 and o_2. This represents an interaction between the query and documents.

2.4 Retrieval

The objects are conceived as being a network of artificial neurons in which a spreading of activation takes place according to a winner takes all strategy.

The activation is initiated at the query, say o_j, and spreads over along the strongest connection thus passing on to another neuron, and so on. The strength of the connection between the query and another object o_i is be defined as follows:

$$\sum_{p=1}^{n_j} w_{jpi} + \sum_{k=1}^{n_i} w_{jik}$$

This summation is made possible by the meaning associated to w_{jpi} and w_{jik} (see point 2.2), and it is a measure of the extent to which the query, represented by o_j, 'identifies' o_i. After a number of steps the spreading of activation reaches an object already affected, i.e., a reverberative circle is formed: the activation spreads in a circle of objects. This is analogous to a local memory recalled by the query. The reverberative circle can be interpreted as an adaptive cluster associated to the query when this is present. Those objects are said to be retrieved in response to the query which belong to the same reverberative circle, and they are ranked in the order of maximal activation. The same objects may not form the same cluster for a different query.

In our example: the activation is spread from the query o_3 towards the other two objects o_1 and o_2. The strength of connection with o_1 is equal to 0.33 + 0.33 + 0 + 0.47 + 0 + 0.3 = 1.43, whilst with o_2 to 0 + 0.3 + 0 + 0 + 0 + 0.3 =0.6. Because 1.43 > 0.6 o_1 gets the highest activation, thus o_1 is selected in the next step, and becomes active whilst all the others, i.e. o_3 and o_2 will become passive. Activity is spread now from o_1 towards o_2 and o_3. o_2 is activated by the value 0 + 0.33 + 0.33 + 0 + 0.3 + 0.47 = 1.43 whilst o_3 by the value 0.33 + 0 + 0.33 + 0.47 + 0.3 + 0 =1.43. As a result of the reverberative circle formed between o_3 and o_1, o_1 is selected first as an answer to the query. Because the activation spreads along equal values from o_1 towards o_2, too, in the next step o_2 becomes active, all the others inactive, and o_2 will spread the activation: o_1 is activated by the value 1.43, while o_3 by 0.63. Thus

another reverberative circle is formed between o_1 and o_2, and thus the next retrieved document is o_2. As there are no other reverberative circles the retrieval process ends.

3 Evaluation of AI^{2}R

The AI^{2}R-based retrieval has been evaluated by performing three types of research: simulations, theoretical, and tests.

3.1 Simulations

The simulations were carried out to monitor the number of non-zero weights (as these determine the sum of point 2.4), of retrieved objects (to see whether their number is within reasonable limits so that the user can assess them), and of multiple maximums (as these influence the number of reverberative circles, see point 2.4).

Number of non-zero links

As it could be seen in parts 2.4 and 3.1 the number of non-zero links plays an important role in the AI^{2}R model in that it influences both memory and running time. Simulations were conducted (in MathCAD Plus 8 Professional) with different number of documents and index terms in order to observe the number of non-zero links between objects, as well as the number of retrieved documents.

The results are as follows.

Let v denote the number of index terms, which is varied in general from 40 to 1000. The number n of documents varies, and it is a multiple of v. Let l denote the length of a document expressed as the number of index terms it contains, l is also varied in general from 3 to 20.

The simulation results show that the number of non-zero links varies (approximately) quadratically with the number of documents (Figure 4).

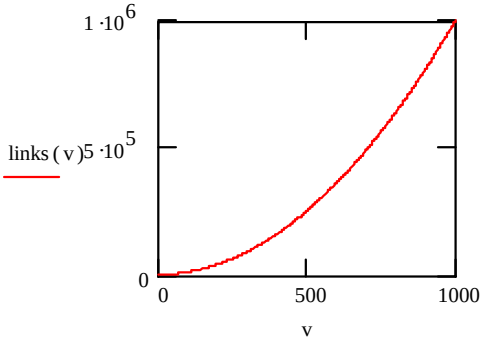

Figure 4. The number of non-zero links varies quadratically with the number of documents.

Number of retrieved objects

The simulation parameters and the (average) number of retrieved documents are shown in Table 1. The number n of documents varies to ten thousands, ten runs were carried out for each n. l and v vary as shown in the table.

Table 1. Simulation of the number of retrieved documents in AI^{2}R.

Simulation parameters	**Number of retrieved documents**
$n = v$ = constant $l = 2n$	under twenty
$n = v$ = constant $l < v$	under twenty
n = varies $v = 10l$	around thirty-forty
n, l, v vary	around forty-sixty

When different values were taken than those given above overflow error was encountered. Nevertheless the simulations results so obtained seemed to be relevant enough for applications. (See part 4.)

The simulation results show that the number of retrieved documents remains under a manageable limit in general. This is very important when the user assesses the retrieved documents, and also because the user is not frustrated by a large number of retrieved objects (such as, e.g., in the case of Internet search engines).

Number of multiple maximums

After weights summations (see point 2.4) there are $M - 1$ links (weights) — s_1, ..., s_{i-1}, s_{i+1}, ...,s_M — from an object o_i to all the other objects. Because i varies from 1 to M there are at most $M(M - 1) = O(M^2)$ links to be evaluated in all (in a search). Depending on the multiplicity (i.e., unique, double, triple maximum, or higher) of the maximum of the series s_1, ..., s_{i-1}, s_{i+1}, ...,s_M the number of reverberative circles becomes larger. Thus the multiplicity of maximums is an important parameter, and its change in number is monitored in the simulation below. Because the total number of links in the entire network (after they have been summed) varies quadratically with the number of objects (see part 2), different values for the number of objects were taken, and the summed weights were generated at random. Several runs were carried out in each case, and Table 2 shows the averaged results of the simulations: the number of weights, and the average multiplicity of maximums.

Table 2. Simulation of the multiplicity of maximums.

	Multiplicity of maximums Number of weights			
Total number of weights	**1**	**2**	**3**	**4**
1 000	985	14	1	0
2 500	2469	30	1	0
10 000	9532	439	26	2

The simulation results show that there always are a few multiple maximums, and their number is expected to increase with the number of weights. Moreover it can also be expected that higher order multiplicity (5, ...) is also to be expected at higher numbers of weights. Interestingly enough the proportion of the multiple links is slightly increasing: 0.015, 0.0124, and 0.052, respectively. This can account for the fact that the number of retrieved objects does not increase drastically with the number of objects to be searched (as perhaps expected).

Based on the above results, in average, in 3% there is double, in 0.2% there is triple, and in the rest there is single maximum. Drawing the empirical density function this has exponential shape and is thus approximated by the function $f(x) = u^2 e^{-u^{0.7}x}$. After curve fitting this becomes $f(x) = 3.864e^{-1.605x}$ (Figure 5), and thus the probability to have multiple (2 or 3) maximum in a random series s_1, ..., s_{i-1}, s_{i+1}, ...,s_M of weights is estimated to be $\int_2^3 3.864e^{-1.605x}dx = 0.078$.

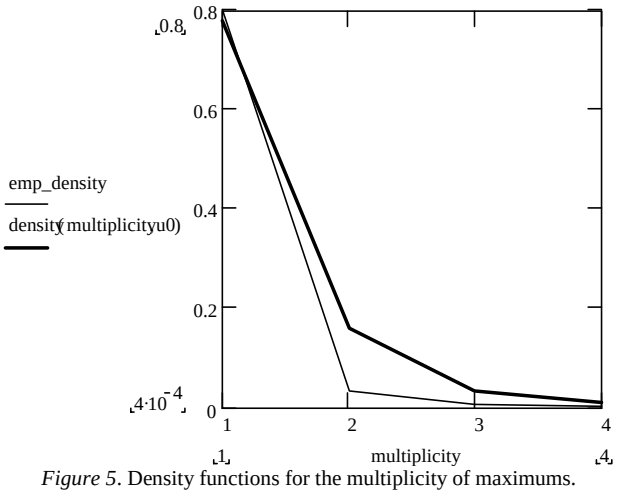

Figure 5. Density functions for the multiplicity of maximums.

As conclusions of the simulations it can be said that the number of non-zero links varies quadratically with the number of objects, and the number of retrieved objects remains within manageable limits.

3.1 Theoretical Results

As it could be seen in part 2 an algorithm which implements AI^{2}R should compute a huge number of weights. Thus the question of the complexity of such a computation arises, and it is answered in the following theorem.

THEOREM 1. The complexity of weights computation is polynomial.

Proof. As it can be seen from the formulas of point 2.2, there are $2(n_i + n_j)$ number of weights between every pair (o_i, o_j) of which there are ${}^MC_2$ (combinations of M taken by 2), hence

$$2(n_i + n_j)\,{}^MC_2 = O(NM^2)$$

is an upper bound for weights computation, where $N = \max_{i,j}(n_i, n_j)$, i.e., the largest of object lengths.

The computation of the sums of the weights (point 2.4) between a given object o_i and all the other objects o_j, of which there are $M - 1$, is $(n_i + n_j)(M - 1)$, and thus an upper bound for the computation of all sums is

$$(n_i + n_j)(M - 1)^2 = O(NM^2)$$

because i can vary, too, at most M - 1 times.

An upper bound to find the strongest connection from a given object and all the others is $O(M)$ (finding the maximum from a sequence of numbers), and thus an upper bound for the selection of all strongest connections in the emtire network is

$$O(M)(M - 1) = O(M^2)$$

Hence an overall upper bound for the weights computation is

$$O(NM^2) + O(NM^2) + O(M^2) = O(NM^2) = O(K^3)$$

where $K = \max(N, M)$. ■

Once the weights have been calculated, the complexity of the retrieval process is given by:

THEOREM 2. The retrieval process takes polynomial time.

Proof. The spreading of activation starts at o_q representing the query, and means finding

$$\max_i w_{iq}\ ,\ i = 1, ..., M - 1$$

i.e., from all the weights towards all the other objects of which there are $M - 1$). Finding this maximum has complexity $O(M)$. Let $o_{m'}$ denote the winner object, i.e., the object to which the activation spreads, and let L denote a list keeping all the winner objects. It should be checked whether m' has already been a winner or not. This is accomplished by checking whether m' is in L or not: if it is we have a reverberative circle, and we stop, but if it is not in L it is written into L in the next available location. Checking L takes $O(\text{length}(L))$ which in the worst case means $O(M)$. Because the spreading of activation is carried out at most M times, the spreading of activation takes

$$M \cdot O(M) = O(M^2)$$

time.

If all the weights are unique there only is one reverberative circle, but if there are objects, say o_i ($i = j_1, ..., j_k$) from which there are more than one (i.e., multiple), say n_i maximal weights the number of reverberative circles will be

$$\prod_i n_i = O(n^k),$$

where $n = \max_i n_i$, and thus an upper bound for the overall complexity of retrieval is

$$O(n^k) \cdot O(M^2) = O(n^k \cdot M^2) = O((\max(n^k, M^2))^2). \blacksquare$$

Although these results do not offer tight bounds for the computation of weights or retrieval they are important in that it means that the computations in AI^{2}R have polynomial complexity, and thus the method is tractable. Moreover taking into account that not all the weights are to be re-computed but only those given by the inverse document frequency formula, an implementation of such a computation should be fast enough (especially together with different programming techniques to speed it up; see part 4).

3.3 Tests

The AI^{2}R technique was implemented in C++, and tested on the ADI and MED standard test collections. Index terms were obtained automatically using a standard technique (stoplist and stemming). Running I^{2}R for the ADI test collections, the following practical results were obtained.

ADI Results

The statistics for the ADI test collection is shown in Table 2.

Table 2. Statistics for the ADI test collection.

Subject Area	Information Science
Type	Homogeneous
No. of Documents	82
No. of Queries	35
No. of Terms	736
Mean no. terms/Document	11
Mean no. of Terms/Query	6

The standard 11-point recall–precision plot is shown in Figure 6 (the line with squares), which also shows, for comparison purposes, the graph obtained by a method called correlated search (CS) for the same ADI test collection (Bodner and Song, 1996). The correlated search method performs much better than the standard vector space method (SMART), and uses query expansion: the query vector is expanded with related terms extracted from appropriate thesauri.

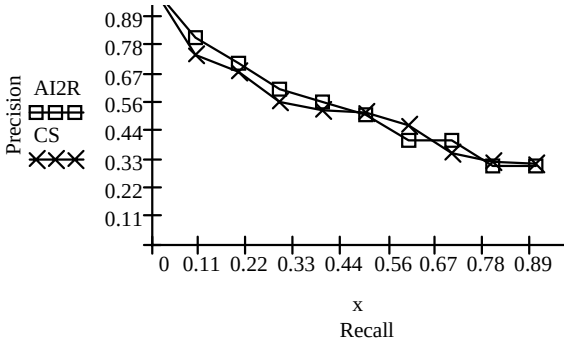

Figure 6. Recall–precision plot of AI^{2}R for the ADI test collection.

The average precision at seen relevant documents is 0.555 in the AI^{2}R search, and outperforms the correlated search by a factor of 3% in terms of average precision. Just like the correlated search the AI^{2}R method too favours high precision (low to middle recall) where it performs better with 7%.

MED Results

The statistics for the MED test collection is shown in Table 3.

Table 3. Statistics for the MEDLINE test collection.

Subject Area	Medical Sciences
Type	Homogeneous
No. of Documents	1033
No. of Queries	30
No. of Terms	5732
Mean no. terms/Document	55
Mean no. of Terms/Query	9

The standard 11-point recall–precision plot is shown in Figure 7 (the line with squares), which also shows, for comparison purposes, the graphs obtained using the SMART and LSI (Latent Semantic Indexing) methods (Deerwester et al., 1990).

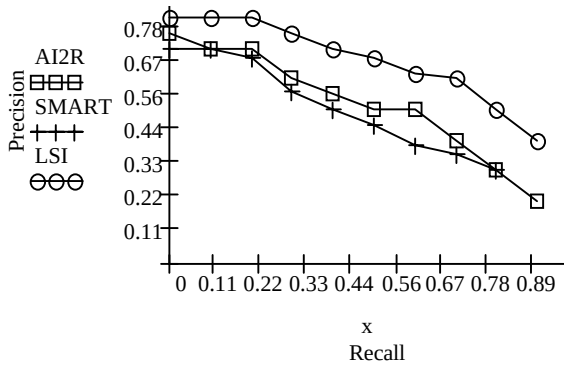

Figure 7. Recall–precision plot of AI^{2}R for the MED test collection.

The average precision at seen relevant documents is 0.51 in the AI^{2}R search, whereas 0.43 using SMART search, and 0.66 in the LSI method. If we take, as usual, the SMART results as a reference, the AI^{2}R outperforms the SMART method with 18% but is weaker than the LSI method by 30% (as regards average precision).

Conclusions of Evaluations

The simulation results show that the AI^{2}R method keeps the number of the retrieved documents within a manageable limit, i.e., the user will not be frustrated by a large quantity of returned documents, say in the thousands or millions (like, say, the Web search engines do).

The standard tests show that the AI^{2}R search outperforms (at low and middle recall values) the standard SMART search with 23% in average in terms of average precision, the correlated search with 7% at low to middle recall levels.

As a result of these considerations a real application was developed, implemented, and evaluated using the AI^{2}R method. This is described in the next part.

4 Application

4.1 Short Description

An application was developed which uses the AI^{2}R method, and makes it possible to search the M.Sc. theses written in the School of Technical Informatics at the University of Veszprem (Hungary). The application has been running for almost two years, and can be found and accessed at the following Internet address http://dcs.vein.hu/CIR/i2rapp/index.html.

The entire application consists of a number of programs written in several languages (C, HTML, Visual Basic). The communication between the Web server and the search programs is based on the CGI protocol.

The application consists of several modules of which the most important are as follows (they are not described in details as this would go beyond the scope of the paper):

(1) *Object Editor*. This module is used off-line, and makes it possible to create/edit the objects, i.e. it is used when new documents are added, or when existing ones are modified. The documents can be typed in directly or can be transferred from files previously created.

(2) *Object Base Editor*. This module is used off-line, and makes it possible to create and modify object bases, i.e. special databases which contain the documents. An object base corresponds to a network of documents.

(3) *Validation Module*. This module consists of several programs which carry out the necessary formal and consistency validations, as well as statistical analyses.

(4) *Search Module*. This module is used online on the Web. It consists of a series of user interfaces which are seen by the user (see figures below), and of a set of search programs which carry out the retrieval. The following example shows a routine session with AI^{2}R. Figure 8 shows the title page of the application which has two versions: an English and a Hungarian version. By clicking on the appropriate title the desired version is selected.

Figure 8. The title page of the AI^{2}R application.

Figure 9 shows the page on which the user can enter the query, and can select the object base to be searched. The query should be entered as terms separated by commas; there is no other restriction (although commas are not needed either).

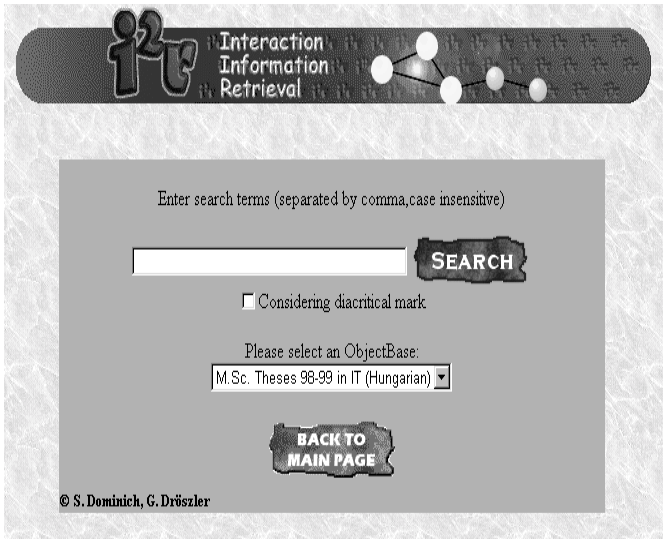

Figure 9. Figure Query formulation and object base selection interface.

Figure 10 shows those two pages which contain the answer to the query. The first page (top one) contains a list of hits showing the major parameters of the documents (title, author, year) If the user wishes to find out more about a specific document he/she can click on the respective list element, and a next page is shown on the screen (bottom one), from which he/she is pointed to further links.

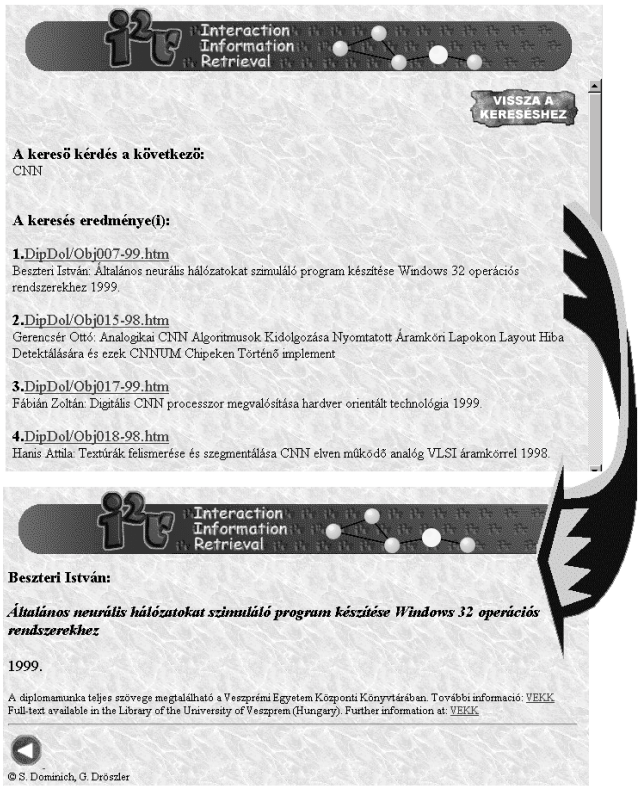

Figure 10. Pages showing retrieved objects.

As we have seen earlier the number of non-zero links grows quadratically with the number of objects, which also means that the number of zero links is much higher. Hence only the non-zero links are effectively stored, and this yields a considerably less storage capacity.

We have also seen that the retrieval in AI^{2}R is very computation demanding. This means that appropriate techniques should be used in order to speed up the response time. Using a special numeric encoding the time needed for the computation of connection strengths was reduced more than ten times compared to the time when string comparison algorithms were used.

4.2 Evaluation

In order to evaluate the application methods based on standard measures (recall, precision, coverage, novelty, relative recall, recall effort; Baeza-Yates and Bibeiro-Neto, 1999) cannot be used. This is a real application, and hence the set of relevant documents to a query is not known in advance, it is not independent of the user, and the queries cannot be envisaged either. Also different users have different perceptions of which documents are relevant and which are not.

Although the application can be accessed by anybody on the WWW, its primary target audience are the final year undergraduate students who are in the

process of choosing a topic and supervisor for their MSc theses (final year projects), and are thus interested in theses already written and in which topics as well as in who the supervisors were (as potential supervisors for them).

The application was evaluated by distributing questionnaires to final year students who were asked to fill them in after having performed effective searches.

The first question asked the students to qualify the screens of the user interface (Search Module) in terms of colours used, structure, usability, aspect, on a four-point scale (don't like = 1, so-so = 2, good = 3, very good = 4).

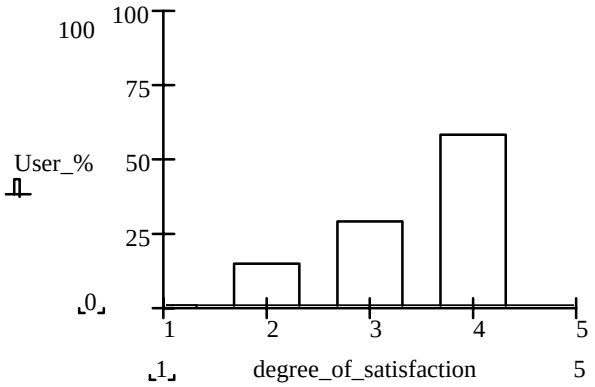

Figure 11. Graph showing user satisfaction with interface screens.

The results are shown in Figure 11. As it can be seen the majority of users were content with the interface screens which they qualified as very good (more than 60%), good (almost 30%), or so-so (the rest of about 10%). There was no student qualifying these screens as ‘don't like’.

The next question asked the students to qualify the returned documents as to how relevant they found them on a scale of four values as follows: not satisfied = 1, too few relevant documents = 2, satisfied = 3, very satisfied = 4. The results are shown in Figure 12.

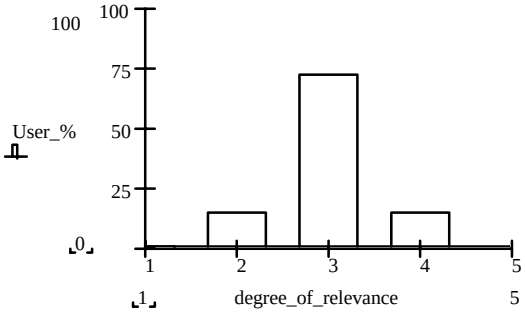

Figure 12. Graph showing user satisfaction with relevance in AI^{2}R.

The results show that almost 75% of the users were satisfied with the returned documents. 13% were very satisfied, and 12% thought that there were too few relevant documents in the answers. There were other questions too on the questionnaire such as, for example, What would you change in the screens?, Was the response time reasonable?, and other questions, which can help enhance the application.

These practical evaluations have confirmed what could be expected based on the results of the simulation and test collections, namely that the users are satisfied with the retrieval application based on AI^{2}R which favours high precision and low recall, and if we were to qualify the application we could say that it performs at an effectiveness level of about 75%.

4 Conclusions

A novel adaptive clustering technique for information retrieval was presented based on the interaction IR method.

It was shown that the complexity of computations involved is polynomial, hence the method is tractable. The number of retrieved objects in response to a query remains within manageable limits which allows the user to effectively assess them if wanted. Thus the AI^{2}R retrieval lends itself to be 'an automatic pre-processor' for a relevance feedback process (i.e., to retrieve an initial set of documents to start relevance feedback with).

Standard test collections were used to evaluate the classical effectiveness of the AI^{2}R retrieval. The results show that it useful when high precision is favoured at low to middle recall values.

A real application based on the AI^{2}R method was also presented, and evaluated using feedback from real users given on questionnaires. The results of this inquiry show that the application meets very well users' satisfaction.

Acknowledgements

The author would like to thank Endre Jeges for helping carry out the tests, Adam Nagy for helpful discussions, Gabor Droszler for helping write the application.

References

Baeza-Yates, R., and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*, Addison Wesley.

Belew, R. K. (1989). Adaptive information retrieval: Using a connectionist representation to retrieve and learn about documents. *Proceedings of the 12th ACM SIGIR '89*, pp. 11-20, Cambridge, MA, ACM Press.

Bodner, R., and Song, F. (1996). Knowledge-based approaches to query expansion in information retrieval. In McCalla, G. (ed.) *Advances in Artificial Intelligence*, Springer, 146-158.

Carrick, C. and Watters, C. (1997) Automatic Association of News Items. *Information Processing and Management*. 33(5): 615-632.

Chen, H., Hsu, P., Orwig, R., Hoopes, L., and Nunamaker, J.F. (1994). Automatic concept classification of text from electronic meetings. *Communications of the ACM*, **37**(10): 56-73.

Chen, H. (1995). Machine Learning for information retrieval: Neural networks, symbolic learning and genetic algorithms. *Journal of the American Society for Information Science*, **46**: 194-216.

Cohen, P., and Kjeldson, R. (1987). Information retrieval by constrained spreading activation in semantic networks. *Information Processing and Management*, **23**: 255-268.

Crawford, S.L., Fung, R., Appelbaum, L.A., and Tong, R.M. (1991). Classification trees for information retrieval. *Proceedings of the 8th Workshop on Machine Learning*, Morgan Kaufmann, 245-249.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, **41**: 391-407.

Dominich, S. (1994) Interaction Information Retrieval. *Journal of Documentation*, 50(3): 197-212.

Dominich, S. (1999). Associative Database for Information Retrieval. *Proceedings of the 3rd International Austrian-Israeli Technion Symposium*, RISC Hagenberg Castle, Linz, Austria, pp. 205-209.

Dominich, S. (2000). *Mathematical Foundations of Information Retrieval*. Kluwer Academic Publishers (to appear).

Doszkocs, T., Reggia, J., and Lin, X. (1990). Connectionist models and information retrieval. *Annual Review of Information Science & Technology*, **25**: 209-260.

Fisher, D.H., and McKusick, K.B. (1989). An empirical comparison of ID3 and back-propagation. *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI-89)*, Detroit, MI, 788-793.

http://dcs.vein.hu/CIR. *AI^{2}R Application*.

Johnson, A., Fotouhi, F., and Goel, N. (1994). Adaptive clustering of scientific data. *Proceedings of the 13th IEEE International Phoenix Conference on Computers and Communication*, Tempe, Arizona, pp. 241-247.

Johson, A., and Fotouhi, F. (1996). Adaptive clustering of hypermedia documents. *Information Systems*, **21**: 549-473.

Lebowitz, M. (1987). Concept learning in a rich input domain: Generalization-based memory. In Carbonell, J.G., Michalski, R.S., and Mitchell, T.M. (eds.) *Machine Learning, An Artificial Intelligence Approach, Vol. II*., Morgan Kaufmann, 193-214.

Mather, L. A. (2000). A Linear Algebra Measure of Cluster Quality. *Journal of the American Society for Information Science*, **51**: 602-613.

Mobasher, B., Cooley, R., and Srivastava, J. (1998). Creating Adaptive Web Sites Through Usage-Based Clustering of URLs. Proceedings of the 1999 Workshop on Knowledge and Data Engineering Exchange,

Mock, K.J. and Vemuri, V.R. (1997) Information Filtering via Hill Climbing, Wordnet and Index Patterns. *Information Processing and Management*. 33(5): 633-644.

Pearce, C. and Nicolas, C. (1996) TELLTALE: Experiments in a Dynamic Hypertext Environment for Degraded and Multilingual Data. *Journal of the American Society for Information Science*, 47(4): 263-275.

Salton, G., Allan, J. and Singhall, A. (1996) Automatic Text Decomposition and Structuring. *Information Processing and Management*. 32(2): 127-138.

Salton, G., and McGill, M. (1983). *Introdcution to Modern Information Retrieval*. McGraw Hill, New York.

Salton, G., Singhall, A., Mitra, M. and Buckley, C. (1997) Automatic Text Structuring and Summarization. *Information Processing and Management*. 33(2): 193-207.

Sebastiani, F. (1994) A probabilistic termonological logic for information retrieval. *ACM SIGIR 17th International Conference on Research and Development in Information Retrieval*, Dublin, Ireland, Springer, London, 122-130.

Shaw, W., Burgiu, R., and Howell, P. (1997). Performance standards and evaluations in IR test collections: Cluster-based retrieval models. *Information Processing and Management*, **33**: 1-14.

Tanaka, H., Kumano, T., Uratani, N., and Ehara, T. (1999). An efficient document clustering algorithm and its application to a document browser. *Information Processing and Management*, **35**(4): 541-557.

van Rijsbergen, C. J. (1979). *Information Retrieval*. Butterworth, London.

Vorhees, E. (1985). The cluster hypothesis revisited. *SIGIR*, 188-196.

Willett, P. (1988). Recent trends in hierarchic document clustering. *Information Processing and Management*, **24**: 577-597.

Yu, C.T., Suen, C., Lam, K., and Siu, M.K. (1985). Adaptive record clustering. *ACM Transactions on Database Systems*, **10**(2): 180-204.